EdgeGallery Documentation

发布 *0.0.1*

EdgeGallery Authors

2021 年 11 月 08 日

Contents

1 Getting Started **3**

1.1 EdgeGallery 界面一览 . . . 3
1.2 APP 开发流程 . . . 6
1.3 Demo 介绍 . . . 6
1.4 开发者平台容器部署流程介绍 . . . 6
1.5 开发者平台虚机部署流程介绍 . . . 10

2 Release Notes **17**

2.1 EdgeGallery Introduction . . . 17
2.2 Version: v1.3.0 . . . 17
2.3 Version: v1.2.0 . . . 23
2.4 Version: v1.1.0 . . . 29
2.5 Version: v1.0.0 . . . 33
2.6 Version: v0.9.1 . . . 36
2.7 Version: v0.9 . . . 40

3 EdgeGallery v1.1 版本架构图 **45**

3.1 参与 EdgeGallery 架构讨论 . . . 46

4 开发者指导 **47**

5 安装指南 **49**

5.1 EdgeGallery 安装指导 . . . 49
5.2 EdgeGallery 证书更新 . . . 49

6 AppStore **51**

6.1 概述 . . . 51
6.2 特性设计 . . . 52
6.3 接口设计 . . . 53

6.4 数据库设计 . . . 104
6.5 如何贡献 . . . 108

7 ATP **123**
7.1 概述 . . . 123
7.2 整体框架说明 . . . 124
7.3 测试套、测试场景和测试用例的对应关系 . . . 124
7.4 特性设计 . . . 125
7.5 atp Interfaces . . . 127
7.6 atp v2 Interfaces . . . 157
7.7 数据库设计 . . . 190
7.8 ATP Contribution . . . 192
7.9 ATP Guide . . . 196

8 Developer **205**
8.1 Developer Overview . . . 205
8.2 Developer 特性清单 . . . 207
8.3 Developer Interfaces . . . 223
8.4 Developer Dependency . . . 335
8.5 Developer DataBase Design . . . 344
8.6 Developer Contribution . . . 354

9 MECM **361**
9.1 **Overview 概述** . . . 361
9.2 **Architecture** . . . 362
9.3 **Interface Designs 界面设计** . . . 364
9.4 **Database Designs 数据库设计** . . . 463
9.5 **User Guides 使用指导** . . . 470
9.6 **How to Contribute 如何贡献** . . . 482
9.7 **Port Matrix 端口矩阵** . . . 492

10 MEP **493**
10.1 概述 . . . 493
10.2 特性设计 . . . 495
10.3 接口设计 . . . 506
10.4 依赖 . . . 598
10.5 数据库设计 . . . 599
10.6 **How to Contribute 如何贡献** . . . 600

11 User Management **603**
11.1 User Overview . . . 603
11.2 特性设计 . . . 605
11.3 User Interfaces . . . 617

11.4 数据库设计 633
11.5 如何参与贡献 636

12 Kubernetes NodePort List **637**

13 ServiceComb NodePort List **639**

EdgeGallery 社区聚焦 5G 边缘计算场景，通过开源协作构建起 MEC 边缘的资源、应用、安全、管理的基础框架和网络开放服务的事实标准，并实现同公有云的互联互通。在兼容边缘基础设施异构差异化的基础上，构建起统一的 MEC 应用生态系统。

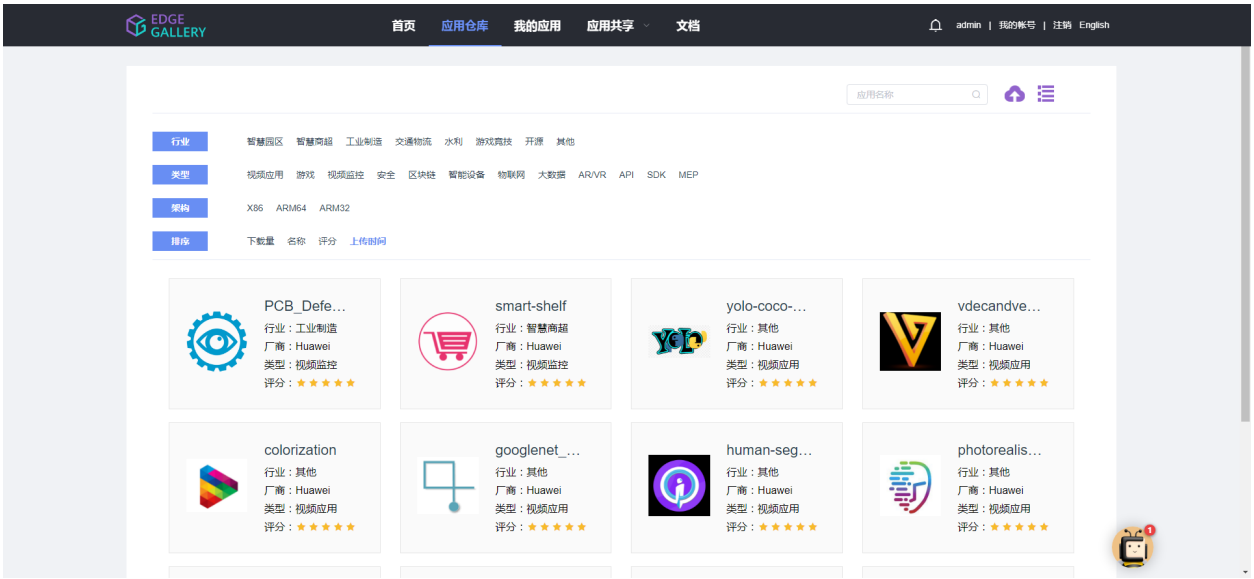
CHAPTER 1

Getting Started

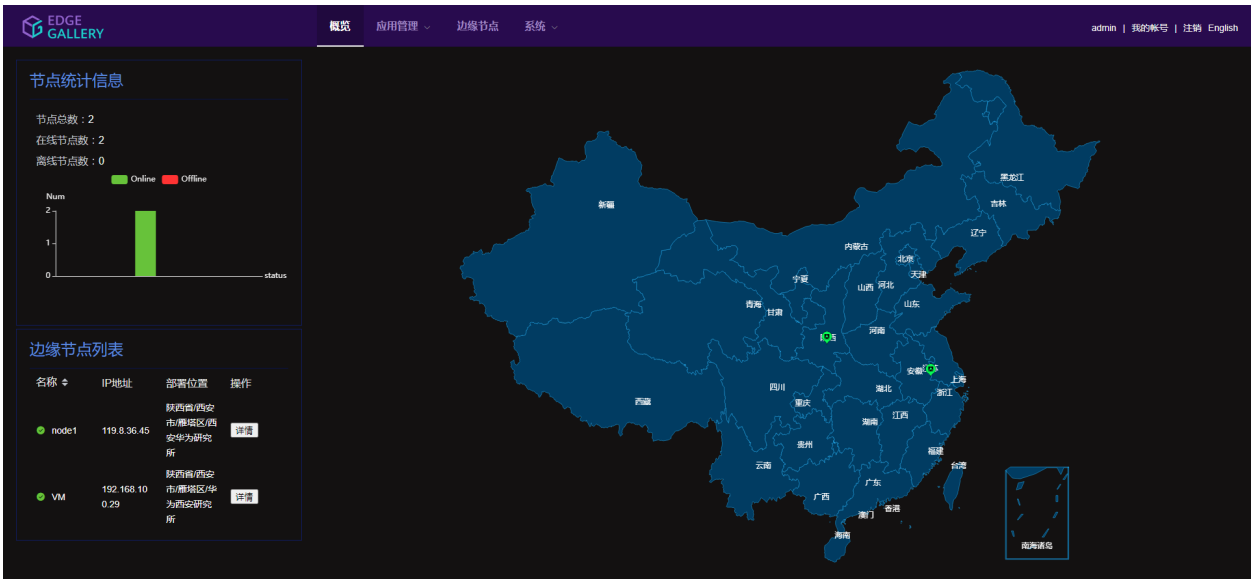
1.1 EdgeGallery 界面一览

1.1.1 开发者平台

1.1.2 应用商城

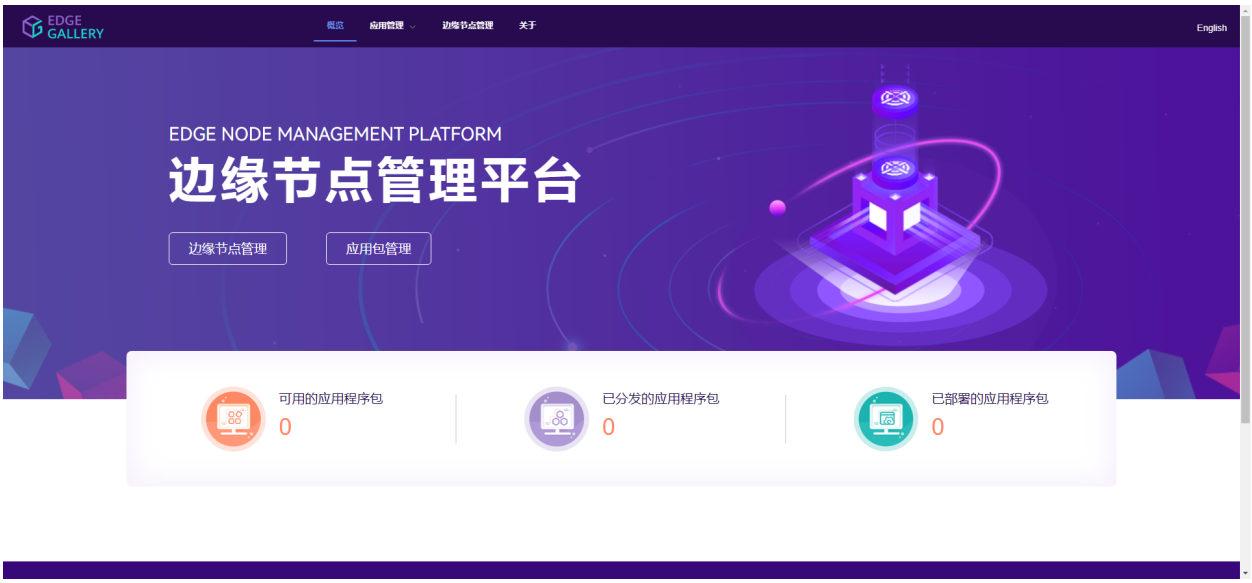


1.1.3 MEC 管理平台

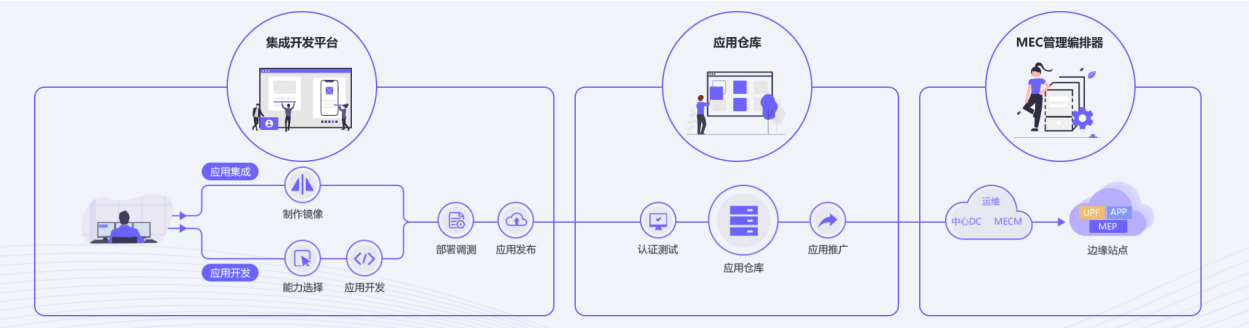


1.1.4 应用测试认证平台

1.1.5 边缘节点管理平台



1.2 APP 开发流程

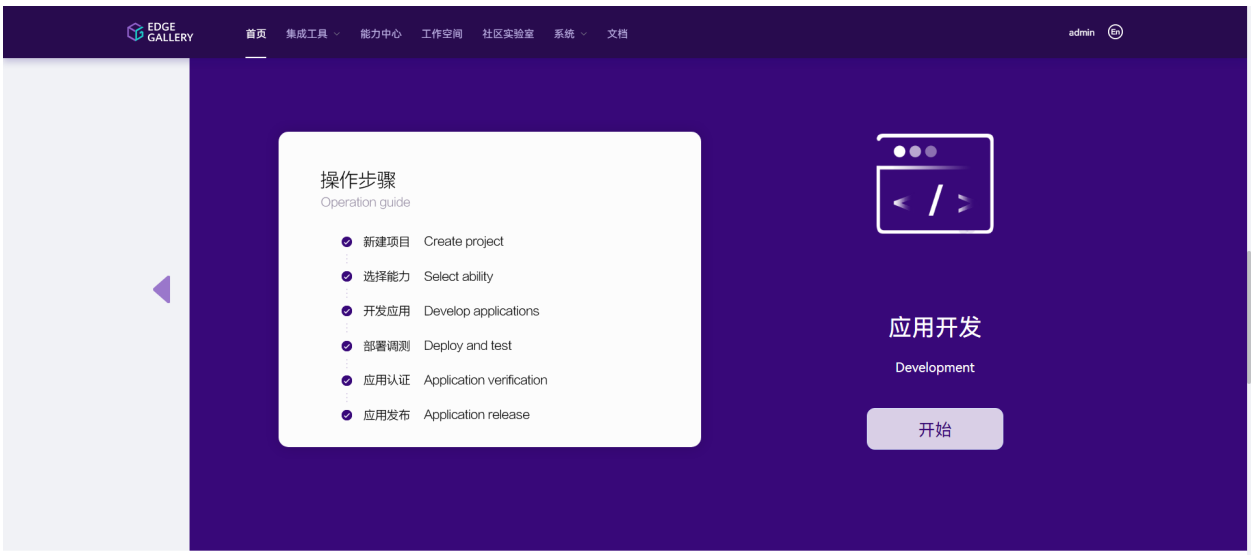


1.3 Demo 介绍

基于 EdgeGallery 的定位服务开发，展示如何基于 EdgeGallery 开发者平台进行边缘应用开发

- 定位服务应用作为服务消费者依赖 EdgeGallery 平台的“人脸识别服务”
- 定位服务应用作为服务供应者发布“定位服务”，对外提供生态 API，其他服务可以消费“定位服务”

1.4 开发者平台容器部署流程介绍



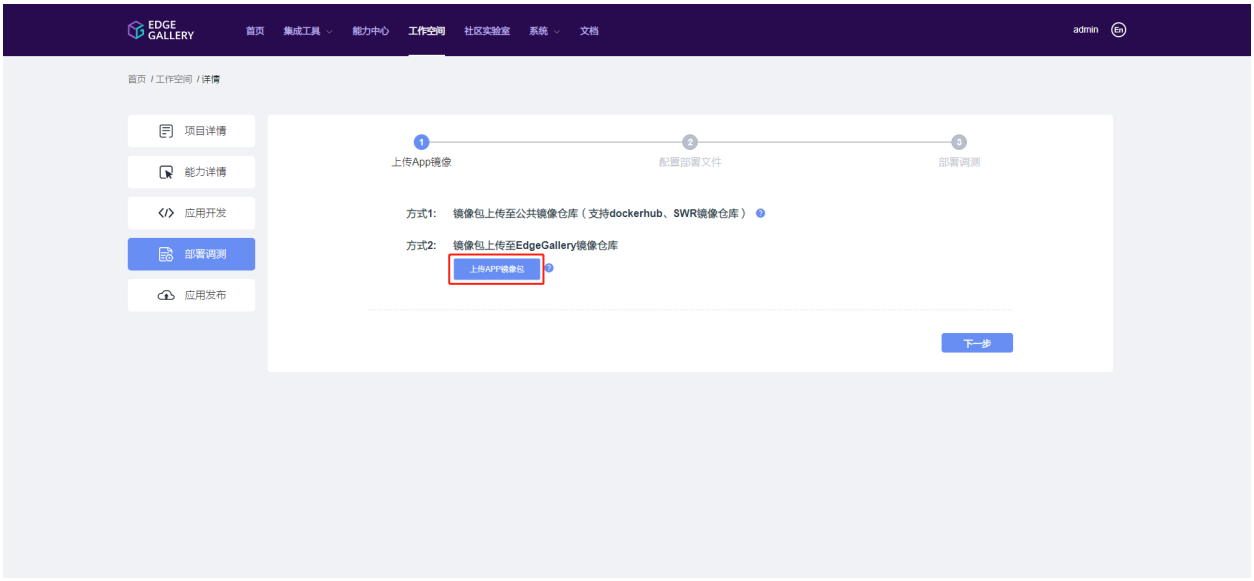
1.4.1 开发准备

- 新建一个项目

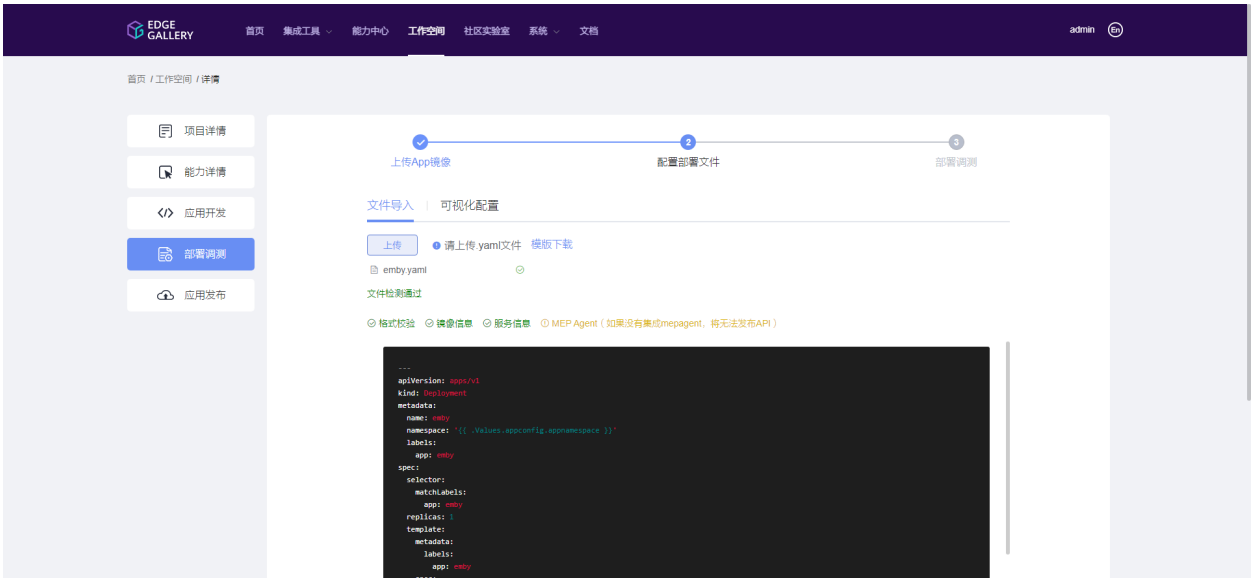
- 选择依赖的平台能力，勾选“服务发现”

1.4.2 部署调测

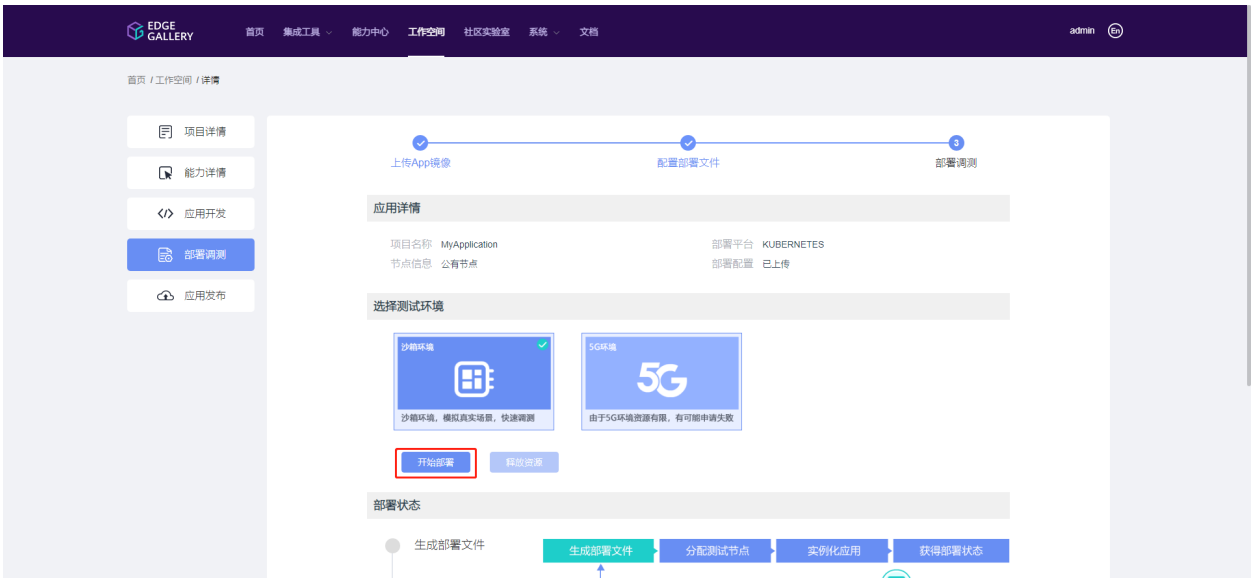
第一步上传 App 镜像



第二步上传 yaml 文档

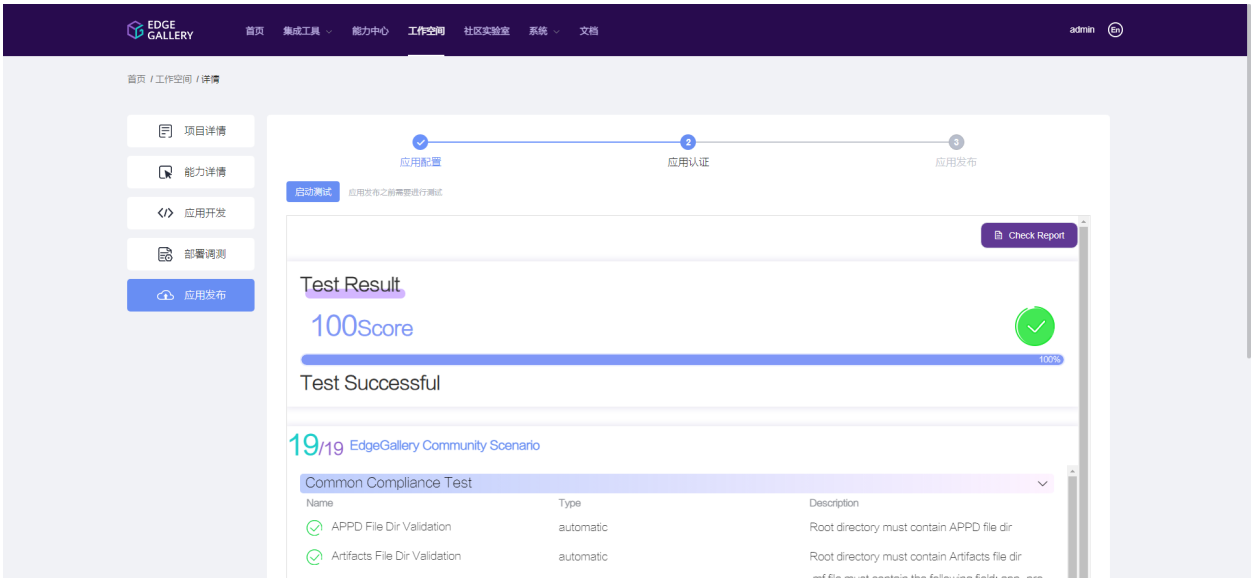


第三步部署调测

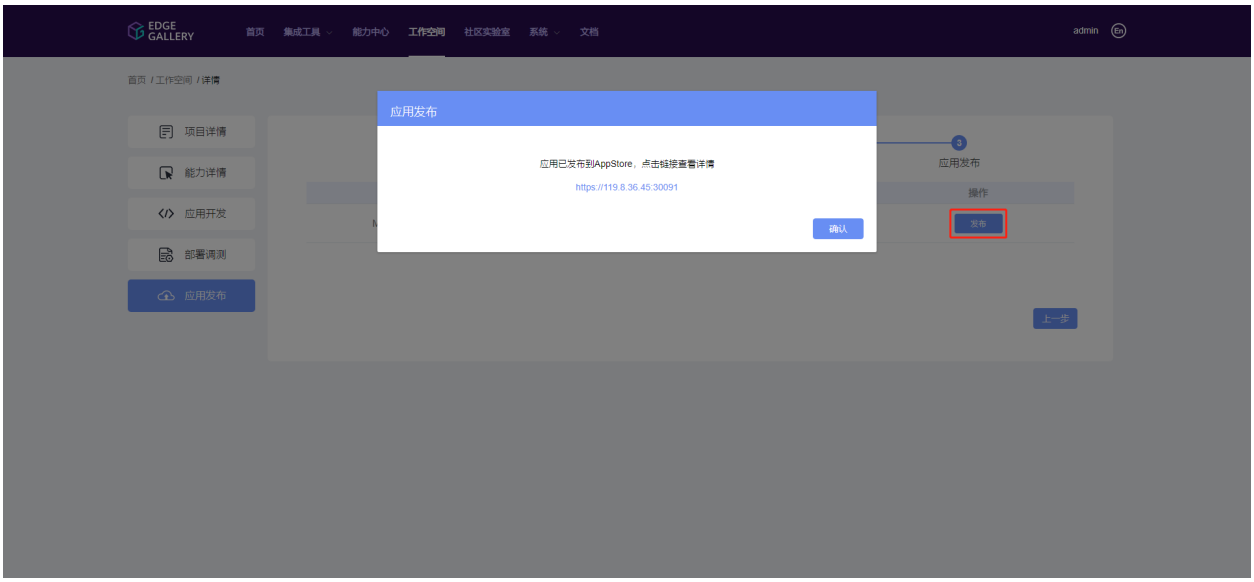


1.4.3 应用发布

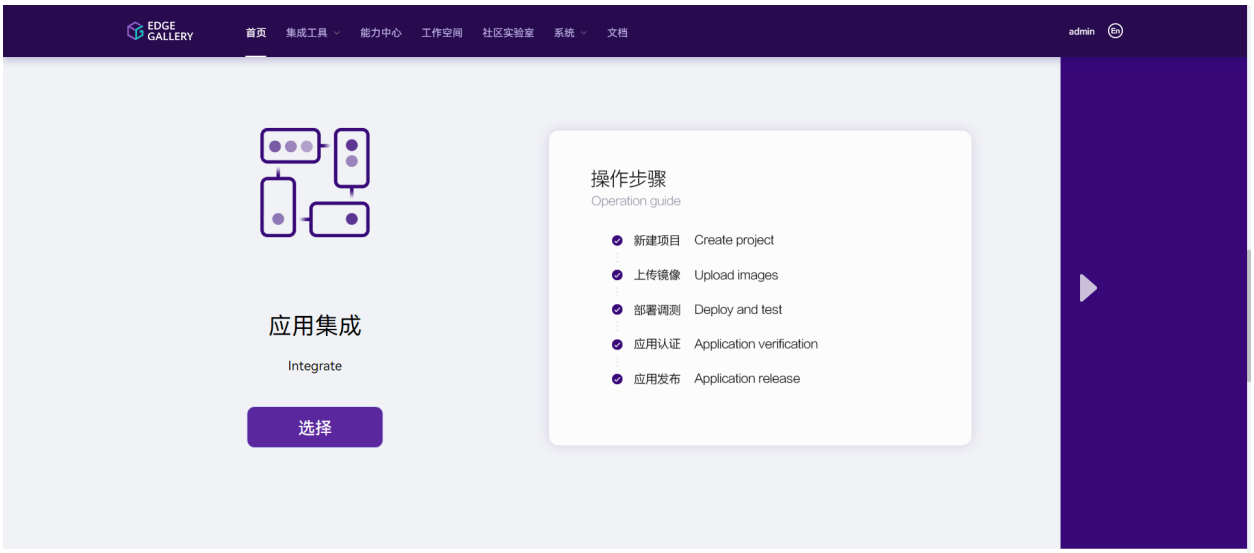
第一步 ATP 测试



第二步应用发布

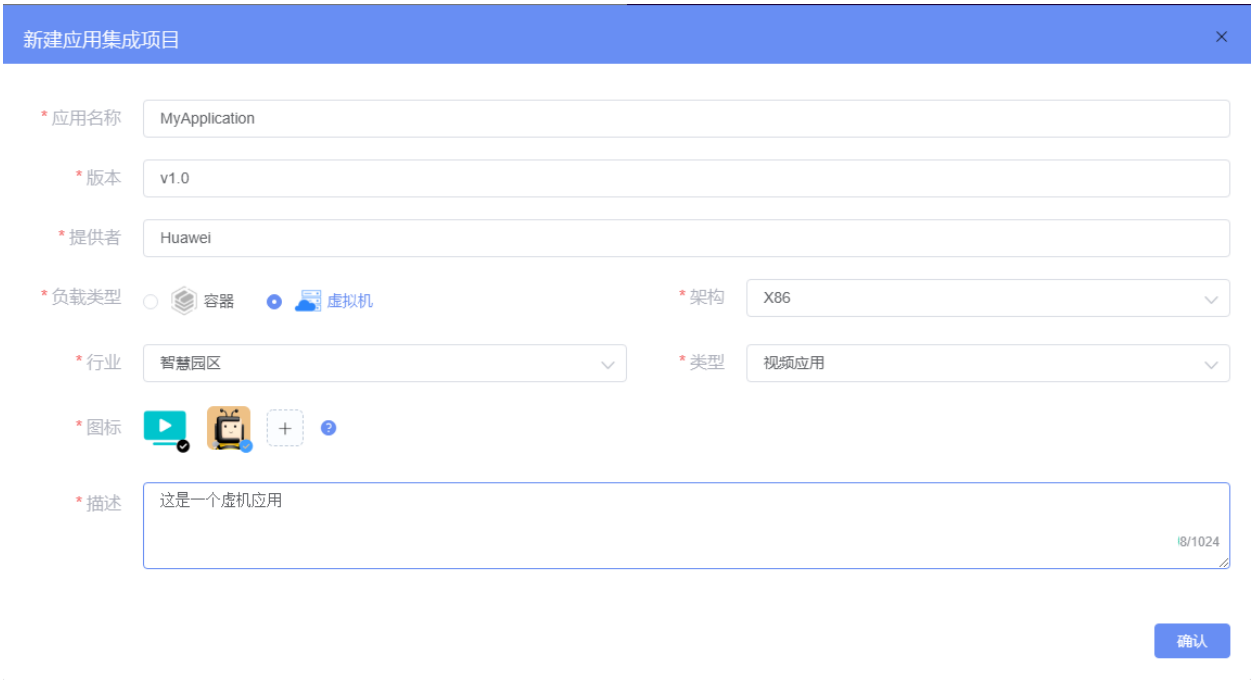


1.5 开发者平台虚机部署流程介绍



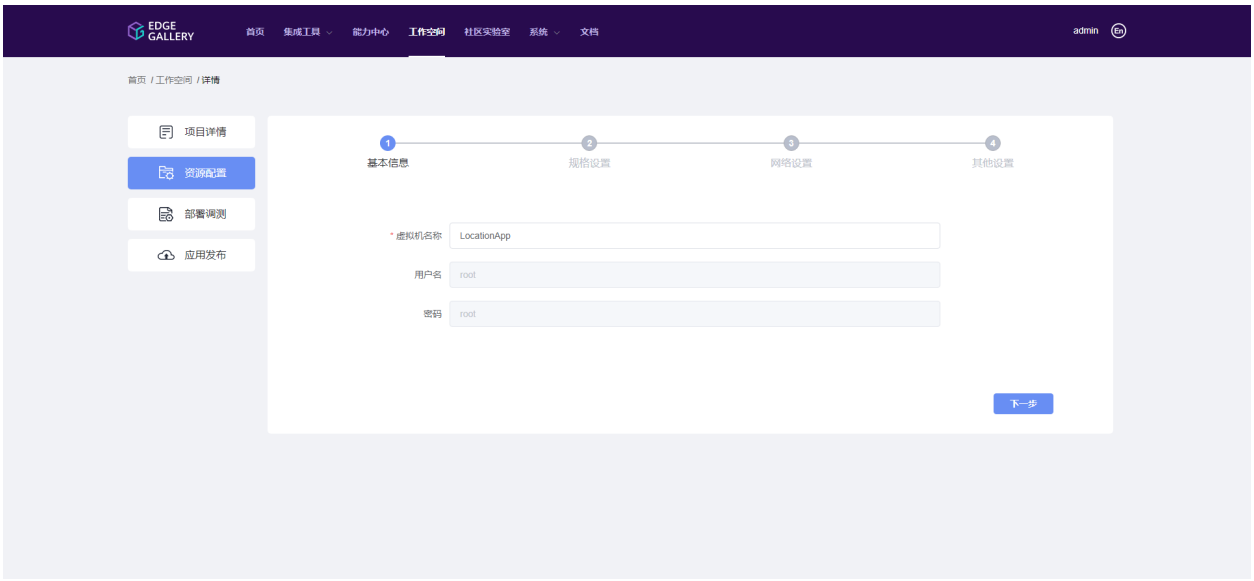
1.5.1 开发准备

新建一个项目

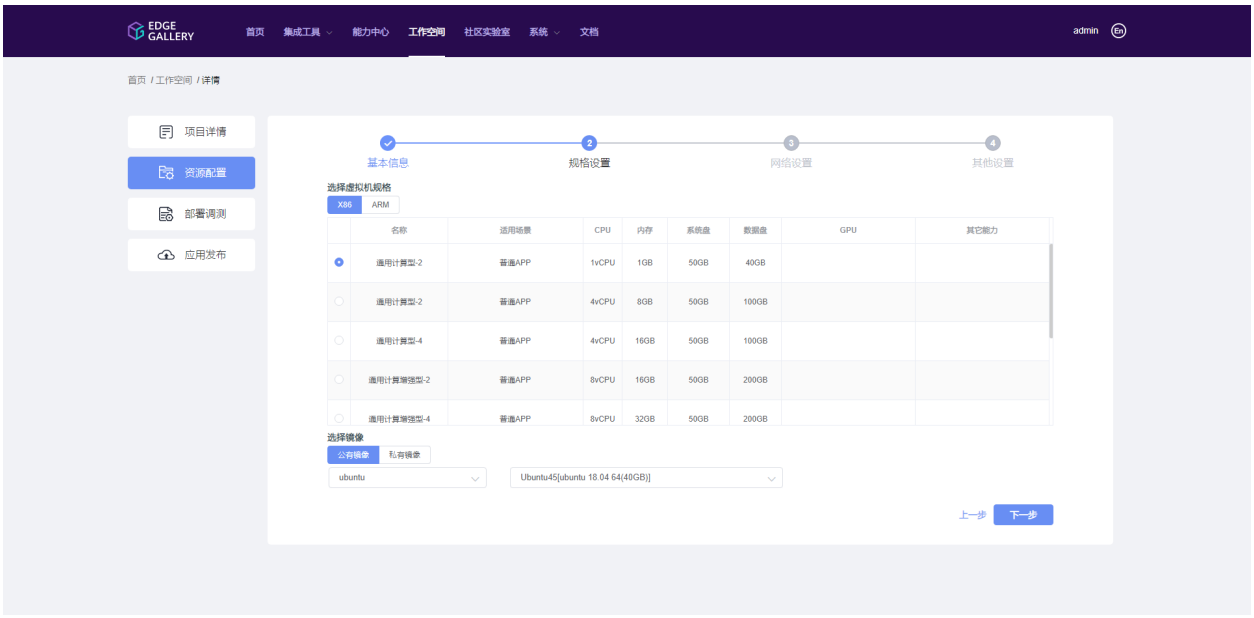


1.5.2 资源配置

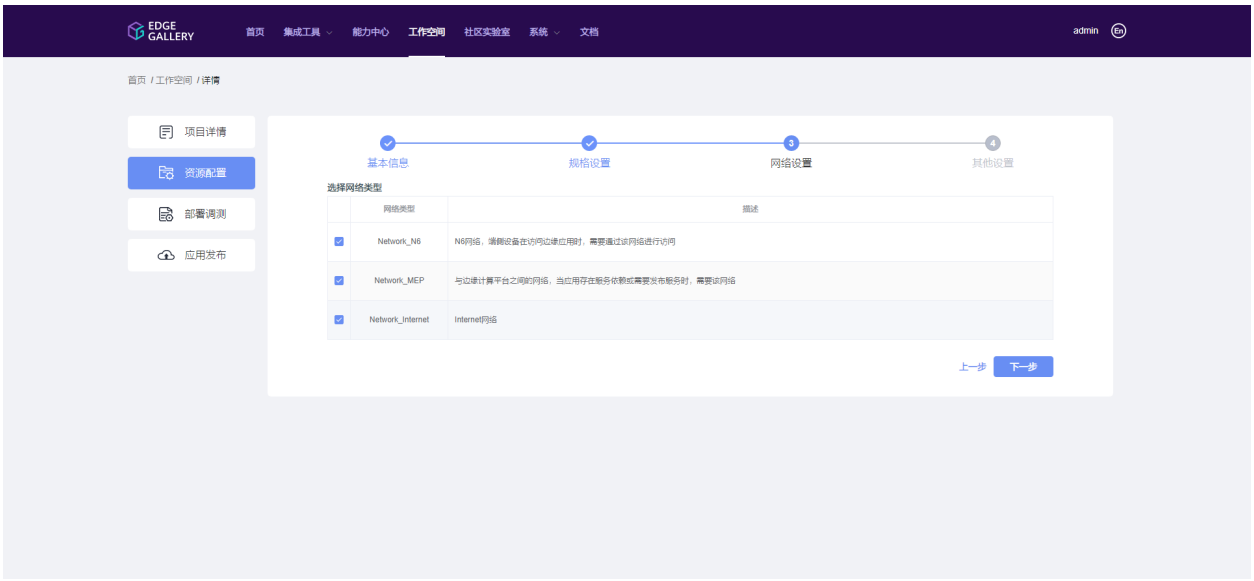
第一步基本信息配置



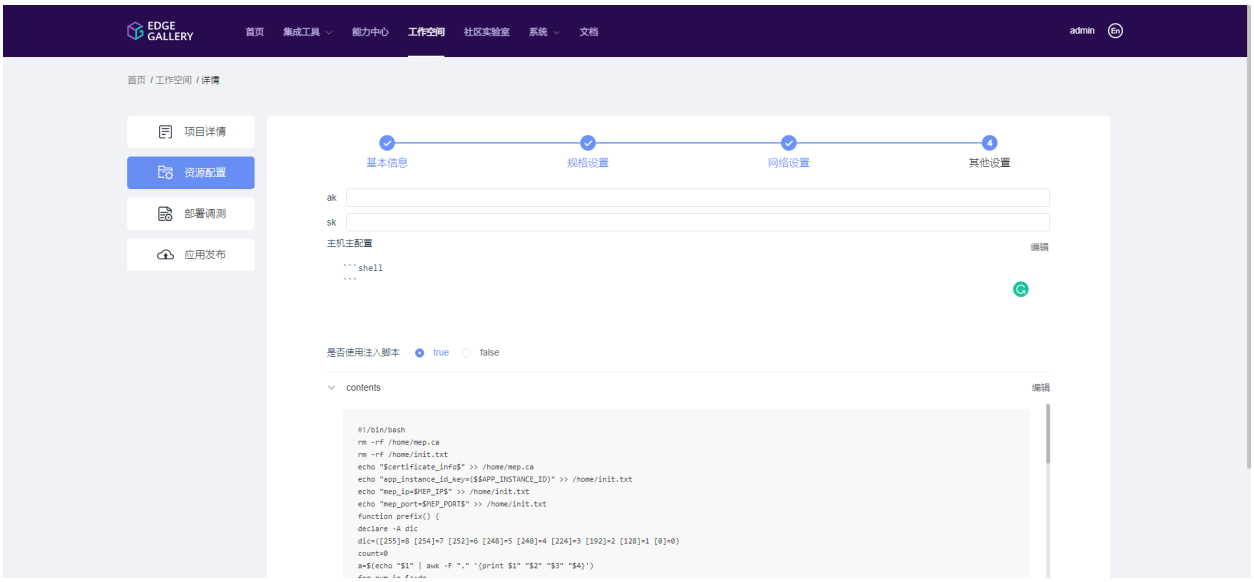
第二步规格设置



第三步网络设置

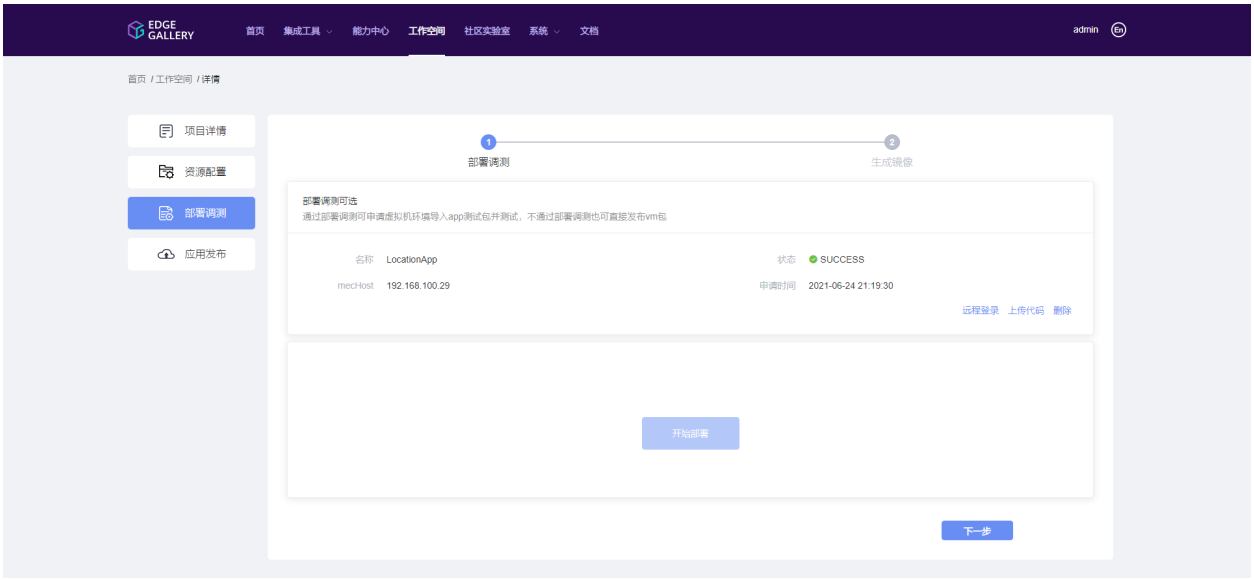


第四步其他设置



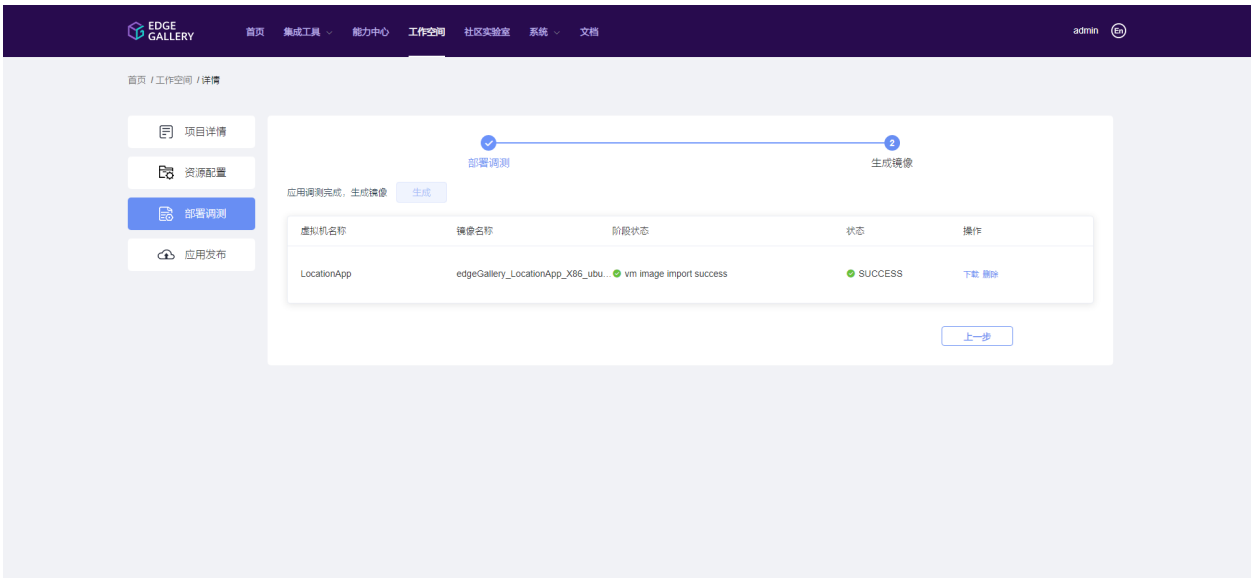
1.5.3 部署调测

第一步开始部署



部署成功后，可上传应用代码文件，使用 VNC 功能进行运维操作

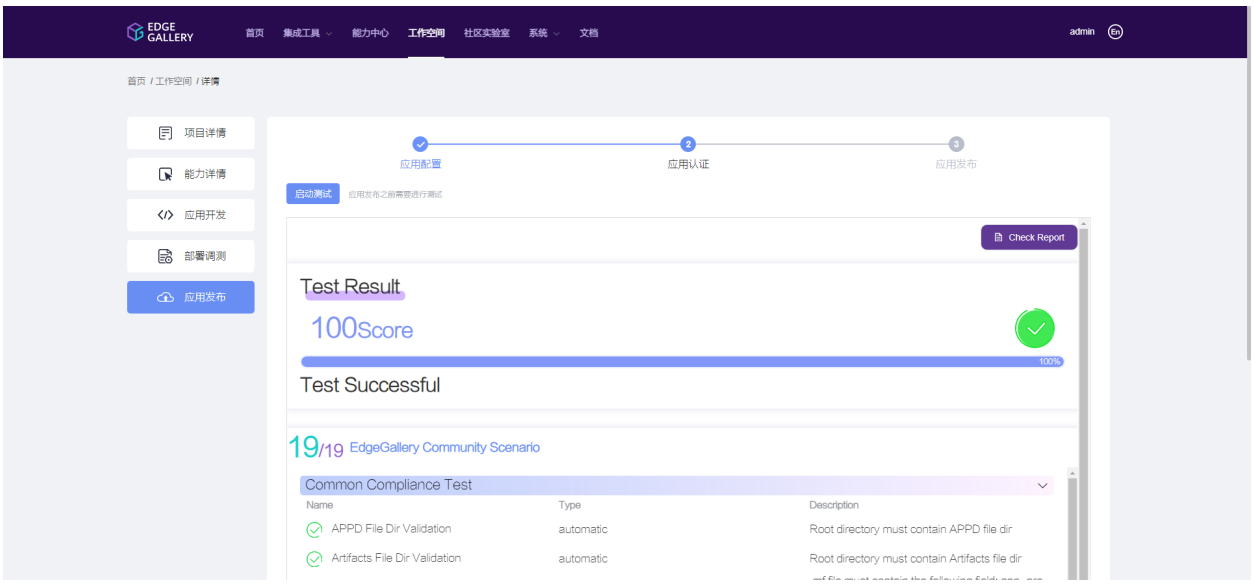
第二步生成镜像



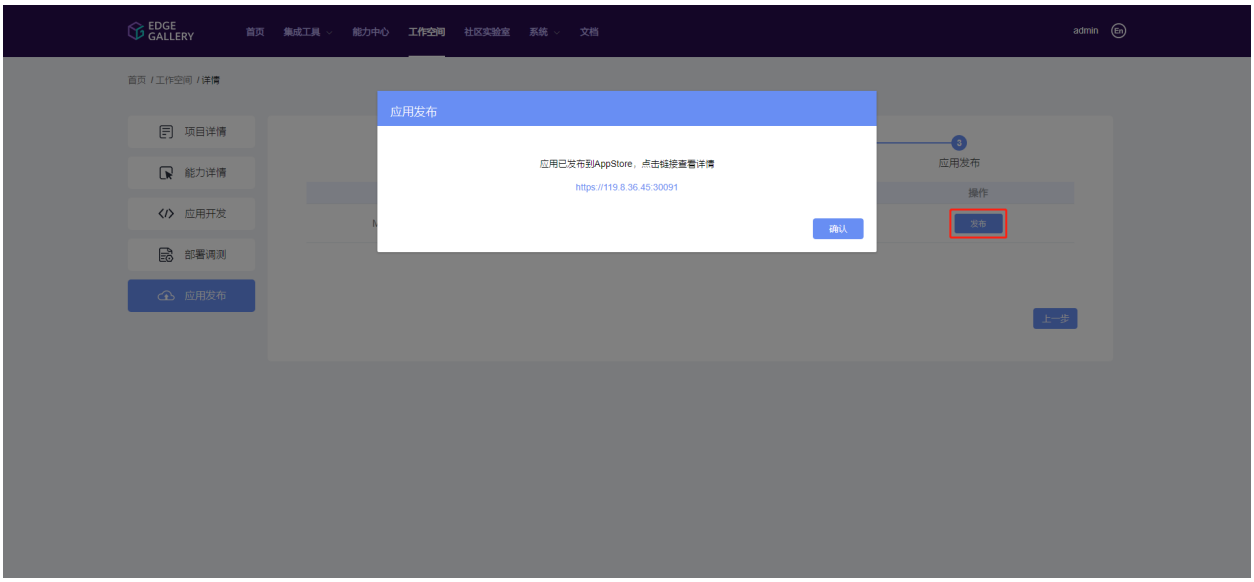
生成镜像成功后，可下载镜像文件

1.5.4 应用发布

第一步 ATP 测试

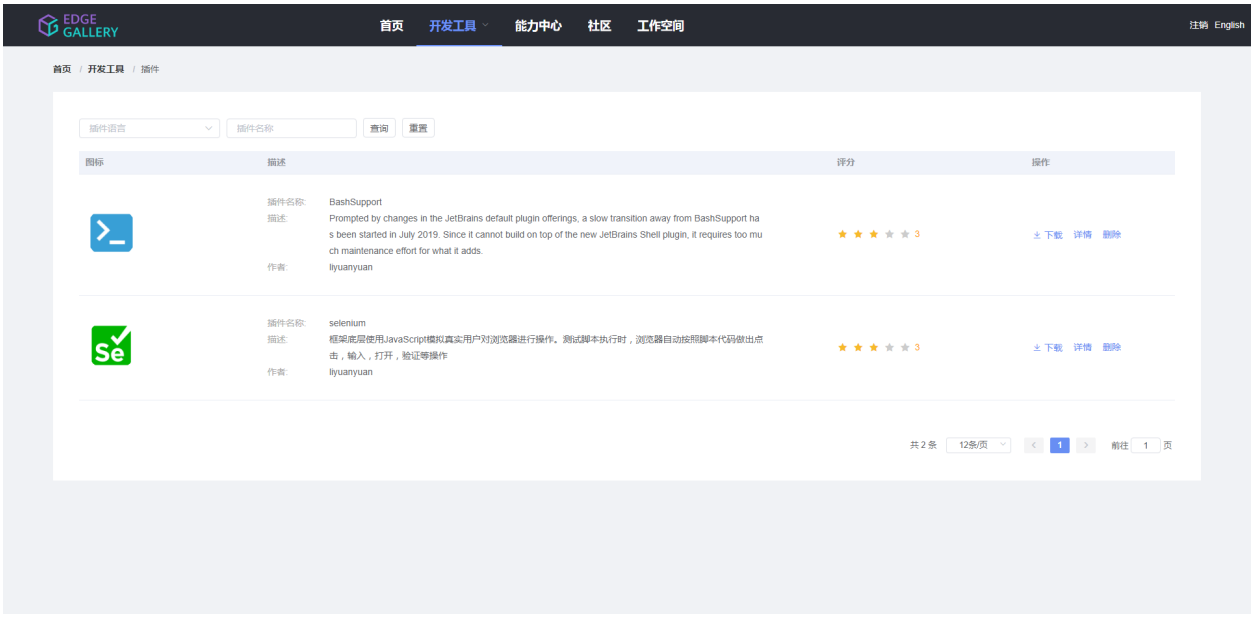


第二步应用发布

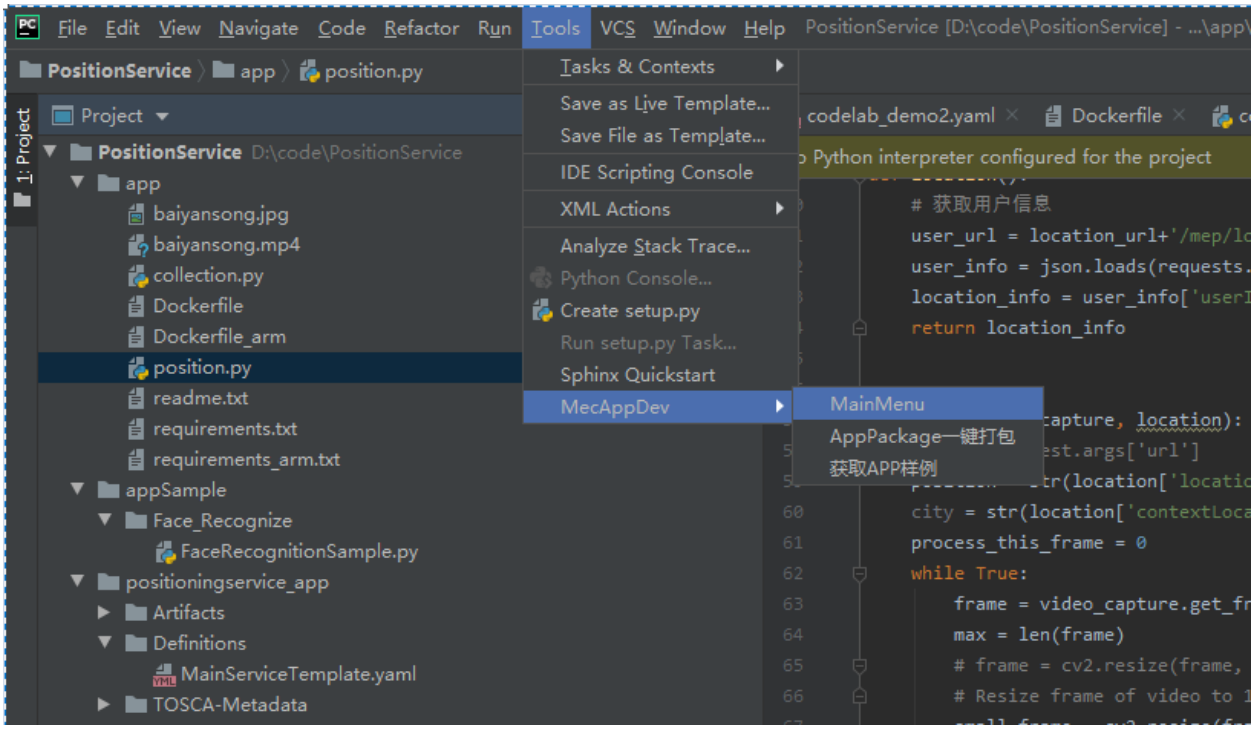


1.5.5 本地开发

- 下载依赖的插件-BashSupport，并导入到本地 IDE



- 执行插件，会自动生成示例代码



- 编写 APP 代码
- 构建 docker image

.. This work is licensed under a Creative Commons Attribution 4.0 International License. .. http://creativecommons.org/licenses/by/4.0 .. Copyright 2019-2021 Huawei Technologies Co., Ltd.

CHAPTER 2

Release Notes

2.1 EdgeGallery Introduction

EdgeGallery 是一个 5G MEC 开源边缘计算平台，它是一个面向应用和应用开发者的端到端解决方案，对于 APP 应用来说除支持应用运行的基础 MEP 平台外，还提供 APP 应用管理系统（MEC Manager，极简管理面，包括 MEPM 以及 MEAO 的部分功能），能够对应用进行端到端生命周期管理，提供 X86 应用向 ARM 迁移的工具链，支持应用平滑的跨平台迁移和调优。对应用开发者来说，提供完整的应用开发 IDE 环境和集成验证环境，为应用开发者提供一站式开发体验，同时通过 APP Store 功能，对应用开发者开发的 APP 提供标准的准入验证流程。

- 项目简介
- 开发者指南

2.2 Version: v1.3.0

- Release Date: 2021-10-12
- Image Version: v1.3.0

新开源仓库列表

None

需求/发布计划详细说明

- **亮点功能**

  - 南向 IoT 设备 Profile Management 完善
    * 边缘侧集成 Kuiper，Flink 等 Stream analyzer 进行数据的清洗：如整形与格式化，上报大数据系统内；也可以通过云边接口上传至公有云进行进一步的数据计算，如机器学习等
    * 集成业界 IoT 协议栈，如 Fledge，EdgeX，KubeEdge 等开源组件，实现 MQTT、OPCUA 等典型协议支持
    * 支持大数据平台接入，支持标准化工业平台配置等
  - 大应用传输体验优化
    * 大包制作效率提升：支持二次大包制作，虚机镜像瘦身 60%，减少应用包制作时间 3-4 小时。
    * Appstore 支持分片下载，即 MEAO 可以通过分片下载、上传机制进行应用包传递，提升应用传输稳定性

- **模块增强**
  - ATP 能力自动化测试能力增强
    * 增强安全性测试含病毒、端口和漏洞扫描，以及时延等性能测试
    * 增加应用使能测试，含服务注册、更新、下线等指标测试
    * 支持测试用例灵活可定义的用例参数配置
  - 界面全面优化升级，提升用户体验
    * Developer、AppStore、MECM 和边缘自治平台界面全面优化，界面风格统一，全面提升用户使用体验
    * 边缘自治界面增加用户管理功能
    * Developer 平台增加“统一界面组件”，帮助开发人员快速参与 EdgeGallery 社区开发
    * MECM 平台增加节点资源和应用资源的展示
  - Developer 平台虚机能力增强
    * 增加应用包签名功能
    * 增加容器镜像管理功能
    * 增加虚机镜像瘦身功能，针对上传镜像和生成的镜像完成自动瘦身，镜像大小减少 60%
  - AppStore 平台能力增强
    * 增加 APPD 转换功能
  - MEP 完全实现 ETSI 接口规范
  - Edge AI 能力全面优化
    * 集成 Tensorflow Lite，MindSpore Lite，增强 EdgeGallery AI 相关框架能力
  - 安全增强

        * 对所有流量添加了访问频率限制，短时间访问频率达到限定值会禁止访问，极大降低了遭受 DOS 攻击和暴力破解的风险

- **其他**
  - 应用创新: 完成 150+ 应用集成、落地 30+ 创新基地
  - 体验优化: AppStore、Developer、MECM各平台界面全面升级优化
  - 1.4 功能预告
    * 通过 wasmEdge 进行 AI 能力大幅瘦身以及定制优化
    * E2E 边缘应用自助订购以及开发集成优化
    * 边缘自治系统自助服务增强
      · 支持企业用户查看资源使用，应用运行以及 upf 网络连接等状态

Bug Fixes

参见 EdgeGallery v1.3.0 Test Report

Security Notes

Fixed/Known Security Issues

参见 EdgeGallery v1.3.0 Security Test Report

Known Vulnerabilities in Used Modules

参见 EdgeGallery v1.3.0 Security Test Report

EdgeGallery v1.3.0 版本第三方开源组件存在以下漏洞需要用户自行修复:

User-mgmt

Component	Version	CVE
vertx-core	3.9.4	CVE-2019-17640
foundation-ssl	1.3.2	CVE-2004-0009 CVE-2021-21501
mybatis-spring	2.0.6	CVE-2020-26945
mybatis-spring-boot-starter	2.2.0	CVE-2020-26945
spring-core	5.2.8.RELEASE	CVE-2021-22118
spring-security-core	5.3.10.RELEASE	CVE-2018-1258
spring-security-core	5.3.8.RELEASE	CVE-2018-1258
tomcat-embed-core	9.0.44	CVE-2021-30639

Appstore

Component	Version	CVE
log4j	1.2.17	CVE-2019-17571
netty-transport	4.1.36.Final	CVE-2019-20444 CVE-2019-20445 CVE-2019-16869 CVE-2020-11612
vertx-core	3.9.4	CVE-2019-17640
docker-java-core	3.2.10	CVE-2019-13139 CVE-2019-13509 CVE-2019-16884 CVE-2019-5736
foundation-ssl	1.3.2	CVE-2004-0009 CVE-2021-21501
mybatis-spring	2.0.6	CVE-2020-26945
mybatis-spring-boot-starter	2.2.0	CVE-2020-26945
spring-jdbc	5.1.8.RELEASE	CVE-2020-5398
spring-security-core	5.1.5.RELEASE	CVE-2018-1258

Developer

Component	Version	CVE
netty-transport	4.1.36.Final	CVE-2019-20444 CVE-2019-20445 CVE-2019-16869 CVE-2020-11612
vertx-core	3.9.4	CVE-2019-17640
swagger-codegen-cli	3.0.21	CVE-2018-8088 CVE-2016-6199 CVE-2019-15052 CVE-2020-25649 CVE-2019-10086 CVE-2017-1000207 CVE-2017-1000208 CVE-2017-18640 CVE-2020-11979 CVE-2021-29428
commons-compress	1.19	CVE-2021-35515 CVE-2021-35516 CVE-2021-35517 CVE-2021-36090
docker-java-core	3.2.10	CVE-2019-13139 CVE-2019-13509 CVE-2019-16884 CVE-2019-5736
foundation-ssl	1.3.2	CVE-2004-0009 CVE-2021-21501
mybatis-spring	2.0.6	CVE-2020-26945
mybatis-spring-boot-starter	2.2.0	CVE-2020-26945
sprint-jdbc	5.2.8.RELEASE	CVE-2021-22118
spring-security-core	5.3.10.RELEASE	CVE-2018-1258
tomcat-embed-core	9.0.44	CVE-2021-30639

MECM

Component	Version	CVE
bcprov-jdk15on-1.56.jar	1.56	CVE-2018-1000613 CVE-2018-1000180
netty-transport-native-epoll	4.1.36.Final	CVE-2019-20444 CVE-2019-20445 CVE-2019-16869 CVE-2020-11612
log4j	1.2.17	CVE-2019-17571
vertx-core	3.6.3	CVE-2019-17640
foundation-ssl	1.3.0	CVE-2004-0009
hibernate-core	5.3.10.Final	CVE-2020-25638
netty-transport-native-epoll	4.1.36.Final	CVE-2019-20444 CVE-2019-20445 CVE-2019-16869 CVE-2020-11612
spring-security-core	5.3.8.RELEASE	CVE-2018-1258
spring-tx	5.1.8.RELEASE	CVE-2020-5398
tomcat-embed-core	9.0.21	CVE-2020-1938 CVE-2019-12418 CVE-2019-17563 CVE-2020-11996 CVE-2020-13934 CVE-2020-13935 CVE-2020-17527 CVE-2020-8022 CVE-2020-9484 CVE-2021-25122 CVE-2021-25329
jackson-databind	2.9.9	CVE-2019-14379 CVE-2019-14540 CVE-2019-14892 CVE-2019-14893 CVE-2019-16335 CVE-2019-16942 CVE-2019-16943 CVE-2019-17267 CVE-2019-17531 CVE-2019-20330 CVE-2020-8840 CVE-2020-9546 CVE-2020-9547 CVE-2020-9548 CVE-2019-14439 CVE-2020-10672 CVE-2020-10673 CVE-2020-10968 CVE-2020-10969 CVE-2020-11111 CVE-2020-11112 CVE-2020-11113 CVE-2020-11619 CVE-2020-11620 CVE-2020-14060 CVE-2020-14061 CVE-2020-14062 CVE-2020-14195 CVE-2020-24616 CVE-2020-24750 CVE-2020-25649 CVE-2020-35490 CVE-2020-35491 CVE-2020-35728 CVE-2020-36179 CVE-2020-36180 CVE-2020-36181 CVE-2020-36182 CVE-2020-36183 CVE-2020-36184 CVE-2020-36185 CVE-2020-36186 CVE-2020-36187 CVE-2020-36188 CVE-2020-36189 CVE-2020-36190
mybatis	3.4.4	CVE-2020-26945

ATP

Component	Version	CVE
netty-transport	4.1.36.Final	CVE-2019-20444 CVE-2019-20445 CVE-2019-16869 CVE-2020-11612
vertx-core	3.9.4	CVE-2019-17640
commons-compress	1.19	CVE-2021-35515 CVE-2021-35516 CVE-2021-35517 CVE-2021-36090
foundation-ssl	1.3.2	CVE-2004-0009 CVE-2021-21501
mybatis-spring	2.0.6	CVE-2020-26945
mybatis-spring-boot-starter	2.2.0	CVE-2020-26945
spring-core	5.1.8.RELEASE	CVE-2020-5398
spring-security-core	5.1.5.RELEASE	CVE-2018-1258 CVE-2021-22112 CVE-2018-1258 CVE-2021-22112
tomcat-embed-core	9.0.44	CVE-2021-30639

MEP

Component	Version	CVE
golang.org/x/net	v0.0.0-20200301022130	CVE-2021-33194

2.3 Version: v1.2.0

- Release Date: 2021-07-07
- Image Version: v1.2.0

新开源仓库列表

模块	名称	类型	URL	说明
installer	installer	配置仓	https://gitee.com/edgegallery/installer	EdgeGallery 离线包制作以及离线安装、在线安装、docker compose 方式安装脚本
filesystem	filesystem	产品仓	https://gitee.com/edgegallery/filesystem	EdgeGallery 平台内的应用镜像管理服务

New Features

需求/发布计划详细说明

- **开放能力**
  - AI 软硬件能力
    * 支撑🄴腾/Atlas 硬件以及开放 API：支持🄴腾 20+ 开放能力，包括图像修复、目标定位、超分辨图像算法等
    * AI 换脸、OCR、智慧货架、姿态识别等
  - 行业套件 支持行业组合能力
    * PCB 质量检测：提供开放能力和 PCB 板质量检测的样例应用
    * 工业现场南向设备管理通信：提供基于 Fledge 的工业南向设备管理方案以及部署脚本
    * 视频会议、在线课堂：提供视频开源样例应用
- **安全增强**
  - 用户管理模块增加动态验证码校验功能，解决恶意注册和邮件轰炸等安全漏洞问题
  - 应用调测过程中，VNC 远程登录沙箱环境设置非 root 用户操作，限制访问权限
- **新增模块特性**
  - Developer、AppStore 和 MECM 平台增加管理员操作界面
  - AppStore 平台提供应用在线体验功能
  - 全流程支持虚机应用的快速集成、测试和发布，MECM 平台支持虚机应用的分发和部署
  - 虚机应用集成能力增强，支持虚机规格配置、网络配置和环境变量配置
  - 增加镜像文件管理服务，解决 EdgeGallery 中心节点模块间镜像文件共享问题
  - 提供边缘自治管理界面和边缘节点健康检查功能
  - 全模块支持数据持久化能力
- **按需部署**
  - 提供基于 Ansible 的离线部署脚本与离线安装包，支持用户进行自定义的按需部署
    * IaaS 层提供 k8s 与 k3s 两种部署方式
    * EdgeGallery 各子模块支持单独部署与卸载
    * 支持基于 k8s NFS 的数据持久化功能
  - 支持以 docker-compose 方式（不依赖 k8s/k3s）部署 EdgeGallery 各子模块
  - [PoC] 支持树莓派部署
- **其他**
  - 应用创新: 完成 130+ 应用集成、落地 25+ 创新基地

  - 体验优化: AppStore、Developer、MECM各平台界面全面升级优化

Bug Fixes

参见 EdgeGallery v1.2.0 Test Report

总缺陷数量：267（含 34 个安全类缺陷），已解决数量：265 个；未解决 bug 16 个（含 16 个 known issue，遗留到下一版本解决）

- 按缺陷类型分析：安全类 bug 35 个；其他类 bug 246 个。
- 按缺陷严重级别分析：严重 bug 31 个；主要 bug 70 个；次要 bug 129 个；不重要 bug 9 个；不指定 42 个。
- 按需求点分析
- 缺陷收敛性分析：总计 281 个 bug（含 14 个无效 bug；含安全类 bug 34）；迭代一 39 个 bug（含 3 个无效 bug）；迭代二 65 个 bug（含 5 个无效 bug）；迭代三 143 个 bug（含 4 个无效 bug）；未制定迭代 34 个 bug（含 2 个无效 bug）

Security Notes

参见 EdgeGallery v1.2.0 Security Test Report

Fixed/Known Security Issues

No.	Issue 标题	状态	级别
1	【mecm】【Security】lcm，rule，apm，appo，inventory，k8xplugin 等 pod 中 usr/app/log 目录下没有日志文件	已修复	主要
2	【user-mgmt】【Security】user-mgmt pod 中 usr/app/log 目录下没有日志文件	**遗留**	主要
3	【developer】【Security】developer-be 中 usr/app/log 目录下没有日志文件	已修复	主要
4	【user-mgmt】【Security】日志记录敏感信息（ssoSessionId）	已修复	主要
5	【developer】【Security】虚机部署时，日志中记录了敏感信息（lcmToken）	已修复	主要
6	【atp】【Security】用户贡献测试用例功能存在重放攻击漏洞	已修复	主要
7	【user-mgmt】【Security】首次登陆 admin 账户未强制修改密码	已修复	主要
8	【atp】【Security】在 appstore 进行 atp 测试-> 社区场景-> 安全测试用例执行失败	已修复	主要
9	【developer】【Security】新增沙箱环境时，鉴权应按角色判断，不应按 userId 判断	已修复	主要

Known Vulnerabilities in Used Modules

EdgeGallery v1.2.0 版本第三方开源组件存在以下漏洞需要用户自行修复：

User-mgmt

Component	Version	CVE
vertx-core	3.9.4	CVE-2019-17640
foundation-ssl	1.3.0	CVE-2004-0009
mybatis-spring	2.0.6	CVE-2020-26945
mybatis-spring-boot-starter	2.1.4	CVE-2020-26945
spring-security-core	5.3.8.RELEASE	CVE-2018-1258

Appstore

Component	Version	CVE
log4j	1.2.17	CVE-2019-17571
netty-transport	4.1.36.Final	CVE-2019-20444 CVE-2019-20445 CVE-2019-16869 CVE-2020-11612
vertx-core	3.9.4	CVE-2019-17640
docker-java-core	3.2.7	CVE-2019-13139 CVE-2019-13509 CVE-2019-16884 CVE-2019-5736
foundation-ssl	1.3.0	CVE-2004-0009
mybatis-spring	2.0.6	CVE-2020-26945
mybatis-spring-boot-starter	2.1.4	CVE-2020-26945
spring-jdbc	5.1.8.RELEASE	CVE-2020-5398
spring-security-core	5.1.5.RELEASE	CVE-2018-1258

Developer

Component	Version	CVE
netty-transport	4.1.36.Final	CVE-2019-20444 CVE-2019-20445 CVE-2019-16869 CVE-2020-11612
vertx-core	3.9.4	CVE-2019-17640
swagger-codegen-cli	3.0.21	CVE-2018-8088 CVE-2016-6199 CVE-2019-15052 CVE-2020-25649 CVE-2019-10086 CVE-2017-1000207 CVE-2017-1000208 CVE-2017-18640 CVE-2020-11979 CVE-2021-29428
docker-java-core	3.2.7	CVE-2019-13139 CVE-2019-13509 CVE-2019-16884 CVE-2019-5736
foundation-ssl	1.3.0	CVE-2004-0009
mybatis-spring	2.0.6	CVE-2020-26945
mybatis-spring-boot-starter	2.1.4	CVE-2020-26945
spring-security-core	5.3.8.RELEASE	CVE-2018-1258
sprint-jdbc	5.2.8.RELEASE	CVE-2021-22118

MECM

Component	Version	CVE
bcprov-jdk15on-1.56.jar	1.56	CVE-2018-1000613 CVE-2018-1000180
log4j	1.2.17	CVE-2019-17571
vertx-core	3.6.3	CVE-2019-17640
foundation-ssl	1.3.0	CVE-2004-0009
hibernate-core	5.3.10.Final	CVE-2020-25638
netty-transport	4.1.36.Final	CVE-2019-20444 CVE-2019-20445 CVE-2019-16869 CVE-2020-11612
spring-security-core	5.3.8.RELEASE	CVE-2018-1258
spring-tx	5.1.8.RELEASE	CVE-2020-5398
tomcat-embed-core	9.0.21	CVE-2020-1938 CVE-2019-12418 CVE-2019-17563 CVE-2020-11996 CVE-2020-13934 CVE-2020-13935 CVE-2020-17527 CVE-2020-8022 CVE-2020-9484 CVE-2021-25122 CVE-2021-25329
jackson-databind	2.9.9	CVE-2019-14379 CVE-2019-14540 CVE-2019-14892 CVE-2019-14893 CVE-2019-16335 CVE-2019-16942 CVE-2019-16943 CVE-2019-17267 CVE-2019-17531 CVE-2019-20330 CVE-2020-8840 CVE-2020-9546 CVE-2020-9547 CVE-2020-9548 CVE-2019-14439 CVE-2020-10672 CVE-2020-10673 CVE-2020-10968 CVE-2020-10969 CVE-2020-11111 CVE-2020-11112 CVE-2020-11113 CVE-2020-11619 CVE-2020-11620 CVE-2020-14060 CVE-2020-14061 CVE-2020-14062 CVE-2020-14195 CVE-2020-24616 CVE-2020-24750 CVE-2020-25649 CVE-2020-35490 CVE-2020-35491 CVE-2020-35728 CVE-2020-36179 CVE-2020-36180 CVE-2020-36181 CVE-2020-36182 CVE-2020-36183 CVE-2020-36184 CVE-2020-36185 CVE-2020-36186 CVE-2020-36187 CVE-2020-36188 CVE-2020-36189 CVE-2020-36190
mybatis	3.4.4	CVE-2020-26945

ATP

Component	Version	CVE
netty-transport	4.1.36.Final	CVE-2019-20444 CVE-2019-20445 CVE-2019-16869 CVE-2020-11612
vertx-core	3.9.4	CVE-2019-17640
foundation-ssl	1.3.0	CVE-2004-0009
mybatis-spring	2.0.6	CVE-2020-26945
mybatis-spring-boot-starter	2.1.4	CVE-2020-26945
spring-security-core	5.1.5.RELEASE	CVE-2018-1258
spring-core	5.1.8.RELEASE	CVE-2020-5398

MEP

Component	Version	CVE
golang.org/x/net	v0.0.0-20200301022130	CVE-2021-33194

2.4 Version: v1.1.0

- Release Date: 2020-04-09
- Image Version: v1.1.0

新开源仓库列表

模块	名称	类型	URL	说明
Example Applications	example-apps	PoC 仓	https://gitee.com/edgegallery/example-apps	提供基于 EdgeGallery 的样例 app
EdgeGallery Frontend Component	eg-view	产品仓	https://gitee.com/edgegallery/eg-view	提供归一化的 EdgeGallery 前端组件
EdgeGallery Testing Platform	edgeT	PoC 仓	https://gitee.com/edgegallery/edgeT	为用户提供本地测试的工具链

New Features

需求/发布计划详细说明

EDGE GALLERY
EdgeGallery V1.1
Dove Release
8大版本特性
AI硬件能力增强
1
支持昇腾硬件和应用
支持Atlas硬件和应用
开放API：AI图像修复、边缘检测、卡通图像生成、图像上色、对象分类、目标检测
跨平台支持
2
支持基于Openstack/fushionsphere的虚机应用和基于K8S容器应用的孵化集成
多样化呈现
3
APP Store：APP分类、APP推荐、爆款动态刷新
EdgeNative架构
4
IaaS/PaaS/应用都支持按需部署、应用框架自动生成
MEP微服务管理架构可视化
提供EdgeGallery小程序、设计态等
API管理
5
ETSI MEC API能力支持情况匹配
3GPP CAPIF框架遵从支持情况匹配
提供昇腾能力支持
安全 用户管理
6
分权分域，对关键操作和数据进行权限控制
应用远程登录（如VNC）支持安全协议登录
社区实验室增强
7
提供真实E2E 5G MEC验证环境
资源一键申请、一键释放、可视化管理
应用生态
8
完成80+应用集成
合作伙伴基于EdgeGallery落地20+创新基地
AppStore联邦，可注册外部AppStore
应用在不同AppStore间可相互推送和拉取
扫码进入代码托管地址
获取新版本更多信息

- **AI 软硬件能**支撑🄴腾/Atlas 硬件以及开放 API
  - AI 图像修复、边缘检测、卡通图像生成、图像上色、对象分类、目标检测等；
- **跨平台支持**支持基于 Openstack 的虚机应用和基于 K8S 容器应用的孵化集成
- **体验优化**
  - APP 分类、APP 推荐、爆款动态刷新
- **EdgeNative 支持**
  - IaaS/PaaS/应用都支持按需部署、应用框架自动生成
  - MEP 微服务管理架构可视化
  - 提供 EdgeGallery 小程序、设计态等
- **API 管理** ETSI MEC API、3GPP CAPIF 以及🄴腾能力支持情况匹配
- **安全、用户管理**
  - 分权分域，对关键操作和数据进行权限控制
  - 应用远程登录（如 VNC）支持安全协议登录
- **社区实验室增强**
  - 提供真实 E2E 5G 验证环境
  - 资源一键申请，一键释放，可视化管理
- **应用创新**
  - AppStore 联邦，可注册外部 AppStore，应用在不同 AppStore 间可相互推送和拉取
  - 完成 80+ 应用集成、落地 17+ 创新基地、西安电子科技大学研究生课程

Bug Fixes

参见 EdgeGallery v1.1.0 Test Report

Known Issues

参见 EdgeGallery v1.1.0 Test Report

Security Notes

Fixed Security Issues

参见 EdgeGallery v1.1.0 Test Report

Known Security Issues

参见 EdgeGallery v1.1.0 Security Test Report

Known Vulnerabilities in Used Modules

EdgeGallery v1.1.0 版本第三方开源组件存在以下漏洞需要用户自行修复：

Appstore

Component	Version	CVE
log4j	1.2.17	CVE-2019-17571
netty-transport	4.1.36.Final	CVE-2019-20444 CVE-2019-20445 CVE-2019-16869 CVE-2020-11612
vertx-core	3.9.4	CVE-2019-17640
docker-java-core	3.2.7	CVE-2019-13139 CVE-2019-13509 CVE-2019-16884 CVE-2019-5736
foundation-ssl	1.3.0	CVE-2004-0009
mybatis-spring	2.0.6	CVE-2020-26945
mybatis-spring-boot-starter	2.1.4	CVE-2020-26945
spring-jdbc	5.1.8.RELEASE	CVE-2020-5398
spring-security-core	5.1.5.RELEASE	CVE-2018-1258

Developer

Component	Version	CVE
netty-transport	4.1.36.Final	CVE-2019-20444 CVE-2019-20445 CVE-2019-16869 CVE-2020-11612
vertx-core	3.9.4	CVE-2019-17640
swagger-codegen-cli	3.0.21	CVE-2018-8088 CVE-2016-6199 CVE-2019-15052 CVE-2020-25649 CVE-2019-10086 CVE-2017-1000207 CVE-2017-1000208 CVE-2017-18640 CVE-2020-11979
docker-java-core	3.2.7	CVE-2019-13139 CVE-2019-13509 CVE-2019-16884 CVE-2019-5736
foundation-ssl	1.3.0	CVE-2004-0009
mybatis-spring	2.0.6	CVE-2020-26945
mybatis-spring-boot-starter	2.1.4	CVE-2020-26945
spring-security-core	5.1.5.RELEASE	CVE-2018-1258

ATP

Component	Version	CVE
netty-transport	4.1.36.Final	CVE-2019-20444 CVE-2019-20445 CVE-2019-16869 CVE-2020-11612
vertx-core	3.9.4	CVE-2019-17640
foundation-ssl	1.3.0	CVE-2004-0009
mybatis-spring	2.0.6	CVE-2020-26945
mybatis-spring-boot-starter	2.1.4	CVE-2020-26945
spring-security-core	5.1.5.RELEASE	CVE-2018-1258
spring-core	5.1.8.RELEASE	CVE-2020-5398

User-mgmt

Component	Version	CVE
vertx-core	3.9.4	CVE-2019-17640
foundation-ssl	1.3.0	CVE-2004-0009
mybatis-spring	2.0.6	CVE-2020-26945
mybatis-spring-boot-starter	2.1.4	CVE-2020-26945
spring-security-core	5.3.8.RELEASE	CVE-2018-1258

Upgrade Notes

N/A

Deprecation Notes

N/A

Other

N/A

2.5 Version: v1.0.0

- Release Date: 2020-12-31
- Image Version: v1.0.0

新开源仓库列表

模块	名称	类型	URL	说明
ATP(Application Test Platform)	atp	产品仓	https://gitee.com/edgegallery/atp	应用测试平台，为应用认证提供测试支撑
ATP(Application Test Platform)	atp-fe	产品仓	https://gitee.com/edgegallery/atp-fe	提供应用测试认证平台前台支撑
Developer	crane-framework	产品仓	https://gitee.com/edgegallery/crane-framework	为应用开发提供轻量化，插件是的开发框架
MECM	mecm-apprulemgr	产品仓	https://gitee.com/edgegallery/mecm-apprulemgr	管理面提供应用规则配置下发和管理的能力

New Features

需求/发布计划详细说明

1. 版本规划特性
   - **Developer** 优化 APP 开发测试发布流程，优化 API 界面，集成多语言 SDK
     - 优化首页设计，增加应用开发生命周期流程图；
     - 优化项目开发流程，全新界面设计；
     - 优化部署调测环节，增加可视化展示部署过程展示，增加部署结果 POD 展示；
     - 优化能力中心界面，增加能力分类选择；
     - 应用发布环节对接认证测试平台；
   - **AppStore** 界面易用性优化，EdgeGallery 生态能力优化
   - **MECM** 优化拓扑、应用管理及边缘节点管理，新增获取 MEP 能力功能，AK/SK 配置接口
   - **MEP** 支持 APP 心跳检测，新增获取 MEP 能力接口，AK/SK 配置接口，MEP-agent 新增获取 token 接口
   - **ATP** APP 应用的测试能力增强
   - **Test** 提供健康检查功能，优化安装包，开发测试平台支持测试例管理
2. PoC 特性
   - MEP 服务治理页面展示
   - 引入插件式应用框架Crane-framework
   - AppStore 联邦，可注册外部 AppStore，应用在不同 AppStore 间可相互推送和拉取

PoC Test Report

参见EdgeGallery v1.0.0 PoC Test Report

Bug Fixes

参见 EdgeGallery v1.0.0 Test Report

Known Issues

参见 EdgeGallery v1.0.0 Test Report

Security Notes

Fixed Security Issues

参见 EdgeGallery v1.0.0 Test Report

Known Security Issues

参见 EdgeGallery v1.0.0 Test Report

Known Vulnerabilities in Used Modules

EdgeGalelry v1.0.0 版本第三方开源组件存在以下漏洞需要用户自行修复:

Appstore

Component	Version	CVE
log4j (indirect dependency)	1.2.17	CVE-2019-17571
bcprov-jdk15on-1.66.jar	1.66	CVE-2020-28052
jackson-databind	2.10.5	CVE-2020-25649
spring-security-oauth2	2.3.3.RELEASE	CVE-2018-15758
tomcat-embed-core	9.0.37	CVE-2020-17525

Developer

Component	Version	CVE
spring-beans (indirect dependency)	5.1.8.RELEASE	CVE-2020-5398
spring-security-core	5.3.4.RELEASE	CVE-2018-1258
jackson-databind	2.10.5	CVE-2020-25649
mybatis-spring-boot-starter	2.1.1	CVE-2020-26945
tomcat-embed-core	9.0.37	CVE-2020-17527

ATP

Component	Version	CVE
mybatis-spring-boot-starter	2.1.1	CVE-2020-26945

User-mgmt

Component	Version	CVE
spring-security-core	5.3.4.RELEASE	CVE-2018-1258
bcprov-jdk15on	1.66	CVE-2020-28052
jackson-databind	2.10.5	CVE-2020-25649
mybatis-spring-boot-starter	2.1.1	CVE-2020-26945
spring-security-oauth2	2.3.3.RELEAS	CVE-2018-15758

Website-gateway

Component	Version	CVE
spring-security-core	5.2.6.RELEASE	CVE-2018-1258
jackson-databind	2.10.5	CVE-2020-26945
spring-security-oauth2	2.3.3	CVE-2018-15758
tomcat-embed-core	9.0.37	CVE-2020-17527

MECM

Component	Version	CVE
lodash	4.17.13	CVE-2020-8203
minimatch	3.0.4	NPM-118

Upgrade Notes

N/A

Deprecation Notes

N/A

Other

N/A

2.6 Version: v0.9.1

- Release Date: 2020-11-11
- Image Version: v0.9.1

新开源仓库列表

模块	名称	类型	URL	说明
Developer	api-emulator	产品仓	https://gitee.com/edgegallery/api-emulator	提供基本能力的模拟 api
Mep	Plugins	产品仓	https://gitee.com/edgegallery/plugins	plugins 仓
Developer	toolchain	产品仓	https://gitee.com/edgegallery/toolchain	提供 x86 平台应用上车 ARM 时源码分析功能

New Features

需求/发布计划详细说明

- 安全设计排查问题汇总
- api 模拟器和工具链、plugin 仓库开源
- **MEP/MEP Agent**ak/sk 配置接口

Bug Fixes

参见 EdgeGallery v0.9.1 Test Report

Known Issues

参见 EdgeGallery v0.9.1 Test Report

Security Notes

Fixed Security Issues

参见 EdgeGallery v0.9.1 Test Report

Known Security Issues

参见 EdgeGallery v0.9.1 Test Report

Known Vulnerabilities in Used Modules

EdgeGallery v0.9.1 版本第三方开源组件存在以下漏洞需要用户自行修复：

Appstore

Component	Version	CVE
netty-transport (indirect dependency)	4.1.36.Final	CVE-2019-20444, CVE-2019-20445, CVE-2020-11612, CVE-2019-16869
log4j (indirect dependency)	1.2.17	CVE-2019-17571
spring-jdbc (indirect dependency)	5.1.8.RELEASE	CVE-2020-5398
spring-security-core (indirect dependency)	5.1.5.RELEASE	CVE-2018-1258
bcprov-jdk15on-1.66.jar	1.66	CVE-2020-28052
foundation-ssl	1.3.0	CVE-2004-0009
jackson-databind	2.10.5	CVE-2020-25649
mybatis	3.5.3	CVE-2020-25645
spring-security-oauth2	2.3.3.RELEASE	CVE-2018-15758
tomcat-embed-core	9.0.37	CVE-2020-17525
vertx-core	3.6.3	CVE-2019-17640

Developer

Component	Version	CVE
netty-transport (indirect dependency)	4.1.36.Final	CVE-2019-20444, CVE-2019-20445, CVE-2020-11612, CVE-2019-16869
spring-beans (indirect dependency)	5.1.8.RELEASE	CVE-2020-5398
spring-security-config (indirect dependency)	5.1.5.RELEASE	CVE-2018-1258, CVE-2020-5398
spring-security-core	5.3.4.RELEASE	CVE-2018-1258
lodash	4.17.13	CVE-2020-8203
minimatch	3.0.4	NPM-118
spring-security-oauth2	2.3.3	CVE-2018-15758，CVE-2019-11269，CVE-2018-3778
swagger-codegen-cli (shaded: com.fasterxml.jackson.core.jackson-databind:2.10.3)	3.0.21	CVE-2020-25649
foundation-ssl	1.3.0	CVE-2004-0009
jackson-databind	2.10.5	CVE-2020-25649
mybatis-spring	2.0.3	CVE-2020-26945
mybatis-spring-boot-starter	2.1.1	CVE-2020-26945
swagger-codegen-cli	3.0.21	CVE-2017-1000207，CVE-2017-1000208，CVE-2019-10086，CVE-2018-8088，CVE-2017-18640，CVE-2016-6199，CVE-2019-11065，CVE-2019-15052，
tomcat-embed-core	9.0.37	CVE-2020-17527
vertx-core	3.6.3	CVE-2019-17640

User-mgmt

Component	Version	CVE
rxnetty (indirect dependency)	0.4.9	CVE-2015-2156
netty-tcnative-boringssl-static (indirect dependency)	2.0.31.Final	CVE-2015-2156
spring-security-core	5.2.6.RELEASE	CVE-2018-1258
lodash	4.17.20	NPM-1065, NPM-557, NPM-7823
minimatch	3.0.4	NPM-118

Website-gateway

Component	Version	CVE
log4j	1.2.17	CVE-2019-17571
spring-security-core	5.2.6.RELEASE	CVE-2018-1258

MECM

Component	Version	CVE
lodash	4.17.13	CVE-2020-8203
minimatch	3.0.4	NPM-118

Upgrade Notes

N/A

Deprecation Notes

N/A

Other

经过安全工作组评审，Plugins 仓库遗留问题单I1XA0N将在 v0.9 版本发布后通过补丁形式修复，详细情况请参见安全工作组会议纪要

2.7 Version: v0.9

- Release Date: 2020-09-30
- Image Version: v0.9

新开源仓库列表

模块	名称	类型	URL	说明
helm-charts	helm-charts	部署	https://gitee.com/edgegallery/helm-charts	EdgeGallery 部署脚本
website-gateway	website-gateway	产品仓	https://gitee.com/edgegallery/website-gateway	前台封装
user-mgmt	user-mgmt	产品仓	https://gitee.com/edgegallery/user-mgmt	usermgmt 后台
user-mgmt	user-mgmt-fe	产品仓	https://gitee.com/edgegallery/user-mgmt-fe	usermgmt 前台
mecm	mecm-fe	产品仓	https://gitee.com/edgegallery/mecm-fe	MECM 前台
mecm	mecm-applcm	产品仓	https://gitee.com/edgegallery/mecm-applcm	MECM 应用生命周期管理
mecm	mecm-apm	产品仓	https://gitee.com/edgegallery/mecm-apm	MECM 应用包管理
mecm	mecm-inventory	产品仓	https://gitee.com/edgegallery/mecm-inventory	MECM 存量管理
mecm	mecm-appo	产品仓	https://gitee.com/edgegallery/mecm-appo	MECM 应用编排
developer	developer-be	产品仓	https://gitee.com/edgegallery/developer-be	developer 后台
developer	developer-fe	产品仓	https://gitee.com/edgegallery/developer-fe	developer 前台
appstore	appstore-fe	产品仓	https://gitee.com/edgegallery/appstore-fe	appstore 前台
appstore	appstore-be	产品仓	https://gitee.com/edgegallery/appstore-be	appstore 后台
platform-mgmt	platform-mgmt	产品仓	https://gitee.com/edgegallery/platform-mgmt	安装 k8/docker
docs	docs	产品仓	https://gitee.com/edgegallery/docs	文档

New Features

需求/发布计划详细说明

- All-in-One MEP/MECM 网络隔离
- MEP 支持 DNS 配置
- 支持 All-in-One 部署
- 多节点离线部署
- 九大维度安全优化

- MECM 界面/拓扑管理/资源管理优化
- （PoC）kubeedge 集成验证

Bug Fixes

参见 EdgeGallery v0.9 Test Report

Known Issues

参见 EdgeGallery v0.9 Test Report

Security Notes

Fixed Security Issues

参见 EdgeGallery v0.9 Test Report

Known Security Issues

参见 EdgeGallery v0.9 Test Report

Known Vulnerabilities in Used Modules

EdgeGallery v0.9 版本第三方开源组件存在以下漏洞需要用户自行修复：

Appstore

Component	Version	CVE
netty-transport (indirect dependency)	4.1.36.Final	CVE-2019-20444, CVE-2019-20445, CVE-2020-11612, CVE-2019-16869
log4j (indirect dependency)	1.2.17	CVE-2019-17571
spring-jdbc (indirect dependency)	5.1.8.RELEASE	CVE-2020-5398
spring-security-core (indirect dependency)	5.1.5.RELEASE	CVE-2018-1258

Developer

Component	Version	CVE
netty-transport (indirect dependency)	4.1.36.Final	CVE-2019-20444, CVE-2019-20445, CVE-2020-11612, CVE-2019-16869
spring-beans (indirect dependency)	5.1.8.RELEASE	CVE-2020-5398
spring-security-config (indirect dependency)	5.1.5.RELEASE	CVE-2018-1258, CVE-2020-5398
spring-security-core	5.3.4.RELEASE	CVE-2018-1258
lodash	4.17.13	CVE-2020-8203
minimatch	3.0.4	NPM-118

User-mgmt

Component	Version	CVE
rxnetty (indirect dependency)	0.4.9	CVE-2015-2156
netty-tcnative-boringssl-static (indirect dependency)	2.0.31.Final	CVE-2015-2156
spring-security-core	5.2.6.RELEASE	CVE-2018-1258
lodash	4.17.20	NPM-1065, NPM-557, NPM-7823
minimatch	3.0.4	NPM-118

Website-gateway

Component	Version	CVE
log4j	1.2.17	CVE-2019-17571
spring-security-core	5.2.6.RELEASE	CVE-2018-1258

MECM

Component	Version	CVE
lodash	4.17.13	CVE-2020-8203
minimatch	3.0.4	NPM-118

Upgrade Notes

N/A

Deprecation Notes

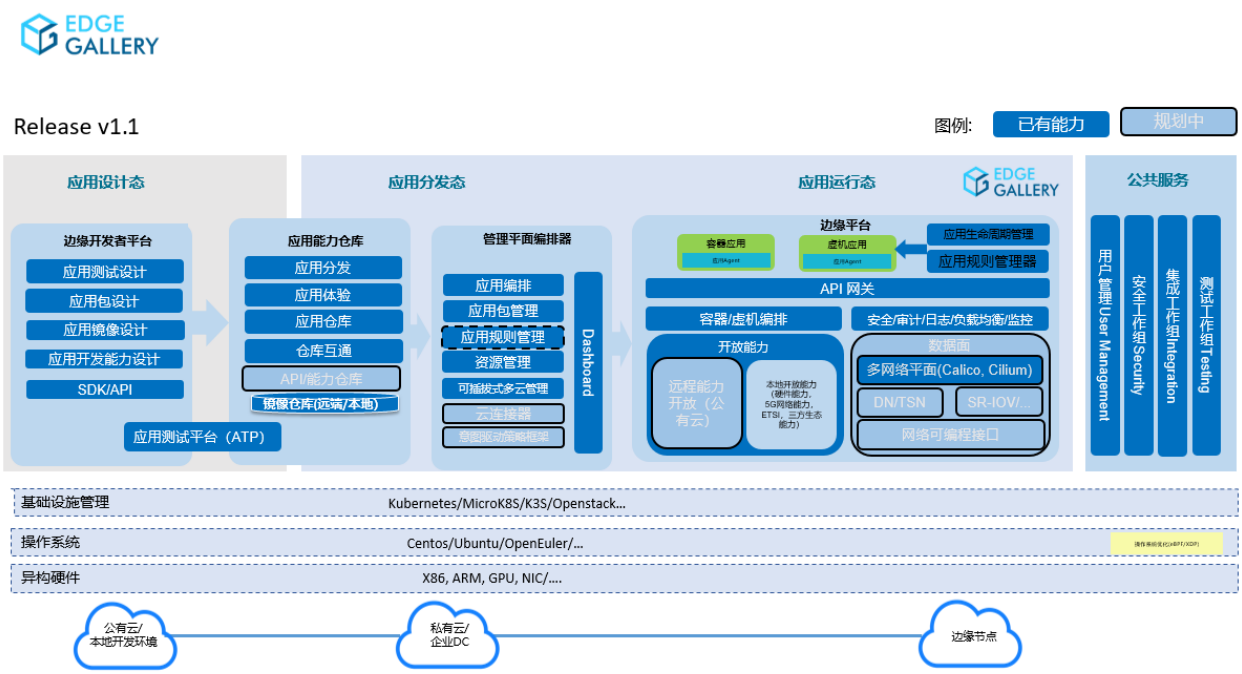
N/A

Other

经过安全工作组评审，Plugins 仓库遗留问题单I1XA0N将在 v0.9 版本发布后通过补丁形式修复，详细情况请参见安全工作组会议纪要

CHAPTER 3

EdgeGallery v1.1 版本架构图



3.1 参与 EdgeGallery 架构讨论

- 架构工作组会议纪要
- 架构工作组需求讨论
- 项目架构 Review

CHAPTER 4

开发者指导

- *APPStore*
- *Developer*
- MECM
  - *APM*
  - *APPO*
  - *Inventory*
  - *APPLCM*
- *MEP*
- *User Management*
- *ATP*

CHAPTER 5

安装指南

5.1 EdgeGallery 安装指导

EdgeGallery 离线安装指导

5.2 EdgeGallery 证书更新

EdgeGallery 证书更新指导

CHAPTER 6

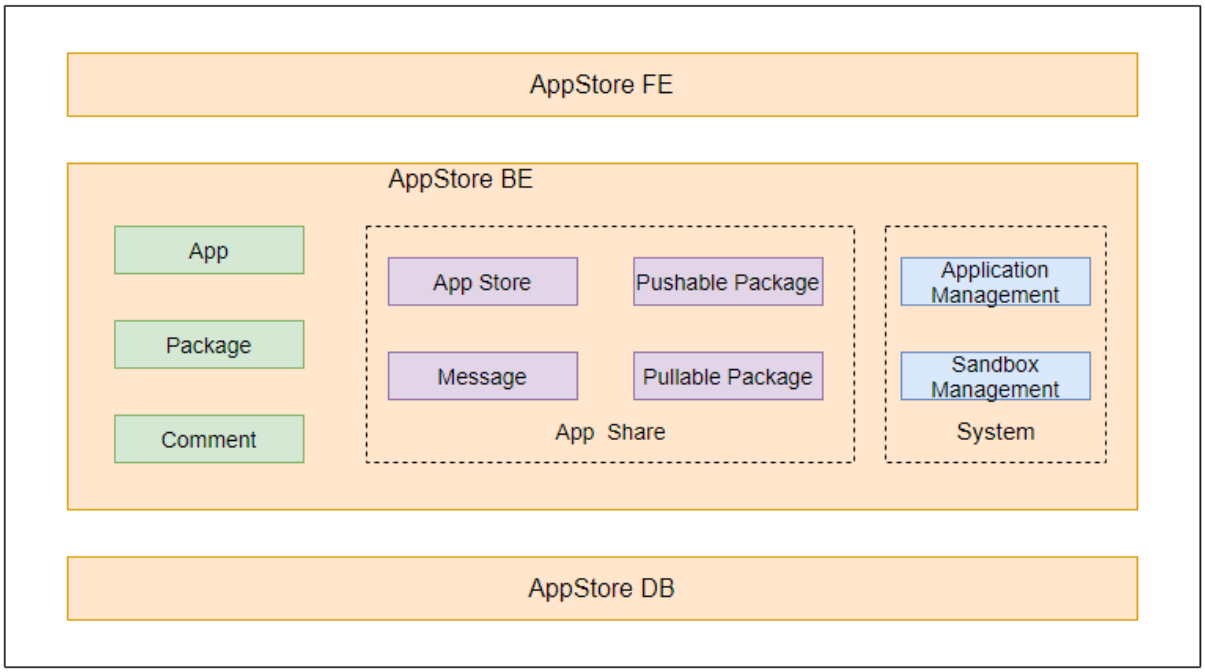
AppStore

6.1 概述

AppStore 是开发者发布和上线 App 应用的边缘应用市场，由 Developer 平台开发的边缘应用，经过 ATP 测试后可以直接发布到 AppStore 应用商店。用户也可以在 AppStore 直接上传边缘应用，经过 ATP 测试后可以发布到 AppStore，对外展示。

AppStore 分为前后台两个部分，AppStore BE是后台部分，提供主要功能接口供前台或其他三方系统调用，AppStore FE是前台部分，提供界面展示。

6.1.1 代码架构



1. AppStore FE：开发者平台前台框架，使用 VUE 开发。
2. AppStore BE：开发者平台后台框架，使用 SpringBoot+ServiceComb 开发。（有关 ServiceComb 请参考这里：https://servicecomb.apache.org/cn/）
3. AppStore DB：后台依赖 Postgres 数据库。

6.2 特性设计

AppStore 是 EdgeGallery 的应用仓库模块，主要负责 5G 边缘应用的存储与管理等工作，当前版本进行了分权控制，不同的用户角色包含不同的特性。

管理员用户包含应用上传、应用测试、应用发布、应用查询、应用评论、下载/删除所有应用、外部仓库管理、应用推送、应用拉取、消息管理、操作分析、应用管理、沙箱管理、应用在线体验、应用同步、文档中心。

租户用户包含应用上传、应用测试、应用发布、应用查询、应用评论、下载/删除本用户应用、应用在线体验、应用同步、文档中心。

游客用户包含应用查询、文档中心。

特性名称	特性描述
应用上传	AppStore 支持用户上传应用包。
应用测试	AppStore 集成了 ATP 测试平台，用户上传应用包后可以通过 ATP 测试平台对应用包进行测试。
应用发布	AppStore 允许已通过 ATP 测试的应用进行发布，发布后的应用会展示给所有用户。
应用查询	用户可以在“应用仓库”查看到所有已发布的应用，可以在”我的应用”页面查看自身上传的所有应用。
应用评论	用户可以对“应用仓库”看到的应用进行评论、打分，不能评论自己的应用。
应用下载	用户可以从 AppStore 下载应用包，租户仅可以下载本用户上传的应用包，管理员用户可以下载所有已发布的应用包和本用户上传的应用包。
应用删除	用户可以从 AppStore 删除应用包，租户仅可以删除本用户上传的应用包，管理员用户可以删除所有已发布的应用包和本用户上传的应用包。
外部应用商店管理	用户可以增删改查外部应用仓库。
应用推送	用户可以将允许共享的应用推送到已添加的外部应用仓库。
应用拉取	用户可以从已添加的外部应用仓库中拉取允许共享的应用到本仓库。
消息管理	用户在”应用共享”中执行的推送、下载等操作均会记录一条消息日志。
应用获取	用户可以从收到的推送消息中获取应用，上传到当前平台。
应用管理	用户可以管理近期热门应用、在应用仓库可展示的应用列表、可推广的应用列表。
沙箱管理	用户可以增删改查沙箱环境。
应用在线体验	用户可以基于已有的沙箱环境实例化应用，进行在线体验。
应用同步	用户可以将应用同步部署到配置的周边系统，如 MEAO
文档中心	用户可以在文档中心了解 appstore 的主要功能、接口规范以及应用包规范

6.3 接口设计

文档中介绍了 appstore-be 模块的接口设计详情，共有六个部分，分别对应于 app 管理、包管理、评论管理、应用推送、仓库管理、消息中心。

- *AppStore 接口列表*
  - *1. 应用*
    - *1.1 获取应用列表*
    - *1.2 获取应用*
    - *1.3 注册应用*
    - *1.4 删除应用*
    - *1.5 获取应用图标*

    * *1.6* 分片上传应用包
    * *1.7* 合并应用包
    * *1.8* 注册虚机应用
    * *1.9* 获取应用视频
    * *1.10* 修改热门应用
    * *1.11* 获取应用列表*-v2*
    * *1.12* 获取应用*-v2*
    * *1.13* 注册应用*-v2*
- *2.* 应用包
    * *2.1* 通过应用 *ID* 获取应用包列表
    * *2.2* 获取应用包
    * *2.3* 删除应用包
    * *2.4* 下载应用包
    * *2.5* 获取文件内容
    * *2.6* 应用发布
    * *2.7* 应用测试
    * *2.8* 获取应用包列表
    * *2.9* 修改应用
    * *2.10* 应用发布*-v2*
    * *2.11* 通过应用 *ID* 获取应用包列表*-v2*
    * *2.12* 获取应用包*-v2*
    * *2.13* 获取应用包列表*-v2*
    * *2.14* 下载应用图标
- *3.* 评论
    * *3.1* 获取评论列表
    * *3.2* 提交评论
    * *3.3* 获取评论列表*-v2*
- *4.* 应用推送
    * *4.1* 查询可推送应用列表
    * *4.2* 获取推送应用包

    * *4.3* 推送应用包
    * *4.4* 下载推送应用包
    * *4.5* 下载推送应用图标
    * *4.6* 查询可推送应用列表*-v2*
  - *5.* 仓库管理
    * *5.1* 添加仓库
    * *5.2* 删除仓库
    * *5.3* 修改仓库
    * *5.4* 查询仓库列表
    * *5.5* 查询仓库
    * *5.6* 分页查询仓库列表*-v2*
  - *6.* 消息
    * *6.1* 添加消息
    * *6.2* 获取所有消息
    * *6.3* 获取消息
    * *6.4* 删除消息
    * *6.5* 接收消息
    * *6.6* 更新状态
    * *6.7* 添加消息*-v2*
    * *6.8* 获取消息列表*-v2*
  - *7.* 应用拉取
    * *7.1* 查询可拉取应用列表
    * *7.2* 获取外部仓库可拉取应用列表
    * *7.3* 拉取应用包
    * *7.4* 分页查询可拉取应用列表*-v2*
    * *7.5* 获取外部仓库可拉取应用列表*-v2*
  - *8.* 沙箱管理
    * *8.1* 查询沙箱环境列表
    * *8.2* 查询沙箱环境
    * *8.3* 添加沙箱环境

    * *8.4* 删除沙箱环境
    * *8.5* 修改沙箱环境
    * *8.6* 上传沙箱环境配置文件
  - *9.* 应用在线体验
    * *9.1* 实例化应用
    * *9.2* 释放资源
    * *9.3* 获取应用工作状态

6.3.1 1. 应用

应用是应用商店基本的发布单位。

1.1 获取应用列表

根据入参查询应用列表。

```
URI:  /mec/appstore/v1/apps
METHOD: GET
```

请求参数:

名称	描述	IN	必选
name	应用名	request param	yes
provider	应用提供商	request param	yes
type	应用类型	request param	yes
affinity	架构	request param	yes
userId	应用创建人	request param	yes

响应示例:

```
200 OK
[
  {
    "appId": "string",
    "iconUrl": "string",
    "name": "string",
    "provider": "string",
    "type": "string",
    "shortDesc": "string",
    "createTime": "string",
```

(下页继续)

(续上页)

```
    "details": "string",
    "downloadCount": 0,
    "affinity": "string",
    "industry": "string",
    "contact": "string",
    "score": 0,
    "userId": "string",
    "userName": "string",
    "status": "string",
    "showType": "string",
    "deployMode": "string",
    "hotApp": false
  }
]
```

1.2 获取应用

根据应用 ID 获取应用详细信息。

```
URI:  /mec/appstore/v1/apps/{appId}
METHOD: GET
```

名称	描述	IN	必选
appId	app id	path	yes

响应示例:

```
200 OK
  {
    "appId": "string",
    "iconUrl": "string",
    "name": "string",
    "provider": "string",
    "type": "string",
    "shortDesc": "string",
    "createTime": "string",
    "details": "string",
    "downloadCount": 0,
    "affinity": "string",
    "industry": "string",
    "contact": "string",
    "score": 0,
```

(下页继续)

(续上页)

```
    "userId": "string",
    "userName": "string"
    "status": "string",
    "showType": "string",
    "deployMode": "string",
    "hotApp": false
  }
```

1.3 注册应用

注册应用。

```
URI:  /mec/appstore/v1/apps
METHOD: POST
```

名称	描述	IN	必选
userId	用户 ID	request param	yes
userName	用户名	request param	yes
file	应用包文件	request body form	yes
icon	应用图标文件	request body form	yes
demoVideo	应用视频文件	request body form	no
type	应用类型	request body form	yes
shortDesc	应用简述	request body form	yes
affinity	架构	request body form	yes
industry	应用所属行业	request body form	yes
testTaskId	测试任务 id	request body form	no
showType	显示类型：public、inner-public、private	request body form	no

响应示例:

```
200 OK
  {
    "appName": "string",
    "appId": "string",
    "packageId": "string",
    "provider": "string",
    "version": "string"
  }
```

1.4 删除应用

根据应用 ID 删除应用。

```
URI:  /mec/appstore/v1/apps/{appId}
METHOD: DELETE
```

名称	描述	IN	必选
userId	用户 ID	request param	yes
userName	应用名	request param	yes
appId	应用 ID	path	yes

响应示例:

```
200 OK
  "delete App success."
```

1.5 获取应用图标

根据应用 ID 获取应用图标。

```
URI:  /mec/appstore/v1/apps/{appId}/icon
METHOD: GET
```

名称	描述	IN	必选
appId	应用 ID	path	yes

响应示例:

```
200 OK
  binary output.
```

1.6 分片上传应用包

分片上传应用包。

```
URI:  /mec/appstore/v1/apps/upload
METHOD: POST
```

名称	描述	IN	必选
id	分包 ID	request body form	no
chunkNumber	当前文件块编号	request body form	yes
chunkSize	分块大小	request body form	no
currentChunkSize	当前分块大小	request body form	no
totalSize	总大小	request body form	no
identifier	文件表示	request body form	yes
filename	文件名	request body form	no
relativePath	文件相对路径	request body form	no
totalChunks	总块数	request body form	no
type	文件类型	request body form	no
file	文件	request body form	yes

响应示例:

```
200 OK
  {
    "upload package block success"
  }
```

1.7 合并应用包

合并应用包。

```
URI:  /mec/appstore/v1/apps/merge
METHOD: GET
```

名称	描述	IN	必选
fileName	文件名	request param	yes
guid	文件标识	request param	yes

响应示例:

```
200 OK
  {
    "file address"
  }
```

1.8 注册虚机应用

注册虚机应用。

```
URI:  /mec/appstore/v1/apps/vm-register
METHOD: POST
```

名称	描述	IN	必选
userId	用户 ID	request param	yes
userName	应用名	request param	yes
fileAddress	应用包文件相对路径	request body form	yes
icon	应用图标文件	request body form	yes
demoVideo	应用视频文件	request body form	no
type	应用类型	request body form	yes
shortDesc	应用简述	request body form	yes
affinity	架构	request body form	yes
industry	应用所属行业	request body form	yes
testTaskId	测试任务 id	request body form	no
showType	显示类型：public、inner-public、private	request body form	no

响应示例:

```
200 OK
  {
    "appName": "string",
    "appId": "string",
    "packageId": "string",
    "provider": "string",
    "version": "string"
  }
```

1.9 获取应用视频

根据应用 ID 获取应用视频。

```
URI:  /mec/appstore/v1/apps/{appId}/demoVideo
METHOD: GET
```

名称	描述	IN	必选
appId	应用 ID	path	yes

响应示例:

```
200 OK
  binary output.
```

1.10 修改热门应用

根据应用 ID 更改应用是否为热门应用。

```
URI:  /mec/appstore/v2/apps/hotapps
METHOD: PUT
```

名称	描述	IN	必选
appId	应用 ID	path	yes

响应示例:

```
200 OK
  set hot apps success.
```

1.11 获取应用列表-v2

分页查询应用列表。

```
URI:  /mec/appstore/v2/query/apps
METHOD: POST
```

请求参数:

名称	描述	IN	必选
QueryAppReqDto	查询应用条件	request body	yes

QueryAppReqDto

字段名	描述	字段类型	必选
types	应用类型	List	no
showType	应用显示类型	List	no
affinity	应用架构	List	no
industry	应用行业	List	no
workloadType	应用负载类型	List	no
userId	用户 Id	String	no
status	应用状态	String	no
appName	应用名称	String	no
queryCtrl	查询条件	QueryAppCtrlDto	yes

QueryAppCtrlDto

字段名	描述	字段类型	必选
limit	分页查询每页查询数量 [1,500]	int	yes
offset	分页查询起始页，从 0 开始	int	yes
sortItem	查询排序字段	String	no
sortType	查询排序方式升序/降序	String	no

响应示例:

```
200 OK
{
    "results": [
        {
            "appId": "string",
            "iconUrl": "string",
            "name": "string",
            "provider": "string",
            "type": "string",
            "shortDesc": "string",
            "showType": "string",
            "createTime": "string",
            "details": "string",
            "downloadCount": int,
            "affinity": "string",
            "industry": "string",
            "contact": "string",
            "score": 5.0,
            "userId": "string",
            "userName": "string",
            "status": "string",
```

(下页继续)

(续上页)

```
            "deployMode": "string",
            "hotApp": false
        }
    ],
    "limit": 1,
    "offset": 0,
    "total": 9
}
```

1.12 获取应用-v2

根据应用 ID 获取应用详细信息。

```
URI:  /mec/appstore/v2/apps/{appId}
METHOD: GET
```

名称	描述	IN	必选
appId	app id	path	yes

响应示例:

```
200 OK
  {
    "data":  {
        "appId": "string",
        "iconUrl": "string",
        "name": "string",
        "provider": "string",
        "type": "string",
        "shortDesc": "string",
        "showType": "string",
        "createTime": "string",
        "details": "string",
        "downloadCount": int,
        "affinity": "string",
        "industry": "string",
        "contact": "string",
        "score": 5.0,
        "userId": "string",
        "userName": "string",
        "status": "string",
        "deployMode": "string",
```

(下页继续)

(续上页)

```
        "hotApp": false
    },
    "retCode": 0,
    "params": "[string]",
    "message": "string"
}
```

1.13 注册应用-v2

注册应用。

```
URI:  /mec/appstore/v2/apps
METHOD: POST
```

名称	描述	IN	必选
userId	用户 ID	request param	yes
userName	应用名	request param	yes
file	应用包文件	request body form	yes
icon	应用图标文件	request body form	yes
demoVideo	应用视频文件	request body form	no
type	应用类型	request body form	yes
shortDesc	应用简述	request body form	yes
affinity	架构	request body form	yes
industry	应用所属行业	request body form	yes
testTaskId	测试任务 id	request body form	no
showType	显示类型：public、inner-public、private	request body form	no

响应示例:

```
200 OK
  {
    "data": {
        "appName": "string",
        "appId": "string",
        "packageId": "string",
        "provider": "string",
        "version": "string"
    },
    "retCode": 0,
    "params": "[string]",
```

(下页继续)

(续上页)

```
    "message": "string"
  }
```

6.3.2 2. 应用包

应用包属于应用，每个应用包含至少一个应用包。

2.1 通过应用 ID 获取应用包列表

通过应用 ID 获取应用包列表。

```
URI:  /mec/appstore/v1/apps/{appId}/packages
METHOD: GET
```

名称	描述	IN	必选
appId	应用 ID	path	yes
userId	用户 ID	request param	yes
access_token	请求 token	request header	yes

响应示例:

```
200 OK
[
  {
    "packageId": "string",
    "userName": "string",
    "status": "string",
    "shortDesc": "string",
    "provider": "string",
    "testTaskId": "string",
    "size": "string",
    "format": "string",
    "createTime": "string",
    "name": "string",
    "version": "string",
    "type": "string",
    "details": "string",
    "affinity": "string",
    "industry": "string",
    "contact": "string",
    "appId": "string",
```

(下页继续)

(续上页)

```
    "userId": "string",
    "showType": "string",
    "deployMode": "string"
  }
]
```

2.2 获取应用包

通过应用 ID 和应用包 ID 获取应用包。

```
URI:  /mec/appstore/v1/apps/{appId}/packages/{packageId}
METHOD: GET
```

名称	描述	IN	必选
appId	应用 ID	path	yes
packageId	应用包 ID	path	yes

响应示例:

```
200 OK
  {
    "packageId": "string",
    "userName": "string",
    "status": "string",
    "shortDesc": "string",
    "provider": "string",
    "testTaskId": "string",
    "size": "string",
    "format": "string",
    "createTime": "string",
    "name": "string",
    "version": "string",
    "type": "string",
    "details": "string",
    "affinity": "string",
    "industry": "string",
    "contact": "string",
    "appId": "string",
    "userId": "string",
    "showType": "string",
    "deployMode": "string"
  }
```

2.3 删除应用包

通过应用 ID 和应用包 ID 删除应用包。

```
URI:  /mec/appstore/v1/apps/{appId}/packages/{packageId}
METHOD: DELETE
```

名称	描述	IN	必选
userId	用户 Id	request param	yes
userName	用户名	request param	yes
appId	应用 ID	path	yes
packageId	应用包 ID	path	yes

响应示例:

```
200 OK
   delete package success.
```

2.4 下载应用包

通过应用 ID 和应用包 ID 下载应用包。

```
URI:  /mec/appstore/v1/apps/{appId}/packages/{packageId}/action/download
METHOD: GET
```

名称	描述	IN	必选
appId	应用 ID	path	yes
packageId	应用包 ID	path	yes

响应示例:

```
200 OK
  binary output.
```

2.5 获取文件内容

通过应用 ID 和应用包 ID 以及文件路径获取应用包中文件内容。

```
URI:  /mec/appstore/v1/apps/{appId}/packages/{packageId}/files
METHOD: POST
```

名称	描述	IN	必选
packageId	package id	path	yes
filePath	file path	request body	yes
appId	应用 ID	path	yes

响应示例:

```
200 OK
  file content output.
```

2.6 应用发布

通过应用 ID 和应用包 ID 发布应用。

```
URI:  /mec/appstore/v1/apps/{appId}/packages/{packageId}/action/publish
METHOD: POST
```

名称	描述	IN	必选
appId	应用 ID	path	yes
packageId	应用包 ID	path	yes

响应示例:

```
200 OK
  Publish Success.
```

2.7 应用测试

通过应用 ID 和应用包 ID 测试应用。

```
URI:  /mec/appstore/v1/apps/{appId}/packages/{packageId}/action/test
METHOD: POST
```

名称	描述	IN	必选
appId	应用 ID	path	yes
packageId	应用包 ID	path	yes
access_token	请求 token	request header	yes

响应示例:

```
200 OK
  {
  "atpTaskId": "string",
  "status": "string"
  }
```

2.8 获取应用包列表

通过用户 ID 获取应用包列表。

```
URI:  /mec/appstore/v1/packages
METHOD: GET
```

名称	描述	IN	必选
ueserId	用户 id	request param	yes
access_token	请求 token	request header	yes

响应示例:

```
200 OK
[
  {
    "packageId": "string",
    "userName": "string",
    "status": "string",
    "shortDesc": "string",
    "provider": "string",
    "testTaskId": "string",
    "size": "string",
    "format": "string",
    "createTime": "string",
    "name": "string",
    "version": "string",
    "type": "string",
```

(下页继续)

(续上页)

```
    "details": "string",
    "affinity": "string",
    "industry": "string",
    "contact": "string",
    "appId": "string",
    "userId": "string",
    "showType": "string",
    "deployMode": "string"
  }
]
```

2.9 修改应用

通过应用 ID 和应用包 ID 修改应用。

```
URI:  /mec/appstore/v1/apps/{appId}/package/{packageId}
METHOD: PUT
```

名称	描述	IN	必选
appId	应用 ID	path	yes
packageId	应用包 ID	path	yes
industry	应用行业	request body form	no
type	应用类型	request body form	no
icon	应用图标	request body form	no
video	应用视频	request body form	no
affinity	应用架构	request body form	no
shortDesc	应用描述	request body form	no
showType	应用显示类型	request body form	no

响应示例:

```
200 OK
  {
    "packageId": "string",
    "userName": "string",
    "status": "string",
    "shortDesc": "string",
    "provider": "string",
    "testTaskId": "string",
    "size": "string",
    "format": "string",
```

(下页继续)

(续上页)

```
    "createTime": "string",
    "name": "string",
    "version": "string",
    "type": "string",
    "details": "string",
    "affinity": "string",
    "industry": "string",
    "contact": "string",
    "appId": "string",
    "userId": "string",
    "showType": "string",
    "deployMode": "string"
  }
```

2.10 应用发布-v2

通过应用 ID 和应用包 ID 发布应用。

```
URI:  /mec/appstore/v2/apps/{appId}/packages/{packageId}/action/publish
METHOD: POST
```

名称	描述	IN	必选
appId	应用 ID	path	yes
packageId	应用包 ID	path	yes

响应示例:

```
200 OK
  Publish Success.
  {
    "data": "Publish Success",
    "retCode": 0,
    "params": "[string]",
    "message": "string"
  }
```

2.11 通过应用 ID 获取应用包列表-v2

通过应用 ID 分页查询应用包列表。

```
URI:  /mec/appstore/v2/apps/{appId}/packages
METHOD: GET
```

名称	描述	IN	必选
appId	应用 ID	path	yes
userId	用户 ID	request param	yes
access_token	请求 token	request header	yes
limit	分页查询每页查询数量，建议设置较大值，一次性查询完该应用的所有应用包	request param	yes
offset	分页查询起始页，从 0 开始	request param	yes

响应示例:

```
200 OK
[
  {
    "packageId": "string",
    "userName": "string",
    "status": "string",
    "shortDesc": "string",
    "provider": "string",
    "testTaskId": "string",
    "size": "string",
    "format": "string",
    "createTime": "string",
    "name": "string",
    "version": "string",
    "type": "string",
    "details": "string",
    "affinity": "string",
    "industry": "string",
    "contact": "string",
    "appId": "string",
    "userId": "string",
    "showType": "string",
```

(下页继续)

(续上页)

```
    "deployMode": "string"
  }
]
```

2.12 获取应用包-v2

通过应用 ID 和应用包 ID 获取应用包。

```
URI:  /mec/appstore/v2/apps/{appId}/packages/{packageId}
METHOD: GET
```

名称	描述	IN	必选
appId	应用 ID	path	yes
packageId	应用包 ID	path	yes

响应示例:

```
200 OK
  {
    "data":
    {
        "packageId": "string",
        "userName": "string",
        "status": "string",
        "shortDesc": "string",
        "provider": "string",
        "testTaskId": "string",
        "size": "string",
        "format": "string",
        "createTime": "string",
        "name": "string",
        "version": "string",
        "type": "string",
        "details": "string",
        "affinity": "string",
        "industry": "string",
        "contact": "string",
        "appId": "string",
        "userId": "string",
        "showType": "string",
        "deployMode": "string"
    },
```

(下页继续)

(续上页)

```
    "retCode": 0,
    "params": "[string]",
    "message": "string"
  }
```

2.13 获取应用包列表-v2

分页获取应用包列表。

```
URI:  /mec/appstore/v2/packages
METHOD: GET
```

名称	描述	IN	必选
ueserId	用户 id	request param	yes
access_token	请求 token	request header	yes
offset	分页查询起始页，从 0 开始	request param	yes
limit	分页查询每页查询数量 [1,500]	request param	yes
appName	应用名称	request param	no
status	应用状态	request param	no
sortType	查询排序方式，升序/降序	request param	no
sortItem	查询排序字段	request param	no

响应示例:

```
200 OK
{
    "results": [
        {
            "packageId": "string",
            "size": "string",
            "format": "string",
            "createTime": "string",
            "name": "string",
            "version": "string",
            "type": "string",
            "details": "string",
            "affinity": "string",
            "industry": "string",
            "contact": "string",
            "appId": "string",
            "userId": "string",
```

(下页继续)

(续上页)


```
            "userName": "string",
            "status": "string",
            "shortDesc": "string",
            "showType": "string",
            "testTaskId": "string",
            "provider": "string",
            "demoVideoName": "string",
            "deployMode": "string"
        }
    ],
    "limit": 1,
    "offset": 0,
    "total": 12
}
```


2.14 下载应用图标

通过应用 ID 和应用包 ID 下载应用图标。

```
URI:  /mec/appstore/v1/apps/{appId}/packages/{packageId}/icon
METHOD: GET
```

名称	描述	IN	必选
appId	应用 ID	path	yes
packageId	应用包 ID	path	yes

响应示例:

```
200 OK
  binary output.
```

6.3.3 3. 评论

用户可以评论应用.

3.1 获取评论列表

根据应用 ID 获取应用的评论列表。

```
URI:  /mec/appstore/v1/apps/{appId}/comments
METHOD: GET
```

名称	描述	IN	必选
appId	应用 ID	path	yes

响应示例:

```
200 OK
[
  {
    "commentId": 0,
    "user": {
      "userId": "string",
      "userName": "string",
      "permissions": [
        {
          "platform": "string",
          "role": "string",
          "pageIds": [
            "string"
          ]
        }
      ],
      "company": "string",
      "gender": "string",
      "telephone": "string"
    },
    "appId": "string",
    "body": "string",
    "score": 0,
    "commentTime": "string"
  }
]
```

3.2 提交评论

用户根据应用 ID 提交应用评分及评论。

```
URI:  /mec/appstore/v1/apps/{appId}/comments
METHOD: POST
```

名称	描述	IN	必选
userId	用户 ID	request param	yes
userName	应用名	request param	yes
appId	应用 ID	path	yes
entity	评论内容和打分	request body	yes

entity 请求体示例:

```
{
  "body": "string",
  "score": 0
}
```

响应示例:

```
200 OK
 "comments success."
```

3.3 获取评论列表-v2

根据应用 ID 分页获取应用的评论列表。

```
URI:  /mec/appstore/v2/apps/{appId}/comments
METHOD: GET
```

名称	描述	IN	必选
appId	应用 ID	path	yes
offset	分页查询起始页，从 0 开始	request param	yes
limit	分页查询每页查询数量 [1,500]	request param	yes

响应示例:

```
200 OK
{
    "results": [
        {
            "commentId": 0,
            "user": {
                "userId": "string",
                "userName": "string",
                "permissions": [
                    {
                        "platform": "string",
                        "role": "string",
                        "pageIds": [
                            "string"
                        ]
                    }
                ],
                "company": "string",
                "gender": "string",
                "telephone": "string"
            },
            "appId": "string",
            "body": "string",
            "score": 0,
            "commentTime": "string"
        }
    ],
    "limit": 1,
    "offset": 0,
    "total": 1
}
```

6.3.4 4. 应用推送

用户可以推送应用到运营商或第三方。

4.1 查询可推送应用列表

查询可推送的应用列表。

```
URI:  /mec/appstore/v1/packages/pushable
METHOD: GET
```

响应示例:

```
200 OK
[
  {
    "appId": "string",
    "packageId": "string",
    "name": "string",
    "provider": "string",
    "version": "string",
    "atpTestStatus": "string",
    "atpTestTaskId": "string",
    "atpTestReportUrl": "string",
    "latestPushTime": "string",
    "pushTimes": "string",
    "targetPlatform": "string",
    "affinity": "string",
    "shortDesc": "string",
    "industry": "string",
    "type": "string",
    "createTime": "string"
  }
]
```

4.2 获取推送应用包

根据 packageId 获取推送应用包。

```
URI:  /mec/appstore/v1/packages/{packageId}/pushable
METHOD: GET
```

名称	描述	IN	必选
packageId	应用包 ID	path	yes

响应示例:

```
200 OK
[
  {
    "appId": "string",
    "packageId": "string",
    "name": "string",
    "provider": "string",
    "version": "string",
    "atpTestStatus": "string",
    "atpTestTaskId": "string",
    "atpTestReportUrl": "string",
    "latestPushTime": "string",
    "pushTimes": "string",
    "targetPlatform": "string",
    "affinity": "string",
    "shortDesc": "string",
    "industry": "string",
    "type": "string",
    "createTime": "string"
  }
]
```

4.3 推送应用包

根据 packageId 推送应用包。

```
URI:  /mec/appstore/v1/packages/{packageId}/action/push
METHOD: POST
```

名称	描述	IN	必选
packageId	应用包 ID	path	yes
dto	运营商列表	request param	yes

响应示例:

```
200 OK
  [true, false, true]
```

4.4 下载推送应用包

根据 packageId 下载推送应用包。

```
URI:  /mec/appstore/v1/packages/{packageId}/action/download-package
METHOD: GET
```

名称	描述	IN	必选
packageId	应用包 ID	path	yes

响应示例:

```
200 OK
  binary output.
```

4.5 下载推送应用图标

根据 packageId 下载推送应用图标。

```
URI:  /mec/appstore/v1/packages/{packageId}/action/download-icon
METHOD: GET
```

名称	描述	IN	必选
packageId	应用包 ID	path	yes

响应示例:

```
200 OK
  binary output.
```

4.6 查询可推送应用列表-v2

分页查询可推送的应用列表。

```
URI:  /mec/appstore/v2/packages/pushable
METHOD: GET
```

名称	描述	IN	必选
limit	分页查询每页查询数量 [1,500]	request param	yes
offset	分页查询起始页，从 0 开始	request param	yes
sortItem	查询排序字段	request param	no
sortType	查询排序方式，升序/降序	request param	no
appName	应用名称	request param	no

响应示例:

```
200 OK
{
    "results": [
        {
            "appId": "string",
            "packageId": "string",
            "name": "string",
            "provider": "string",
            "version": "string",
            "atpTestStatus": "string",
            "atpTestTaskId": "string",
            "atpTestReportUrl": "string",
            "latestPushTime": "string",
            "pushTimes": 2,
            "sourcePlatform": "string",
            "targetPlatform": "string",
            "affinity": "string",
            "shortDesc": "string",
            "industry": "string",
            "type": "string",
            "createTime": "string",
            "deployMode": "string"
        }
    ],
    "limit": 1,
    "offset": 0,
    "total": 9
}
```

6.3.5 5. 仓库管理

对应用仓库的增删改查。

5.1 添加仓库

新增外部应用仓库。

```
URI:  /mec/appstore/v1/appstores
METHOD: POST
```

名称	描述	IN	必选
appStoreName	仓库名称	request param	yes
appStoreVersion	仓库版本	request param	yes
company	公司名称	request param	yes
url	访问地址	request param	yes
schema	访问协议	request param	no
appPushIntf	推送方式	request param	no
appdTransId	转换器 id	request param	yes
description	描述	request param	no

响应示例:

```
200 OK
  {
    "appStoreId": "string",
    "appStoreName": "string",
    "appStoreVersion": "string",
    "company": "string",
    "url": "string",
    "schema": "string",
    "appPushIntf": "string",
    "appdTransId": "string",
    "description": "string",
    "addedTime": "string",
    "modifiedTime": "string"
  }
```

5.2 删除仓库

删除外部应用仓库。

```
URI:  /mec/appstore/v1/appstores/{appStoreId}
METHOD: DELETE
```

名称	描述	IN	必选
appStoreId	仓库 id	path	yes

响应示例:

```
200 OK
  delete success
```

5.3 修改仓库

修改外部应用仓库。

```
URI:  /mec/appstore/v1/appstores/{appStoreId}
METHOD: PUT
```

名称	描述	IN	必选
appStoreId	仓库名称	path	yes
appStoreName	仓库名称	request param	yes
appStoreVersion	仓库版本	request param	yes
company	公司名称	request param	yes
url	访问地址	request param	yes
schema	访问协议	request param	no
appPushIntf	推送方式	request param	no
appdTransId	转换器 id	request param	yes
description	描述	request param	no

响应示例:

```
200 OK
  {
    "appStoreId": "string",
    "appStoreName": "string",
    "appStoreVersion": "string",
    "company": "string",
```

(下页继续)

(续上页)

```
    "url": "string",
    "schema": "string",
    "appPushIntf": "string",
    "appdTransId": "string",
    "description": "string",
    "addedTime": "string",
    "modifiedTime": "string"
  }
```

5.4 查询仓库列表

查询外部应用仓库列表。

```
URI:  /mec/appstore/v1/appstores
METHOD: GET
```

名称	描述	IN	必选
appStoreName	仓库名称	request param	yes
company	公司名称	request param	yes

响应示例:

```
200 OK
[
  {
    "appStoreId": "string",
    "appStoreName": "string",
    "appStoreVersion": "string",
    "company": "string",
    "url": "string",
    "schema": "string",
    "appPushIntf": "string",
    "appdTransId": "string",
    "description": "string",
    "addedTime": "string",
    "modifiedTime": "string"
  }
]
```

5.5 查询仓库

查询外部应用仓库。

```
URI:  /mec/appstore/v1/appstores/{appStoreId}
METHOD: GET
```

名称	描述	IN	必选
appStoreId	仓库名称	path	yes

响应示例:

```
200 OK
  {
    "appStoreId": "string",
    "appStoreName": "string",
    "appStoreVersion": "string",
    "company": "string",
    "url": "string",
    "schema": "string",
    "appPushIntf": "string",
    "appdTransId": "string",
    "description": "string",
    "addedTime": "string",
    "modifiedTime": "string"
  }
```

5.6 分页查询仓库列表-v2

分页查询外部应用仓库列表。

```
URI:  /mec/appstore/v2/appstores
METHOD: GET
```

名称	描述	IN	必选
appStoreName	仓库名称	request param	yes
company	公司名称	request param	yes
limit	分页查询每页查询数量 [1,500]	request param	yes
offset	分页查询起始页，从 0 开始	request param	yes

响应示例:

```
200 OK
{
    "results": [
        {
            "appStoreId": "string",
            "appStoreName": "string",
            "appStoreVersion": "string",
            "company": "string",
            "url": "string",
            "schema": "string",
            "appPushIntf": "string",
            "appdTransId": "string",
            "description": "string",
            "addedTime": "string",
            "modifiedTime": "string"
        }
    ],
    "limit": 1,
    "offset": 0,
    "total": 2
}
```

6.3.6 6. 消息

消息包含应用推送、下载、通知等信息。

6.1 添加消息

新增一条消息。

```
URI:  /mec/appstore/v1/messages
METHOD: POST
```

名称	描述	IN	必选
basicInfo	基本信息	request param	yes
sourceAppStore	源 AppStore	request param	yes
targetAppStore	目标 AppStore	request param	yes
atpTestStatus	ATP 测试结果	request param	yes
atpTestTaskId	ATP 测试任务 Id	request param	yes
atpTestReportUrl	ATP 测试报告	request param	yes
packageDownloadUrl	应用包下载地址	request param	yes
iconDownloadUrl	应用图标下载地址	request param	yes

响应示例:

```
200 OK
  add a message success
```

6.2 获取所有消息

根据消息类型获取消息。

```
URI:  /mec/appstore/v1/messages
METHOD: GET
```

名称	描述	IN	必选
messageType	消息类型	request param	no

响应示例:

```
200 OK
[
  {
    "messageId": "string",
    "basicInfo": {
      "name": "string",
      "provider": "string",
      "version": "string",
      "affinity": "string",
      "shortDesc": "string",
      "industry": "string",
      "type": "string"
    },
    "messageType": "string",
    "sourceAppStore": "string",
    "targetAppStore": "string",
    "time": "string",
    "description": "string",
    "atpTestStatus": "string",
    "atpTestTaskId": "string",
    "atpTestReportUrl": "string",
    "iconDownloadUrl": "string",
    "readed": true
  }
]
```

6.3 获取消息

根据 message id 获取消息。

```
URI:  /mec/appstore/v1/messages/{messageId}
METHOD: GET
```

名称	描述	IN	必选
messageId	消息 id	request param	yes

响应示例:

```
200 OK
  {
    "messageId": "string",
    "basicInfo": {
      "name": "string",
      "provider": "string",
      "version": "string",
      "affinity": "string",
      "shortDesc": "string",
      "industry": "string",
      "type": "string"
    },
    "messageType": "string",
    "sourceAppStore": "string",
    "targetAppStore": "string",
    "time": "string",
    "description": "string",
    "atpTestStatus": "string",
    "atpTestTaskId": "string",
    "atpTestReportUrl": "string",
    "iconDownloadUrl": "string",
    "readed": true
  }
```

6.4 删除消息

根据 message id 删除消息。

```
URI:  /mec/appstore/v1/messages/{messageId}
METHOD: DELETE
```

名称	描述	IN	必选
messageId	消息 id	request param	yes

响应示例:

```
200 OK
  delete success
```

6.5 接收消息

根据 message id 下载 app 相关信息。

```
URI:  /mec/appstore/v1/messages/{messageId}/action/download
METHOD: GET
```

名称	描述	IN	必选
messageId	消息 id	request param	yes
request	请求体	request param	yes

响应示例:

```
200 OK
  success
```

6.6 更新状态

根据 message id 更新消息状态。

```
URI:  /mec/appstore/v1/messages/{messageId}/action/readed
METHOD: PUT
```

名称	描述	IN	必选
messageId	消息 id	request param	yes

响应示例:

```
200 OK
  success
```

6.7 添加消息-v2

新增一条消息。

```
URI:  /mec/appstore/v2/messages
METHOD: POST
```

名称	描述	IN	必选
basicInfo	基本信息	request param	yes
sourceAppStore	源 AppStore	request param	yes
targetAppStore	目标 AppStore	request param	yes
atpTestStatus	ATP 测试结果	request param	yes
atpTestTaskId	ATP 测试任务 Id	request param	yes
atpTestReportUrl	ATP 测试报告	request param	yes
packageDownloadUrl	应用包下载地址	request param	yes
iconDownloadUrl	应用图标下载地址	request param	yes

响应示例:

```
200 OK
  {
    "data": "add a message success",
    "retCode": 0,
    "params": "[string]",
    "message": "string"
  }
```

6.8 获取消息列表-v2

根据查询条件分页获取消息。

```
URI:  /mec/appstore/v2/messages/action/query
METHOD: POST
```

请求参数:

名称	描述	IN	必选
QueryMessageReqDto	查询消息条件	request body	yes

QueryMessageReqDto

字段名	描述	字段类型	必选
messageType	消息类型	String	no
limit	分页查询每页查询数量 [1,500]	int	yes
offset	分页查询起始页，从 0 开始	int	yes
sortItem	查询排序字段	String	no
sortType	查询排序方式，升序/降序	String	no
appName	应用名称	String	no
timeFlag	消息时间标志，枚举值有：今天、一周内、一月内、更早	enum	no
allMessage	是否获取所有消息	boolean	yes
readable	不获取所有消息时，获取已读/未读消息	boolean	yes

响应示例:

```
200 OK
{
    "results": [
        {
            "messageId": "string",
            "basicInfo": {
                "name": "string",
                "provider": "string",
                "version": "string",
                "affinity": "string",
                "shortDesc": "string",
                "industry": "string",
                "type": "string"
            },
            "messageType": "string",
            "sourceAppStore": "string",
            "targetAppStore": "string",
            "time": "string",
            "description": "string",
            "atpTestStatus": "string",
            "atpTestTaskId": "string",
            "atpTestReportUrl": "string",
            "iconDownloadUrl": "string",
            "readed": true
```

(下页继续)

(续上页)

```
        }
    ],
    "limit": 8,
    "offset": 0,
    "total": 2
}
```

6.3.7 7. 应用拉取

用户可以从运营商或第三方仓库拉取应用。

7.1 查询可拉取应用列表

查询可拉取的应用列表。

```
URI:  /mec/appstore/v1/packages/pullable
METHOD: GET
```

响应示例:

```
200 OK
[
  {
    "appId": "string",
    "packageId": "string",
    "name": "string",
    "provider": "string",
    "version": "string",
    "atpTestStatus": "string",
    "atpTestTaskId": "string",
    "atpTestReportUrl": "string",
    "latestPushTime": "string",
    "pushTimes": "string",
    "targetPlatform": "string",
    "affinity": "string",
    "shortDesc": "string",
    "industry": "string",
    "type": "string",
    "createTime": "string"
  }
]
```

7.2 获取外部应用仓库可拉取应用列表

根据 platfromId 获取外部仓库可拉取应用列表。

```
URI:  /mec/appstore/v1/packages/{platformId}/pullable
METHOD: GET
```

名称	描述	IN	必选
platformId	外部应用仓库 ID	path	yes

响应示例:

```
200 OK
[
  {
    "appId": "string",
    "packageId": "string",
    "name": "string",
    "provider": "string",
    "version": "string",
    "atpTestStatus": "string",
    "atpTestTaskId": "string",
    "atpTestReportUrl": "string",
    "latestPushTime": "string",
    "pushTimes": "string",
    "targetPlatform": "string",
    "affinity": "string",
    "shortDesc": "string",
    "industry": "string",
    "type": "string",
    "createTime": "string"
  }
]
```

7.3 拉取应用包

根据 packageId 拉取应用包。

```
URI:  /mec/appstore/v1/packages/{packageId}/action/pull
METHOD: POST
```

名称	描述	IN	必选
packageId	应用包 ID	path	yes
PullAppReqDto	拉取请求结构	request body	yes

PullAppReqDto：

字段	描述	类型	必选
sourceStoreId	源仓库平台 ID	string	yes
sourceStoreName	源仓库平台名称	string	yes
name	应用名称	string	yes
provider	应用供应商	string	yes
version	应用版本	string	yes
atpTestStatus	应用测试状态	string	yes
affinity	应用架构	string	yes
shortDesc	应用描述	string	yes
showType	应用显示类型	string	yes
industry	应用行业	string	yes
type	应用类型	string	yes
userId	用户 ID	string	yes
userName	用户名称	string	yes

响应示例:

```
200 OK
  true
```

7.4 分页查询可拉取应用列表-v2

分页查询可拉取的应用列表。

```
URI:  /mec/appstore/v2/packages/pullable
METHOD: GET
```

名称	描述	IN	必选
limit	分页查询每页查询数量 [1,500]	request param	yes
offset	分页查询起始页，从 0 开始	request param	yes
sortItem	查询排序字段	request param	no
sortType	查询排序方式，升序/降序	request param	no
appName	应用名称	request param	no

响应示例:

```
200 OK
{
    "results": [
        {
            "appId": "string",
            "packageId": "string",
            "name": "string",
            "provider": "string",
            "version": "string",
            "atpTestStatus": "string",
            "atpTestTaskId": "string",
            "atpTestReportUrl": "string",
            "latestPushTime": "string",
            "pushTimes": "string",
            "targetPlatform": "string",
            "affinity": "string",
            "shortDesc": "string",
            "industry": "string",
            "type": "string",
            "createTime": "string"
        }
    ],
    "limit": 1,
    "offset": 0,
    "total": 2
}
```

7.5 获取外部应用仓库可拉取应用列表-v2

根据 platfromId 获取外部仓库可拉取应用列表。

```
URI:  /mec/appstore/v2/packages/{platfromId}/pullable
METHOD: GET
```

名称	描述	IN	必选
platformId	外部应用仓库 ID	path	yes
limit	分页查询每页查询数量 [1,500]	request param	yes
offset	分页查询起始页，从 0 开始	request param	yes
sortItem	查询排序字段	request param	no
sortType	查询排序方式，升序/降序	request param	no
appName	应用名称	request param	no

响应示例:

```
200 OK
{
    "results": [
        {
            "appId": "string",
            "packageId": "string",
            "name": "string",
            "provider": "string",
            "version": "string",
            "atpTestStatus": "string",
            "atpTestTaskId": "string",
            "atpTestReportUrl": "string",
            "latestPushTime": "string",
            "pushTimes": "string",
            "targetPlatform": "string",
            "affinity": "string",
            "shortDesc": "string",
            "industry": "string",
            "type": "string",
            "createTime": "string"
        }
    ],
    "limit": 1,
    "offset": 0,
    "total": 2
}
```

6.3.8 8. 沙箱管理

管理沙箱环境。

8.1 查询沙箱环境列表

查询沙箱环境列表。

```
URI:  /mec/appstore/v1/system/hosts
METHOD: GET
```

名称	描述	IN	必选
userId	用户 ID	request param	no
name	沙箱环境名称	request param	no
ip	沙箱环境 ip	request param	no

响应示例:

```
200 OK
[
  {
    "hostId": "string",
    "name": "string",
    "address": "string",
    "architecture": "string",
    "status": "NORMAL",
    "lcmIp": "string",
    "mecHost": "string",
    "protocol": "string",
    "port": 0,
    "os": "string",
    "parameter":"string",
    "portRangeMin": 0,
    "portRangeMax": 0
  }
]
```

8.2 查询沙箱环境

根据 hostId 查询沙箱环境。

```
URI:  /mec/appstore/v1/system/hosts/{hostId}
METHOD: GET
```

名称	描述	IN	必选
hostId	沙箱环境 id	path	yes

响应示例:

```
200 OK


 {
   "hostId": "string",
```

(下页继续)

(续上页)

```
   "name": "string",
   "address": "string",
   "architecture": "string",
   "status": "NORMAL",
   "lcmIp": "string",
   "mecHost": "string",
   "protocol": "string",
   "port": 0,
   "os": "string",
   "parameter":"string",
   "portRangeMin": 0,
   "portRangeMax": 0
 }
```

8.3 添加沙箱环境

新增一个沙箱环境。

```
URI: /mec/appstore/v1/system/hosts
METHOD: POST
```

名称	描述	IN	必选
MepHost	沙箱管理请求结构体	request body	yes

```
MepHost
{
  "name": "string",
  "address": "string",
  "architecture": "string",
  "status": "NORMAL",
  "lcmIp": "string",
  "mecHost": "string",
  "protocol": "string",
  "port": 0,
  "os": "string",
  "parameter":"string",
  "portRangeMin": 0,
  "portRangeMax": 0
}
```

响应示例:

```
200 OK
true
```

8.4 删除沙箱环境

根据 hostId 删除沙箱环境。

```
URI: /mec/appstore/v1/system/hosts/{hostId}
METHOD: DELETE
```

名称	描述	IN	必选
hostId	沙箱环境 id	path	yes

响应示例:

```
200 OK
true
```

8.5 修改沙箱环境

根据 hostId 修改沙箱环境。

```
URI: /mec/appstore/v1/system/hosts/{hostId}
METHOD: PUT
```

名称	描述	IN	必选
hostId	沙箱环境 id	path	yes
MepHost	沙箱管理请求结构体	request body	yes

```
MepHost
{
  "hostId": "string",
  "name": "string",
  "address": "string",
  "architecture": "string",
  "status": "NORMAL",
  "lcmIp": "string",
  "mecHost": "string",
  "protocol": "string",
  "port": 0,
```

(下页继续)

(续上页)

```
  "os": "string",
  "parameter":"string",
  "portRangeMin": 0,
  "portRangeMax": 0
}
```

响应示例:

```
200 OK
true
```

8.6 上传沙箱环境配置文件

上传沙箱环境配置文件。

```
URI: /mec/appstore/v1/system/host/files
METHOD: POST
```

名称	描述	IN	必选
uploadFile	上传文件	request body form	yes
userId	用户 ID	request param	yes

响应示例:

```
200 OK
{
  "fileId": "string",
  "fileName": "string",
  "url": "string",
  "userId": "string",
  "uploadDate": "2020-09-14T09:03:17.084Z",
  "filePath": "string",
  "temp": false
}
```

6.3.9 9. 应用在线体验

可以在线体验应用

9.1 实例化应用

实例化应用，开始应用在线体验。

```
URI:  /mec/appstore/v1/experience/deploy
METHOD: GET
```

名称	描述	IN	必选
appId	应用 ID	request param	no
packageId	应用包 ID	request param	no
userId	用户 ID	request param	no
name	沙箱环境名称	request param	no
ip	沙箱环境 ip	request param	no

响应示例:

```
200 OK
{
    "data": "[{"serviceName":"testApp","serviceName":"30258","mecHost":"127.0.0.1"}]",
    "retCode": 0,
    "params": "[string]",
    "message": "string"
}
```

9.2 释放资源

体验完成，释放资源。

```
URI:  /mec/appstore/v1/experience/clean
METHOD: POST
```

名称	描述	IN	必选
packageId	应用包 ID	request param	yes
userId	用户 ID	request param	no
name	沙箱环境名称	request param	no
ip	沙箱环境 ip	request param	no

响应示例:

```
200 OK
true
```

9.3 获取应用工作状态

获取应用工作状态。

```
URI:  /mec/appstore/v1/experience/container/workStatus
METHOD: POST
```

名称	描述	IN	必选
packageId	应用包 ID	request param	no
userId	用户 ID	request param	no
name	沙箱环境名称	request param	no
ip	沙箱环境 ip	request param	no

响应示例:

```
200 OK
{
    "data": "[{"serviceName":"testApp","serviceName":"30258","mecHost":"127.0.0.1"}]",
    "retCode": 0,
    "params": "[string]",
    "message": "string"
}
```

6.4 数据库设计

AppStore 数据库使用了开源的 PostgreSQL 数据库，推荐版本为 12.2。目前设计并应用了如下数据库表格:

- **app_table** 存储 app 信息，如名字、提供者、描述等

```
create TABLE if not exists app_table (
        APPID                   VARCHAR(200)        NOT NULL,
        APPNAME                 VARCHAR(100)        NULL,
        APPLICATIONTYPE         VARCHAR(300)        NULL,
        DEPLOYMODE              VARCHAR(100)        NULL,
        SHORTDESC               TEXT                NULL,
        PROVIDER                VARCHAR(300)        NULL,
```

(下页继续)

(续上页)

```
        APPINTRODUCTION          TEXT              NULL,
        DOWNLOADCOUNT            INT               NULL,
        AFFINITY                 VARCHAR(100)      NULL,
        INDUSTRY                 VARCHAR(100)      NULL,
        CONTACT                  VARCHAR(100)      NULL,
        USERID                   VARCHAR(100)      NULL,
        USERNAME                 VARCHAR(100)      NULL,
        CREATETIME               TIMESTAMP         NULL,
        MODIFYTIME               TIMESTAMP         NULL,
        SCORE                    NUMERIC(2,1)      NULL,
        STATUS                   VARCHAR(50)       NULL,
        SHOWTYPE                 VARCHAR(100)      NOT NULL DEFAULT 'public',
        ISHOTAPP                 boolean           DEFAULT false,
        EXPERIENCEABLE           boolean           DEFAULT false,
        CONSTRAINT app_table_pkey PRIMARY KEY (APPID)
    );
```

- **catalog_package_table** 存储应用的 package 信息，一个应用可以拥有多个 package

```
create TABLE if not exists catalog_package_table (
        PACKAGEID                VARCHAR(200)      NOT NULL,
        PACKAGEADDRESS           VARCHAR(200)      NULL,
        ICONADDRESS              VARCHAR(200)      NULL,
        DEMOVIDEOADDRESS         VARCHAR(200)      NULL,
        SIZE                     VARCHAR(100)      NULL,
        FILESTRUCTURE            TEXT              NULL,
        CREATETIME               TIMESTAMP         NULL,
        SHORTDESC                TEXT              NULL,
        APPNAME                  VARCHAR(100)      NULL,
        VERSION                  VARCHAR(20)       NULL,
        APPLICATIONTYPE          VARCHAR(300)      NULL,
        DEPLOYMODE               VARCHAR(100)      NULL,
        MARKDOWNCONTENT          TEXT              NULL,
        AFFINITY                 VARCHAR(100)      NULL,
        INDUSTRY                 VARCHAR(100)      NULL,
        CONTACT                  VARCHAR(100)      NULL,
        APPID                    VARCHAR(100)      NULL,
        USERID                   VARCHAR(100)      NULL,
        USERNAME                 VARCHAR(100)      NULL,
        TESTTASKID               VARCHAR(100)      NULL,
        STATUS                   VARCHAR(100)      NULL,
        PROVIDER                 VARCHAR(100)      NULL,
        SHOWTYPE                 VARCHAR(100)      NOT NULL DEFAULT 'public',
        APPINSTANCEID            VARCHAR(100)      NULL,
```

(下页继续)

(续上页)

```
        INSTANCETENENTID         VARCHAR(100)       NULL,
        INSTANCEPACKAGEID        VARCHAR(255)       NULL,
        EXPERIENCEABLE           boolean            DEFAULT false,
        STARTEXPTIME             VARCHAR(100)       NULL,
        EXPERIENCEABLEIP         VARCHAR(100)       NULL,
        MECHOST                  VARCHAR(100)       NULL,
        CONSTRAINT catalog_package_table_pkey PRIMARY KEY (PACKAGEID)
    );
```

- **csar_package_score** 存储应用的评论信息

```
create TABLE if not exists csar_package_score (
        COMMENTID                serial,
        USERID                   VARCHAR(100)       NULL,
        USERNAME                 VARCHAR(100)       NULL,
        APPID                    VARCHAR(200)       NOT NULL,
        COMMENTS                 TEXT               NULL,
        SCORE                    NUMERIC(2,1)       NULL,
        COMMENTTIME              TIMESTAMP          NULL,
        CONSTRAINT csar_package_score_pkey PRIMARY KEY (COMMENTID)
    );
```

- **app_store_table** 存储外部 appstore 信息

```
create TABLE if not exists app_store_table (
        APPSTOREID               VARCHAR(64)        NOT NULL,
        APPSTORENAME             VARCHAR(128)       NULL,
        APPSTOREVERSION          VARCHAR(64)        NOT NULL,
        COMPANY                  VARCHAR(128)       NULL,
        URL                      VARCHAR(256)       NOT NULL,
        SCHEMA                   VARCHAR(16)        NULL,
        APPPUSHINTF              VARCHAR(256)       NULL,
        APPDTRANSID               VARCHAR(64)        NULL,
        DESCRIPTION              VARCHAR(256)       NULL,
        ADDEDTIME                TIMESTAMP          NOT NULL,
        MODIFIEDTIME             TIMESTAMP          NULL,
        CONSTRAINT app_store_table_pkey PRIMARY KEY (APPSTOREID)
    );
```

- **pushable_package_table** 存储可推送应用包的信息

```
create TABLE if not exists pushable_package_table (
        PACKAGEID  VARCHAR(64) NOT NULL,
        ATPTESTREPORTURL VARCHAR(100) NOT NULL,
```

(下页继续)

(续上页)

```
        LATESTPUSHTIME TIMESTAMP NOT NULL,
        PUSHTIMES INTEGER NOT NULL DEFAULT 0,
        SOURCEPLATFORM VARCHAR(100) NOT NULL,
        CONSTRAINT PUSHABLE_PACKAGE_TABLE_PKEY PRIMARY KEY (PACKAGEID)
    );
```

- **message_table** 存储 appstore 间互通时的消息

```
create TABLE if not exists message_table (
        MESSAGEID                VARCHAR(100)       NOT NULL,
        RESULT                   VARCHAR(100)       NULL,
        READED                   boolean            default false,
        NAME                     VARCHAR(100)       NULL,
        PROVIDER                 VARCHAR(100)       NULL,
        VERSION                  VARCHAR(100)       NULL,
        MESSAGETYPE              VARCHAR(20)        NULL,
        SOURCEAPPSTORE           VARCHAR(100)       NULL,
        TARGETAPPSTORE           VARCHAR(100)       NULL,
        TIME                     VARCHAR(100)       NULL,
        DESCRIPTION              VARCHAR(255)       NULL,
        ATPTESTSTATUS            VARCHAR(50)        NULL,
        ATPTESTTASKID            VARCHAR(100)       NULL,
        ATPTESTREPORTURL         VARCHAR(255)       NULL,
        PACKAGEDOWNLOADURL       VARCHAR(255)       NULL,
        ICONDOWNLOADURL          VARCHAR(255)       NULL,
        DEMOVIDEODOWNLOADURL     VARCHAR(255)       NULL,
        AFFINITY                 VARCHAR(100)       NULL,
        SHORTDESC                TEXT               NULL,
        INDUSTRY                 VARCHAR(100)       NULL,
        TYPE                     VARCHAR(50)        NULL,
        CONSTRAINT message_table_pkey PRIMARY KEY (MESSAGEID)
    );
```

- **tbl_service_host** 存储沙箱环境信息

```
create TABLE if not exists tbl_service_host (
        HOST_ID VARCHAR(50) NOT NULL,
        USER_ID VARCHAR(50) DEFAULT NULL,
        NAME VARCHAR(100) DEFAULT NULL,
        ADDRESS VARCHAR(255) DEFAULT NULL,
        ARCHITECTURE VARCHAR(100) DEFAULT NULL,
        STATUS VARCHAR(20) DEFAULT NULL,
        PROTOCOL VARCHAR(20) DEFAULT NULL,
        LCM_IP VARCHAR(20) DEFAULT NULL,
```

(下页继续)

(续上页)

```
        MEC_HOST VARCHAR(20) DEFAULT NULL,
        OS VARCHAR(255) DEFAULT NULL,
        PORT_RANGE_MIN INTEGER DEFAULT 0,
        PORT_RANGE_MAX INTEGER DEFAULT 0,
        PORT INTEGER DEFAULT 0,
        VNC_PORT INTEGER DEFAULT NULL,
        PARAMETER text DEFAULT NULL,
        DELETE BOOLEAN DEFAULT NULL
    );
```

- **tbl_uploaded_file** 存储沙箱环境配置文件信息

```
create TABLE if not exists tbl_uploaded_file (
        FILE_ID VARCHAR(50) NOT NULL DEFAULT NULL,
        FILE_NAME VARCHAR(255) DEFAULT NULL,
        IS_TEMP BOOL DEFAULT NULL,
        USER_ID VARCHAR(50) DEFAULT NULL,
        UPLOAD_DATE TIMESTAMPTZ(6) DEFAULT NULL,
        FILE_PATH VARCHAR(255) DEFAULT NULL,
        CONSTRAINT TBL_UPLOADED_FILE_PKEY PRIMARY KEY (FILE_ID)
    );
```

6.5 如何贡献

本文将向用户展示如何参与到 AppStore 项目的贡献中，内容分为两个部分，将分别介绍如何搭建本地开发环境和 AppStore 的操作指导。

6.5.1 搭建本地开发环境-Release 1.1

本节将指导用户如何在本地搭建 AppStore-be 和 AppStore-fe 模块，从而帮助开发者快速理解、开发代码。

其中 AppStore-be 的本地启动依赖于：

```
- Service center
- DataBase
- User Management
```

其中 AppStore-fe 的本地启动依赖于:

```
- Website Gateway
- User Management
- AppStore-be
```

所以为了搭建本地开发环境，最终的启动顺序为 Service Center -> DataBase -> UserManagement -> AppStore-be -> WebsiteGateway(AppStore-fe)

Service Center

1、下载Service Center

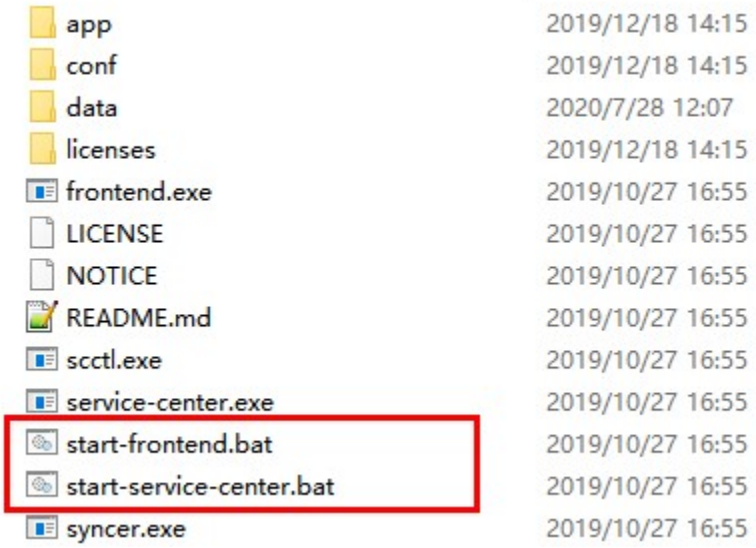
ServiceComb Service-Center下载

发布包

版本号	源码	Windows版本	Linux版本	Darwin版本	与之兼容的Java-Chassis版本
Apache ServiceComb Service-Center 2.0.0 (LATEST)	[src] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	Java-Chassis 1.0.0 or above
Apache ServiceComb Service-Center 1.3.0	[src] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	Java-Chassis 1.0.0 or above
Apache ServiceComb Service-Center 1.2.0	[src] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	Java-Chassis 1.0.0 or above
Apache ServiceComb Service-Center 1.1.0	[src] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	Java-Chassis 1.0.0 or above
Apache ServiceComb (incubating) Service-Center 1.0.0	[src] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	Java-Chassis 1.0.0 or above
Apache ServiceComb (incubating) Service-Center 1.0.0-m2	[src] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	NA	Java-Chassis 1.0.0-m2 or above
Apache ServiceComb (incubating) Service-Center 1.0.0-m1	[src] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	NA	Java-Chassis 1.0.0-m1 or above

!

2、解压后运行 start-service-center.bat 和 start-frontend.bat

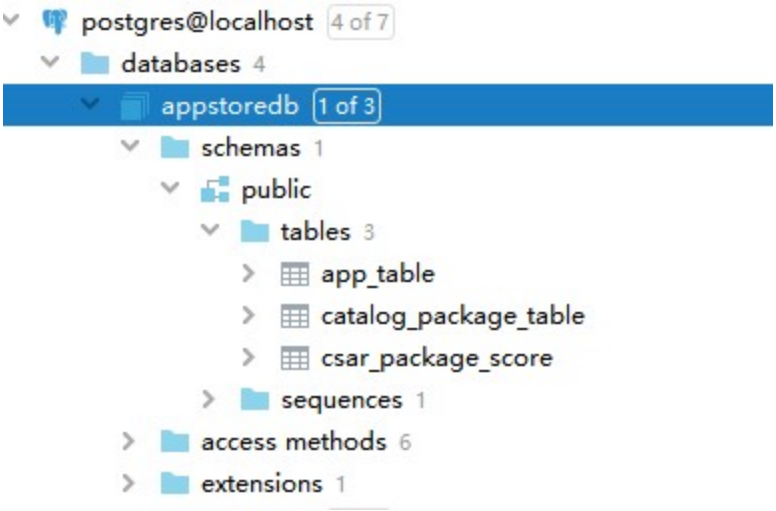


DataBase

1、下载、安装、启动 Postgresql 12.3

2、设置登录名和密码，例如 postgresql/root

3、创建名为 appstoredb 的数据库



4、创建名为 usermgmtdb 的数据库

User Management

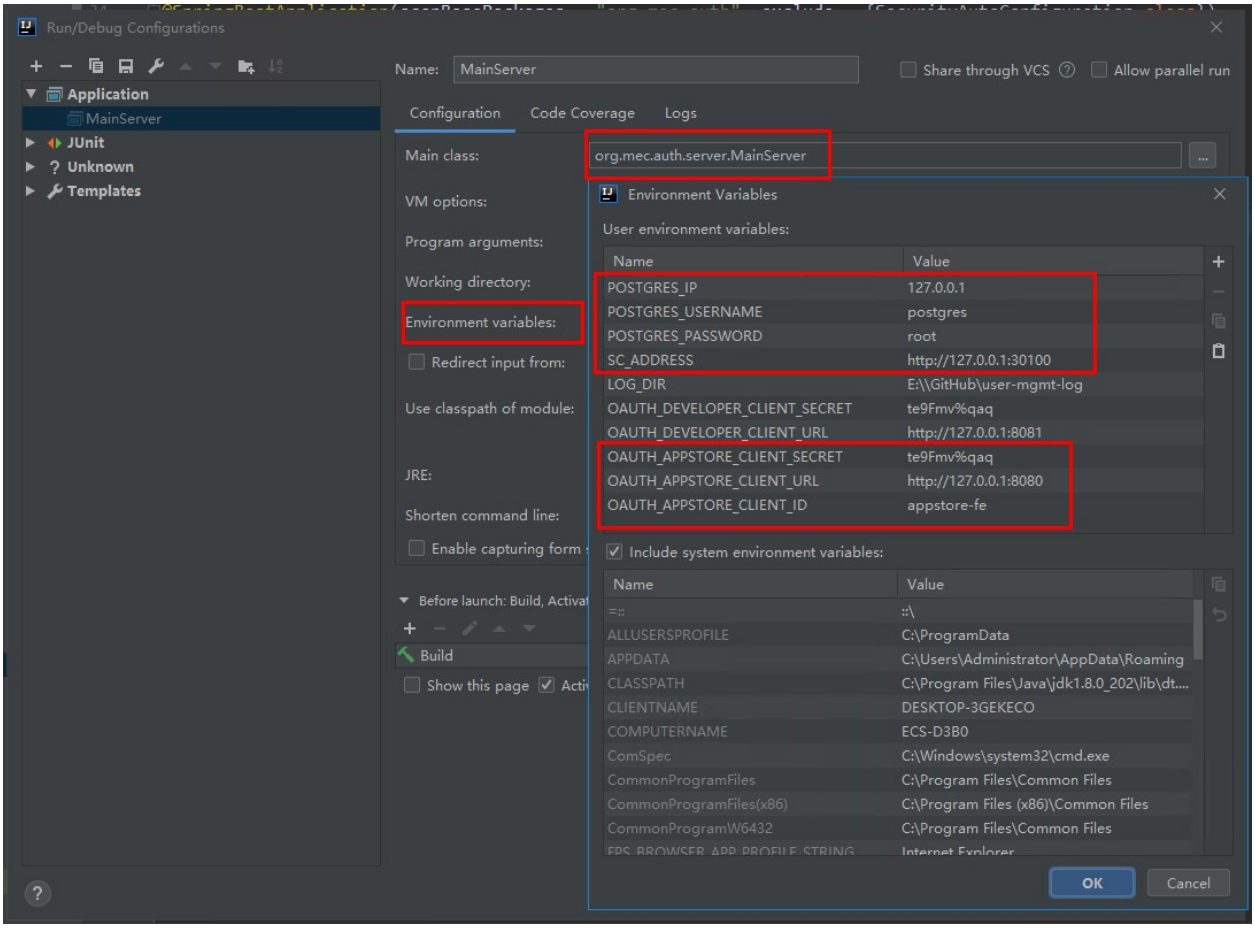
1. 下载 user-mgmt-be 代码和 user-mgmt-fe 代码

2、安装 node.js 和 npm，进入 user-mgmt-fe 文件夹，运行 `npm install` 和 `npm run build`

3、在 user-mgmt-be 的 src/main/resources/文件夹中创建 static 文件夹

4、将 user-mgmt-fe 编译出的 dist 文件夹中的内容拷贝到 user-mgmt-be 的 static 文件夹中

5、在编译器（此处以 IDEA 为例）中配置 MainServer 的运行时环境变量



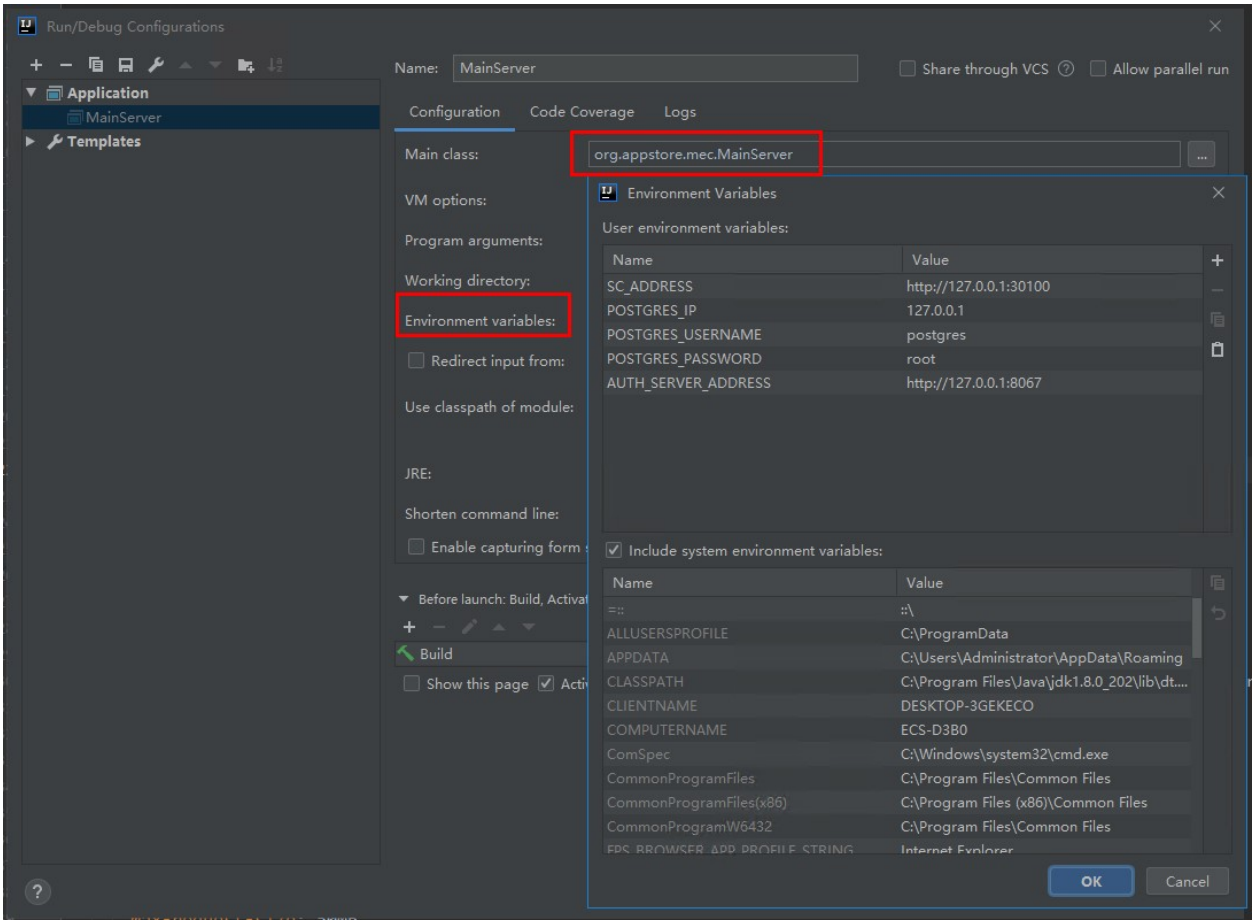
6、运行 MainServer 函数，启动 UserManagement

AppStore-be

1、下载 appstore-be 代码

2、修改 appstore-be 中/src/main/resources/application.yaml 文件内的 `servicecomb.rest.address` 为 `127.0.0.1:8099`

3、在编译器（此处以 IDEA 为例）中配置 MainServer 的运行时环境变量



4. 运行 MainServer 函数，启动 appstore-be

AppStore-fe

1、下载 appstore-fe 代码

2、进入 appstore-fe 文件夹，运行 `npm install` 和 `npm run build`

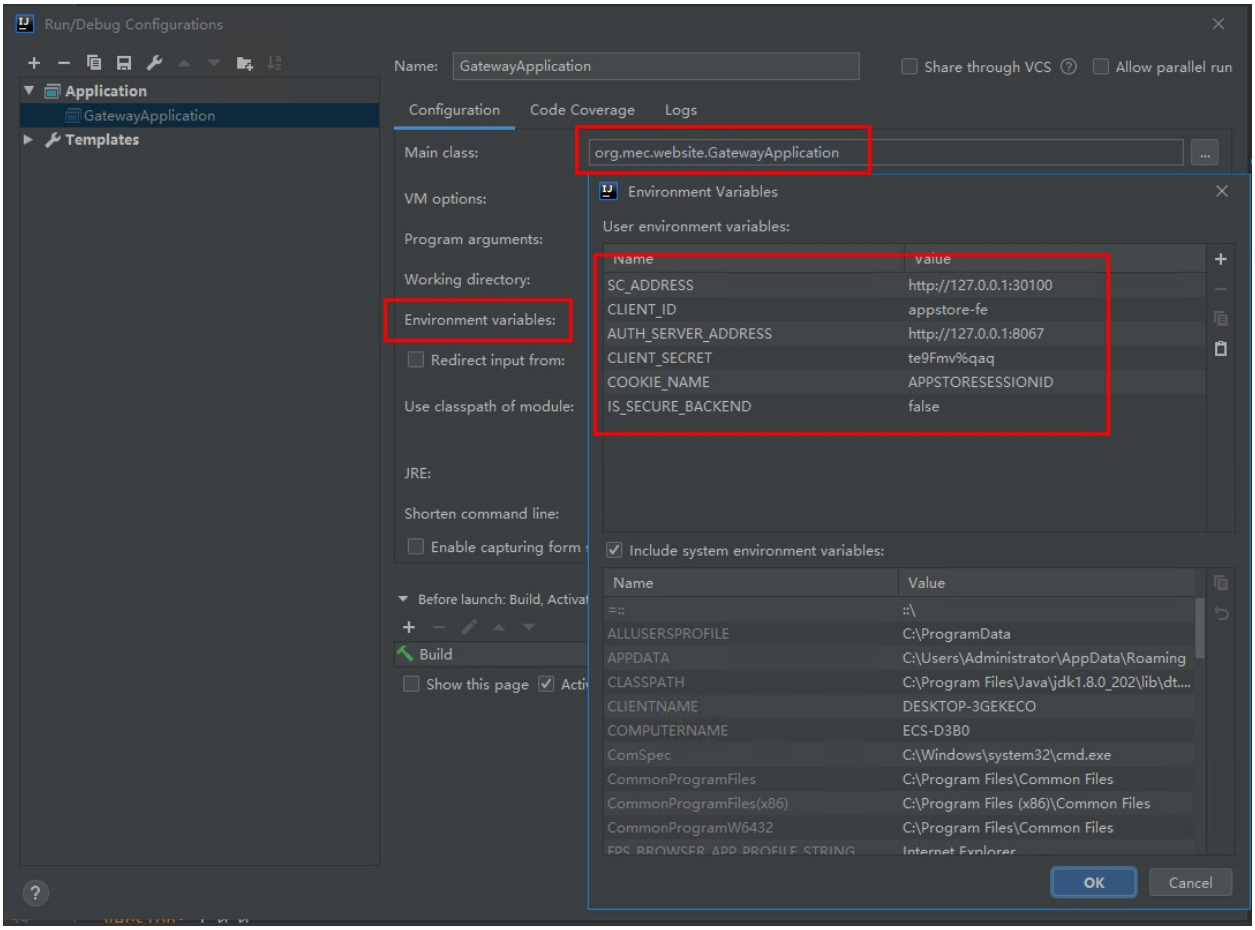
3、下载 website-gateway 代码

4、在 website-gateway 的 src/main/resources/文件夹中创建 static 文件夹

5、将 appstore-fe 编译出的 dist 文件夹中的内容拷贝到 website-gateway 的 static 文件夹中

6、修改 website-gateway 中/src/main/resources/application.yaml 文件内的 `server.port` 为 `8080`(与 usermgmt 启动时填写的 `OAUTH_APPSTORE_CLIENT_URL` 变量中的端口保持一致)

7、在编译器（此处以 IDEA 为例）中配置 GatewayApplication 的运行时环境变量



8、运行 GatewayApplication 启动 WebsiteGateway 与 AppStore-fe，浏览器访问 http://127.0.0.1:8080/ 即可访问到 AppStore 首页

6.5.2 操作指导

AppStore 是开发者发布和上线 App 应用的市场，由 Developer 平台开发的边缘应用，经过测试可以直接分享到 AppStore 应用商店。也可直接上传符合平台规范的应用包，应用包文件类型为 csar。一个 APP 对应多个 Package，使用者可直接下载 APP 的最新版本 Package 或者下载指定版本的 Package。

登录应用平台

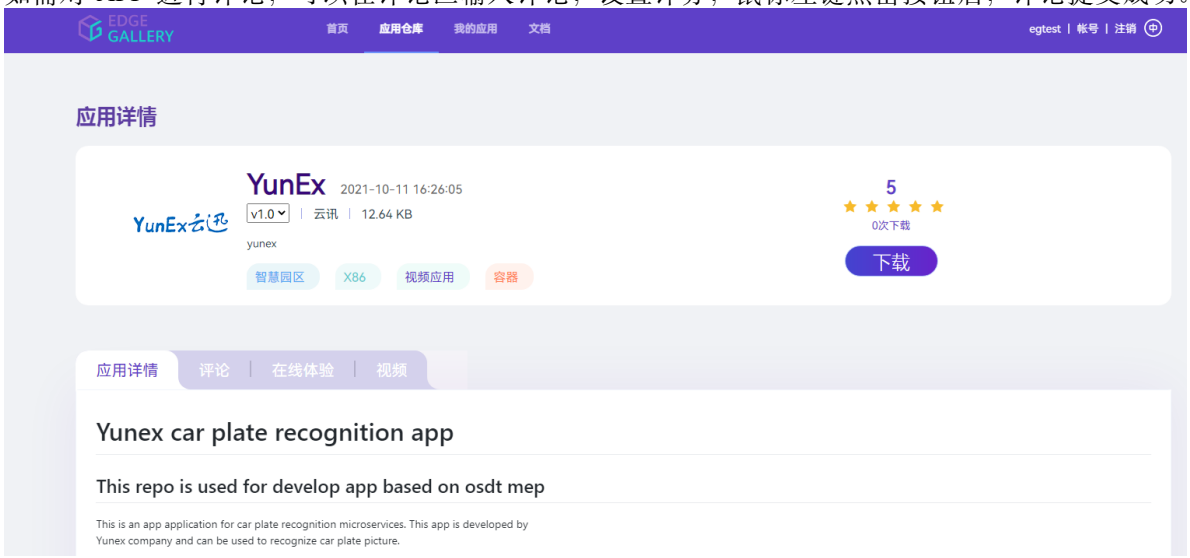
进入网址后，默认为 guest 用户，可以查看信息。如果要进行更多操作（如上传应用）还需注册账号，注册完成，使用该账户登录之后便可进入应用平台首页：

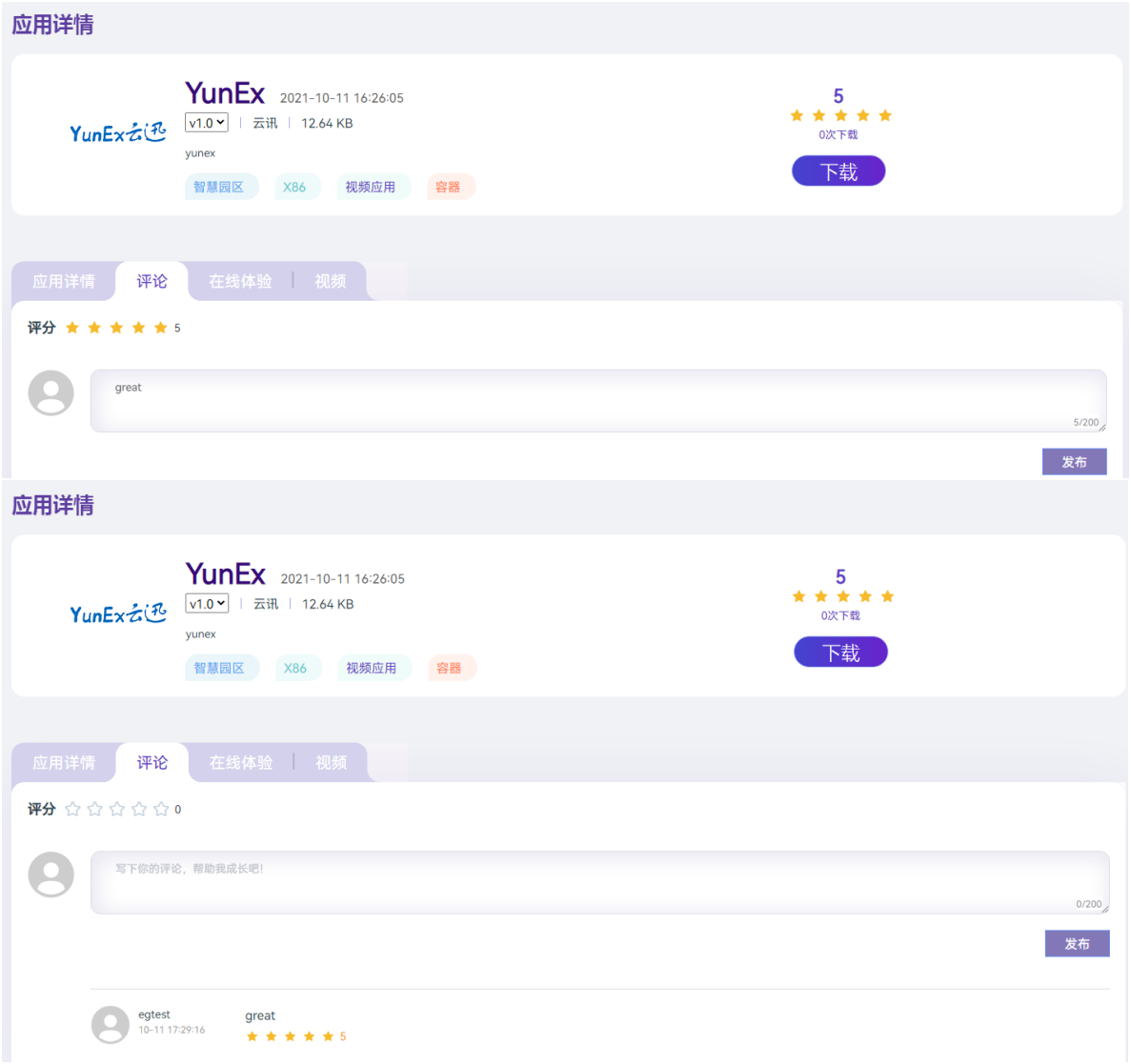
EDGE GALLERY
首页
应用仓库
文档
登录
开放连接行业应用
汇聚各行业最前沿的边缘应用
共享5G智慧互联生活
上传应用
EDGE GALLERY
English
欢迎注册EdgeGallery
用户名
密码
确认密码
邮箱地址
手机号码
同意协议并注册
取消
EDGE GALLERY
帐号登录
用户名 / 邮箱 / 手机
密码
图形验证码
9410
登录
免费注册

查看应用

1. 点击应用查看应用详情，详情界面展示应用版本，行业，类型等基本信息，下侧展示具体介绍，基本信息中的右侧操作中，右侧下载按钮有权限设置，只能下载自己上传的包，管理员用户可以下载所有发布的包和自己上传的包。

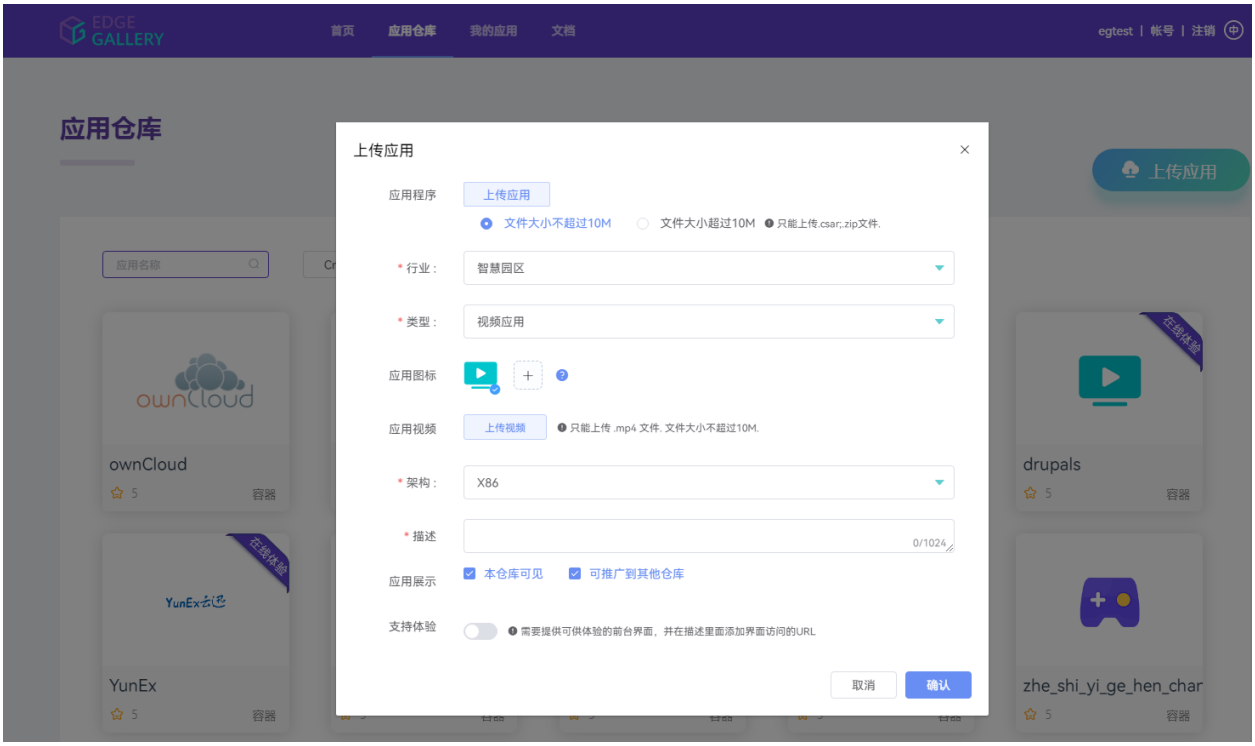
2. 如需对 APP 进行评论，可以在评论区输入评论，设置评分，鼠标左键点击按钮后，评论提交成功。



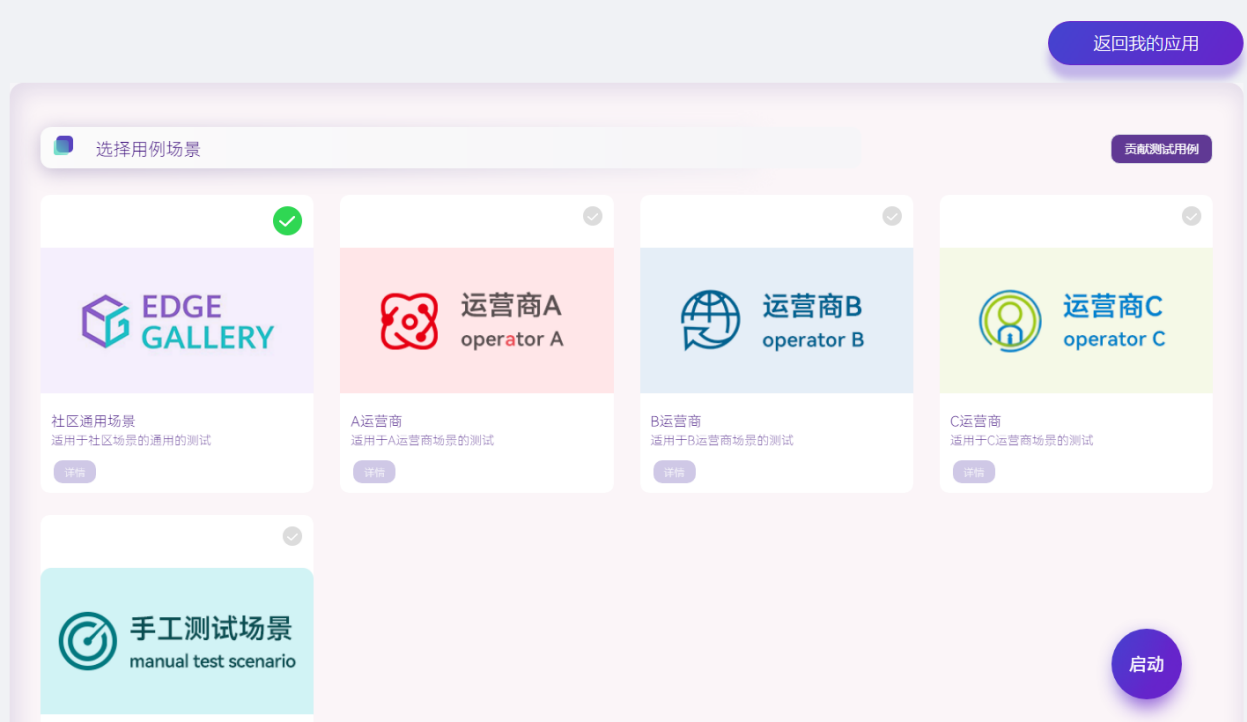


发布应用

1. 上传 APP，打开 APP Store 主界面，点击右侧上传图标，打开上传 APP 对话框。在上传对话框中选择需要上传的 APP，上传或选择系统默认提供的 APP 图标，选择 APP 行业、类型、架构，填写 APP 简单描述，点击确认完成上传。



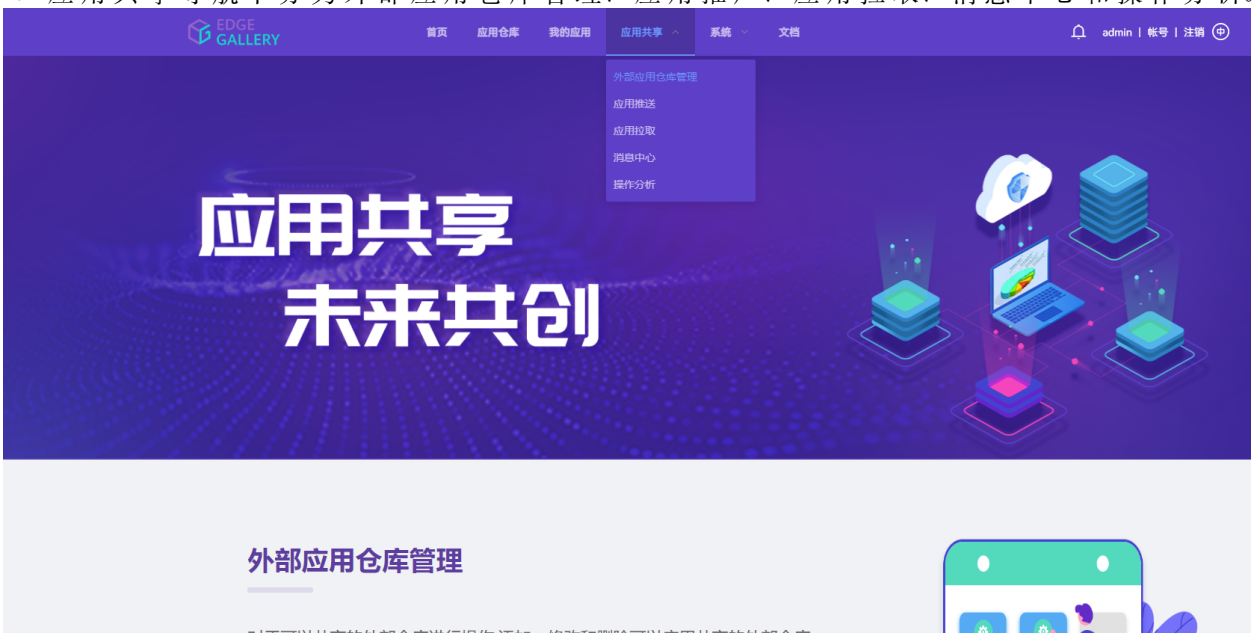
2. 应用上传后，会自动跳转至我的应用界面下，上传的应用需要经过平台的 atp 测试之后才可以发布到仓库。我的应用下是上传的包的基本信息，表格右侧为基本操作，点击测试按钮，会开始创建测试任务，页面为嵌套的 atp 测试页面，点击开始测试按钮，解析包的依赖，可以看到具体的测试用例有多少条，点击确定后开始测试，页面为测试进展页面，可以返回到我的应用页面，等待测完成

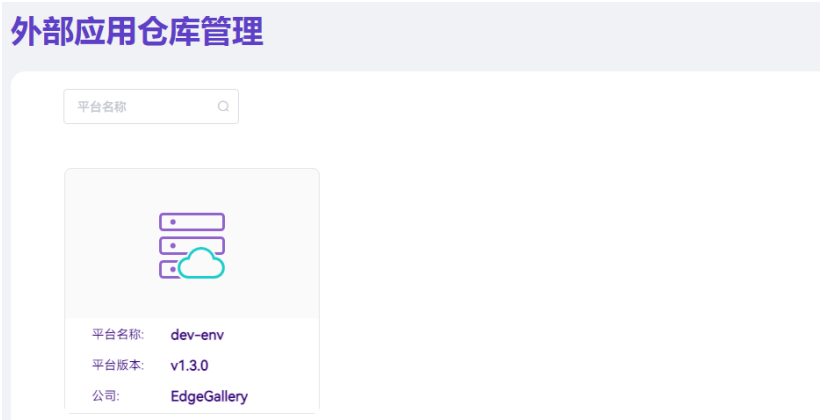


3. 我的应用中测试状态为 Test_success（即测试成功）时才能发布应用，点击右侧的发布按钮，发布成功后，在首页可以看到你上传的应用。

共享应用

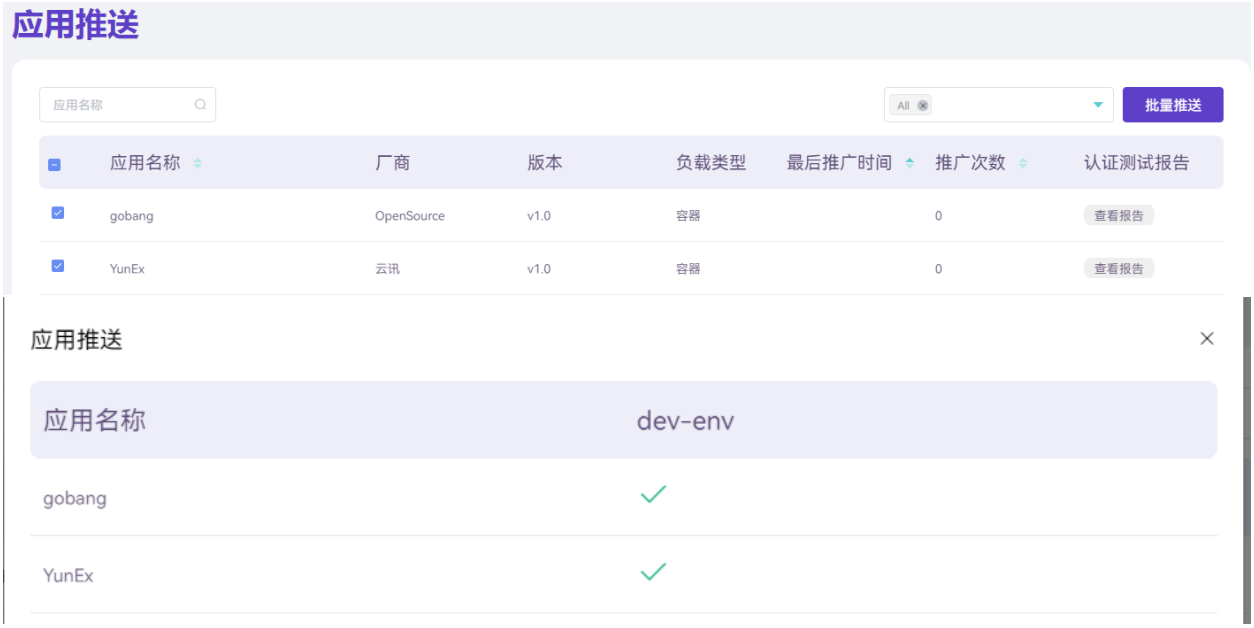
1. 应用共享导航下分为外部应用仓库管理、应用推广、应用拉取、消息中心和操作分析。



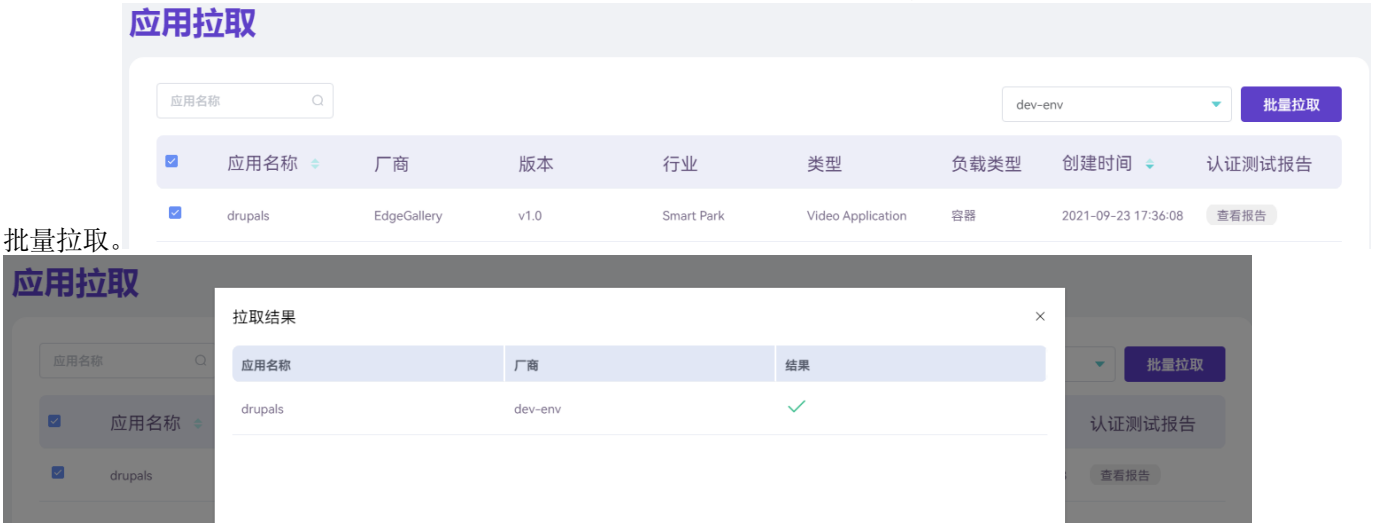


2. 外部应用仓库管理页面,可以添加、修改和删除仓库。

3. 点击应用共享主菜单下的应用推送菜单查看可推送的应用信息，选择需要推送给运营商或者第三方的应用，打开应用推送操作对话框。点击推送操作对话框左下角执行按钮，触发推送任务，然后查看应用推送到运营商或者第三方的结果。

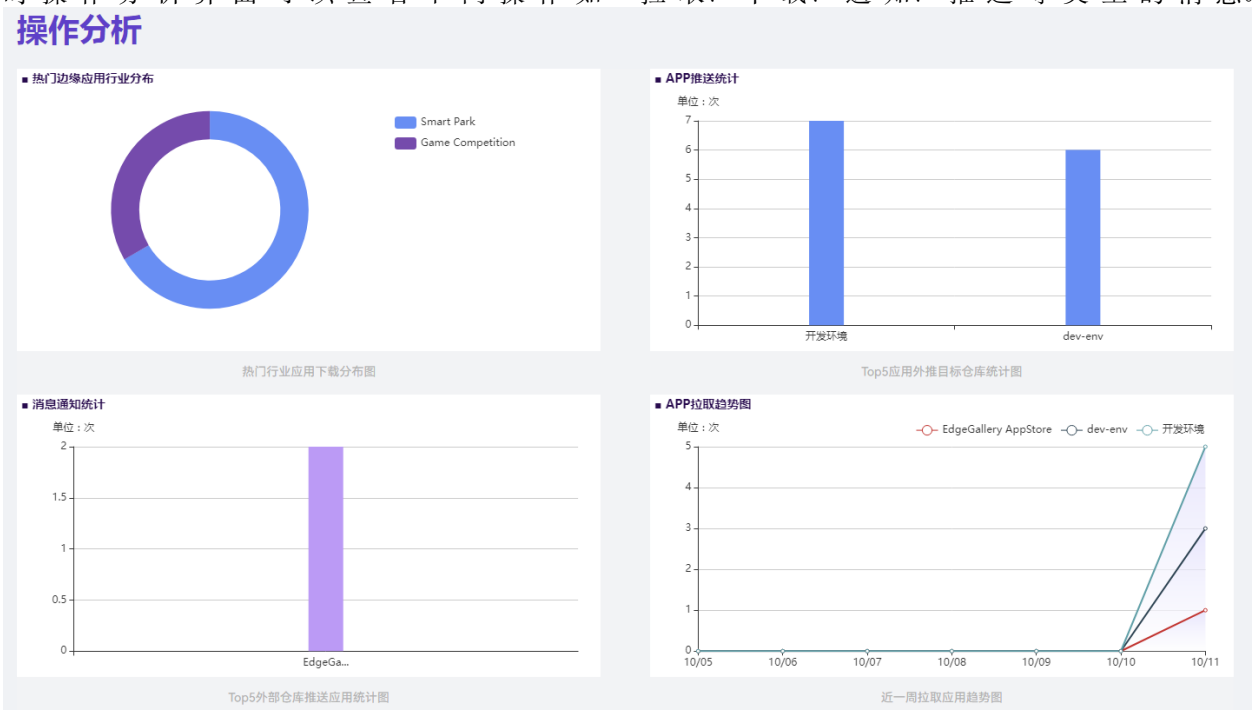


4. 点击应用共享主菜单下的应用拉取菜单查看各个外部仓库可拉取的应用信息，选择需要拉取的应用可进行批量拉取。



5. 点击应用共享主菜单下的消息中心菜单，查看通知信息，可以接受来自不同运营商需要推送的 APP 信息。通过单击消息中心右侧面板右上角的接收按钮可以拉取（下载）该应用的信息，拉取成功的 APP 可以在“我的应用”界面查看，点击删除按钮则删除该条消息，双击可查看消息详情。

6. 应用推送成功后切换到右边操作分析 tab，在操作分析界面查看对应的信息，同时操作分析界面可以查看不同操作如：拉取、下载、通知、推送等类型的消息。



应用名称
应用名称
厂商
版本
消息类型
描述
时间
操作
drupals EdgeGallery v1.0 拉取 EdgeGallery AppStore pull this app fro... 2021-10-11 19:44:08 查看
drupals EdgeGallery v1.0 拉取 EdgeGallery AppStore pull this app fro... 2021-10-11 19:43:48 查看
gobang OpenSource v1.0 推送 from EdgeGallery AppStore to dev-en... 2021-10-11 19:42:13 查看
YunEx 云讯 v1.0 推送 from EdgeGallery AppStore to dev-en... 2021-10-11 19:42:13 查看
gobang OpenSource v1.0 推送 from EdgeGallery AppStore to dev-en... 2021-10-11 19:41:18 查看
YunEx 云讯 v1.0 推送 from EdgeGallery AppStore to dev-en... 2021-10-11 19:41:18 查看
ownCloud OpenSource v1.0 推送 from EdgeGallery AppStore to dev-en... 2021-10-11 19:37:51 查看

CHAPTER 7

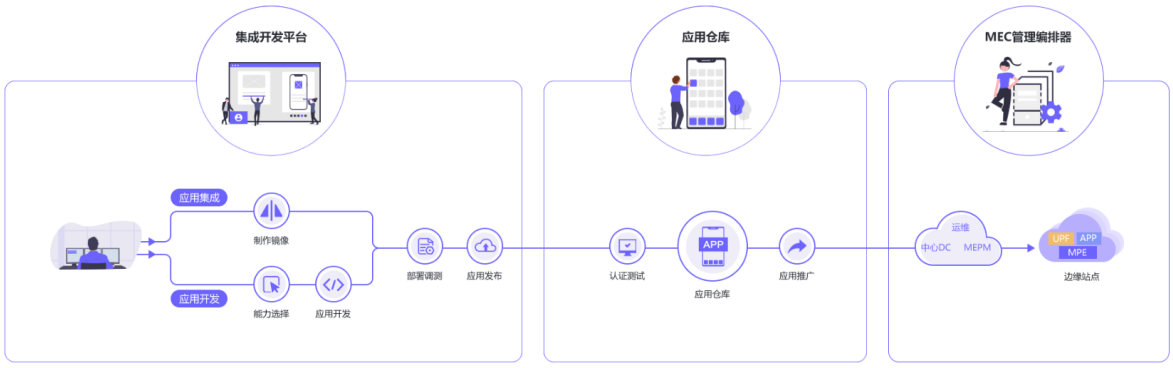
ATP

7.1 概述

- 应用测试服务对于应用包，提供了检测的功能，只有通过了应用测试服务的测试用例，才能将应用包发布到应用商城中。
- 应用测试服务分为前后台两个部分，atp是后台部分，提供主要功能接口供前台或其他三方系统调用，atp-fe是前台部分，提供界面展示。
- 应用测试服务目前分为管理面功能和用户面功能，管理面功能包括测试场景的管理、测试套的管理、测试用例的管理、模型模型批量导入、测试任务的管理、贡献的管理以及配置项的管理，其中贡献管理的菜单仅管理员可见。管理员可以在管理面动态的新增测试场景、测试套和测试用例，修改测试任务中手工用例的状态，还可以下载脚本类型贡献的测试用例，对于合理的用例，将加入到平台的用例集中。用户面功能包括选择要测试的场景、测试过程可视化、测试报告展示、贡献测试用例以及上传自测报告。目前用户面的功能主要集成在开发者平台和应用商店中，对于生成的应用包进行测试。
- 目前测试用例涵盖安全性测试（漏洞扫描、签名校验、端口扫描、病毒扫描等）、遵从性测试（应用包结构校验）、沙箱测试（应用部署、mep 平台服务注册更新等）以及性能测试（时延、cpu 占用率、内存占用率等）。

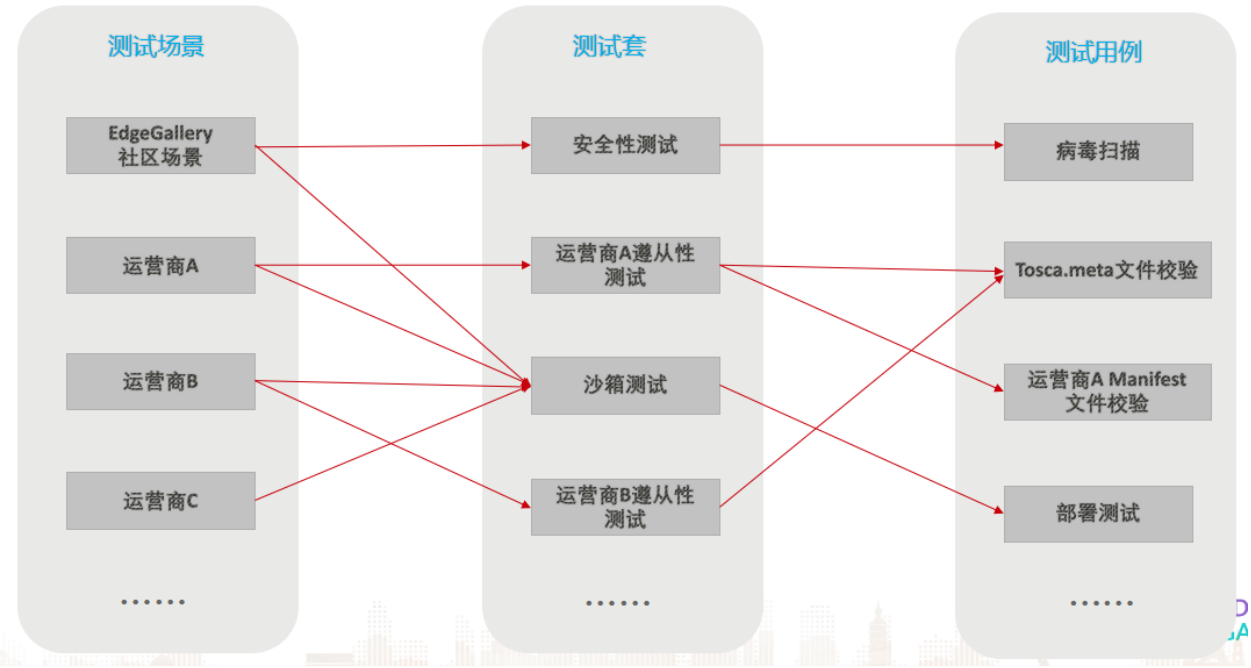
7.2 整体框架说明

EdgeGallery 整体框架流程如下，应用测试平台主要对应图中的认证测试部分。



- atp-fe：开发者平台前台框架，使用 VUE 开发。
- atp：开发者平台后台框架，使用 SpringBoot+ServiceComb 开发。（有关 ServiceComb 请参考这里：https://servicecomb.apache.org/cn/）
- atp DB：后台依赖 Postgres 数据库。

7.3 测试套、测试场景和测试用例的对应关系



7.4 特性设计

ATP 当前提供的特性分为用户面特性和管理面特性。用户面特性包含：选择测试场景、测试任务进展展示、测试报告分析、测试报告下载、贡献测试用例以及用户自测报告导入。管理面特性包含：测试场景管理、测试套管理、测试用例管理、测试模型一键导入、测试任务分析、测试任务管理、贡献管理以及配置项管理。目前测试用例涵盖安全性测试（漏洞扫描、签名校验、端口扫描、病毒扫描等）、遵从性测试（应用包结构校验）、沙箱测试（应用部署、mep 平台服务注册更新等）以及性能测试（时延、cpu 占用率、内存占用率等）。

7.4.1 选择测试场景

用户可以根据需要，选择要进行的测试场景，ATP 服务会执行对应场景的测试用例。

7.4.2 测试任务进展展示

动态展示一条测试任务的执行进展，可以动态看到各个场景下面各个用例的执行情况。

7.4.3 测试报告分析

展示测试报告结果。报告会给出社区场景及所选场景的测试用例结果分析，显示出各个测试套的通过率，用例数量以及具体测试用例的失败原因。

7.4.4 测试报告下载

提供测试报告下载功能，将测试报告作为 pdf 格式下载。

7.4.5 贡献测试用例

用户面支持用户贡献测试用例的入口，支持用户贡献测试用例脚本，或者贡献测试用例的文本描述。

7.4.6 自测报告导入

在自动执行完测试用例后，支持用户上传自测报告。如果用户有功能性测试等应用特定的测试，可以上传自测报告，应用测试平台会将平台用例执行报告和用户上传自测报告整合成一份测试报告展示。

7.4.7 测试场景管理

包含测试场景的展示、新增、删除、修改操作。

7.4.8 测试套管理

包含测试套的展示、新增、删除、修改操作。

7.4.9 测试用例管理

包含测试用例的展示、新增、删除、修改操作。

7.4.10 测试模型一键导入

可以将批量的测试场景、测试套、测试用例定义到 excel 中，测试用例脚本和测试场景图标放到文件夹中，最终将这些文件压缩成 zip 包，在管理面一键式导入批量的测试模型数据。

7.4.11 测试任务分析

ATP 管理面首页展示近 6 个月测试任务数量的分布情况以及总数。

7.4.12 测试任务管理

包含测试任务的展示、以及测试任务的批量删除操作。

7.4.13 贡献管理

包含脚本类型的贡献用例下载、贡献的批量删除操作。

7.4.14 配置项管理

可以针对某个测试用例配置参数，使测试用例更灵活。

7.5 atp Interfaces

- *atp Interfaces*
  - *1. Task*
    - *1.1 POST create test task*
    - *1.2 POST run test task*
    - *1.3 GET get task list*
    - *1.4 GET get one task*
    - *1.5 PUT modify test case status*
    - *1.6 POST batch delete test tasks*
    - *1.7 GET get test tasks analysis*
    - *1.8 DELETE delete one test task*
    - *1.9 POST upload self test report*
  - *2. Test case*
    - *2.1 GET query all test cases*
    - *2.2 POST create test case*
    - *2.3 PUT update test case*
    - *2.4 DELETE delete test case*
    - *2.5 GET query one test case*
    - *2.6 GET download test case*
  - *3. Test scenario*
    - *3.1 GET query all test scenarios*
    - *3.2 POST create test scenarios*
    - *3.3 PUT update test scenarios*
    - *3.4 DELETE delete test scenarios*
    - *3.5 GET query one test case*
    - *3.6 GET query all test cases under one scenario*
  - *4. Test suite*
    - *4.1 GET query all test suites*
    - *4.2 POST create test suite*

        * *4.3 PUT update test suite*
        * *4.4 DELETE delete test suite*
        * *4.5 GET query one test suite*
    - *5. Contribution*
        * *5.1 GET query all contributions*
        * *5.2 POST create contribution*
        * *5.3 POST batch delete contributions*
        * *5.4 GET download contribution script*
    - *6. File*
        * *6.1 GET query one file*
    - *7. Test model*
        * *7.1 POST import test model*
    - *8. configuration*
        * *8.1 POST create config*
        * *8.2 PUT update config*
        * *8.3 DELETE delete config*
        * *8.4 GET query one config*
        * *8.5 GET query all configs*

7.5.1 1. Task

Test task for application package

1.1 POST create test task

create test task for application package.

```
Resource URI: /edgegallery/atp/v1/tasks
```

Name	Definition	Type	Required
file	file	requestpart	yes

Example response:

```
200 OK
{
   "id":"",
   "packagePath":"",
   "appName":"",
   "appVersion":"",
   "providerId":"",
   "status":"",
   "createTime":123456789,
   "user":{
       "userId":"",
       "userName":""
   }
}
```

1.2 POST run test task

Run test task by taskId

```
Resource URI: /edgegallery/atp/v1/tasks/{taskId}/action/run
```

Name	Definition	Type	Required
taskId	taskId	path param	yes
scenarioIdList	scenarioIdList	request param	yes

Example response:

```
200 OK
{
  "id": "8ac6e62d-1d20-4193-942e-ece6deff5e4e",
  "appName": "monitor-service",
  "appVersion": "1.0",
  "status": "running",
  "createTime": "2021-03-01T02:42:29.306+0000",
  "endTime": "2021-03-01T02:43:03.943+0000",
  "reportPath":"/report/68016a29-75da-4f3e-aa00-42ed731dbc3f.pdf",
  "user": {
    "userId": "4eed814b-5d29-4e4c-ba73-51fc6db4ed86",
    "userName": "atp"
  },
  "providerId": "Huawei",
```

(下页继续)

(续上页)


```
  "testScenarios": [
    {
      "id": "e71718a5-864a-49e5-855a-5805a5e9f97d",
      "nameCh": "中国联通",
      "nameEn": "China Unicom Scenario",
      "label": "China Unicom",
      "testSuites": [
        {
          "id": "6d04da1b-1f36-4295-920a-8074f7f9d942",
          "nameCh": "沙箱测试",
          "nameEn": "Sandbox Test",
          "testCases": [
            {
              "id": "4d203173-5555-4f62-aabb-8ebcec357f87",
              "nameCh": "应用实例化",
              "nameEn": "Application Instantiation",
              "descriptionCh": "将应用包部署到边缘节点",
              "descriptionEn": "Instantiate application and its dependency␣
↪application on one edge host",
              "type": "automatic",
              "result": "failed",
              "reason": "host ip is empty"
            }
          ]
        }
      ]
    }
  ]
}
```


1.3 GET get task list

download one plugin

```
Resource URI: /edgegallery/atp/v1/tasks
```

Name	Definition	Type	Required
appName	appName	query param	no
status	status	query param	no
appVersion	appVersion	query param	no
providerId	providerId	query param	no

Example response:

```
200 OK
[{
  "id": "8ac6e62d-1d20-4193-942e-ece6deff5e4e",
  "appName": "monitor-service",
  "appVersion": "1.0",
  "status": "running",
  "createTime": "2021-03-01T02:42:29.306+0000",
  "endTime": "2021-03-01T02:43:03.943+0000",
  "reportPath":"/report/68016a29-75da-4f3e-aa00-42ed731dbc3f.pdf",
  "user": {
    "userId": "4eed814b-5d29-4e4c-ba73-51fc6db4ed86",
    "userName": "atp"
  },
  "providerId": "Huawei",
  "testScenarios": [
    {
      "id": "e71718a5-864a-49e5-855a-5805a5e9f97d",
      "nameCh": "中国联通",
      "nameEn": "China Unicom Scenario",
      "label": "China Unicom",
      "testSuites": [
        {
          "id": "6d04da1b-1f36-4295-920a-8074f7f9d942",
          "nameCh": "沙箱测试",
          "nameEn": "Sandbox Test",
          "testCases": [
            {
              "id": "4d203173-5555-4f62-aabb-8ebcec357f87",
              "nameCh": "应用实例化",
              "nameEn": "Application Instantiation",
              "descriptionCh": "将应用包部署到边缘节点",
              "descriptionEn": "Instantiate application and its dependency␣
↪application on one edge host",
              "type": "automatic",
              "result": "failed",
              "reason": "host ip is empty"
            }
          ]
        }
      ]
    }
  ]
}]
```

1.4 GET get one task

get task by taskId

```
Resource URI: /edgegallery/atp/v1/tasks/{taskId}
```

Name	Definition	Type	Required
taskId	taskId	path param	yes

Example response:

```
200 OK
{
  "id": "8ac6e62d-1d20-4193-942e-ece6deff5e4e",
  "appName": "monitor-service",
  "appVersion": "1.0",
  "status": "running",
  "createTime": "2021-03-01T02:42:29.306+0000",
  "endTime": "2021-03-01T02:43:03.943+0000",
  "reportPath":"/report/68016a29-75da-4f3e-aa00-42ed731dbc3f.pdf",
  "user": {
    "userId": "4eed814b-5d29-4e4c-ba73-51fc6db4ed86",
    "userName": "atp"
  },
  "providerId": "Huawei",
  "testScenarios": [
    {
      "id": "e71718a5-864a-49e5-855a-5805a5e9f97d",
      "nameCh": "中国联通",
      "nameEn": "China Unicom Scenario",
      "label": "China Unicom",
      "testSuites": [
        {
          "id": "6d04da1b-1f36-4295-920a-8074f7f9d942",
          "nameCh": "沙箱测试",
          "nameEn": "Sandbox Test",
          "testCases": [
            {
              "id": "4d203173-5555-4f62-aabb-8ebcec357f87",
              "nameCh": "应用实例化",
              "nameEn": "Application Instantiation",
              "descriptionCh": "将应用包部署到边缘节点",
              "descriptionEn": "Instantiate application and its dependency␣
↪application on one edge host",
```

(下页继续)

(续上页)

```
                "type": "automatic",
                "result": "failed",
                "reason": "host ip is empty"
            }
          ]
        }
      ]
    }
  ]
}
```

1.5 PUT modify test case status

modify manual test case status

```
Resource URI: edgegallery/atp/v1/tasks/{taskId}/testcase
```

Name	Definition	Type	Required
taskId	taskId	path param	yes

Example request:

```
[{
      "testScenarioId":"",
      "testSuiteId":"",
      "testCaseId":"",
      "result":"" ,
      "reason":""
}]
```

Example response:

```
true
```

1.6 POST batch delete test tasks

batch delete test tasks

```
Resource URI: edgegallery/atp/v1/tasks/batch_delete
```

Example request:

```
{
  "taskIds":["id1","id2"]
}
```

Example response:

```
{
   "failed":["id3"]
}
```

1.7 GET get test tasks analysis

get test tasks analysis

```
Resource URI: edgegallery/atp/v1/tasks/action/analysize
```

Example response:

```
200 OK
 {
   "total":426,
   "currentMonth":11,
   "oneMonthAgo":111,
   "twoMonthAgo":12,
   "threeMonthAgo":13,
   "fourMonthAgo":14,
   "fiveMonthAgo":15
}
```

1.8 DELETE delete one test task

delete one test task

```
Resource URI: edgegallery/atp/v1/tasks/{taskId}
```

Example response:

```
true
```

1.9 POST upload self test report

upload self test report

```
Resource URI: edgegallery/atp/v2/tasks/{taskId}/action/upload-report
```

Example request:

```
{
    file: file
}
```

Example response:

```
{
    "data":{
        "path":"reportPath"
    },
    "retCode":0,
    "message":"errorMsg",
    "params":[
        "param1 “
    ]
}
```

7.5.2 2. Test case

test cases of csar package

2.1 GET query all test cases

query all test cases

```
Resource URI: /edgegallery/atp/v1/testcases
```

Name	Definition	Type	Required
type	type	query param	no
name	name	query param	no
locale	locale	query param	no
testSuiteIdList	testSuiteIdList	query param	no

Example response:

```
200 OK
[{
  "id": "4d203173-9999-4f62-aabb-8ebcec357f87",
  "nameCh": "APPD 文件目录校验",
  "nameEn": "APPD File Dir Validation",
  "type": "automatic",
  "descriptionCh": "根目录下必须包含 APPD 文件目录",
  "descriptionEn": "Root directory must contain APPD file dir",
  "hashCode": "",
  "codeLanguage": "java",
  "expectResultCh": "根目录下存在 APPD 文件目录",
  "expectResultEn": "Root directory contains APPD file dir",
  "testStepCh": "1. 打开 csar 包 2. 校验根目录下存在 APPD 目录",
  "testStepEn": "1.open csar package 2.validate root directory contains APPD directory
↪",
  "testSuiteIdList": [
    "522684bd-d6df-4b47-aab8-b43f1b4c19c0"
  ],
  "configIdList": [
    "522684bd-9999-4b47-aab8-b43f1b4c19c0"
  ]
}]
```

2.2 POST create test case

create test case

```
Resource URI: /edgegallery/atp/v1/testcases
```

Name	Definition	Type	Required
nameCh	nameCh	request param	yes
nameEn	nameEn	request param	no
type	type	request param	yes
descriptionCh	descriptionCh	request param	yes
descriptionEn	descriptionEn	request param	no
codeLanguage	codeLanguage	request param	yes
expectResultCh	expectResultCh	request param	yes
expectResultEn	expectResultEn	request param	no
testSuiteIdList	testSuiteIdList	request param	yes
testStepCh	testStepCh	request param	yes
testStepEn	testStepEn	request param	no
file	file	request part	yes
configIdList	configIdList	request param	no

Example response:

```
200 OK
{
  "id": "4d203173-9999-4f62-aabb-8ebcec357f87",
  "nameCh": "APPD 文件目录校验",
  "nameEn": "APPD File Dir Validation",
  "type": "automatic",
  "descriptionCh": "根目录下必须包含 APPD 文件目录",
  "descriptionEn": "Root directory must contain APPD file dir",
  "hashCode": "",
  "codeLanguage": "java",
  "expectResultCh": "根目录下存在 APPD 文件目录",
  "expectResultEn": "Root directory contains APPD file dir",
  "testStepCh": "1. 打开 csar 包 2. 校验根目录下存在 APPD 目录",
  "testStepEn": "1.open csar package 2.validate root directory contains APPD directory
↪",
  "testSuiteIdList": [
    "522684bd-d6df-4b47-aab8-b43f1b4c19c0"
  ],
  "configIdList": [
```

(下页继续)

(续上页)

```
    "522684bd-9999-4b47-aab8-b43f1b4c19c0"
  ]
}
```

2.3 PUT update test case

update test case

```
Resource URI: /edgegallery/atp/v1/testcases
```

Name	Definition	Type	Required
id	id	request param	yes
type	type	request param	no
descriptionCh	descriptionCh	request param	no
descriptionEn	descriptionEn	request param	no
codeLanguage	codeLanguage	request param	no
expectResultCh	expectResultCh	request param	no
expectResultEn	expectResultEn	request param	no
testSuiteIdList	testSuiteIdList	request param	no
testStepCh	testStepCh	request param	no
testStepEn	testStepEn	request param	no
file	file	request part	no
configIdList	configIdList	request param	no

Example response:

```
200 OK
{
  "id": "4d203173-9999-4f62-aabb-8ebcec357f87",
  "nameCh": "APPD 文件目录校验",
  "nameEn": "APPD File Dir Validation",
  "type": "automatic",
  "descriptionCh": "根目录下必须包含 APPD 文件目录",
  "descriptionEn": "Root directory must contain APPD file dir",
  "hashCode": "",
  "codeLanguage": "java",
  "expectResultCh": "根目录下存在 APPD 文件目录",
  "expectResultEn": "Root directory contains APPD file dir",
  "testStepCh": "1. 打开 csar 包 2. 校验根目录下存在 APPD 目录",
  "testStepEn": "1.open csar package 2.validate root directory contains APPD directory
↪",
```

(下页继续)

(续上页)

```
  "testSuiteIdList": [
    "522684bd-d6df-4b47-aab8-b43f1b4c19c0"
  ],
  "configIdList": [
    "522684bd-9999-4b47-aab8-b43f1b4c19c0"
  ]
}
```

2.4 DELETE delete test case

delete test case

```
Resource URI: /edgegallery/atp/v1/testcases/{id}
```

Name	Definition	Type	Required
id	id	path param	yes

Example response:

```
200 OK
true
```

2.5 GET query one test case

query one test case

```
Resource URI: /edgegallery/atp/v1/testcases/{id}
```

Name	Definition	Type	Required
id	id	path param	yes

Example response:

```
200 OK
{
  "id": "4d203173-9999-4f62-aabb-8ebcec357f87",
  "nameCh": "APPD 文件目录校验",
  "nameEn": "APPD File Dir Validation",
  "type": "automatic",
  "descriptionCh": "根目录下必须包含 APPD 文件目录",
```

(下页继续)

(续上页)


```
  "descriptionEn": "Root directory must contain APPD file dir",
  "hashCode": "",
  "codeLanguage": "java",
  "expectResultCh": "根目录下存在 APPD 文件目录",
  "expectResultEn": "Root directory contains APPD file dir",
  "testStepCh": "1. 打开 csar 包 2. 校验根目录下存在 APPD 目录",
  "testStepEn": "1.open csar package 2.validate root directory contains APPD directory
↪",
  "testSuiteIdList": [
    "522684bd-d6df-4b47-aab8-b43f1b4c19c0"
  ],
  "configIdList": [
    "522684bd-9999-4b47-aab8-b43f1b4c19c0"
  ]
}
```


2.6 GET download test case

download test case

```
Resource URI: edgegallery/atp/v1/testcases/{id}/action/download
```

Name	Definition	Type	Required
id	id	path param	yes

Example response:

```
200 OK
binary stream
```

7.5.3 3. Test scenario

test scenario management

3.1 GET query all test scenarios

query all test scenarios

```
Resource URI: edgegallery/atp/v1/testscenarios
```

Name	Definition	Type	Required
name	name	query param	no
locale	locale	query param	no

Example response:

```
200 OK
[
  {
    "id": "4d203111-1111-4f62-aabb-8ebcec357f87",
    "nameCh": "社区场景",
    "nameEn": "EdgeGallery Community Scenario",
    "descriptionCh": "适用于社区场景的测试",
    "descriptionEn": "suite for EdgeGallery community test",
    "label": "EdgeGallery"
  }
]
```

3.2 POST create test scenario

create test scenario

```
Resource URI: edgegallery/atp/v1/testscenarios
```

Name	Definition	Type	Required
nameCh	nameCh	request param	yes
nameEn	nameEn	request param	no
descriptionCh	descriptionCh	request param	yes
descriptionEn	descriptionEn	request param	no
label	label	request param	yes
icon	icon	request part	yes

Example response:

```
200 OK
  {
    "id": "4d203111-1111-4f62-aabb-8ebcec357f87",
    "nameCh": "社区场景",
    "nameEn": "EdgeGallery Community Scenario",
    "descriptionCh": "适用于社区场景的测试",
    "descriptionEn": "suite for EdgeGallery community test",
    "label": "EdgeGallery"
  }
```

3.3 PUT update test scenario

update test scenario

```
Resource URI: edgegallery/atp/v1/testscenarios/{id}
```

Name	Definition	Type	Required
id	id	path param	yes
nameCh	nameCh	request param	no
nameEn	nameEn	request param	no
descriptionCh	descriptionCh	request param	no
descriptionEn	descriptionEn	request param	no
label	label	request param	no
icon	icon	request part	no

Example response:

```
200 OK
  {
    "id": "4d203111-1111-4f62-aabb-8ebcec357f87",
    "nameCh": "社区场景",
    "nameEn": "EdgeGallery Community Scenario",
    "descriptionCh": "适用于社区场景的测试",
    "descriptionEn": "suite for EdgeGallery community test",
    "label": "EdgeGallery"
  }
```

3.4 DELETE delete test scenario

delete test scenario

```
Resource URI: edgegallery/atp/v1/testscenarios/{id}
```

Name	Definition	Type	Required
id	id	path param	yes

Example response:

```
200 OK
true
```

3.5 GET query one test scenario

query one test scenario

```
Resource URI: edgegallery/atp/v1/testscenarios/{id}
```

Name	Definition	Type	Required
id	id	path param	yes

Example response:

```
200 OK
  {
    "id": "4d203111-1111-4f62-aabb-8ebcec357f87",
    "nameCh": "社区场景",
    "nameEn": "EdgeGallery Community Scenario",
    "descriptionCh": "适用于社区场景的测试",
    "descriptionEn": "suite for EdgeGallery community test",
    "label": "EdgeGallery"
  }
```

3.6 GET query all test cases under one scenario

query all test cases under one scenario

```
Resource URI: edgegallery/atp/v1/testscenarios/testcases
```

Name	Definition	Type	Required
scenarioids	scenarioids	request param	yes

Example request:

```
200 OK
"scenarioIds":["id", "id2"]
```

Example response:

```
200 OK
[
  {
    "id": "96a82e85-d40d-4ce5-beec-2dd1c9a3d41d",
    "nameCh": "中国电信",
    "nameEn": "China Telecom Scenario",
    "descriptionCh": "适用于中国电信场景的测试",
    "descriptionEn": "suite for China Telecom test",
    "label": "China Telecom",
    "testSuites": [
      {
        "id": "6d04da1b-1f36-4295-920a-8074f7f9d942",
        "nameCh": "沙箱测试",
        "nameEn": "Sandbox Test",
        "descriptionCh": "应用包部署测试",
        "descriptionEn": "App package deployment test",
        "scenarioIdList": [
          "4d203111-1111-4f62-aabb-8ebcec357f87",
          "e71718a5-864a-49e5-855a-5805a5e9f97d",
          "6fe8581c-b83f-40c2-8f5b-505478f9e30b",
          "96a82e85-d40d-4ce5-beec-2dd1c9a3d41d"
        ],
        "testCases": [
          {
            "id": "4d203173-5555-4f62-aabb-8ebcec357f87",
            "nameCh": "应用实例化",
            "nameEn": "Application Instantiation",
            "type": "automatic",
```

(下页继续)

(续上页)

```
            "descriptionCh": "将应用包部署到边缘节点",
            "descriptionEn": "Instantiate application and its dependency application␣
↪on one edge host",
            "hashCode": "",
            "codeLanguage": "jar",
            "expectResultCh": "应用包可以成功部署",
            "expectResultEn": "app can instantiate successfully.",
            "testStepCh": "部署应用包到对应的边缘节点",
            "testStepEn": "Deploy application package to edge node",
            "testSuiteIdList": [
              "6d04da1b-1f36-4295-920a-8074f7f9d942"
            ]
          }
        ]
      }
    ]
  }
]
```

7.5.4 4. Test suite

test suite management

4.1 GET query all test suite

query all test suite

```
Resource URI: edgegallery/atp/v1/testsuites
```

Name	Definition	Type	Required
name	name	query param	no
locale	locale	query param	no
scenarioIdList	scenarioIdList	query param	no

Example response:

```
200 OK
[
  {
    "id": "522684bd-d6df-4b47-aab8-b43f1b4c19c0",
    "nameCh": "遵从性测试",
```

(下页继续)

(续上页)

```
    "nameEn": "Compliance Test",
    "descriptionCh": "遵从社区 APPD 标准、ETSI 标准对应用包结构进行校验",
    "descriptionEn": "Validate app package structure according to commnunity and ETSI␣
↪standard",
    "scenarioIdList": [
      "4d203111-1111-4f62-aabb-8ebcec357f87"
    ]
  }
]
```

4.2 POST create test suite

create test suite

```
Resource URI: edgegallery/atp/v1/testsuites
```

Name	Definition	Type	Required
nameCh	nameCh	request param	yes
nameEn	nameEn	request param	no
descriptionCh	descriptionCh	request param	yes
descriptionEn	descriptionEn	request param	no
scenarioIdList	scenarioIdList	request param	yes

Example response:

```
200 OK
{
  "id": "743abd93-57a3-499d-9591-fa7db86a4778",
  "nameCh": "安全性测试",
  "nameEn": "Security Test",
  "descriptionCh": "应用包安全测试",
  "descriptionEn": "App package security test",
  "scenarioIdList": [
    "4d203111-1111-4f62-aabb-8ebcec357f87"
  ]
}
```

4.3 PUT update test suite

update test suite

```
Resource URI: edgegallery/atp/v1/testsuites/{id}
```

Name	Definition	Type	Required
id	id	path param	yes
nameCh	nameCh	request param	no
nameEn	nameEn	request param	no
descriptionCh	descriptionCh	request param	no
descriptionEn	descriptionEn	request param	no
scenarioIdList	scenarioIdList	request param	no

Example response:

```
200 OK
{
  "id": "743abd93-57a3-499d-9591-fa7db86a4778",
  "nameCh": "安全性测试",
  "nameEn": "Security Test",
  "descriptionCh": "应用包安全测试",
  "descriptionEn": "App package security test",
  "scenarioIdList": [
    "4d203111-1111-4f62-aabb-8ebcec357f87"
  ]
}
```

4.4 DELETE delete test suite

delete test suite

```
Resource URI: edgegallery/atp/v1/testsuites/{id}
```

Name	Definition	Type	Required
id	id	path param	yes

Example response:

```
200 OK
true
```

4.5 GET query one test suite

query one test suite

```
Resource URI: edgegallery/atp/v1/testsuites/{id}
```

Name	Definition	Type	Required
id	id	path param	yes

Example response:

```
200 OK
{
  "id": "743abd93-57a3-499d-9591-fa7db86a4778",
  "nameCh": "安全性测试",
  "nameEn": "Security Test",
  "descriptionCh": "应用包安全测试",
  "descriptionEn": "App package security test",
  "scenarioIdList": [
    "4d203111-1111-4f62-aabb-8ebcec357f87"
  ]
}
```

7.5.5 5. Contribution

contribution

5.1 GET query all contribution

query all contribution

```
Resource URI: edgegallery/atp/v1/contributions
```

Example response:

```
200 OK
[
  {
    "id": "872b55f8-bfb8-4fb4-be4e-0b643ff375e2",
    "name": "test",
    "objective": "test",
    "step": "test",
```

(下页继续)

(续上页)

```
    "expectResult": "test",
    "type": "text",
    "createTime": "2021-03-01T09:17:31.915+0000"
  }
]
```

5.2 POST create contribution

create contribution

```
Resource URI: edgegallery/atp/v1/contributions
```

Name	Definition	Type	Required
name	name	request param	yes
objective	objective	request param	no
step	step	request param	yes
expectResult	expectResult	request param	no
type	type	request param	yes
file	file	requestpart	yes

Example response:

```
200 OK
  {
    "id": "872b55f8-bfb8-4fb4-be4e-0b643ff375e2",
    "name": "test",
    "objective": "test",
    "step": "test",
    "expectResult": "test",
    "type": "text",
    "createTime": "2021-03-01T09:17:31.915+0000"
  }
```

5.3 POST batch delete contributions

batch delete contributions

```
Resource URI: edgegallery/atp/v1/contributions/batch_delete
```

Name	Definition	Type	Required
ids	ids	request param	yes

Example request:

```
{
  "ids":["id1","id2"]
}
```

Example response:

```
200 OK
{
    "failed":["id3"]
}
```

5.4 GET download contribution script

download contribution script

```
Resource URI: edgegallery/atp/v1/contributions/{id}/action/download
```

Example response:

```
200 OK
binary stream
```

7.5.6 6. File

file

6.1 GET query one file

query one file

```
Resource URI: edgegallery/atp/v1/files/{id}
```

Name	Definition	Type	Required
id	id	path param	yes
type	type	query param	no

Example response:

```
200 OK
binary stream
```

7.5.7 7. Test model

test model

7.1 POST import test model

import test model

```
Resource URI: edgegallery/atp/v1/testmodels/action/import
```

Name	Definition	Type	Required
file	file	request param	yes

Example response:

```
200 OK
{
  "retCode" : 0
  "failures" : []
}

206 partial success
{
  "retCode" : 5000,
  "failures" : [{
       "id" : "string",
```

(下页继续)

(续上页)

```
        "nameEn" : "string",
        "type" : "testScenario",
        "errCode": 1000,
         "errMsg":  “string” ,
        "params": "string"
      },
      {
        "id" : "string",
        "nameEn" : "string",
        "type" : "testSuite",
        "errCode": 1000,
       "errMsg":  “string” ,
        "params": "string"
      },
      {
        "id" : "string",
        "nameEn" : "string",
        "type" : "testCase",
        "errCode": 1000,
        "errMsg":  “string” ,
        "params": "string"
      }
  ]
}
```

7.5.8 8. Configuration

config

8.1 POST create config

create config

```
Resource URI: edgegallery/atp/v2/configs
```

Example request:

```
{
    "nameCh":"虚机应用实例化",
    "nameEn":"Instantiation for vm app",
    "descriptionCh":"配置虚机应用实例化所需参数",
```

(下页继续)

(续上页)

```
    "descriptionEn":"Configure params for instantiation of vm app",
    "configuration":"DC_ID = FS+M;az_dc=nova;"
}
```

Example response:

```
200 OK
{
    "data":{
        "id":"uuid",
        "nameCh":"虚机应用实例化",
        "nameEn":"Instantiation for vm app",
        "descriptionCh":"配置虚机应用实例化所需参数",
        "descriptionEn":"Configure params for instantiation of vm app",
        "configuration":"DC_ID = FS+M;az_dc=nova;",
        "createTime":""
    },
    "retCode":0,
    "message":"errorMsg",
    "params":[
        "param1",
        "param2"
    ]
}
```

8.2 PUT update config

update config

```
Resource URI: edgegallery/atp/v2/configs/{id}
```

Example request:

```
{
    "nameCh":"虚机应用实例化",
    "nameEn":"Instantiation for vm app",
    "descriptionCh":"配置虚机应用实例化所需参数",
    "descriptionEn":"Configure params for instantiation of vm app",
    "configuration":"DC_ID = FS+M;az_dc=nova;"
}
```

Example response:

```
200 OK
{
    "data":{
        "id":"uuid",
        "nameCh":"虚机应用实例化",
        "nameEn":"Instantiation for vm app",
        "descriptionCh":"配置虚机应用实例化所需参数",
        "descriptionEn":"Configure params for instantiation of vm app",
        "configuration":"DC_ID = FS+M;az_dc=nova;",
        "createTime":""
    },
    "retCode":0,
    "message":"errorMsg",
    "params":[
        "param1",
        "param2"
    ]
}
```

8.3 DELETE delete config

delete config

```
Resource URI: edgegallery/atp/v2/configs/{id}
```

Name	Definition	Type	Required
id	id	path param	yes

Example request:

```
NA
```

Example response:

```
200 OK
true
```

8.4 GET query one config

query one config

```
Resource URI: edgegallery/atp/v2/configs/{id}
```

Name	Definition	Type	Required
id	id	path param	yes

Example request:

```
NA
```

Example response:

```
200 OK
{
    "data":{
        "id":"uuid",
        "nameCh":"虚机应用实例化",
        "nameEn":"Instantiation for vm app",
        "descriptionCh":"配置虚机应用实例化所需参数",
        "descriptionEn":"Configure params for instantiation of vm app",
        "configuration":"DC_ID = FS+M;az_dc=nova;",
        "createTime":""
    },
    "retCode":0,
    "message":"errorMsg",
    "params":[
        "param1",
        "param2"
    ]
}
```

8.5 GET query all configs

query all configs

```
Resource URI: edgegallery/atp/v2/configs?limit=10&offset=0
```

Name	Definition	Type	Required
name	name	query param	no
locale	locale	query param	no
limit	limit	query param	yes
offset	offset	query param	yes

Example request:

```
NA
```

Example response:

```
200 OK
{
    "results":[
        {
            "id":"uuid",
            "nameCh":"虚机应用实例化",
            "nameEn":"Instantiation for vm app",
            "descriptionCh":"配置虚机应用实例化所需参数",
            "descriptionEn":"Configure params for instantiation of vm app",
            "configuration":"DC_ID = FS+M;az_dc=nova;",
            "createTime":""
        }
    ],
    "total":1,
    "limit":10,
    "offset":0
}
```

7.6 atp v2 Interfaces

- *atp v2 Interfaces*
  - *1. Task*
    - *1.1 POST create test task*
    - *1.2 POST run test task*
    - *1.3 GET get task list*
    - *1.4 GET get one task*
    - *1.5 PUT modify test case status*
    - *1.6 POST batch delete test tasks*
    - *1.7 GET get test tasks analysis*
    - *1.8 DELETE delete one test task*
    - *1.9 POST upload self test report*
  - *2. Test case*
    - *2.1 GET query all test cases*
    - *2.2 POST create test case*
    - *2.3 PUT update test case*
    - *2.4 DELETE delete test case*
    - *2.5 GET query one test case*
    - *2.6 GET download test case*
  - *3. Test scenario*
    - *3.1 GET query all test scenarios*
    - *3.2 POST create test scenarios*
    - *3.3 PUT update test scenarios*
    - *3.4 DELETE delete test scenarios*
    - *3.5 GET query one test case*
    - *3.6 GET query all test cases under one scenario*
  - *4. Test suite*
    - *4.1 GET query all test suites*
    - *4.2 POST create test suite*

        * *4.3 PUT update test suite*
        * *4.4 DELETE delete test suite*
        * *4.5 GET query one test suite*
    - *5. Contribution*
        * *5.1 GET query all contributions*
        * *5.2 POST create contribution*
        * *5.3 POST batch delete contributions*
        * *5.4 GET download contribution script*
    - *6. File*
        * *6.1 GET query one file*
    - *7. Test model*
        * *7.1 POST import test model*
    - *8. configuration*
        * *8.1 POST create config*
        * *8.2 PUT update config*
        * *8.3 DELETE delete config*
        * *8.4 GET query one config*
        * *8.5 GET query all configs*

7.6.1 1. Task

Test task for application package

1.1 POST create test task

create test task for application package.

```
Resource URI: /edgegallery/atp/v2/tasks
```

Name	Definition	Type	Required
file	file	requestpart	yes

Example response:

```
200 OK
{
    "data":{
        "id":"",
        "packagePath":"",
        "appName":"",
        "appVersion":"",
        "providerId":"",
        "status":"",
        "createTime":123456789,
        "user":{
            "userId":"",
            "userName":""
        }
    },
    "retCode":0,
    "message":"errorMsg",
    "params":[
        "param1",
        "param2"
    ]
}
```

1.2 POST run test task

Run test task by taskId

```
Resource URI: /edgegallery/atp/v2/tasks/{taskId}/action/run
```

Name	Definition	Type	Required
taskId	taskId	path param	yes
scenarioIdList	scenarioIdList	request param	yes

Example response:

```
200 OK
{
    "data":{
        "id":"8ac6e62d-1d20-4193-942e-ece6deff5e4e",
        "appName":"monitor-service",
        "appVersion":"1.0",
```

(下页继续)

(续上页)


```
        "status":"running",
        "createTime":"2021-03-01T02:42:29.306+0000",
        "endTime":"2021-03-01T02:43:03.943+0000",
        "reportPath":"/report/68016a29-75da-4f3e-aa00-42ed731dbc3f.pdf",
        "user":{
            "userId":"4eed814b-5d29-4e4c-ba73-51fc6db4ed86",
            "userName":"atp"
        },
        "providerId":"Huawei",
        "testScenarios":[
            {
                "id":"e71718a5-864a-49e5-855a-5805a5e9f97d",
                "nameCh":"中国联通",
                "nameEn":"China Unicom Scenario",
                "label":"China Unicom",
                "testSuites":[
                    {
                        "id":"6d04da1b-1f36-4295-920a-8074f7f9d942",
                        "nameCh":"沙箱测试",
                        "nameEn":"Sandbox Test",
                        "testCases":[
                            {
                                "id":"4d203173-5555-4f62-aabb-8ebcec357f87",
                                "nameCh":"应用实例化",
                                "nameEn":"Application Instantiation",
                                "descriptionCh":"将应用包部署到边缘节点",
                                "descriptionEn":"Instantiate application and its␣
↪dependency application on one edge host",
                                "type":"automatic",
                                "result":"failed",
                                "reason":"host ip is empty"
                            }
                        ]
                    }
                ]
            }
        ]
    },
    "retCode":0,
    "message":"errorMsg",
    "params":[
        "param1"
    ]
```


(下页继续)

(续上页)

```
}
```

1.3 GET get task list

download one plugin

```
Resource URI: /edgegallery/atp/v2/tasks?limit=10&offset=0
```

Name	Definition	Type	Required
appName	appName	query param	no
status	status	query param	no
appVersion	appVersion	query param	no
providerId	providerId	query param	no
limit	limit	query param	yes
offset	offset	query param	yes

Example response:

```
200 OK
{
    "results":[
        {
            "id":"8ac6e62d-1d20-4193-942e-ece6deff5e4e",
            "appName":"monitor-service",
            "appVersion":"1.0",
            "status":"running",
            "createTime":"2021-03-01T02:42:29.306+0000",
            "endTime":"2021-03-01T02:43:03.943+0000",
            "reportPath":"/report/68016a29-75da-4f3e-aa00-42ed731dbc3f.pdf",
            "user":{
                "userId":"4eed814b-5d29-4e4c-ba73-51fc6db4ed86",
                "userName":"atp"
            },
            "providerId":"Huawei",
            "testScenarios":[
                {
                    "id":"e71718a5-864a-49e5-855a-5805a5e9f97d",
                    "nameCh":"中国联通",
                    "nameEn":"China Unicom Scenario",
                    "label":"China Unicom",
```

(下页继续)

(续上页)


```
                    "testSuites":[
                        {
                            "id":"6d04da1b-1f36-4295-920a-8074f7f9d942",
                            "nameCh":"沙箱测试",
                            "nameEn":"Sandbox Test",
                            "testCases":[
                                {
                                    "id":"4d203173-5555-4f62-aabb-8ebcec357f87",
                                    "nameCh":"应用实例化",
                                    "nameEn":"Application Instantiation",
                                    "descriptionCh":"将应用包部署到边缘节点",
                                    "descriptionEn":"Instantiate application and its
↪dependency application on one edge host",
                                    "type":"automatic",
                                    "result":"failed",
                                    "reason":"host ip is empty"
                                }
                            ]
                        }
                    ]
                }
            ]
        }
    ],
    "total":1,
    "limit":10,
    "offset":0
}
```


1.4 GET get one task

get task by taskId

```
Resource URI: /edgegallery/atp/v2/tasks/{taskId}
```

Name	Definition	Type	Required
taskId	taskId	path param	yes

Example response:


```
200 OK
{
```


(下页继续)

(续上页)


```
    "data":{
        "id":"8ac6e62d-1d20-4193-942e-ece6deff5e4e",
        "appName":"monitor-service",
        "appVersion":"1.0",
        "status":"running",
        "createTime":"2021-03-01T02:42:29.306+0000",
        "endTime":"2021-03-01T02:43:03.943+0000",
        "reportPath":"/report/68016a29-75da-4f3e-aa00-42ed731dbc3f.pdf",
        "user":{
            "userId":"4eed814b-5d29-4e4c-ba73-51fc6db4ed86",
            "userName":"atp"
        },
        "providerId":"Huawei",
        "testScenarios":[
            {
                "id":"e71718a5-864a-49e5-855a-5805a5e9f97d",
                "nameCh":"中国联通",
                "nameEn":"China Unicom Scenario",
                "label":"China Unicom",
                "testSuites":[
                    {
                        "id":"6d04da1b-1f36-4295-920a-8074f7f9d942",
                        "nameCh":"沙箱测试",
                        "nameEn":"Sandbox Test",
                        "testCases":[
                            {
                                "id":"4d203173-5555-4f62-aabb-8ebcec357f87",
                                "nameCh":"应用实例化",
                                "nameEn":"Application Instantiation",
                                "descriptionCh":"将应用包部署到边缘节点",
                                "descriptionEn":"Instantiate application and its␣
↪dependency application on one edge host",
                                "type":"automatic",
                                "result":"failed",
                                "reason":"host ip is empty"
                            }
                        ]
                    }
                ]
            }
        ]
    },
    "retCode":0,
```


(下页继续)

(续上页)

```
    "message":"errorMsg",
    "params":[
        "param1"
    ]
}
```

1.5 PUT modify test case status

modify manual test case status

```
Resource URI: edgegallery/atp/v2/tasks/{taskId}/testcase
```

Name	Definition	Type	Required
taskId	taskId	path param	yes

Example request:

```
[{
      "testScenarioId":"",
      "testSuiteId":"",
      "testCaseId":"",
      "result":"" ,
      "reason":""
}]
```

Example response:

```
true
```

1.6 POST batch delete test tasks

batch delete test tasks

```
Resource URI: edgegallery/atp/v2/tasks/batch_delete
```

Example request:

```
{
  "taskIds":["id1","id2"]
}
```

Example response:

```
{
    "retCode":5000,
    "failures":[
        {
            "id":"string",
            "errCode":11111,
            "message":"string",
            "params":[
                "string"
            ]
        }
    ]
}
```

1.7 GET get test tasks analysis

get test tasks analysis

```
Resource URI: edgegallery/atp/v2/tasks/action/analysize
```

Example response:

```
200 OK
 {
    "data":{
        "total":426,
        "currentMonth":11,
        "oneMonthAgo":111,
        "twoMonthAgo":12,
        "threeMonthAgo":13,
        "fourMonthAgo":14,
        "fiveMonthAgo":15
    },
    "retCode":0,
    "message":"errorMsg",
    "params":[
        "param1"
    ]
}
```

1.8 DELETE delete one test task

delete one test task

```
Resource URI: edgegallery/atp/v2/tasks/{taskId}
```

Example response:

```
true
```

1.9 POST upload self test report

upload self test report

```
Resource URI: edgegallery/atp/v2/tasks/{taskId}/action/upload-report
```

Example request:

```
{
    file: file
}
```

Example response:

```
{
    "data":{
        "path":"reportPath"
    },
    "retCode":0,
    "message":"errorMsg",
    "params":[
        "param1 “
    ]
}
```

7.6.2 2. Test case

test cases of csar package

2.1 GET query all test cases

query all test cases

```
Resource URI: /edgegallery/atp/v2/testcases?limit=10&offset=0
```

Name	Definition	Type	Required
type	type	query param	no
name	name	query param	no
locale	locale	query param	no
testSuiteIdList	testSuiteIdList	query param	no
limit	limit	query param	yes
offset	offset	query param	yes

Example response:

```
200 OK
{
    "results":[
        {
            "id":"4d203173-9999-4f62-aabb-8ebcec357f87",
            "nameCh":"APPD 文件目录校验",
            "nameEn":"APPD File Dir Validation",
            "type":"automatic",
            "descriptionCh":"根目录下必须包含 APPD 文件目录",
            "descriptionEn":"Root directory must contain APPD file dir",
            "hashCode":"",
            "codeLanguage":"java",
            "expectResultCh":"根目录下存在 APPD 文件目录",
            "expectResultEn":"Root directory contains APPD file dir",
            "testStepCh":"1. 打开 csar 包 2. 校验根目录下存在 APPD 目录",
            "testStepEn":"1.open csar package 2.validate root directory contains APPD␣
↪directory",
            "testSuiteIdList":[
                "522684bd-d6df-4b47-aab8-b43f1b4c19c0"
            ],
            "configIdList":[
                "522684bd-9999-4b47-aab8-b43f1b4c19c0"
            ]
```

(下页继续)

(续上页)

```
        }
    ],
    "total":1,
    "limit":10,
    "offset":0
}
```

2.2 POST create test case

create test case

```
Resource URI: /edgegallery/atp/v2/testcases
```

Name	Definition	Type	Required
nameCh	nameCh	request param	yes
nameEn	nameEn	request param	no
type	type	request param	yes
descriptionCh	descriptionCh	request param	yes
descriptionEn	descriptionEn	request param	no
codeLanguage	codeLanguage	request param	yes
expectResultCh	expectResultCh	request param	yes
expectResultEn	expectResultEn	request param	no
testSuiteIdList	testSuiteIdList	request param	yes
testStepCh	testStepCh	request param	yes
testStepEn	testStepEn	request param	no
file	file	request part	yes
configIdList	configIdList	request param	no

Example response:

```
200 OK
{
    "data":{
        "id":"4d203173-9999-4f62-aabb-8ebcec357f87",
        "nameCh":"APPD 文件目录校验",
        "nameEn":"APPD File Dir Validation",
        "type":"automatic",
        "descriptionCh":"根目录下必须包含 APPD 文件目录",
        "descriptionEn":"Root directory must contain APPD file dir",
```

(下页继续)

(续上页)

```
        "hashCode":"",
        "codeLanguage":"java",
        "expectResultCh":"根目录下存在 APPD 文件目录",
        "expectResultEn":"Root directory contains APPD file dir",
        "testStepCh":"1. 打开 csar 包 2. 校验根目录下存在 APPD 目录",
        "testStepEn":"1.open csar package 2.validate root directory contains APPD␣
↪directory",
        "testSuiteIdList":[
            "522684bd-d6df-4b47-aab8-b43f1b4c19c0"
        ],
        "configIdList":[
            "522684bd-9999-4b47-aab8-b43f1b4c19c0"
        ]
    },
    "retCode":0,
    "message":"errorMsg",
    "params":[
        "param1"
    ]
}
```

2.3 PUT update test case

update test case

```
Resource URI: /edgegallery/atp/v2/testcases
```

Name	Definition	Type	Required
id	id	request param	yes
type	type	request param	no
descriptionCh	descriptionCh	request param	no
descriptionEn	descriptionEn	request param	no
codeLanguage	codeLanguage	request param	no
expectResultCh	expectResultCh	request param	no
expectResultEn	expectResultEn	request param	no
testSuiteIdList	testSuiteIdList	request param	no
testStepCh	testStepCh	request param	no
testStepEn	testStepEn	request param	no
file	file	request part	no
configIdList	configIdList	request param	no

Example response:

```
200 OK
{
    "data":{
        "id":"4d203173-9999-4f62-aabb-8ebcec357f87",
        "nameCh":"APPD 文件目录校验",
        "nameEn":"APPD File Dir Validation",
        "type":"automatic",
        "descriptionCh":"根目录下必须包含 APPD 文件目录",
        "descriptionEn":"Root directory must contain APPD file dir",
        "hashCode":"",
        "codeLanguage":"java",
        "expectResultCh":"根目录下存在 APPD 文件目录",
        "expectResultEn":"Root directory contains APPD file dir",
        "testStepCh":"1. 打开 csar 包 2. 校验根目录下存在 APPD 目录",
        "testStepEn":"1.open csar package 2.validate root directory contains APPD␣
↪directory",
        "testSuiteIdList":[
            "522684bd-d6df-4b47-aab8-b43f1b4c19c0"
        ],
        "configIdList":[
            "522684bd-9999-4b47-aab8-b43f1b4c19c0"
        ]
    },
    "retCode":0,
    "message":"errorMsg",
    "params":[
        "param1"
    ]
}
```

2.4 DELETE delete test case

delete test case

```
Resource URI: /edgegallery/atp/v2/testcases/{id}
```

Name	Definition	Type	Required
id	id	path param	yes

Example response:

```
200 OK
true
```

2.5 GET query one test case

query one test case

```
Resource URI: /edgegallery/atp/v2/testcases/{id}
```

Name	Definition	Type	Required
id	id	path param	yes

Example response:

```
200 OK
{
    "data":{
        "id":"4d203173-9999-4f62-aabb-8ebcec357f87",
        "nameCh":"APPD 文件目录校验",
        "nameEn":"APPD File Dir Validation",
        "type":"automatic",
        "descriptionCh":"根目录下必须包含 APPD 文件目录",
        "descriptionEn":"Root directory must contain APPD file dir",
        "hashCode":"",
        "codeLanguage":"java",
        "expectResultCh":"根目录下存在 APPD 文件目录",
        "expectResultEn":"Root directory contains APPD file dir",
        "testStepCh":"1. 打开 csar 包 2. 校验根目录下存在 APPD 目录",
        "testStepEn":"1.open csar package 2.validate root directory contains APPD␣
↪directory",
        "testSuiteIdList":[
            "522684bd-d6df-4b47-aab8-b43f1b4c19c0"
        ],
        "configIdList":[
            "522684bd-9999-4b47-aab8-b43f1b4c19c0"
        ]
    },
    "retCode":0,
    "message":"errorMsg",
    "params":[
        "param1"
    ]
}
```

2.6 GET download test case

download test case

```
Resource URI: edgegallery/atp/v2/testcases/{id}/action/download
```

Name	Definition	Type	Required
id	id	path param	yes

Example response:

```
200 OK
binary stream
```

7.6.3 3. Test scenario

test scenario management

3.1 GET query all test scenarios

query all test scenarios

```
Resource URI: edgegallery/atp/v2/testscenarios?limit=10&offset=0
```

Name	Definition	Type	Required
name	name	query param	no
locale	locale	query param	no
limit	limit	query param	yes
offset	offset	query param	yes

Example response:

```
200 OK
{
    "results":[
        {
            "id":"4d203111-1111-4f62-aabb-8ebcec357f87",
            "nameCh":"社区场景",
            "nameEn":"EdgeGallery Community Scenario",
            "descriptionCh":"适用于社区场景的测试",
            "descriptionEn":"suite for EdgeGallery community test",
```

(下页继续)

(续上页)

```
            "label":"EdgeGallery"
        }
    ],
    "total":1,
    "limit":10,
    "offset":0
}
```

3.2 POST create test scenario

create test scenario

```
Resource URI: edgegallery/atp/v2/testscenarios
```

Name	Definition	Type	Required
nameCh	nameCh	request param	yes
nameEn	nameEn	request param	no
descriptionCh	descriptionCh	request param	yes
descriptionEn	descriptionEn	request param	no
label	label	request param	yes
icon	icon	request part	yes

Example response:

```
200 OK
{
    "data":{
        "id":"4d203111-1111-4f62-aabb-8ebcec357f87",
        "nameCh":"社区场景",
        "nameEn":"EdgeGallery Community Scenario",
        "descriptionCh":"适用于社区场景的测试",
        "descriptionEn":"suite for EdgeGallery community test",
        "label":"EdgeGallery"
    },
    "retCode":0,
    "message":"errorMsg",
    "params":[
        "param1"
    ]
}
```

3.3 PUT update test scenario

update test scenario

```
Resource URI: edgegallery/atp/v2/testscenarios/{id}
```

Name	Definition	Type	Required
id	id	path param	yes
nameCh	nameCh	request param	no
nameEn	nameEn	request param	no
descriptionCh	descriptionCh	request param	no
descriptionEn	descriptionEn	request param	no
label	label	request param	no
icon	icon	request part	no

Example response:

```
200 OK
{
    "data":{
        "id":"4d203111-1111-4f62-aabb-8ebcec357f87",
        "nameCh":"社区场景",
        "nameEn":"EdgeGallery Community Scenario",
        "descriptionCh":"适用于社区场景的测试",
        "descriptionEn":"suite for EdgeGallery community test",
        "label":"EdgeGallery"
    },
    "retCode":0,
    "message":"errorMsg",
    "params":[
        "param1"
    ]
}
```

3.4 DELETE delete test scenario

delete test scenario

```
Resource URI: edgegallery/atp/v2/testscenarios/{id}
```

Name	Definition	Type	Required
id	id	path param	yes

Example response:

```
200 OK
true
```

3.5 GET query one test scenario

query one test scenario

```
Resource URI: edgegallery/atp/v2/testscenarios/{id}
```

Name	Definition	Type	Required
id	id	path param	yes

Example response:

```
200 OK
{
    "data":{
        "id":"4d203111-1111-4f62-aabb-8ebcec357f87",
        "nameCh":"社区场景",
        "nameEn":"EdgeGallery Community Scenario",
        "descriptionCh":"适用于社区场景的测试",
        "descriptionEn":"suite for EdgeGallery community test",
        "label":"EdgeGallery"
    },
    "retCode":0,
    "message":"errorMsg",
    "params":[
        "param1"
    ]
}
```

3.6 GET query all test cases under one scenario

query all test cases under one scenario

```
Resource URI: edgegallery/atp/v2/testscenarios/testcases
```

Name	Definition	Type	Required
scenarioids	scenarioids	request param	yes

Example request:

```
200 OK
"scenarioIds":["id", "id2"]
```

Example response:

```
200 OK
[
  {
    "id": "96a82e85-d40d-4ce5-beec-2dd1c9a3d41d",
    "nameCh": "中国电信",
    "nameEn": "China Telecom Scenario",
    "descriptionCh": "适用于中国电信场景的测试",
    "descriptionEn": "suite for China Telecom test",
    "label": "China Telecom",
    "testSuites": [
      {
        "id": "6d04da1b-1f36-4295-920a-8074f7f9d942",
        "nameCh": "沙箱测试",
        "nameEn": "Sandbox Test",
        "descriptionCh": "应用包部署测试",
        "descriptionEn": "App package deployment test",
        "scenarioIdList": [
          "4d203111-1111-4f62-aabb-8ebcec357f87",
          "e71718a5-864a-49e5-855a-5805a5e9f97d",
          "6fe8581c-b83f-40c2-8f5b-505478f9e30b",
          "96a82e85-d40d-4ce5-beec-2dd1c9a3d41d"
        ],
        "testCases": [
          {
            "id": "4d203173-5555-4f62-aabb-8ebcec357f87",
            "nameCh": "应用实例化",
            "nameEn": "Application Instantiation",
            "type": "automatic",
            "descriptionCh": "将应用包部署到边缘节点",
            "descriptionEn": "Instantiate application and its dependency application␣
↪on one edge host",
            "hashCode": "",
            "codeLanguage": "jar",
            "expectResultCh": "应用包可以成功部署",
            "expectResultEn": "app can instantiate successfully.",
            "testStepCh": "部署应用包到对应的边缘节点",
            "testStepEn": "Deploy application package to edge node",
            "testSuiteIdList": [
```

(下页继续)

(续上页)

```
                "6d04da1b-1f36-4295-920a-8074f7f9d942"
            ]
          }
        ]
      }
    ]
  }
]
```

7.6.4 4. Test suite

test suite management

4.1 GET query all test suite

query all test suite

```
Resource URI: edgegallery/atp/v2/testsuites?limit=10&offset=0
```

Name	Definition	Type	Required
name	name	query param	no
locale	locale	query param	no
scenarioIdList	scenarioIdList	query param	no
limit	limit	query param	yes
offset	offset	query param	yes

Example response:

```
200 OK
{
    "results":[
        {
            "id":"522684bd-d6df-4b47-aab8-b43f1b4c19c0",
            "nameCh":"遵从性测试",
            "nameEn":"Compliance Test",
            "descriptionCh":"遵从社区 APPD 标准、ETSI 标准对应用包结构进行校验",
            "descriptionEn":"Validate app package structure according to commnunity␣
↪and ETSI standard",
            "scenarioIdList":[
                "4d203111-1111-4f62-aabb-8ebcec357f87"
```

(下页继续)

(续上页)

```
            ]
        }
    ],
    "total":1,
    "limit":10,
    "offset":0
}
```

4.2 POST create test suite

create test suite

```
Resource URI: edgegallery/atp/v2/testsuites
```

Name	Definition	Type	Required
nameCh	nameCh	request param	yes
nameEn	nameEn	request param	no
descriptionCh	descriptionCh	request param	yes
descriptionEn	descriptionEn	request param	no
scenarioIdList	scenarioIdList	request param	yes

Example response:

```
200 OK
{
    "data":{
        "id":"743abd93-57a3-499d-9591-fa7db86a4778",
        "nameCh":"安全性测试",
        "nameEn":"Security Test",
        "descriptionCh":"应用包安全测试",
        "descriptionEn":"App package security test",
        "scenarioIdList":[
            "4d203111-1111-4f62-aabb-8ebcec357f87"
        ]
    },
    "retCode":0,
    "message":"errorMsg",
    "params":[
        "param1"
    ]
}
```

4.3 PUT update test suite

update test suite

```
Resource URI: edgegallery/atp/v2/testsuites/{id}
```

Name	Definition	Type	Required
id	id	path param	yes
nameCh	nameCh	request param	no
nameEn	nameEn	request param	no
descriptionCh	descriptionCh	request param	no
descriptionEn	descriptionEn	request param	no
scenarioIdList	scenarioIdList	request param	no

Example response:

```
200 OK
{
    "data":{
        "id":"743abd93-57a3-499d-9591-fa7db86a4778",
        "nameCh":"安全性测试",
        "nameEn":"Security Test",
        "descriptionCh":"应用包安全测试",
        "descriptionEn":"App package security test",
        "scenarioIdList":[
            "4d203111-1111-4f62-aabb-8ebcec357f87"
        ]
    },
    "retCode":0,
    "message":"errorMsg",
    "params":[
        "param1"
    ]
}
```

4.4 DELETE delete test suite

delete test suite

```
Resource URI: edgegallery/atp/v2/testsuites/{id}
```

Name	Definition	Type	Required
id	id	path param	yes

Example response:

```
200 OK
true
```

4.5 GET query one test suite

query one test suite

```
Resource URI: edgegallery/atp/v2/testsuites/{id}
```

Name	Definition	Type	Required
id	id	path param	yes

Example response:

```
200 OK
{
    "data":{
        "id":"743abd93-57a3-499d-9591-fa7db86a4778",
        "nameCh":"安全性测试",
        "nameEn":"Security Test",
        "descriptionCh":"应用包安全测试",
        "descriptionEn":"App package security test",
        "scenarioIdList":[
            "4d203111-1111-4f62-aabb-8ebcec357f87"
        ]
    },
    "retCode":0,
    "message":"errorMsg",
    "params":[
        "param1"
```

(下页继续)

(续上页)

```
    ]
}
```

7.6.5 5. Contribution

contribution

5.1 GET query all contribution

query all contribution

```
Resource URI: edgegallery/atp/v2/contributions?limit=10&offset=0
```

Name	Definition	Type	Required
name	name	query param	no
limit	limit	query param	yes
offset	offset	query param	yes

Example response:

```
200 OK
{
    "results":[
        {
            "id":"872b55f8-bfb8-4fb4-be4e-0b643ff375e2",
            "name":"test",
            "objective":"test",
            "step":"test",
            "expectResult":"test",
            "type":"text",
            "createTime":"2021-03-01T09:17:31.915+0000"
        }
    ],
    "total":1,
    "limit":10,
    "offset":0
}
```

5.2 POST create contribution

create contribution

```
Resource URI: edgegallery/atp/v2/contributions
```

Name	Definition	Type	Required
name	name	request param	yes
objective	objective	request param	no
step	step	request param	yes
expectResult	expectResult	request param	no
type	type	request param	yes
file	file	requestpart	yes

Example response:

```
200 OK
{
    "data":{
        "id":"872b55f8-bfb8-4fb4-be4e-0b643ff375e2",
        "name":"test",
        "objective":"test",
        "step":"test",
        "expectResult":"test",
        "type":"text",
        "createTime":"2021-03-01T09:17:31.915+0000"
    },
    "retCode":0,
    "message":"errorMsg",
    "params":[
        "param1"
    ]
}
```

5.3 POST batch delete contributions

batch delete contributions

```
Resource URI: edgegallery/atp/v2/contributions/batch_delete
```

Name	Definition	Type	Required
ids	ids	request param	yes

Example request:

```
{
  "ids":["id1","id2"]
}
```

Example response:

```
200 OK
{
    "retCode":5000,
    "failures":[
        {
            "id":"string",
            "errCode":11111,
            "message":"string",
            "params":[
                "string"
            ]
        }
    ]
}
```

5.4 GET download contribution script

download contribution script

```
Resource URI: edgegallery/atp/v2/contributions/{id}/action/download
```

Example response:

```
200 OK
binary stream
```

7.6.6 6. File

file

6.1 GET query one file

query one file

```
Resource URI: edgegallery/atp/v2/files/{id}
```

Name	Definition	Type	Required
id	id	path param	yes
type	type	query param	no

Example response:

```
200 OK
binary stream
```

7.6.7 7. Test model

test model

7.1 POST import test model

import test model

```
Resource URI: edgegallery/atp/v2/testmodels/action/import
```

Name	Definition	Type	Required
file	file	request param	yes

Example response:

```
200 OK
{
  "retCode" : 0
  "failures" : []
}

206 partial success
{
  "retCode" : 5000,
  "failures" : [{
       "id" : "string",
```

(下页继续)

(续上页)

```
        "nameEn" : "string",
        "type" : "testScenario",
        "errCode": 1000,
         "errMsg":  “string” ,
        "params": "string"
      },
      {
        "id" : "string",
        "nameEn" : "string",
        "type" : "testSuite",
        "errCode": 1000,
       "errMsg":  “string” ,
        "params": "string"
      },
      {
        "id" : "string",
        "nameEn" : "string",
        "type" : "testCase",
        "errCode": 1000,
        "errMsg":  “string” ,
        "params": "string"
      }
  ]
}
```

7.6.8 8. Configuration

config

8.1 POST create config

create config

```
Resource URI: edgegallery/atp/v2/configs
```

Example request:

```
{
    "nameCh":"虚机应用实例化",
    "nameEn":"Instantiation for vm app",
    "descriptionCh":"配置虚机应用实例化所需参数",
```

(下页继续)

(续上页)

```
    "descriptionEn":"Configure params for instantiation of vm app",
    "configuration":"DC_ID = FS+M;az_dc=nova;"
}
```

Example response:

```
200 OK
{
    "data":{
        "id":"uuid",
        "nameCh":"虚机应用实例化",
        "nameEn":"Instantiation for vm app",
        "descriptionCh":"配置虚机应用实例化所需参数",
        "descriptionEn":"Configure params for instantiation of vm app",
        "configuration":"DC_ID = FS+M;az_dc=nova;",
        "createTime":""
    },
    "retCode":0,
    "message":"errorMsg",
    "params":[
        "param1",
        "param2"
    ]
}
```

8.2 PUT update config

update config

```
Resource URI: edgegallery/atp/v2/configs/{id}
```

Example request:

```
{
    "nameCh":"虚机应用实例化",
    "nameEn":"Instantiation for vm app",
    "descriptionCh":"配置虚机应用实例化所需参数",
    "descriptionEn":"Configure params for instantiation of vm app",
    "configuration":"DC_ID = FS+M;az_dc=nova;"
}
```

Example response:

```
200 OK
{
    "data":{
        "id":"uuid",
        "nameCh":"虚机应用实例化",
        "nameEn":"Instantiation for vm app",
        "descriptionCh":"配置虚机应用实例化所需参数",
        "descriptionEn":"Configure params for instantiation of vm app",
        "configuration":"DC_ID = FS+M;az_dc=nova;",
        "createTime":""
    },
    "retCode":0,
    "message":"errorMsg",
    "params":[
        "param1",
        "param2"
    ]
}
```

8.3 DELETE delete config

delete config

```
Resource URI: edgegallery/atp/v2/configs/{id}
```

Name	Definition	Type	Required
id	id	path param	yes

Example request:

```
NA
```

Example response:

```
200 OK
true
```

8.4 GET query one config

query one config

```
Resource URI: edgegallery/atp/v2/configs/{id}
```

Name	Definition	Type	Required
id	id	path param	yes

Example request:

```
NA
```

Example response:

```
200 OK
{
    "data":{
        "id":"uuid",
        "nameCh":"虚机应用实例化",
        "nameEn":"Instantiation for vm app",
        "descriptionCh":"配置虚机应用实例化所需参数",
        "descriptionEn":"Configure params for instantiation of vm app",
        "configuration":"DC_ID = FS+M;az_dc=nova;",
        "createTime":""
    },
    "retCode":0,
    "message":"errorMsg",
    "params":[
        "param1",
        "param2"
    ]
}
```

8.5 GET query all configs

query all configs

```
Resource URI: edgegallery/atp/v2/configs?limit=10&offset=0
```

Name	Definition	Type	Required
name	name	query param	no
locale	locale	query param	no
limit	limit	query param	yes
offset	offset	query param	yes

Example request:

```
NA
```

Example response:

```
200 OK
{
    "results":[
        {
            "id":"uuid",
            "nameCh":"虚机应用实例化",
            "nameEn":"Instantiation for vm app",
            "descriptionCh":"配置虚机应用实例化所需参数",
            "descriptionEn":"Configure params for instantiation of vm app",
            "configuration":"DC_ID = FS+M;az_dc=nova;",
            "createTime":""
        }
    ],
    "total":1,
    "limit":10,
    "offset":0
}
```

7.7 数据库设计

- 在 V1.3 版本中，较 v1.2 版本，数据库表 TEST_CASE_TABLE 新增了 configIdList 字段，表 TASK_TABLE 新增字段 reportPath，需要在上述 2 个数据库表中执行如下 sql 语句：alter table TEST_CASE_TABLE add column configIdList TEXT NULL;alter table TASK_TABLE add column reportPath varchar(200) NULL;
- 新增了配置项的数据库表 CONFIG_TABLE，建表信息见下方表结构信息即可。
- ATP 数据库使用了开源的 PostgreSQL 数据库，推荐版本为 12.2。目前设计并应用的数据库表格如下：
- TABLE TASK_TABLE: 存储测试任务信息

```
CREATE TABLE TASK_TABLE (
    ID                      VARCHAR(200)       NOT NULL,
    APPNAME                 VARCHAR(200)       NULL,
    APPVERSION              VARCHAR(200)       NULL,
    STATUS                  VARCHAR(200)       NULL,
    TESTCASEDETAIL          TEXT               NULL,
    CREATETIME              TIMESTAMP          NULL,
    ENDTIME                 TIMESTAMP          NULL,
    PROVIDERID              VARCHAR(200)       NULL,
    PACKAGEPATH             VARCHAR(200)       NULL,
    USERID                  VARCHAR(200)       NULL,
    USERNAME                VARCHAR(200)       NULL,
    REPORTPATH              VARCHAR(200)       NULL,
    CONSTRAINT task_table_pkey PRIMARY KEY (ID)
);
```

- TEST_CASE_TABLE : 存储测试用例信息

```
CREATE TABLE TEST_CASE_TABLE (
    ID                      VARCHAR(200)       NOT NULL,
    NAMECH                  VARCHAR(200)       NULL,
    NAMEEN                  VARCHAR(200)       NULL,
    TYPE                    VARCHAR(200)       NULL,
    CLASSNAME               VARCHAR(200)       NULL,
    HASHCODE                TEXT               NULL,
    DESCRIPTIONCH           TEXT               NULL,
    DESCRIPTIONEN           TEXT               NULL,
    FILEPATH                VARCHAR(200)       NULL,
    CODELANGUAGE            VARCHAR(200)       NULL,
    EXPECTRESULTCH          VARCHAR(200)       NULL,
    EXPECTRESULTEN          VARCHAR(200)       NULL,
    TESTSUITEIDLIST         TEXT               NULL,
    TESTSTEPCH              TEXT               NULL,
    TESTSTEPEN              TEXT               NULL,
```

(下页继续)

(续上页)

```
    CREATETIME              TIMESTAMP          NULL,
    CONFIGIDLIST            TEXT               NULL,
    CONSTRAINT test_case_table_pkey PRIMARY KEY (ID)
);
```

- TEST_SCENARIO_TABLE: 存储测试场景信息

```
CREATE TABLE TEST_SCENARIO_TABLE (
    ID                      VARCHAR(200)       NOT NULL,
    NAMECH                  VARCHAR(200)       NULL,
    NAMEEN                  VARCHAR(200)       NULL,
    DESCRIPTIONCh           TEXT               NULL,
    DESCRIPTIONEN           TEXT               NULL,
    LABEL                   VARCHAR(200)       NULL,
    CREATETIME              TIMESTAMP          NULL,
    CONSTRAINT test_scenario_table_pkey PRIMARY KEY (ID)
);
```

- TEST_SUITE_TABLE: 存储测试套信息

```
CREATE TABLE TEST_SUITE_TABLE (
    ID                      VARCHAR(200)       NOT NULL,
    NAMECH                  VARCHAR(200)       NULL,
    NAMEEN                  VARCHAR(200)       NULL,
    DESCRIPTIONCh           TEXT               NULL,
    DESCRIPTIONEN           TEXT               NULL,
    SCENARIOIDLIST          VARCHAR(255)       NULL,
    CREATETIME              TIMESTAMP          NULL,
    CONSTRAINT test_suite_table_pkey PRIMARY KEY (ID)
);
```

- FILE_TABLE: 存储图标文件信息

```
CREATE TABLE FILE_TABLE (
    FILEID                  VARCHAR(200)       NOT NULL,
    TYPE                    VARCHAR(200)       NOT NULL,
    CREATETIME              TIMESTAMP          NULL,
    FILEPATH                VARCHAR(200)       NULL
);
alter table FILE_TABLE add constraint file_table_pkey unique(FILEID,TYPE);
```

- CONTRIBUTION_TABLE: 存储贡献的测试用例信息

```
CREATE TABLE CONTRIBUTION_TABLE (
    ID                      VARCHAR(200)       NOT NULL,
    NAME                    VARCHAR(200)       NULL,
    OBJECTIVE               VARCHAR(200)       NULL,
    STEP                    TEXT               NULL,
    EXPECTRESULT            TEXT               NULL,
    TYPE                    VARCHAR(255)       NULL,
    CREATETIME              TIMESTAMP          NULL,
    FILEPATH                VARCHAR(200)       NULL,
    CONSTRAINT contribution_table_pkey PRIMARY KEY (ID)
);
```

- CONFIG_TABLE: 存储配置项信息

```
CREATE TABLE CONFIG_TABLE (
          ID                      VARCHAR(200)       NOT NULL,
          NAMECH                  VARCHAR(200)       NULL,
          NAMEEN                  VARCHAR(200)       NULL,
          DESCRIPTIONCh           TEXT               NULL,
          DESCRIPTIONEN           TEXT               NULL,
          CONFIGURATION           TEXT               NOT NULL,
          CREATETIME              TIMESTAMP          NULL,
          CONSTRAINT config_table_pkey PRIMARY KEY (ID)
 );
```

7.8 ATP Contribution

7.8.1 搭建本地开发环境

本文将指导用户如何在本地搭建 atp 和 atp-fe 模块，从而帮助开发者快速理解、开发代码。

其中 atp 的本地启动依赖于：

```
- Service Center
- DataBase
- UserManagement
```

其中 atp-fe 的本地启动依赖于:

```
- WebsiteGateway
- UserManagement
- atp
```

所以为了搭建本地开发环境，最终的启动顺序为 Service Center -> DataBase -> UserManagement -> atp -> WebsiteGateway(atp-fe)

Service Center

1、下载Service Center

ServiceComb Service-Center下载

发布包

版本号	源码	Windows版本	Linux版本	Darwin版本	与之兼容的Java-Chassis版本
Apache ServiceComb Service-Center 1.3.0 (LATEST)	[src] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	Java-Chassis 1.0.0 or above
Apache ServiceComb Service-Center 1.2.0	[src] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	Java-Chassis 1.0.0 or above
Apache ServiceComb Service-Center 1.1.0	[src] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	Java-Chassis 1.0.0 or above
Apache ServiceComb (incubating) Service-Center 1.0.0	[src] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	Java-Chassis 1.0.0 or above
Apache ServiceComb (incubating) Service-Center 1.0.0-m2	[src] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	NA	Java-Chassis 1.0.0-m2 or above
Apache ServiceComb (incubating) Service-Center 1.0.0-m1	[src] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	NA	Java-Chassis 1.0.0-m1 or above

2、解压后运行 start-service-center.bat 和 start-frontend.bat

名称	修改日期	类型	大小
app	2019/12/18 14:15	文件夹	
conf	2019/12/18 14:15	文件夹	
data	2020/7/28 12:07	文件夹	
licenses	2019/12/18 14:15	文件夹	
frontend.exe	2019/10/27 16:55	应用程序	9,519 KB
LICENSE	2019/10/27 16:55	文件	46 KB
NOTICE	2019/10/27 16:55	文件	3 KB
README.md	2019/10/27 16:55	MD 文件	9 KB
scctl.exe	2019/10/27 16:55	应用程序	15,394 KB
service-center.exe	2019/10/27 16:55	应用程序	43,543 KB
start-frontend.bat	2019/10/27 16:55	Windows 批处理...	1 KB
start-service-center.bat	2019/10/27 16:55	Windows 批处理...	1 KB
syncer.exe	2019/10/27 16:55	应用程序	26,730 KB

DataBase

1、下载、安装、启动 Postgresql 最新版本

2、设置登录名和密码，例如 postgresql/root

3、创建名为 atpdb 的数据库

4、根据atp 部署文件 中的数据库语句创建表结构

5、创建名为 usermgmtdb 的数据库

6、根据usermgmt 部署文件中的数据库语句创建表结构

UserManagement

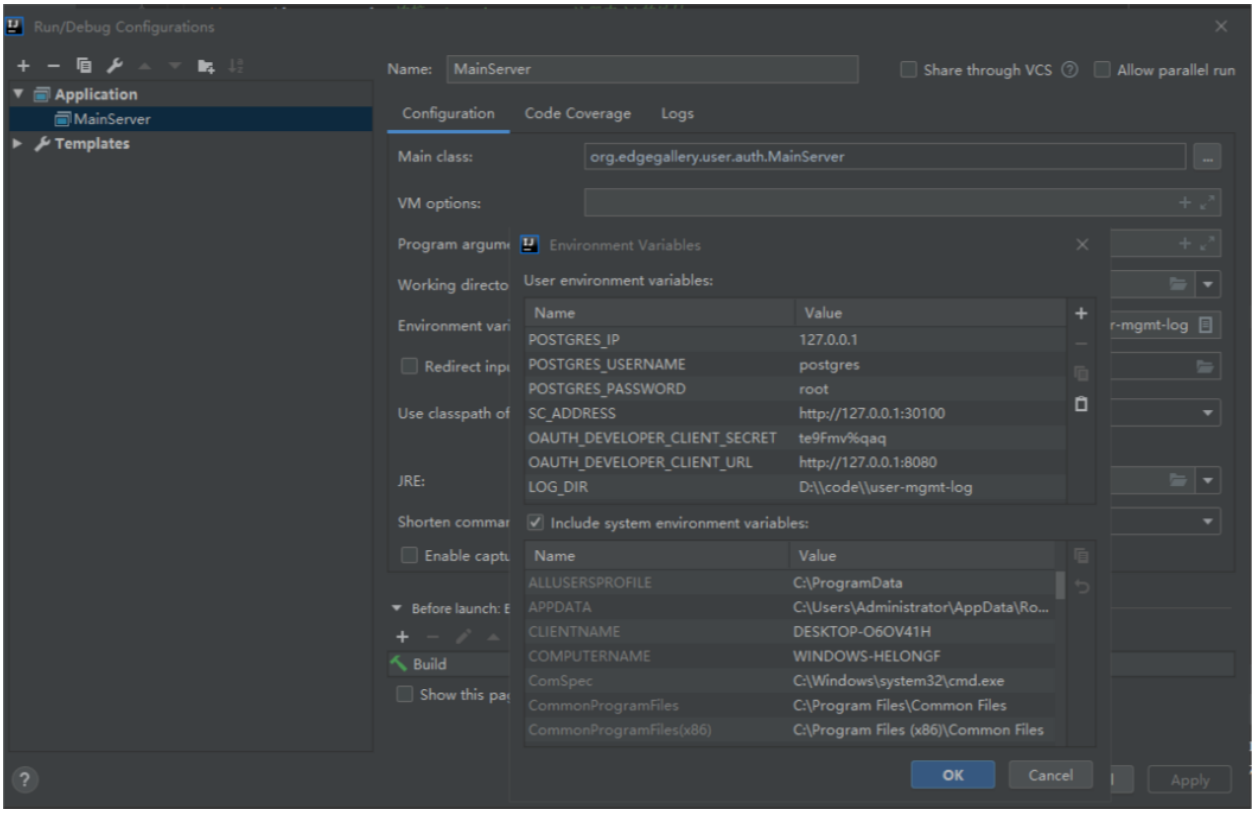
1、下载 user-mgmt-be 代码和 user-mgmt-fe 代码

2、安装 node.js 和 npm，进入 user-mgmt-fe 文件夹，运行 `npm install` 和 `npm run build`

3、在 user-mgmt-be 的 src/main/resources/文件夹中创建 static 文件夹

4、将 user-mgmt-fe 编译出的 dist 文件夹中的内容拷贝到 user-mgmt-be 的 static 文件夹中

5、在编译器（此处以 IDEA 为例）中配置 MainServer 的运行时环境变量

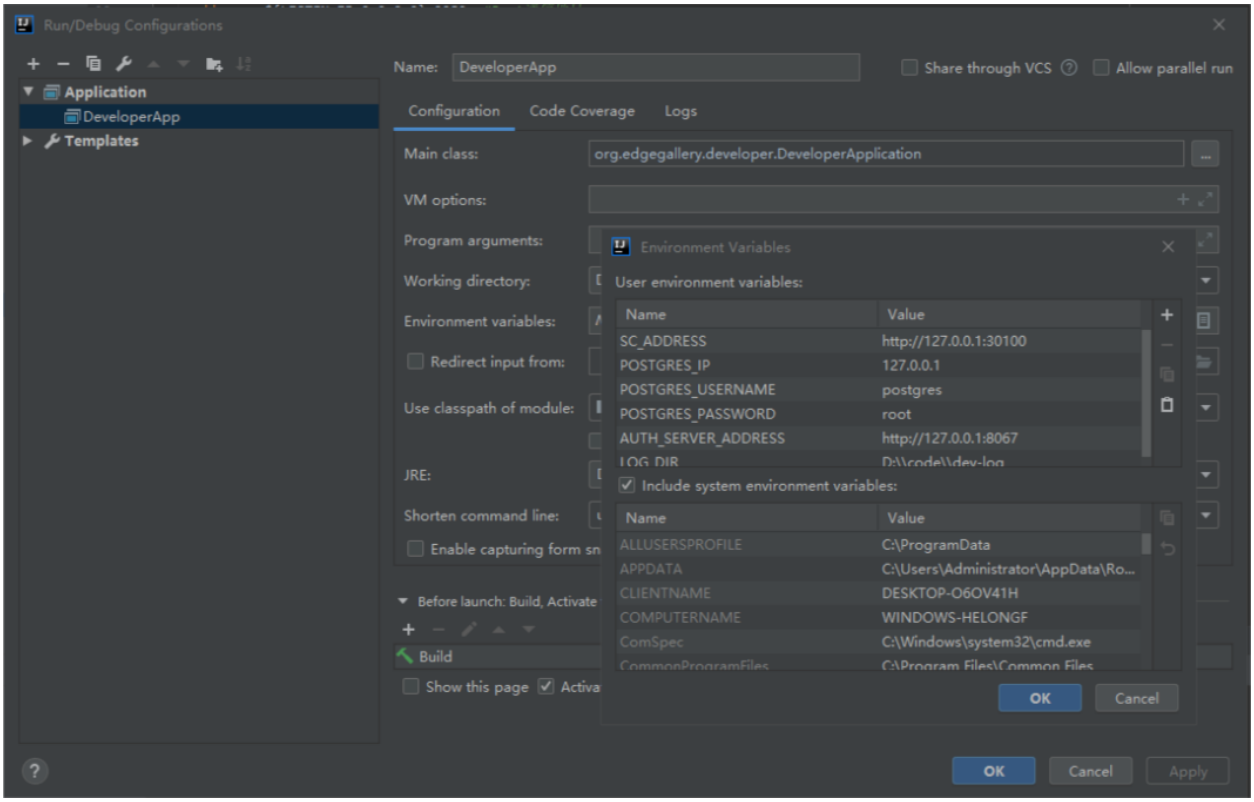


6、运行 MainServer 函数，启动 UserManagement

atp

1、下载 atp 代码

2、在编译器（此处以 IDEA 为例）中配置 atpApp 的运行时环境变量



3、运行 atpApp 函数，启动 atp

atp-fe

1、下载 atp-fe 代码

2、进入 atp-fe 文件夹，运行 `npm install` 和 `npm run build`

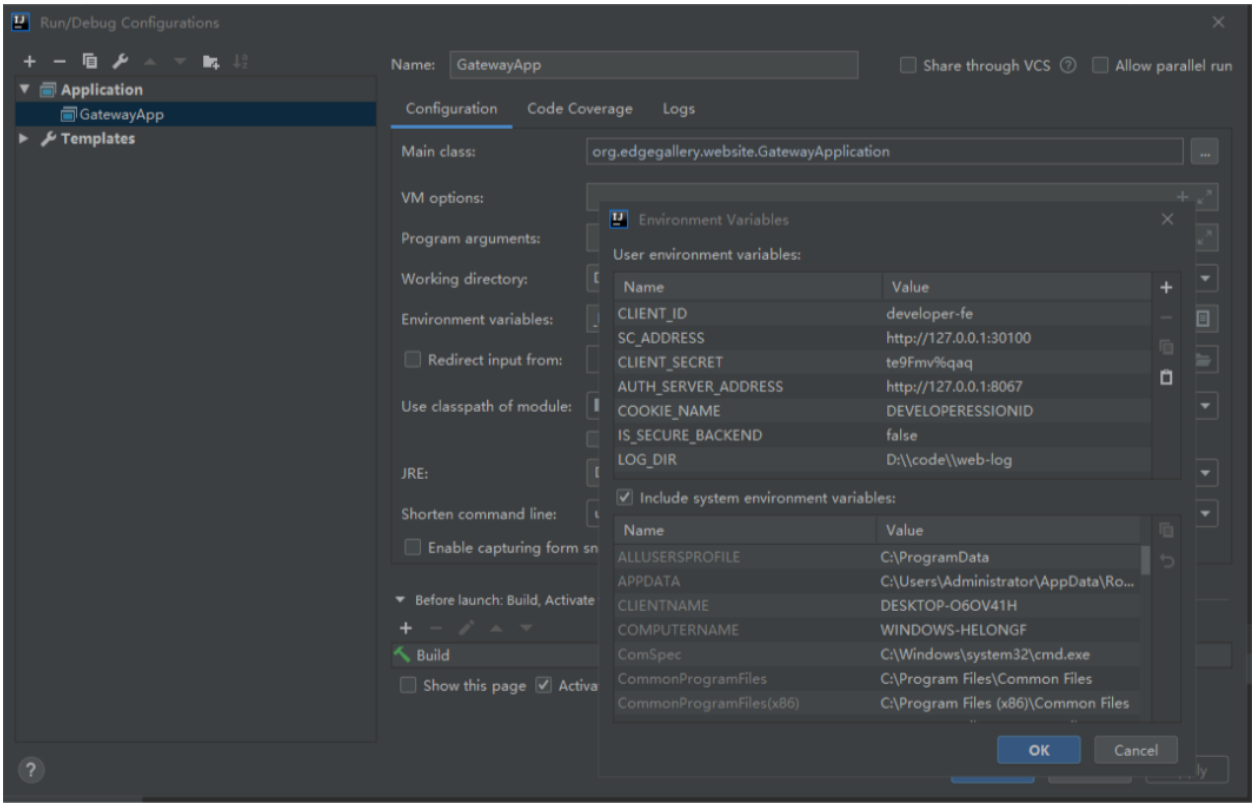
3、下载 website-gateway 代码

4、在 website-gateway 的 src/main/resources/文件夹中创建 static 文件夹

5、将 atp-fe 编译出的 dist 文件夹中的内容拷贝到 website-gateway 的 static 文件夹中

6、修改 website-gateway 中/src/main/resources/application.yaml 文件内的 `server.port` 为 `8080`

7、在编译器（此处以 IDEA 为例）中配置 GatewayApp 的运行时环境变量



8、运行 GatewayApp 启动 WebsiteGateway 与 atp-fe，浏览器访问 http://127.0.0.1:8080/ 即可访问到 atp 首页

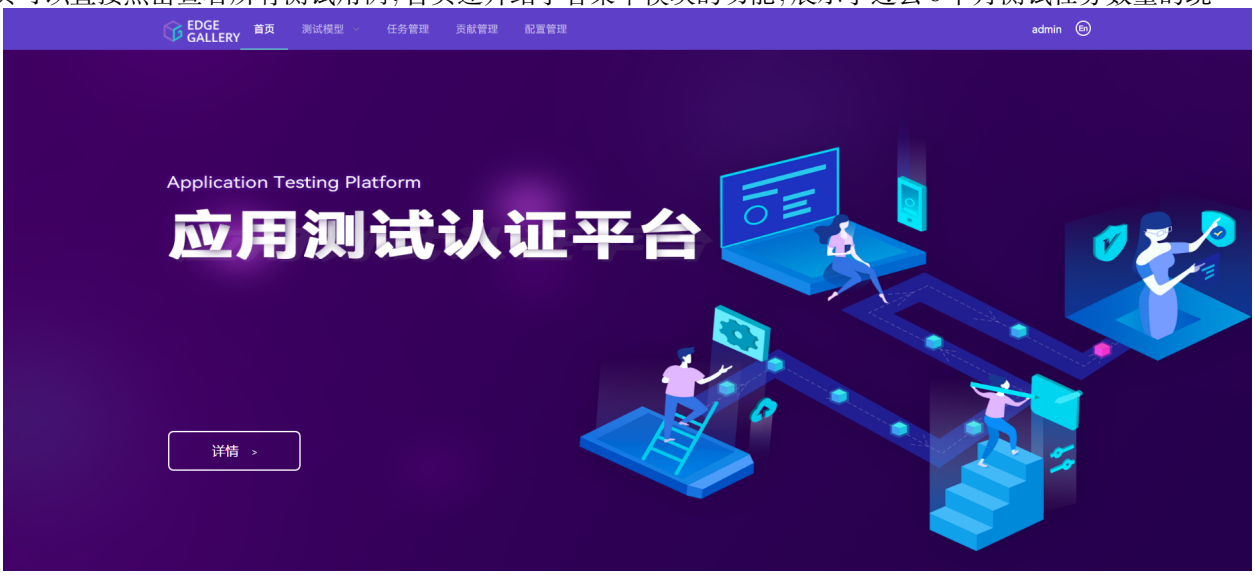
7.9 ATP Guide

7.9.1 概述

- 应用测试认证服务目前分为管理面功能和用户面功能。
- 在管理面，管理员可以管理测试场景、测试套、测试用例、测试任务、贡献的测试用例以及配置项，其中贡献管理的菜单仅管理员可见。平台内置了社区场景以及 3 个样例运营商场景。对于社区场景，平台内置了通用安全性测试测试套、通用遵从性测试测试套和通用沙箱测试测试套，每个测试套下面都有对应的测试用例。平台支持一键式批量导入测试模型，方便批量测试模型的新增。对于有手工用例的测试任务，管理员可以在后台手动更改测试用例状态，完成手工测试用例状态的更新。
- 在用户面（即集成在开发者平台和应用商店中的功能），对于开发者，在开发者平台开发 app 以后，对生成的 csar 包，要经过应用测试认证服务测试通过后才可以发布到应用商店；对于已经有应用包的供应商或者厂商，在应用商店中上传应用包以后，也要经过应用测试认证服务测试通过后才可以发布到应用商店。用户可以选择要进行测试的场景，其中社区场景是必选的，然后系统会动态展示出测试用例的执行过程。当测试完成后，会生成测试报告，给出测试结果的分析，用户也可以下载测试报告，格式为 pdf 格式。用户面也为用户提供了贡献测试用例的入口，用户可以通过填写对测试用例的描述或者上传脚本的方式，完成对测试用例的贡献。

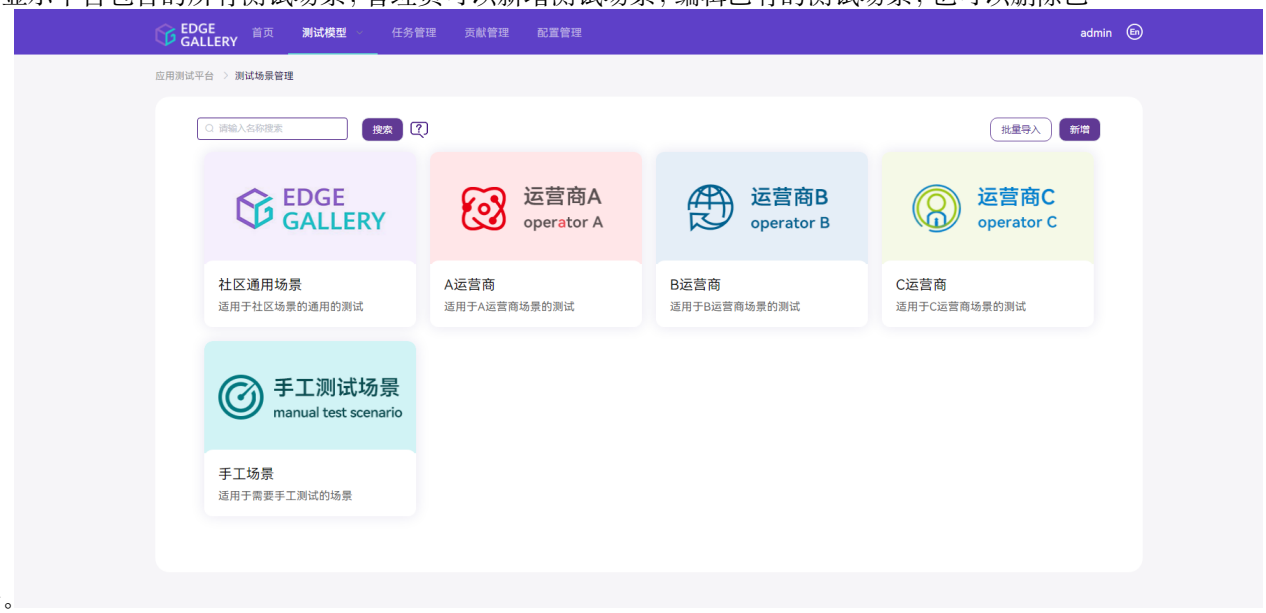
7.9.2 管理面首页

首页可以直接点击查看所有测试用例，首页还介绍了各菜单模块的功能，展示了过去 6 个月测试任务数量的统计。



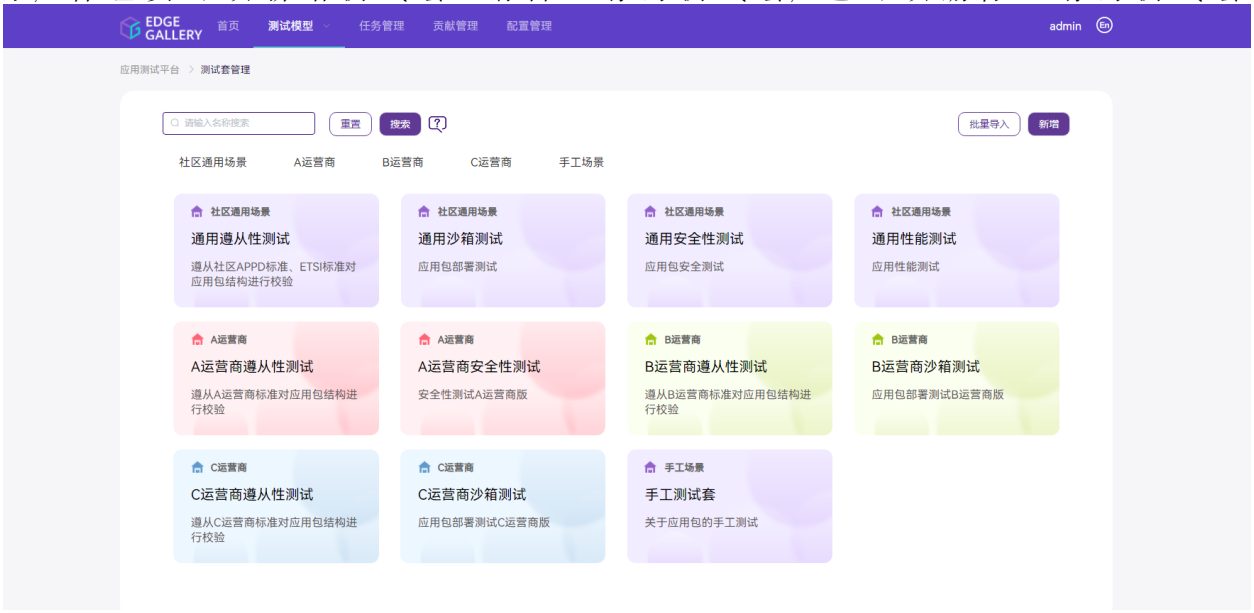
7.9.3 管理面测试场景管理

测试场景页面显示平台包含的所有测试场景，管理员可以新增测试场景，编辑已有的测试场景，也可以删除已有的测试场景。



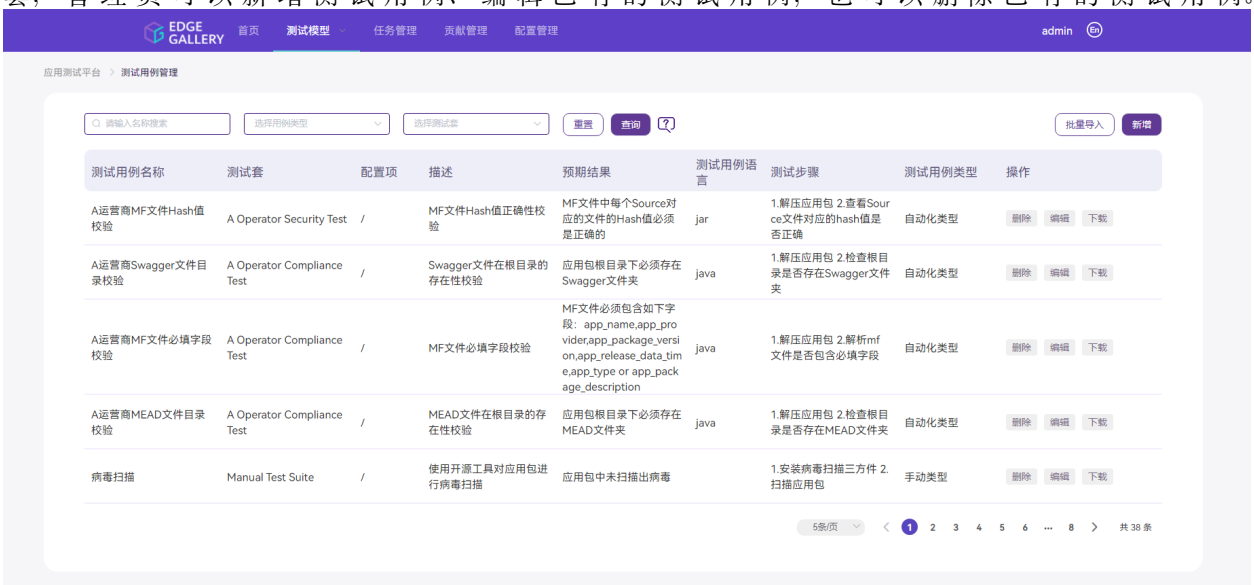
7.9.4 管理面测试套管理

测试套页面显示平台包含的所有测试套，一个测试套属于一个或者多个测试场景，管理员可以新增测试套、编辑已有的测试套，也可以删除已有的测试套。



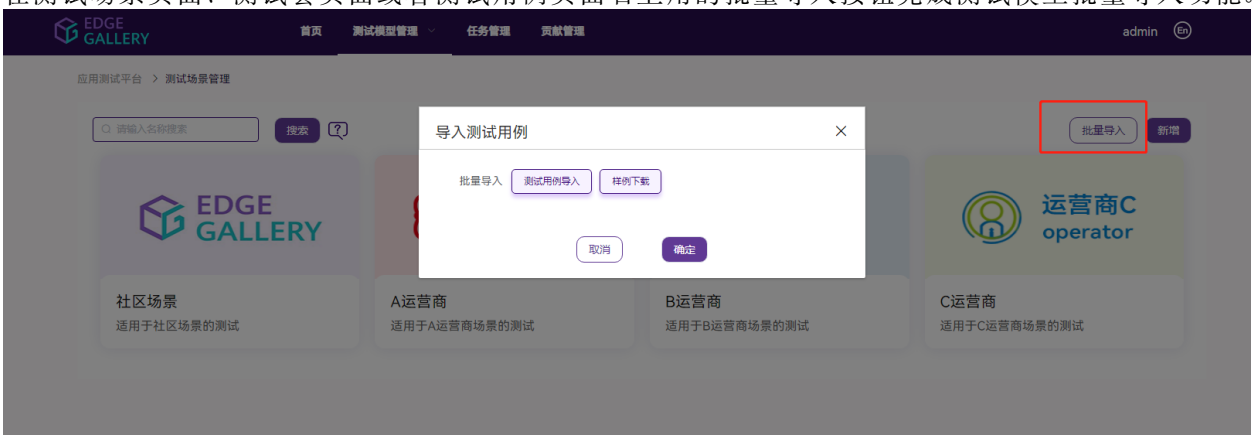
7.9.5 管理面测试用例管理

测试用例页面显示平台包含的所有测试用例，一个测试用例属于一个或者多个测试套，管理员可以新增测试用例、编辑已有的测试用例，也可以删除已有的测试用例。



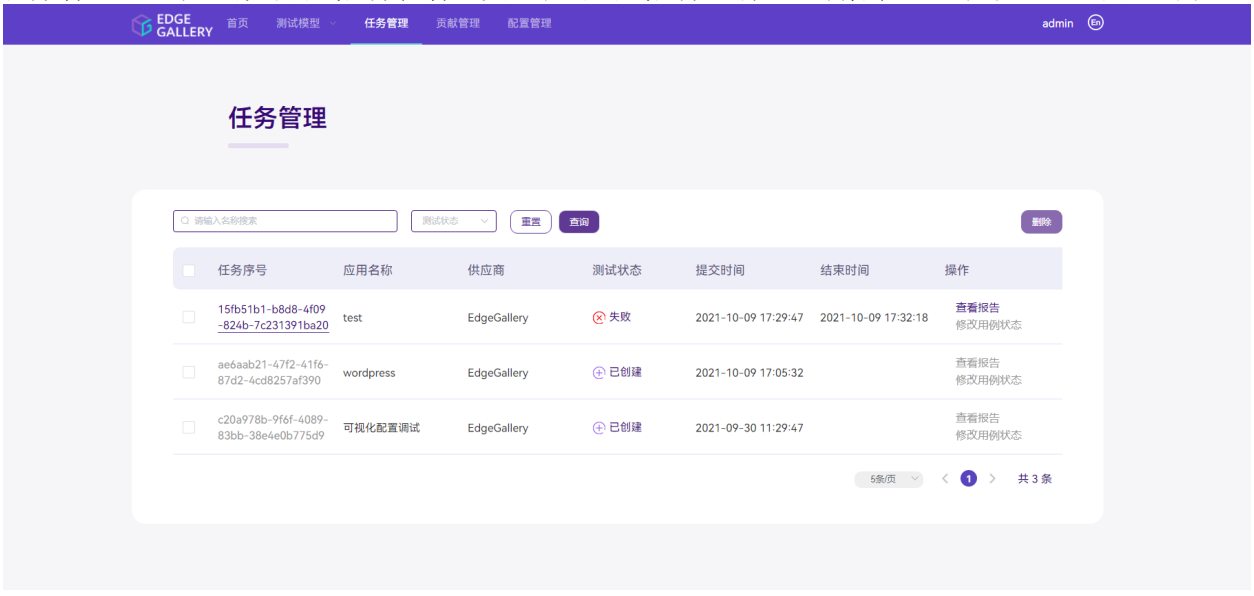
7.9.6 管理面测试模型一键式导入

为了方便管理员批量导入测试场景、测试套和测试用例，管理面提供了测试模型导入功能。管理员可以将测试场景、测试套和测试用例的描述写到 excel 表中，并把测试用例脚本，测试场景图标放到指定文件夹中，最终压缩成 zip 包，通过导入 zip 包的方式完成测试模型的批量导入功能。管理员可以在测试场景页面、测试套页面或者测试用例页面右上角的批量导入按钮完成测试模型批量导入功能。



7.9.7 管理面测试任务管理

任务管理显示了所有的测试任务，管理员可以对测试任务进行批量删除，也可以修改手工用例的状态。

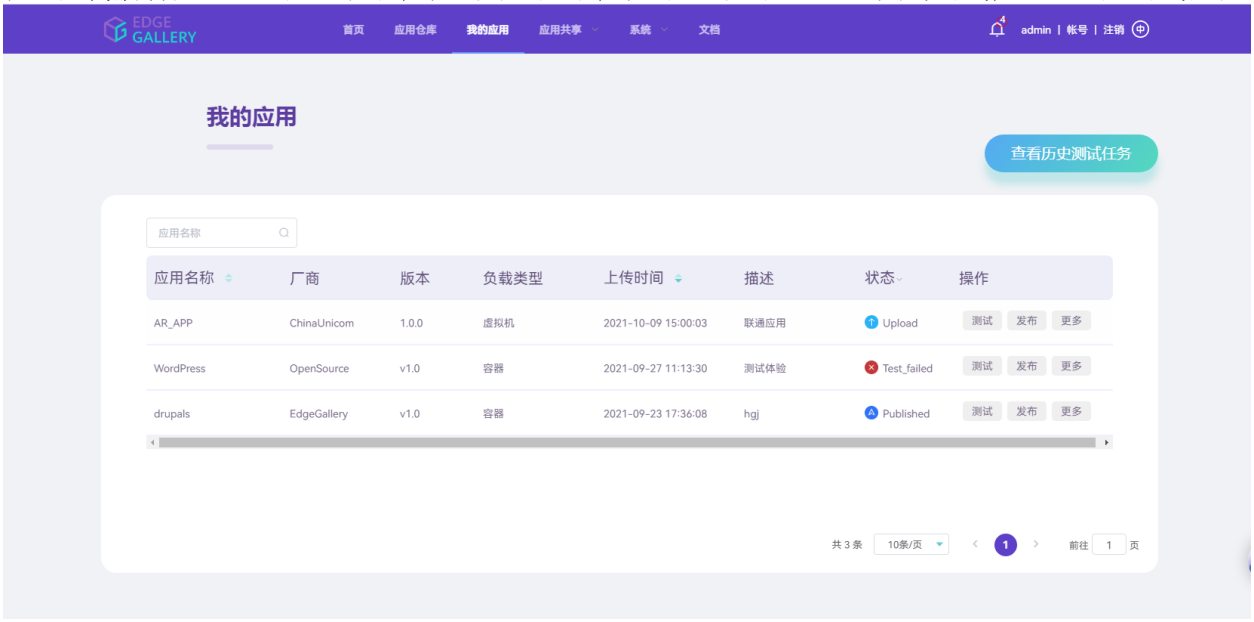


7.9.8 管理面配置项管理

配置项管理显示了所有配置项信息，每个配置项可以关联一个或者多个测试用例，使测试用例参数可配置，更灵活。

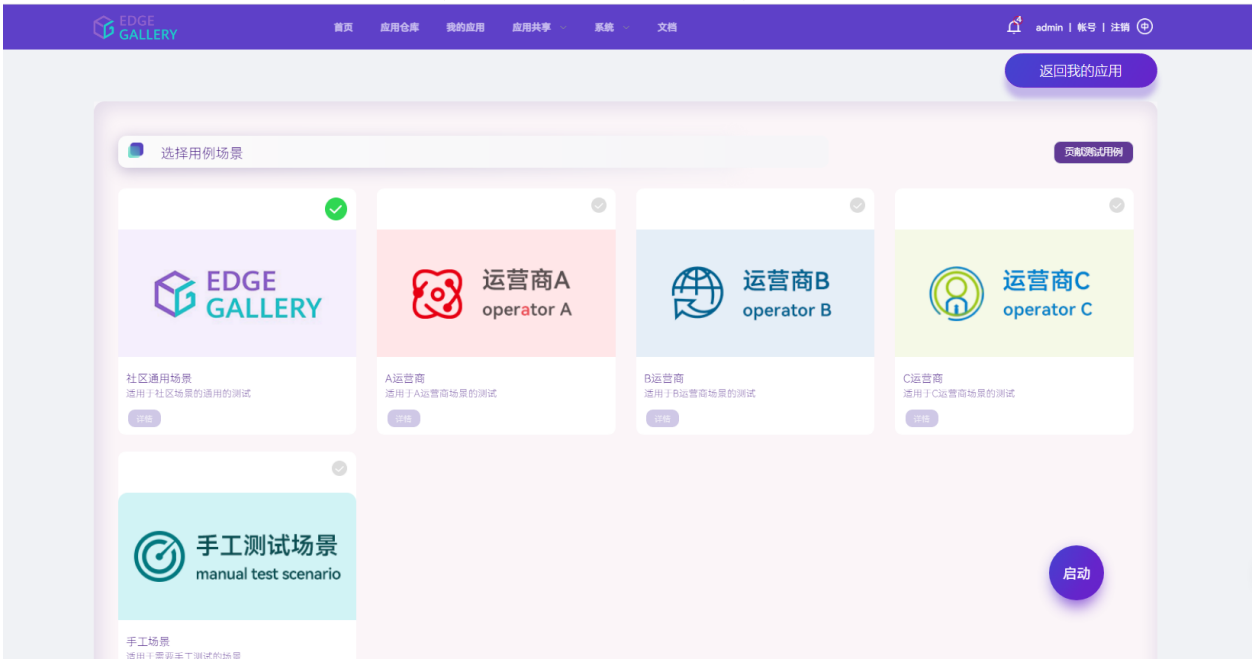
7.9.9 应用商店测试认证入口

在应用商店首页上传应用包以后，在我的应用中，在对应的应用包右边点击测试，进入应用测试流程。



7.9.10 测试场景选择 & 测试用例贡献

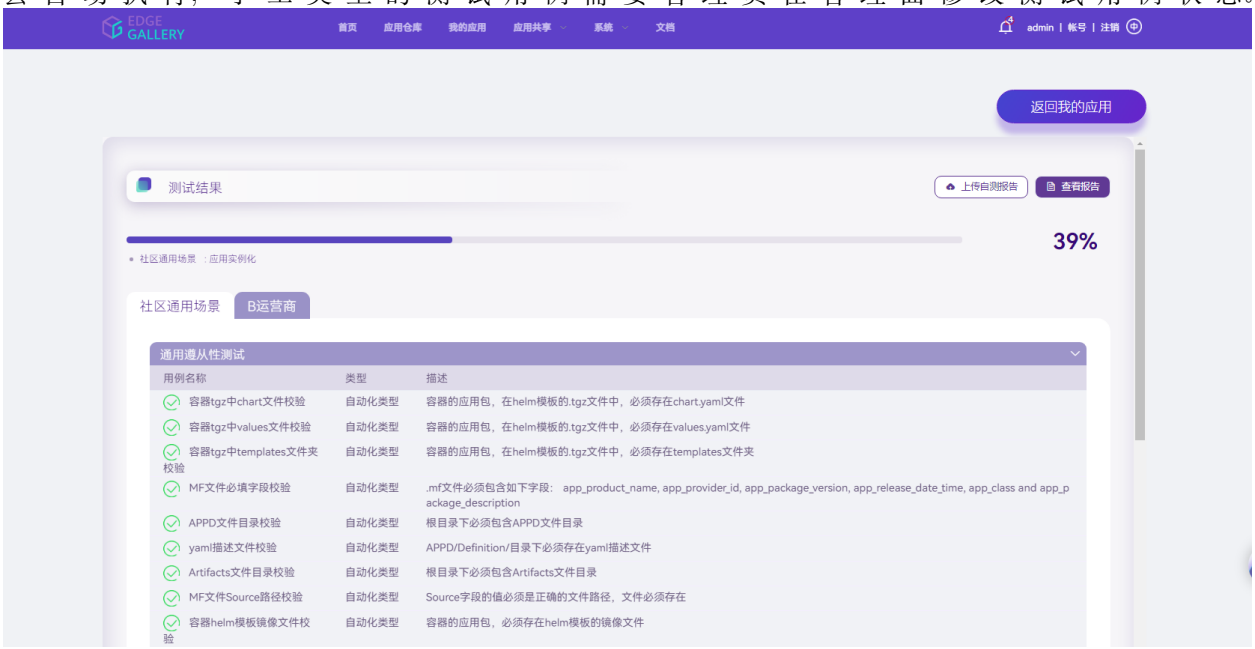
用户可以选择想要测试的场景，社区场景是必选的，其余场景根据用户需要自行选择。



页面右上方有贡献测试用例的入口，点击测试用例，支持用户用描述的方式或者脚本的方式贡献测试用例。

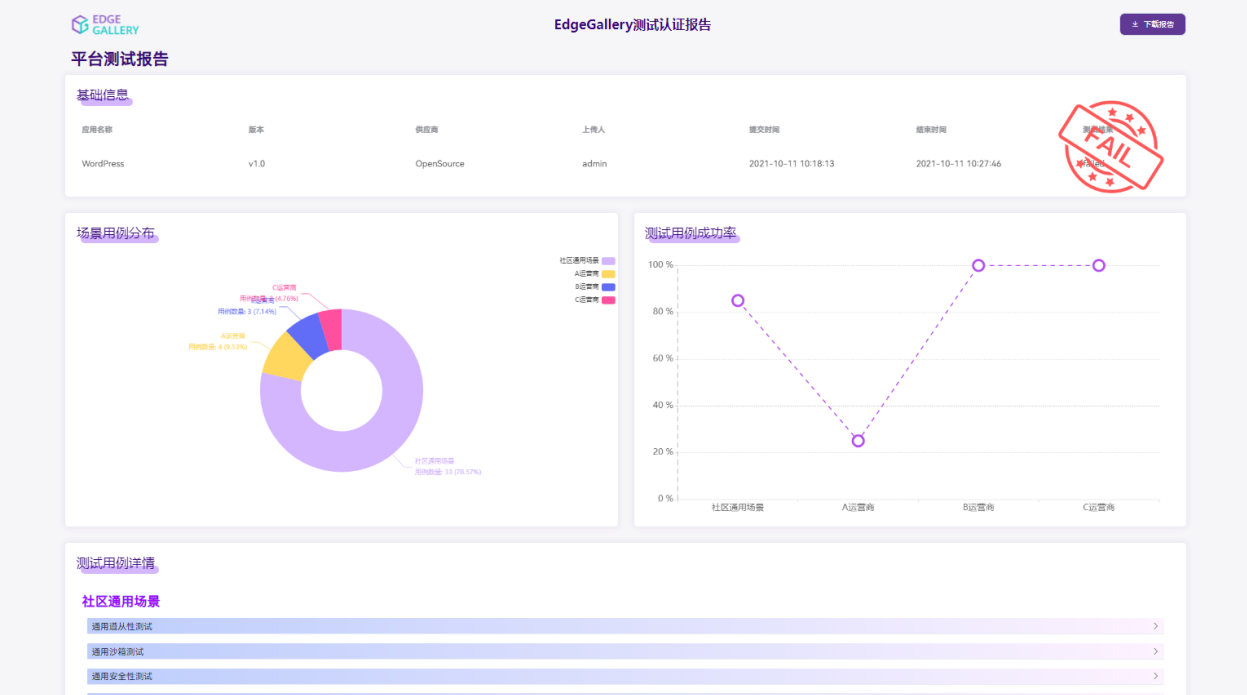
7.9.11 测试进展可视化 & 支持用户上传自测报告 & 报告查看

测试进展页面动态地展示用例的执行情况，所有自动化类型的测试用例都会自动执行，手工类型的测试用例需要管理员在管理面修改测试用例状态。



用户如果有应用相关的独特的测试，可以点击上传自测报告，格式为 pdf。若用户上传了自测报告，最终的测试报告会将自测报告和平台报告整合到一起展示。

点击右上角的查看报告，会进入该测试任务的报告页面，报告页面显示测试任务的基本信息、各测试场景的测试用例通过率以及测试用例的详细信息。


EDGE GALLERY
EdgeGallery测试认证报告
下载报告
平台测试报告
基础信息
应用名称
版本
供应商
上传人
提交时间
结束时间
WordPress
v1.0
OpenSource
admin
2021-10-11 10:18:13
2021-10-11 10:27:46
FAIL
场景用例分布
测试用例成功率
100 %
80 %
60 %
40 %
20 %
0 %
社区通用场景
A运营商
B运营商
C运营商
测试用例详情
社区通用场景
通用道从性测试
通用沙箱测试
通用安全性测试

CHAPTER 8

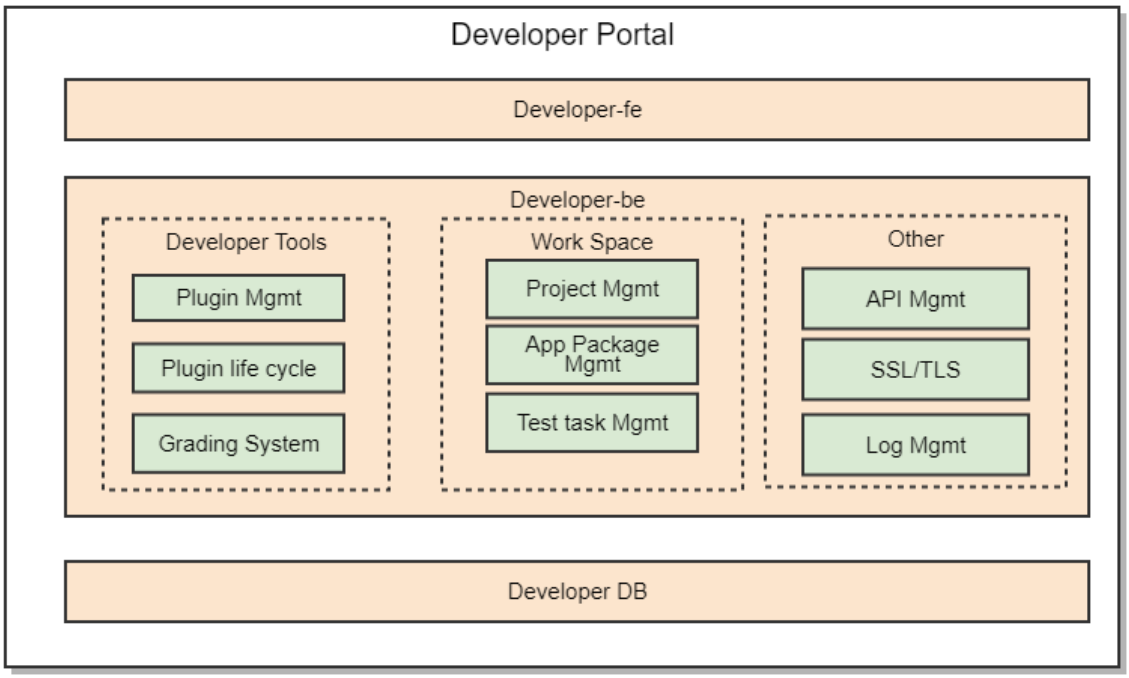
Developer

8.1 Developer Overview

提供开源开发者统一入口，包括开发流程、开发工具、开放的 API 能力、集成测试验证，开发者交流论坛等，使开发者更加方便快捷的开发应用并集成到 edgegallery 平台。Developer 分为前后台两个部分，developere-be 是后台部分，提供主要功能接口供前台或其他三方系统调用，developer-fe 是前台部分，提供界面展示。

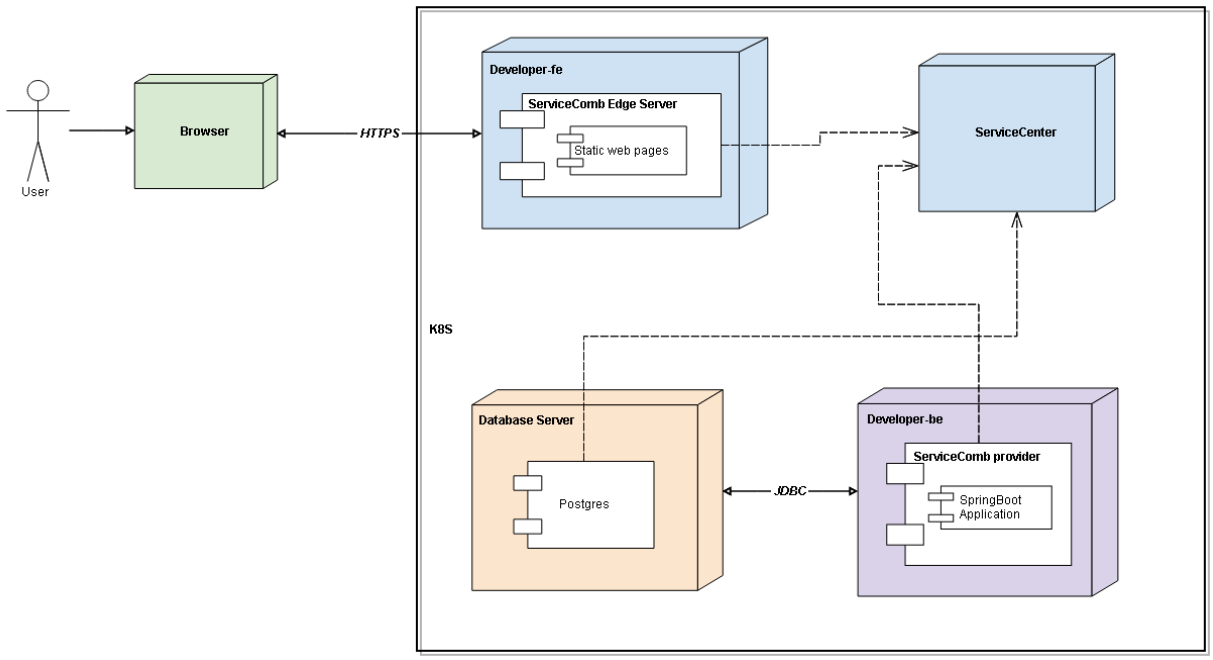
- 架构设计
- 部署视图

8.1.1 架构设计



1. **Developer-fe** 开发者平台前台框架，使用 VUE 开发。
2. **Developer-be** 开发者平台后台框架，使用 SpringBoot+ServiceComb 开发。（有关 ServiceComb 请参考这里：https://servicecomb.apache.org/cn/）
3. **Developer DB** 后台依赖 Postgres 数据库。

8.1.2 部署视图



以上部署视图是按照 K8S 部署设计的，在实际部署过程中，可以直接在虚拟机或者 docker 容器中启动各自的服务，部署过程中需要有以下几点注意：

- Developer-fe，Developer-be 需要注册到 ServiceCenter 中，有关 ServiceCenter 的使用方法请参考：https://servicecomb.apache.org/cn/
- 推荐使用 Chrome 浏览器

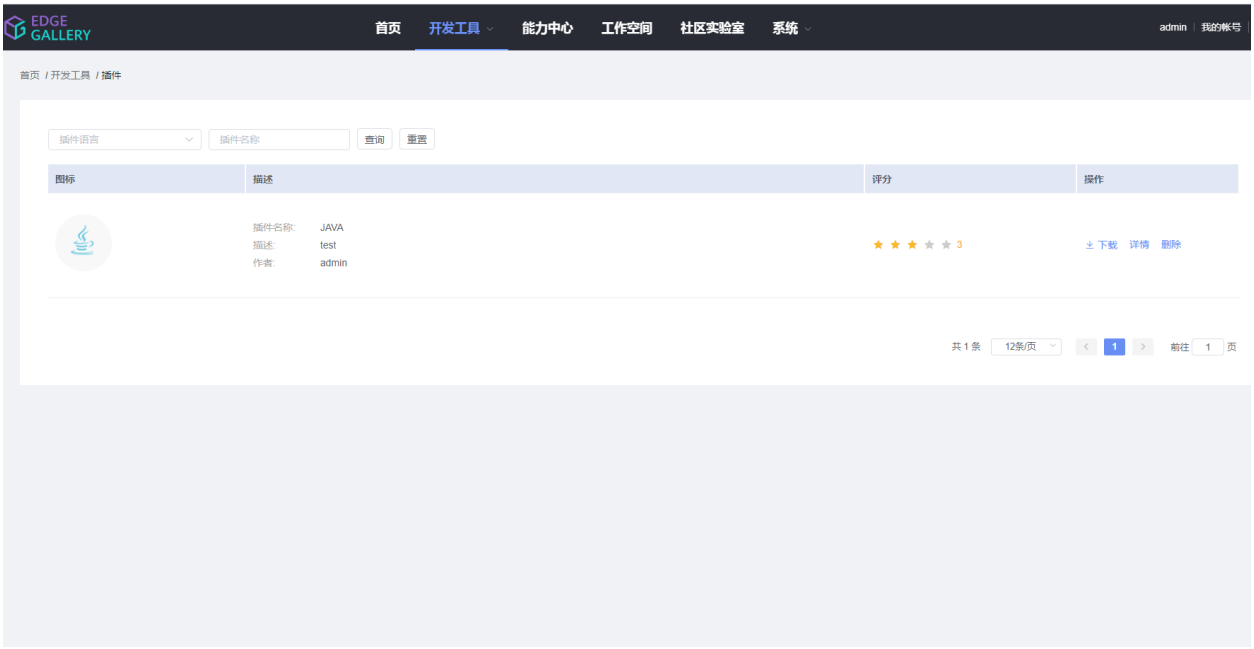
8.2 Developer 特性清单

特性名称	特性描述	设计约束
开发工具管理	提供给开发者使用的 App 开发工具，开发者可以下载使用，也可以将自研的工具分享给社区开发者使用。	
能力中心	展示所有开发者平台提供的能力，包括平台自身提供，开发者共享到此平台的能力，供开发者开发部署项目使用。	
工作空间	工作空间是给开发者提供的项目管理平台，开发者可以通过创建项目，完成 App 的开发-测试-发布的整个流程。支持容器和虚机应用的部署调测	
系统	系统菜单是给管理员提供的管理菜单。主要负责测试沙箱和能力中心（服务）的管理	

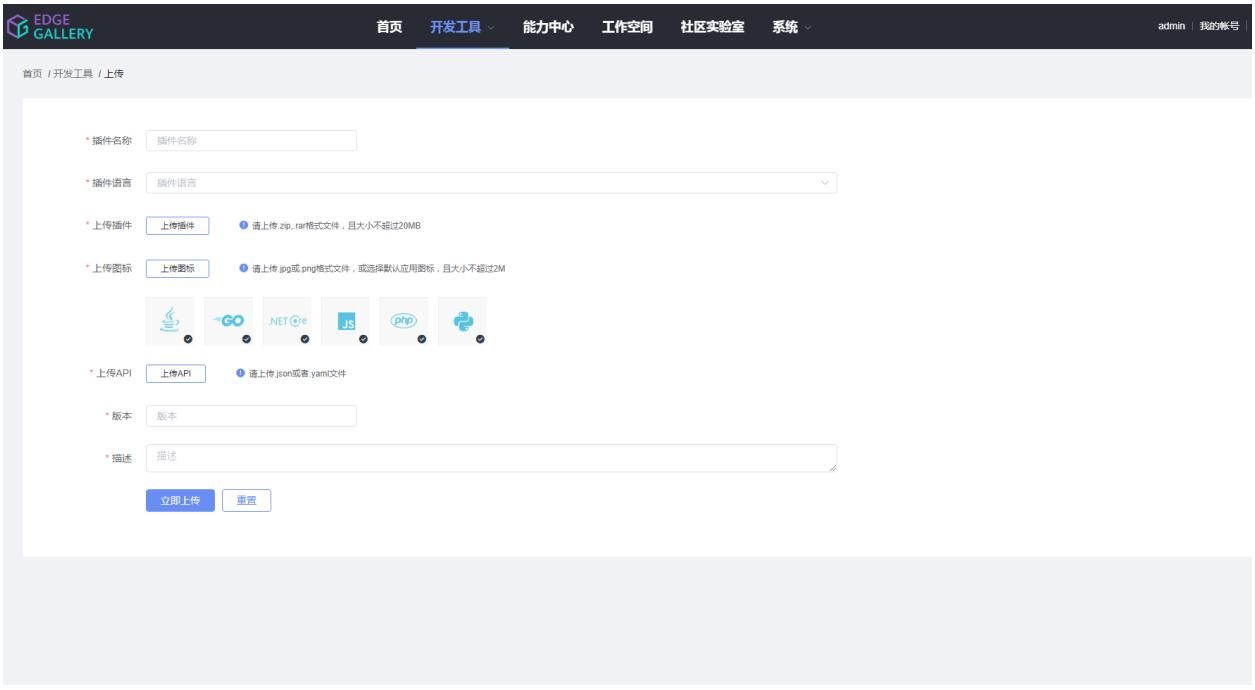
8.2.1 开发工具管理

目前开发工具管理提供生成 API 调用的样例代码和生成 csar 包格式的插件，该插件目前支持 java 和 python。

插件列表



插件上传



工具链

工具链是开发者平台提供给开发者分析源代码，定位出需要修改的地方，以便于在不同平台部署。

文档

文档菜单包含平台的简介，开发工具，以及介绍如何将能力发布到开发者平台。

8.2.2 能力中心

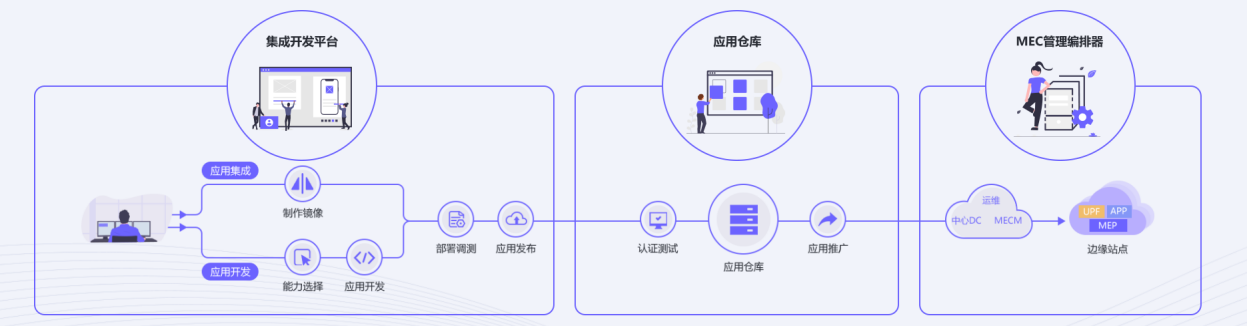
能力中心展示的是开发者平台所有的能力（开发者共享的能力和平台自身提供的能力）。包含详细的 API 描述文档，供开发者在线查看，模拟，体验。

8.2.3 工作空间

开发者在这里通过创建 project 的方式，管理自己的 APP 开发过程，有两种开发模式：

1. **应用开发**从编码开始，开发一个全新的应用
2. **应用集成**已有应用，需要将应用打包成镜像，在开发集成平台进行部署测试，最终发布到应用仓库上。

开发集成平台提供的端到端开发流程如下：



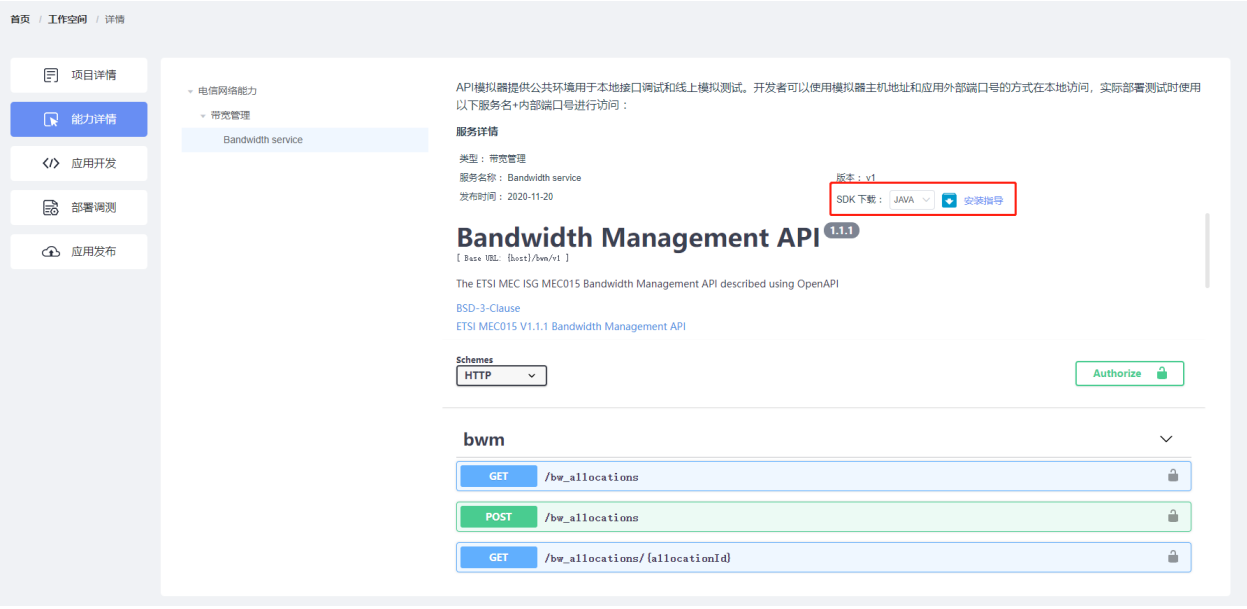
创建项目

创建项目时，需要输入项目基本信息和选择需要依赖的生态能力，当前版本支持容器应用和虚机应用。

能力详情

创建项目时选择的能力，这里会展示对应能力的详细 API，帮助开发者理解如何调用能力服务，并在后台提供 API 模拟器，可以测试 API 的请求和返回值。

针对每个能力服务，还提供 SDK 下载功能，开发者可以将 SDK 直接集成到代码中。



应用开发

针对选择依赖的能力 API，提供调用的样例代码，帮助开发者将依赖 API 快速集成到自己的代码中。

部署调测

容器应用部署调测分为三个部署：

- 上传 APP 镜像

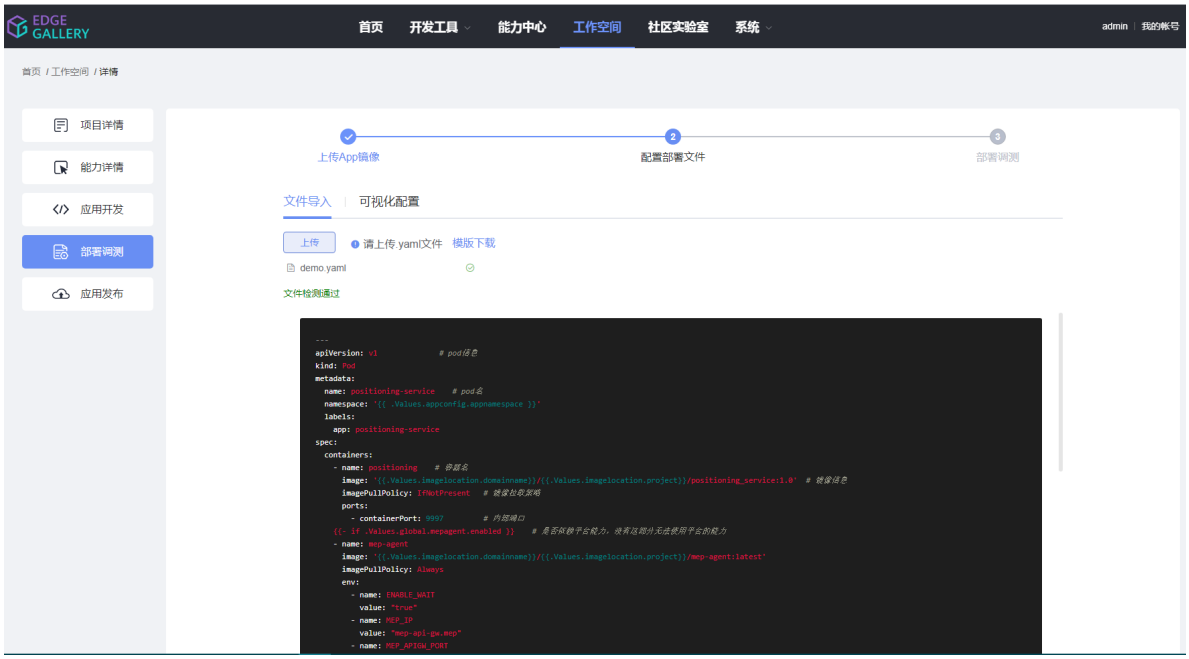
  支持两种镜像上传方式

  - 用户上传镜像打包文件。
  - 上传到 EdgeGallery 的镜像仓库。

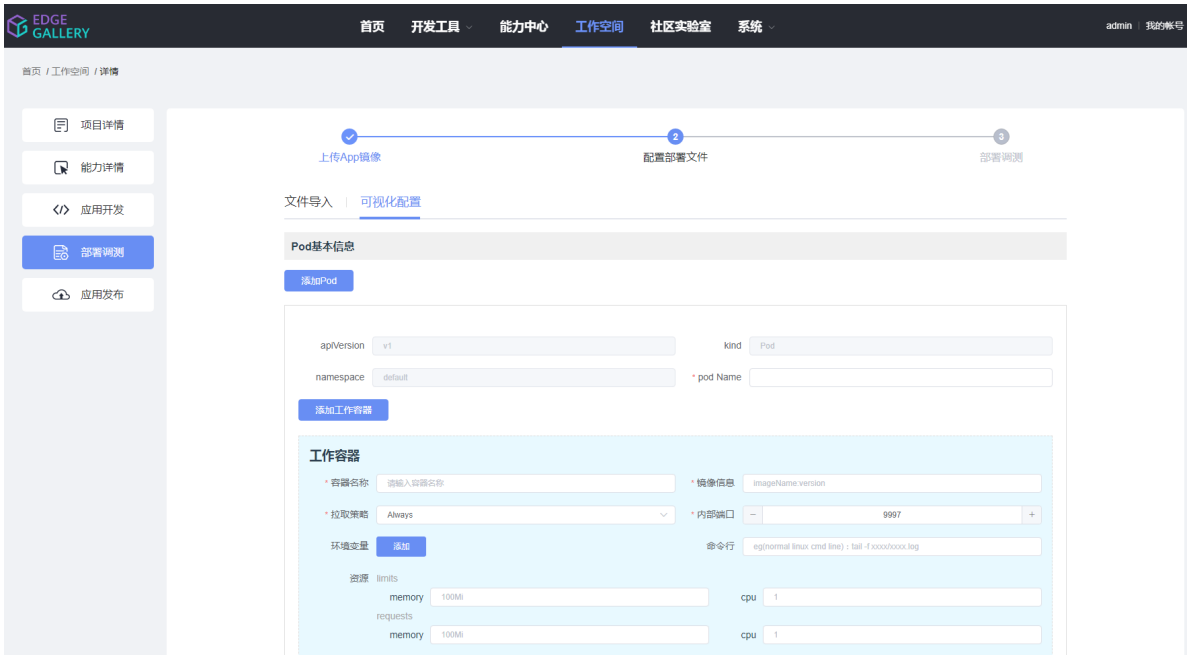
- 配置部署文件

  提供两种配置部署文件方式：

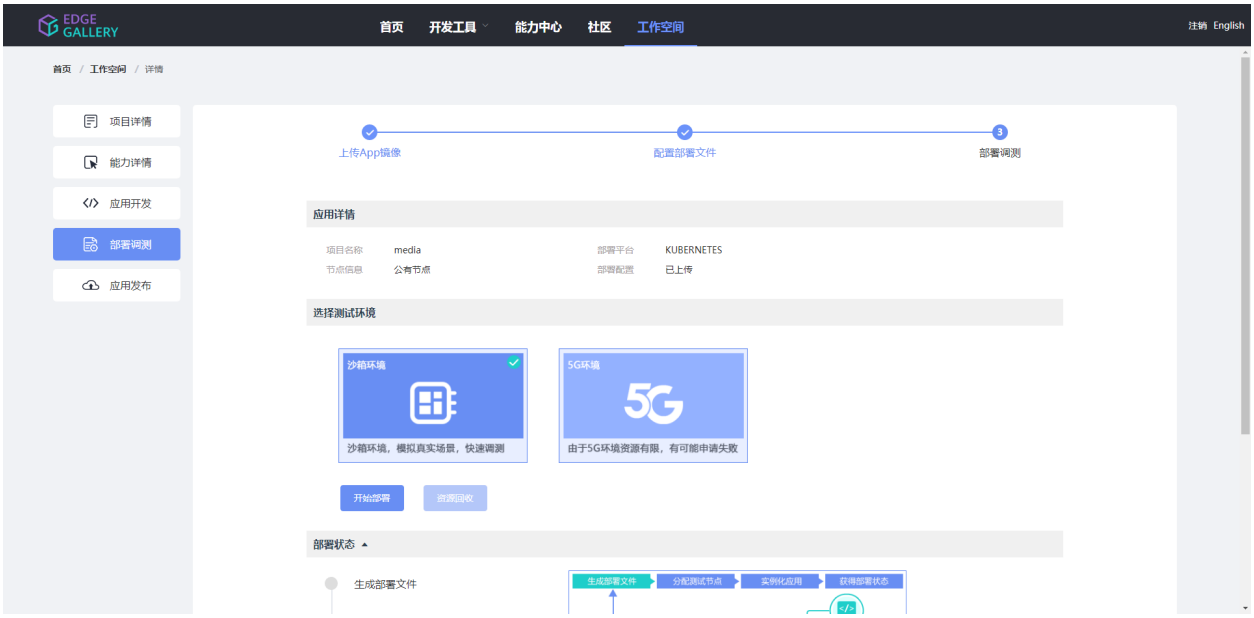
  - 用户在本地编辑好后直接上传，平台提供基本的校验功能；有关配置文件的格式，可以参考界面提供的 demo 样例，该样例可以直接上传使用。



  - 可视化配置文件，目前版本支持的部署类型为 Pod 和 Service。

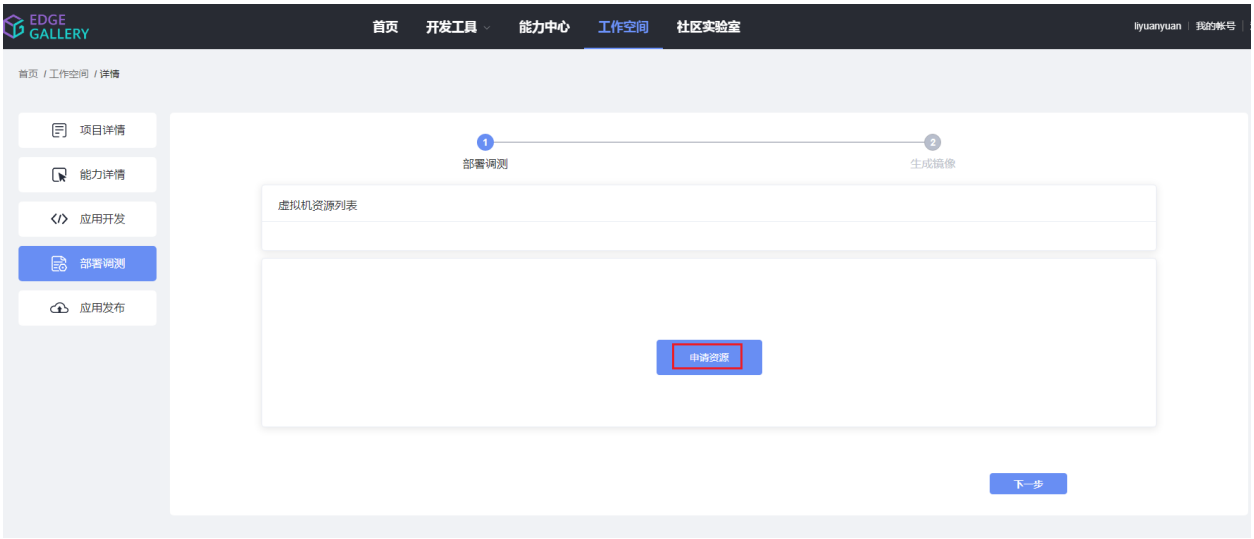


- 部署调测
  - **Step1** 部署调测前可以根据需要选择部署环境，EdgeGallery 提供‘模拟环境’和‘5G 真实’两种测试环境（v1.1 版本只实现模拟环境分配）；
  - **Step2** 启动部署，开始生成部署文件 -> 分配测试节点 -> 实例化应用 -> 获取部署状态
  - **Step3** 部署成功后，会返回部署后台的测试链接，并且展示已经部署完成 POD 信息
  - **Step4** 应用开发者通过链接对应用进行测试
  - **Step5** 测试完成后，通过【资源回收】按钮释放测试环境，如需修改，请在完成镜像修改后，循环以上 Step1->Step2->Step3->Step4 操作，对应用进行反复测试



虚机应用部署调测分为以下三个步骤:

- 选择虚机资源



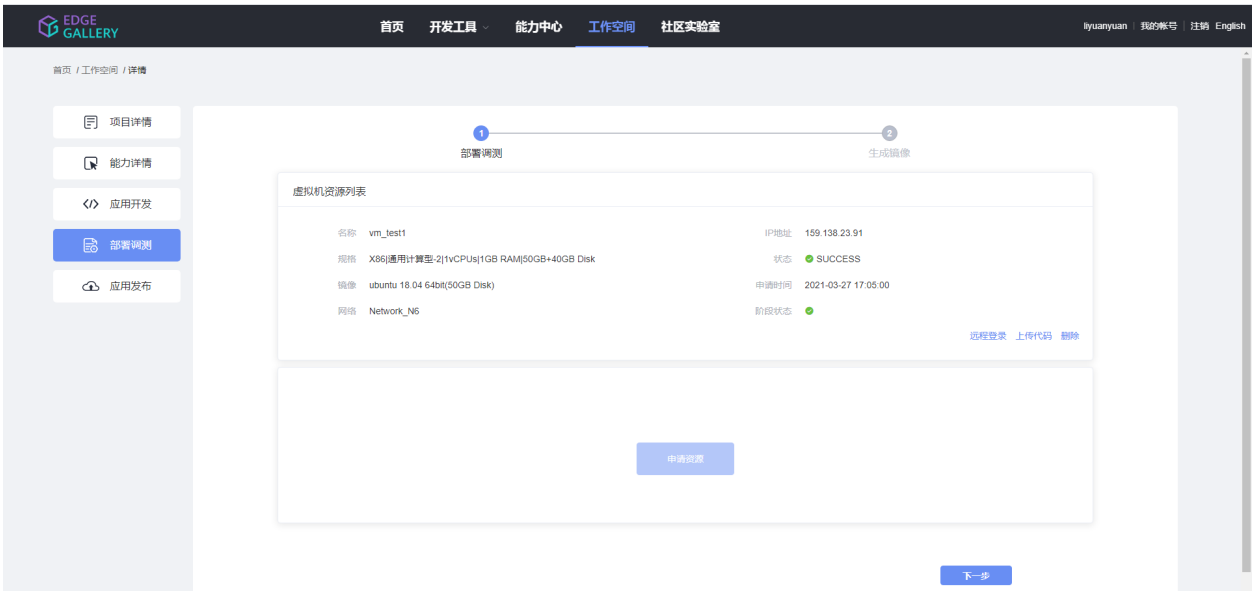
配置虚机的规格、镜像、网络等信息

- 输入虚机名称

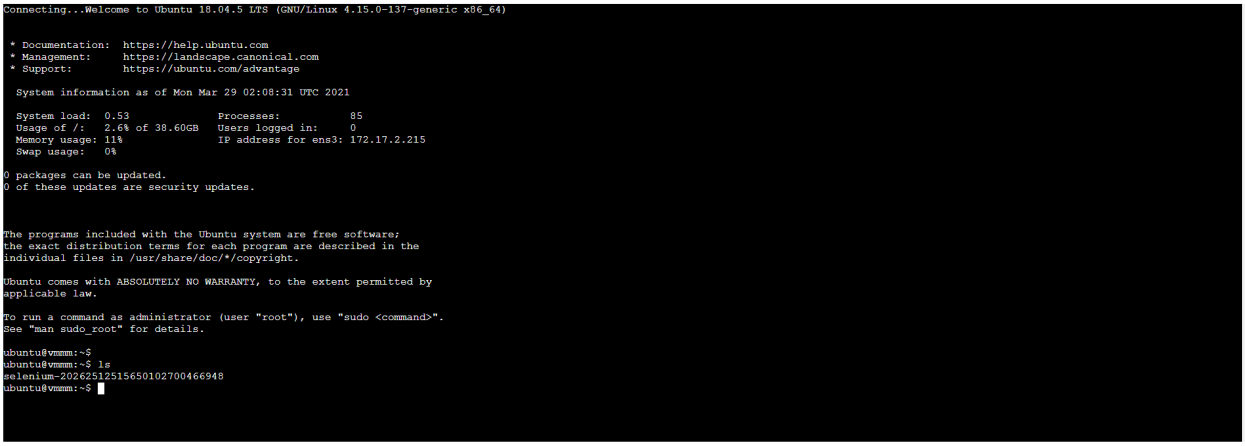
- 配置虚机资源大小，镜像源

- 设置虚机网络类型，默认勾选 N6 网络

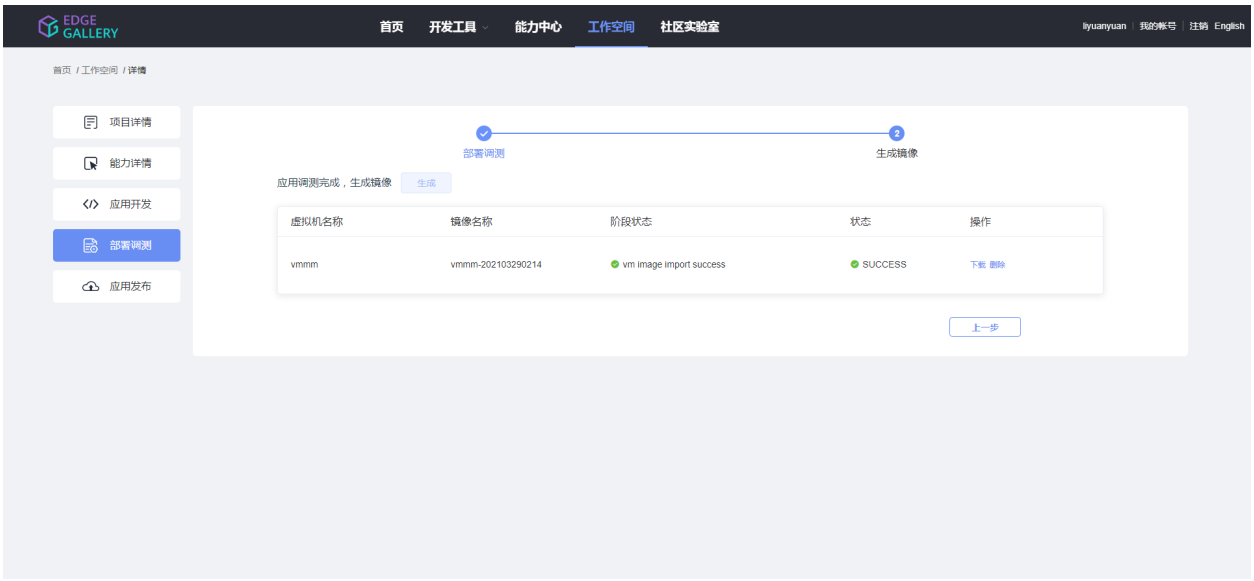
- 申请虚机



申请到的虚机可以远程登录进行应用开发调测：



- 导出镜像
  - 将镜像分片下载并打包在 csar 包中



应用发布

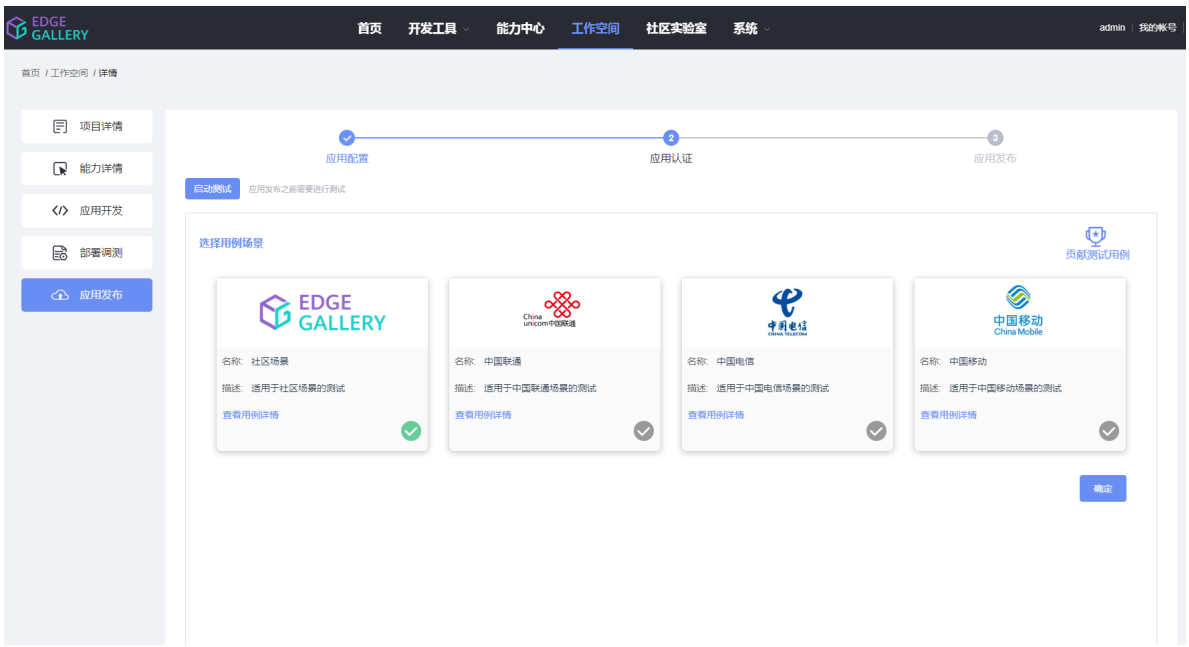
应用发布分为三个步骤：

- 应用配置
  - **可选**流规则配置
  - **可选**应用服务发布配置：如果该应用需要成为生态能力服务，需要进行该配置。需要上传对应的 API 接口文档和 MD 说明文档，具体配置如下

- 应用认证

  对接认证测试平台，所有完成开发的应用都需要通过认证测试平台的认证后才能发布到 AppStore 平台，生成的认证测试报告会附带发送给你 AppStore 平台。

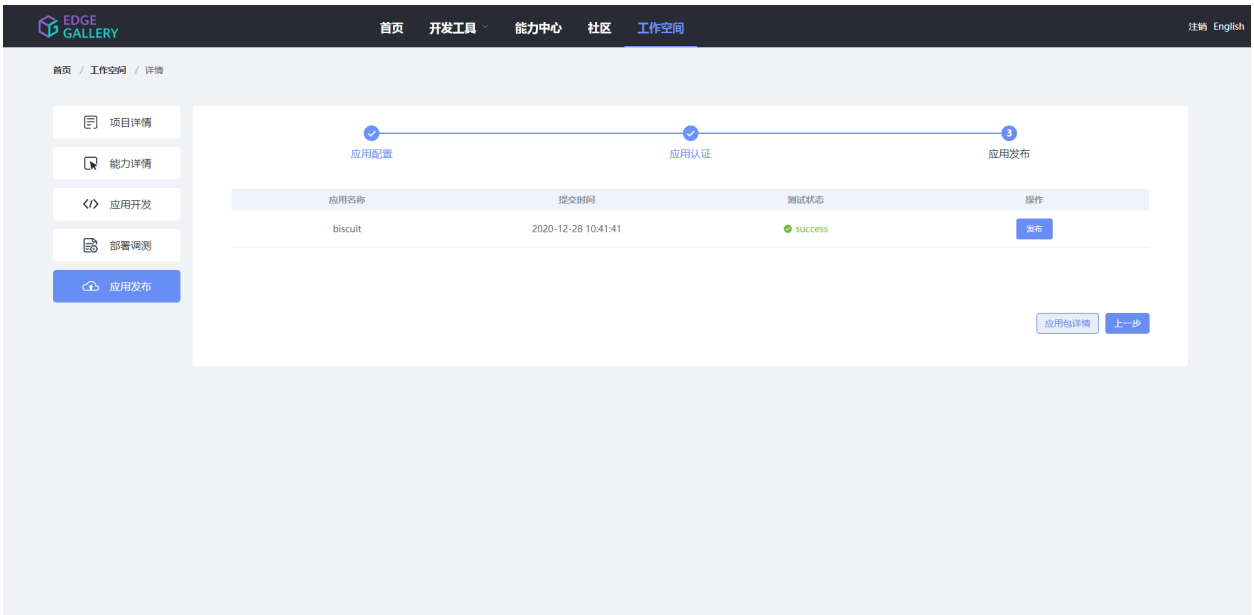
  - 选择测试场景



  - 执行测试用例

- 应用发布

认证通过的应用，通过这里可以直接发布到 EdgeGallery 的 AppStore 平台，同时如果在【应用配置】时填写了应用服务发布配置，该应用同样会做为能力服务，被收录到能力中心里，可以被其他应用开发者使用。



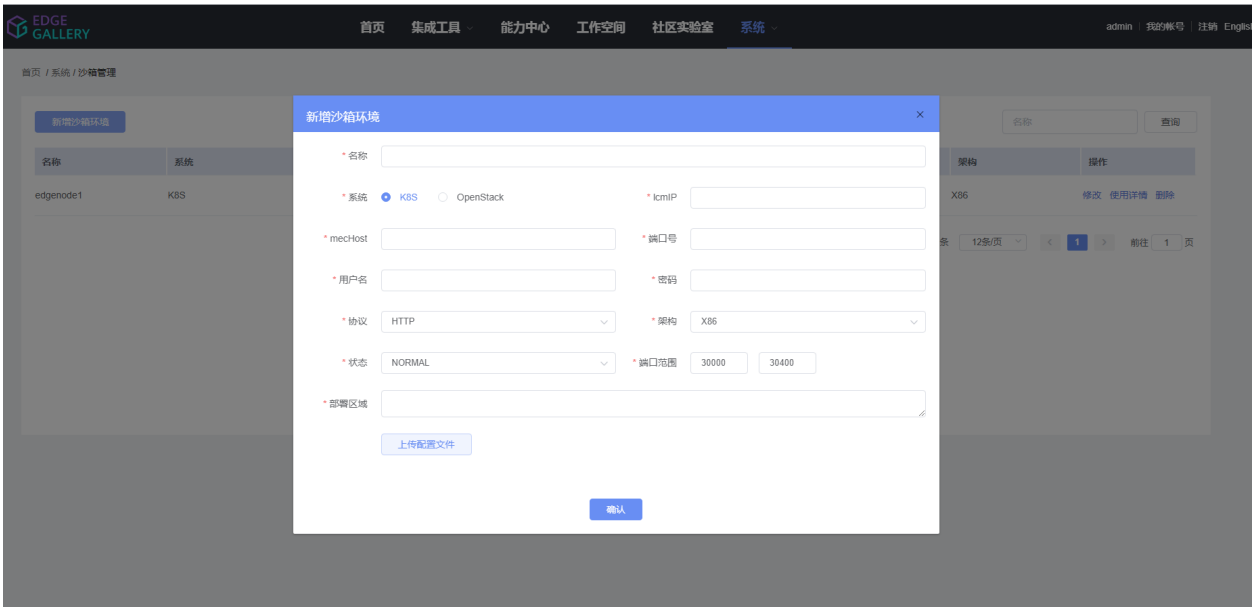
8.2.4 系统

系统菜单主要对管理员权限的用户开放，提供沙箱节点以及平台能力的管理。

沙箱管理

- 沙箱节点列表

- 新增沙箱节点



能力中心管理

- 能力中心列表

- 新增能力中心

8.3 Developer Interfaces

- *Developer Interfaces*
  - *1. Plugin*
    - *1.1 POST upload plugin*
    - *1.2 GET all plugin*
    - *1.3 DELETE one plugin*
    - *1.4 GET download plugin*
    - *1.5 GET download logo*
    - *1.6 GET download api*
    - *1.7 PUT update plugin*
    - *1.8 PUT mark plugin*
    - *1.9 GET api content*
  - *2. Container Image*
    - *2.1 POST upload container image*
    - *2.2 DELETE upload container image*
    - *2.3 POST create image*
    - *2.4 PUT modify image*
    - *2.5 DELETE container image*
    - *2.6 GET download image*
    - *2.7 POST query all image*
    - *2.8 GET merge container image*
    - *2.9 GET synchronize image*
  - *3. App Project*
    - *3.1 GET all project*
    - *3.2 GET one project*
    - *3.3 POST create one project*
    - *3.4 DELETE one project*
    - *3.5 PUT modify one project*
    - *3.6 POST deploy one project*

        * *3.7 POST clean test env*
        * *3.8 POST create test config*
        * *3.9 PUT modify test config*
        * *3.10 GET one test-config*
        * *3.11 POST upload to store*
        * *3.12 GET project atp task*
        * *3.13 GET project workstatus*
        * *3.14 POST terminate project*
    - *4. Host*
        * *4.1 GET all host*
        * *4.2 GET one host*
        * *4.3 POST create one host*
        * *4.4 DELETE one host*
        * *4.5 PUT modify one host*
        * *4.6 GET host list*
        * *4.7 GET host logs*
    - *5. Capability-groups*
        * *5.1 POST create group*
        * *5.2 GET all group*
        * *5.3 GET one group*
        * *5.4 DELETE one group*
    - *6. File*
        * *6.1 GET one file*
        * *6.2 POST upload one file*
        * *6.3 POST upload helm yaml*
        * *6.4 DELETE helm yaml*
        * *6.5 POST get sample code*
        * *6.6 GET api info*
        * *6.7 GET sdk code*
        * *6.8 GET sample content*

        * *6.9 POST pkg structure*
    - *7. Capabilities*
        * *7.1 POST create capability*
        * *7.2 GET all capability*
        * *7.3 GET one capability*
        * *7.4 DELETE one capability*
        * *7.5 PUT one capability*
    - *8. Capability-Groups Query*
        * *8.1 GET group by type*
    - *9. AppRelease*
        * *9.1 GET pkg structure*
        * *9.2 GET file content*
    - *10. ReleaseConfig*
        * *10.1 GET release config*
        * *10.2 POST release config*
        * *10.3 PUT release config*
    - *11. Health*
        * *11.1 GET health check*
    - *12. Deploy*
        * *12.1 GET deploy yaml*
        * *12.2 PUT deploy yaml*
        * *12.3 GET deploy json*
        * *12.4 POST deploy yaml*
    - *13. Capability-group stats*
        * *13.1 GET group stats*
    - *14. VirtualMachine*
        * *14.1 POST one vm*
        * *14.2 DELETE vm config*
        * *14.3 GET all image config*
        * *14.4 POST image config*

        * *14.5 DELETE image config*
        * *14.6 GET image config By ProjectId*
        * *14.7 GET csar pkg*
        * *14.8 GET vm resources*
        * *14.9 POST vm file*
        * *14.10 GET merge image*
        * *14.11 POST clean vm*
        * *14.12 GET vm pkg*
        * *14.13 POST vm pkg*
        * *14.14 DELETE vm pkg*
    - *15. Capability-group stats query*
        * *15.1 GET stats by type*
    - *16. SystemImage*
        * *16.1 POST one system image*
        * *16.2 DELETE system image*
        * *16.3 POST upload system image*
        * *16.4 PUT system image*
        * *16.5 PUT publish system image*
        * *16.6 POST system image to upload file*
        * *16.7 GET system image to merge*
        * *16.8 GET system image to download*
        * *16.9 POST slim image*
        * *16.10 PUT reset image status*
        * *16.11 POST upload image*
        * *16.12 DELETE upload image*
    - *17. Capability query*
        * *17.1 GET capability by groupid*
        * *17.2 GET capability by en name*
        * *17.3 GET capability by name*
        * *17.4 GET capability by projectid*

* *17.5 GET capability by type*

8.3.1 1. Plugin

Development environment plug-in or sdk

1.1 POST upload plugin

Upload one plugin with some params.

```
Resource URI: /mec/developer/v1/plugins/
```

Name	Definition	Type	Required
pluginFile	pluginfile	requestpart	yes
logoFile	pluginlogofile	requestpart	yes
apiFile	pluginapifile	requestpart	yes
pluginName	pluginname	requestparam	yes
codeLanguage	languagerepresentedbytheplugin	requestparam	yes
pluginType	1plugin2sdk	requestparam	yes
version	pluginversion	requestparam	yes
introduction	pluginbrief	requestparam	yes
userId	theauthor’sidofuploadplugin	requestparam	yes
userName	theauthor’snameofuploadplugin	requestparam	yes

Example response

```
200 OK
{
  "pluginId": "string",
  "pluginName": "string",
  "introduction": "string",
  "satisfaction": 0,
  "codeLanguage": "string",
  "pluginType": "string",
  "version": "string",
  "scoreCount": 0,
  "downloadCount": 0,
  "logoFile": {
    "originalFileName": "string",
    "storageAddress": "string",
    "fileSize": 0,
    "hashCode": "string",
```

(下页继续)

(续上页)

```
    "name": "string",
    "size": 0
  },
  "pluginFile": {
    "originalFileName": "string",
    "storageAddress": "string",
    "fileSize": 0,
    "hashCode": "string",
    "name": "string",
    "size": 0
  },
  "uploadTime": "2020-12-29T01:38:05.596Z",
  "user": {
    "userId": "string",
    "userName": "string"
  },
  "apiFile": {
    "originalFileName": "string",
    "storageAddress": "string",
    "fileSize": 0,
    "hashCode": "string",
    "name": "string",
    "size": 0
  }
}
```

1.2 GET all plugin

Query all plugin or sdk

```
Resource URI: /mec/developer/v1/plugins/
```

Name	Definition	Type	Required
pluginType	plugin type	request param	yes
limit	the max count of one page	request param	yes
offset	start index of the page	request param	yes

Example response

```
200 OK
{
```

(下页继续)

(续上页)

```
  "results": [
    {}
  ],
  "limit": 0,
  "offset": 0,
  "total": 0
}
```

1.3 DELETE one plugin

Delete one plugin by pluginId

```
Resource URI: /mec/developer/v1/plugins/{pluginId}
```

Name	Definition	Type	Required
pluginId	plugin id	path param	yes

Example response

```
200 OK
  true
```

1.4 GET download plugin

Download one plugin

```
Resource URI: /mec/developer/v1/plugins/{pluginId}/action/download
```

Name	Definition	Type	Required
pluginId	plugin id	path param	yes

Example response

```
200 OK
  binary output.
```

1.5 GET download logo

Download plugin’s logo file

```
Resource URI: /mec/developer/v1/plugins/{pluginId}/action/get-logofile
```

Name	Definition	Type	Required
pluginId	plugin id	path param	yes

Example response

```
200 OK
  binary output.
```

1.6 GET download api

Download plugin’s api file

```
Resource URI: /mec/developer/v1/plugins/{pluginId}/action/get-apifile
```

Name	Definition	Type	Required
pluginId	plugin id	path param	yes

Example response

```
200 OK
  binary output.
```

1.7 PUT update plugin

Update api by id

```
Resource URI: /mec/developer/v1/plugins/{pluginId}
```

Name	Definition	Type	Required
pluginId	pluginid	pathparam	yes
pluginFile	pluginfile	requestpart	no
logoFile	pluginlogofile	requestpart	no
apiFile	pluginapifile	requestpart	no
pluginName	pluginname	requestpart	no
codeLanguage	languagerepresentedbytheplugin	requestpart	no
pluginType	1plugin2sdk	requestpart	no
version	pluginversion	requestpart	no
introduction	pluginbrief	requestpart	no

Example response

```
200 OK
 {
  "pluginId": "string",
  "pluginName": "string",
  "introduction": "string",
  "satisfaction": 0,
  "codeLanguage": "string",
  "pluginType": "string",
  "version": "string",
  "scoreCount": 0,
  "downloadCount": 0,
  "logoFile": {
    "originalFileName": "string",
    "storageAddress": "string",
    "fileSize": 0,
    "hashCode": "string",
    "name": "string",
    "size": 0
  },
  "pluginFile": {
    "originalFileName": "string",
    "storageAddress": "string",
    "fileSize": 0,
    "hashCode": "string",
    "name": "string",
    "size": 0
  },
  "uploadTime": "2020-12-29T01:38:05.596Z",
  "user": {
    "userId": "string",
```

(下页继续)

(续上页)

```
    "userName": "string"
  },
  "apiFile": {
    "originalFileName": "string",
    "storageAddress": "string",
    "fileSize": 0,
    "hashCode": "string",
    "name": "string",
    "size": 0
  }
}
```

1.8 PUT mark plugin

Update plugin satisfaction by pluginId

```
Resource URI: /mec/developer/v1/plugins/{pluginId}/action/score
```

Name	Definition	Type	Required
pluginId	pluginid	pathparam	yes
score	pluginsatisfaction	requestparam	yes
userId	raterid	requestparam	yes
userName	ratername	requestparam	yes

Example response

```
200 OK
 {
  "pluginId": "string",
  "pluginName": "string",
  "introduction": "string",
  "satisfaction": 0,
  "codeLanguage": "string",
  "pluginType": "string",
  "version": "string",
  "scoreCount": 0,
  "downloadCount": 0,
  "logoFile": {
    "originalFileName": "string",
    "storageAddress": "string",
    "fileSize": 0,
```

(下页继续)

(续上页)


```
    "hashCode": "string",
    "name": "string",
    "size": 0
  },
  "pluginFile": {
    "originalFileName": "string",
    "storageAddress": "string",
    "fileSize": 0,
    "hashCode": "string",
    "name": "string",
    "size": 0
  },
  "uploadTime": "2020-12-29T01:38:05.596Z",
  "user": {
    "userId": "string",
    "userName": "string"
  },
  "apiFile": {
    "originalFileName": "string",
    "storageAddress": "string",
    "fileSize": 0,
    "hashCode": "string",
    "name": "string",
    "size": 0
  }
}
```


1.9 GET api content

Get plugin api content by pluginId

```
Resource URI: /mec/developer/v1/plugins/{pluginId}/action/content
```

Name	Definition	Type	Required
pluginId	pluginid	pathparam	yes

Example response

```
200 OK
 string output.
```

8.3.2 2. Container Image

operate about container app image

2.1 POST upload container image

Upload image with some params.

```
Resource URI: /mec/developer/v2/containerimages/{imageId}/upload
```

Name	Definition	Type	Required
imageId	imageId	path param	yes
id	id	query param	yes
chunkNumber	chunkNumber	path param	yes
chunkSize	chunkSize	query param	yes
currentChunkSize	currentChunkSize	query param	yes
totalSize	totalSize	query param	yes
identifier	identifier	query param	yes
filename	filename	query param	yes
relativePath	relativePath	query param	yes
totalChunks	totalChunks	query param	yes
type	type	query param	yes
file	file	query param	yes

Example response

```
200 OK
{
  "headers": {
    "string": [
      "string"
    ]
  },
  "body": {},
  "statusCode": "CONTINUE",
  "statusCodeValue": 0
}
```

2.2 DELETE upload container image

cancel upload image

```
Resource URI: /mec/developer/v2/containerimages/{imageId}/upload
```

Name	Definition	Type	Required
imageId	imageId	path param	yes

Example response

```
200 OK
{
  "headers": {
    "string": [
      "string"
    ]
  },
  "body": {},
  "statusCode": "CONTINUE",
  "statusCodeValue": 0
}
```

2.3 POST create image

create image record

```
Resource URI: /mec/developer/v2/containerimages/
```

Name	Definition	Type	Required
ContainerImage	ContainerImage	request body param	yes

```
ContainerImage:
{
  "imageId": "string",
  "imageName": "string",
  "imageVersion": "string",
  "userId": "string",
  "userName": "string",
  "uploadTime": "2021-10-12T08:37:38.000+0000",
  "createTime": "2021-10-12T08:37:38.000+0000",
```

(下页继续)

(续上页)

```
  "imageStatus": "UPLOAD_SUCCEED",
  "imageType": "string",
  "imagePath": "string",
  "fileName": "string"
}
```

Example response

```
200 OK
{
  "imageId": "string",
  "imageName": "string",
  "imageVersion": "string",
  "userId": "string",
  "userName": "string",
  "uploadTime": "2021-10-12T08:37:38.000+0000",
  "createTime": "2021-10-12T08:37:38.000+0000",
  "imageStatus": "UPLOAD_SUCCEED",
  "imageType": "string",
  "imagePath": "string",
  "fileName": "string"
}
```

2.4 PUT modify image

modify image record

```
Resource URI: /mec/developer/v2/containerimages/{imageId}
```

Name	Definition	Type	Required
ContainerImage	ContainerImage	request body param	yes
imageId	imageId	path param	yes

```
ContainerImage:
{
  "imageId": "string",
  "imageName": "string",
  "imageVersion": "string",
  "userId": "string",
  "userName": "string",
  "uploadTime": "2021-10-12T08:37:38.000+0000",
```

(下页继续)

(续上页)

```
  "createTime": "2021-10-12T08:37:38.000+0000",
  "imageStatus": "UPLOAD_SUCCEED",
  "imageType": "string",
  "imagePath": "string",
  "fileName": "string"
}
```

Example response

```
200 OK
{
  "imageId": "string",
  "imageName": "string",
  "imageVersion": "string",
  "userId": "string",
  "userName": "string",
  "uploadTime": "2021-10-12T08:37:38.000+0000",
  "createTime": "2021-10-12T08:37:38.000+0000",
  "imageStatus": "UPLOAD_SUCCEED",
  "imageType": "string",
  "imagePath": "string",
  "fileName": "string"
}
```

2.5 DELETE container image

delete image

```
Resource URI: /mec/developer/v2/containerimages/{imageId}
```

Name	Definition	Type	Required
imageId	imageId	path param	yes

Example response

```
200 OK
true
```

2.6 GET download image

download container image

```
Resource URI: /mec/developer/v2/containerimages/{imageId}/download
```

Name	Definition	Type	Required
imageId	imageId	path param	yes

Example response

```
200 OK
binary output.
```

2.7 POST query all image

query all container image

```
Resource URI: /mec/developer/v2/containerimages/list
```

Name	Definition	Type	Required
ContainerImageReq	ContainerImageReq	request body param	yes

```
ContainerImageReq:
{
  "imageName": "string",
  "userId": "string",
  "uploadTimeBegin": "string",
  "uploadTimeEnd": "string",
  "limit": 0,
  "offset": 0,
  "sortBy": "string",
  "sortOrder": "string"
}
```

Example response

```
200 OK
{
  "imageId": "string",
  "imageName": "string",
  "imageVersion": "string",
```

(下页继续)

(续上页)

```
  "userId": "string",
  "userName": "string",
  "uploadTime": "2021-10-12T08:17:01.000Z",
  "createTime": "2021-10-12T08:17:01.000Z",
  "imageStatus": "UPLOAD_WAIT",
  "imageType": "string",
  "imagePath": "string",
  "fileName": "string"
}
```

2.8 GET merge container image

merge container image

```
Resource URI: /mec/developer/v2/containerimages/{imageId}/merge
```

Name	Definition	Type	Required
imageId	imageId	path param	yes
fileName	fileName	query param	yes
guid	guid	query param	yes

Example response

```
200 OK
{
  "headers": {
    "string": [
      "string"
    ]
  },
  "body": {},
  "statusCode": "CONTINUE",
  "statusCodeValue": 0
}
```

2.9 GET synchronize image

synchronize container image

```
Resource URI: /mec/developer/v2/containerimages/synchronize
```

Example response

```
200 OK
{
  "headers": {
    "string": [
      "string"
    ]
  },
  "body": {},
  "statusCode": "CONTINUE",
  "statusCodeValue": 0
}
```

8.3.3 3. App Project

Edge application projects created by the developer platform

3.1 GET all project

Query all app project with userId.

```
Resource URI: /mec/developer/v1/projects/
```

Name	Definition	Type	Required
userId	userId	query param	yes
projectName	projectName	query param	yes
limit	limit	query param	yes
offset	offset	query param	yes

Example response

```
200 OK
[
  {
    "id": "string",
```

(下页继续)

(续上页)


```
    "projectType": "CREATE_NEW",
    "name": "string",
    "version": "string",
    "provider": "string",
    "platform": [
      "string"
    ],
    "deployPlatform": "KUBERNETES",
    "type": "string",
    "industry": [
      "string"
    ],
    "description": "string",
    "iconFileId": "string",
    "status": "ONLINE",
    "capabilityList": [
      "string"
    ],
    "lastTestId": "string",
    "userId": "string",
    "createDate": "string",
    "openCapabilityId": "string",
    "applicationProjectCapabilities": [
      {
        "projectId": "string",
        "capabilityId": "string"
      }
    ]
  }
]
```


3.2 GET one project

Query project by id

```
Resource URI:  /mec/developer/v1/projects/{projectId}
```

Name	Definition	Type	Required
projectId	projectid	path param	yes
userId	userId	query param	yes

Example response

```
200 OK
{
  "id": "string",
  "projectType": "CREATE_NEW",
  "name": "string",
  "version": "string",
  "provider": "string",
  "platform": [
    "string"
  ],
  "deployPlatform": "KUBERNETES",
  "type": "string",
  "industry": [
    "string"
  ],
  "description": "string",
  "iconFileId": "string",
  "status": "ONLINE",
  "capabilityList": [
    "string"
  ],
  "lastTestId": "string",
  "userId": "string",
  "createDate": "string",
  "openCapabilityId": "string",
  "applicationProjectCapabilities": [
    {
      "projectId": "string",
      "capabilityId": "string"
    }
  ]
}
```

3.3 POST create one project

Create one project with some params

```
Resource URI: /mec/developer/v1/projects/
```

Name	Definition	Type	Required
ApplicationProject	Entityclass	body param	yes
userId	userId	query param	yes

```
ApplicationProject:
{
  "id": "string",
  "projectType": "CREATE_NEW",
  "name": "string",
  "version": "string",
  "provider": "string",
  "platform": [
    "string"
  ],
  "deployPlatform": "KUBERNETES",
  "type": "string",
  "industry": [
    "string"
  ],
  "description": "string",
  "iconFileId": "string",
  "status": "ONLINE",
  "capabilityList": [
    "string"
  ],
  "lastTestId": "string",
  "userId": "string",
  "createDate": "string",
  "openCapabilityId": "string",
  "applicationProjectCapabilities": [
    {
      "projectId": "string",
      "capabilityId": "string"
    }
  ]
}
```

Example response

```
{
  "id": "string",
  "projectType": "CREATE_NEW",
  "name": "string",
  "version": "string",
  "provider": "string",
  "platform": [
    "string"
  ],
```

(下页继续)

(续上页)


```
  "deployPlatform": "KUBERNETES",
  "type": "string",
  "industry": [
    "string"
  ],
  "description": "string",
  "iconFileId": "string",
  "status": "ONLINE",
  "capabilityList": [
    "string"
  ],
  "lastTestId": "string",
  "userId": "string",
  "createDate": "string",
  "openCapabilityId": "string",
  "applicationProjectCapabilities": [
    {
      "projectId": "string",
      "capabilityId": "string"
    }
  ]
}
```


3.4 DELETE one project

Delete one project by id

```
Resource URI: /mec/developer/v1/projects/{projectId}
```

Name	Definition	Type	Required
projectId	project id	path param	yes
userId	the author id of create project	request param	yes

Example response


```
200 OK
true
```

3.5 PUT modify one project

Modify one project by id

```
Resource URI: /mec/developer/v1/projects/{projectId}
```

Name	Definition	Type	Required
projectId	projectid	path param	yes
userId	userId	request param	yes
ApplicationProject	entityclass	bodyparam	yes

```
ApplicationProject
{
  "id": "string",
  "projectType": "CREATE_NEW",
  "name": "string",
  "version": "string",
  "provider": "string",
  "platform": [
    "string"
  ],
  "deployPlatform": "KUBERNETES",
  "type": "string",
  "industry": [
    "string"
  ],
  "description": "string",
  "iconFileId": "string",
  "status": "ONLINE",
  "capabilityList": [
    "string"
  ],
  "lastTestId": "string",
  "userId": "string",
  "createDate": "string",
  "openCapabilityId": "string",
  "applicationProjectCapabilities": [
    {
      "projectId": "string",
      "capabilityId": "string"
    }
  ]
}
```

Example response

```
200 OK
{
  "id": "string",
  "projectType": "CREATE_NEW",
  "name": "string",
  "version": "string",
  "provider": "string",
  "platform": [
    "string"
  ],
  "deployPlatform": "KUBERNETES",
  "type": "string",
  "industry": [
    "string"
  ],
  "description": "string",
  "iconFileId": "string",
  "status": "ONLINE",
  "capabilityList": [
    "string"
  ],
  "lastTestId": "string",
  "userId": "string",
  "createDate": "string",
  "openCapabilityId": "string",
  "applicationProjectCapabilities": [
    {
      "projectId": "string",
      "capabilityId": "string"
    }
  ]
}
```

3.6 POST deploy one project

deploy one project by id

```
Resource URI: /mec/developer/v1/projects/{projectId}/action/deploy
```

Name	Definition	Type	Required
projectId	projectid	path param	yes
userId	userId	request param	yes

Example response

```
200 OK
{
  "id": "string",
  "projectType": "CREATE_NEW",
  "name": "string",
  "version": "string",
  "provider": "string",
  "platform": [
    "string"
  ],
  "deployPlatform": "KUBERNETES",
  "type": "string",
  "industry": [
    "string"
  ],
  "description": "string",
  "iconFileId": "string",
  "status": "ONLINE",
  "capabilityList": [
    "string"
  ],
  "lastTestId": "string",
  "userId": "string",
  "createDate": "string",
  "openCapabilityId": "string",
  "applicationProjectCapabilities": [
    {
      "projectId": "string",
      "capabilityId": "string"
    }
  ]
}
```

3.7 POST clean test env

Clean test env

```
Resource URI: /mec/developer/v1/projects/{projectId}/action/clean
```

Name	Definition	Type	Required
projectId	projectid	path param	yes
userId	userId	request param	yes

Example response

```
200 OK
true
```

3.8 POST create test config

Create test config of one project

```
Resource URI: /mec/developer/v1/projects/{projectId}/test-config
```

Name	Definition	Type	Required
projectId	projectid	path param	yes
userId	userId	request param	yes
ProjectTestConfig	entityclass	bodyparam	yes

```
ProjectTestConfig
{
  "testId": "string",
  "projectId": "string",
  "platform": "KUBERNETES",
  "deployFileId": "string",
  "privateHost": false,
  "pods": "string",
  "deployStatus": "NOTDEPLOY",
  "stageStatus": {
    "csar": "Success",
    "hostInfo": "Success",
    "instantiateInfo": "Success",
    "workStatus": "Success"
  },
  "hosts": [
    {
      "hostId": "string",
      "name": "string",
      "address": "string",
      "architecture": "string",
      "status": "NORMAL",
      "lcmIp": "string",
      "mecHost": "string",
      "protocol": "string",
      "port": 0,
```

(下页继续)

(续上页)


```
      "os": "string",
      "portRangeMin": 0,
      "portRangeMax": 0,
      "userId": "string",
      "configId": "string",
      "userName": "string",
      "password": "string",
      "vncPort": 0,
      "parameter": "string",
      "resource": "string"
    }
  ],
  "errorLog": "string",
  "workLoadId": "string",
  "appInstanceId": "string",
  "deployDate": "2021-10-12T08:56:43.051Z",
  "lcmToken": "string",
  "agentConfig": {
    "serviceName": "string",
    "href": "string",
    "port": 0
  },
  "imageFileIds": [
    "string"
  ],
  "appImages": [
    {
      "imageId": "string",
      "serviceName": "string",
      "imageName": "string",
      "version": "string",
      "env": [
        {
          "name": "string",
          "value": "string"
        }
      ],
      "ports": [
        {
          "containerPort": 0,
          "protocol": "string",
          "nodePort": 0
        }
```


(下页继续)

(续上页)

```
      ],
      "imagePullPolicy": "IF_NOT_PRESENT",
      "volumeMounts": [
        {
          "name": "string",
          "mountPath": "string"
        }
      ]
    }
  ],
  "otherImages": [
    {
      "imageId": "string",
      "serviceName": "string",
      "imageName": "string",
      "version": "string",
      "env": [
        {
          "name": "string",
          "value": "string"
        }
      ],
      "ports": [
        {
          "containerPort": 0,
          "protocol": "string",
          "nodePort": 0
        }
      ],
      "imagePullPolicy": "IF_NOT_PRESENT",
      "volumeMounts": [
        {
          "name": "string",
          "mountPath": "string"
        }
      ]
    }
  ],
  "appApiFileId": "string",
  "accessUrl": "string",
  "packageId": "string",
  "nextStage": "string"
}
```

Example response

```
200 OK
{
  "testId": "string",
  "projectId": "string",
  "platform": "KUBERNETES",
  "deployFileId": "string",
  "privateHost": false,
  "pods": "string",
  "deployStatus": "NOTDEPLOY",
  "stageStatus": {
    "csar": "Success",
    "hostInfo": "Success",
    "instantiateInfo": "Success",
    "workStatus": "Success"
  },
  "hosts": [
    {
      "hostId": "string",
      "name": "string",
      "address": "string",
      "architecture": "string",
      "status": "NORMAL",
      "lcmIp": "string",
      "mecHost": "string",
      "protocol": "string",
      "port": 0,
      "os": "string",
      "portRangeMin": 0,
      "portRangeMax": 0,
      "userId": "string",
      "configId": "string",
      "userName": "string",
      "password": "string",
      "vncPort": 0,
      "parameter": "string",
      "resource": "string"
    }
  ],
  "errorLog": "string",
  "workLoadId": "string",
  "appInstanceId": "string",
  "deployDate": "2021-10-12T08:56:43.051Z",
  "lcmToken": "string",
```

(下页继续)

(续上页)

```
  "agentConfig": {
    "serviceName": "string",
    "href": "string",
    "port": 0
  },
  "imageFileIds": [
    "string"
  ],
  "appImages": [
    {
      "imageId": "string",
      "serviceName": "string",
      "imageName": "string",
      "version": "string",
      "env": [
        {
          "name": "string",
          "value": "string"
        }
      ],
      "ports": [
        {
          "containerPort": 0,
          "protocol": "string",
          "nodePort": 0
        }
      ],
      "imagePullPolicy": "IF_NOT_PRESENT",
      "volumeMounts": [
        {
          "name": "string",
          "mountPath": "string"
        }
      ]
    }
  ],
  "otherImages": [
    {
      "imageId": "string",
      "serviceName": "string",
      "imageName": "string",
      "version": "string",
      "env": [
```

(下页继续)

(续上页)

```
        {
          "name": "string",
          "value": "string"
        }
      ],
      "ports": [
        {
          "containerPort": 0,
          "protocol": "string",
          "nodePort": 0
        }
      ],
      "imagePullPolicy": "IF_NOT_PRESENT",
      "volumeMounts": [
        {
          "name": "string",
          "mountPath": "string"
        }
      ]
    }
  ],
  "appApiFileId": "string",
  "accessUrl": "string",
  "packageId": "string",
  "nextStage": "string"
}
```

3.9 PUT modify test config

Modify test config of one project

```
Resource URI: /mec/developer/v1/projects/{projectId}/test-config
```

Name	Definition	Type	Required
projectId	projectid	pathparam	yes
ProjectTestConfig	entityclass	bodyparam	yes

```
ProjectTestConfig
{
  "testId": "string",
  "projectId": "string",
```

(下页继续)

(续上页)


```
  "platform": "KUBERNETES",
  "deployFileId": "string",
  "privateHost": false,
  "pods": "string",
  "deployStatus": "NOTDEPLOY",
  "stageStatus": {
    "csar": "Success",
    "hostInfo": "Success",
    "instantiateInfo": "Success",
    "workStatus": "Success"
  },
  "hosts": [
    {
      "hostId": "string",
      "name": "string",
      "address": "string",
      "architecture": "string",
      "status": "NORMAL",
      "lcmIp": "string",
      "mecHost": "string",
      "protocol": "string",
      "port": 0,
      "os": "string",
      "portRangeMin": 0,
      "portRangeMax": 0,
      "userId": "string",
      "configId": "string",
      "userName": "string",
      "password": "string",
      "vncPort": 0,
      "parameter": "string",
      "resource": "string"
    }
  ],
  "errorLog": "string",
  "workLoadId": "string",
  "appInstanceId": "string",
  "deployDate": "2021-10-12T08:56:43.051Z",
  "lcmToken": "string",
  "agentConfig": {
    "serviceName": "string",
    "href": "string",
    "port": 0
```


(下页继续)

(续上页)

```
  },
  "imageFileIds": [
    "string"
  ],
  "appImages": [
    {
      "imageId": "string",
      "serviceName": "string",
      "imageName": "string",
      "version": "string",
      "env": [
        {
          "name": "string",
          "value": "string"
        }
      ],
      "ports": [
        {
          "containerPort": 0,
          "protocol": "string",
          "nodePort": 0
        }
      ],
      "imagePullPolicy": "IF_NOT_PRESENT",
      "volumeMounts": [
        {
          "name": "string",
          "mountPath": "string"
        }
      ]
    }
  ],
  "otherImages": [
    {
      "imageId": "string",
      "serviceName": "string",
      "imageName": "string",
      "version": "string",
      "env": [
        {
          "name": "string",
          "value": "string"
        }
```

(下页继续)

(续上页)

```
      ],
      "ports": [
        {
          "containerPort": 0,
          "protocol": "string",
          "nodePort": 0
        }
      ],
      "imagePullPolicy": "IF_NOT_PRESENT",
      "volumeMounts": [
        {
          "name": "string",
          "mountPath": "string"
        }
      ]
    }
  ],
  "appApiFileId": "string",
  "accessUrl": "string",
  "packageId": "string",
  "nextStage": "string"
}
```

Example response

```
200 OK
{
  "testId": "string",
  "projectId": "string",
  "platform": "KUBERNETES",
  "deployFileId": "string",
  "privateHost": false,
  "pods": "string",
  "deployStatus": "NOTDEPLOY",
  "stageStatus": {
    "csar": "Success",
    "hostInfo": "Success",
    "instantiateInfo": "Success",
    "workStatus": "Success"
  },
  "hosts": [
    {
      "hostId": "string",
      "name": "string",
```

(下页继续)

(续上页)


```
      "address": "string",
      "architecture": "string",
      "status": "NORMAL",
      "lcmIp": "string",
      "mecHost": "string",
      "protocol": "string",
      "port": 0,
      "os": "string",
      "portRangeMin": 0,
      "portRangeMax": 0,
      "userId": "string",
      "configId": "string",
      "userName": "string",
      "password": "string",
      "vncPort": 0,
      "parameter": "string",
      "resource": "string"
    }
  ],
  "errorLog": "string",
  "workLoadId": "string",
  "appInstanceId": "string",
  "deployDate": "2021-10-12T08:56:43.051Z",
  "lcmToken": "string",
  "agentConfig": {
    "serviceName": "string",
    "href": "string",
    "port": 0
  },
  "imageFileIds": [
    "string"
  ],
  "appImages": [
    {
      "imageId": "string",
      "serviceName": "string",
      "imageName": "string",
      "version": "string",
      "env": [
        {
          "name": "string",
          "value": "string"
        }
```


(下页继续)

(续上页)

```
      ],
      "ports": [
        {
          "containerPort": 0,
          "protocol": "string",
          "nodePort": 0
        }
      ],
      "imagePullPolicy": "IF_NOT_PRESENT",
      "volumeMounts": [
        {
          "name": "string",
          "mountPath": "string"
        }
      ]
    }
  ],
  "otherImages": [
    {
      "imageId": "string",
      "serviceName": "string",
      "imageName": "string",
      "version": "string",
      "env": [
        {
          "name": "string",
          "value": "string"
        }
      ],
      "ports": [
        {
          "containerPort": 0,
          "protocol": "string",
          "nodePort": 0
        }
      ],
      "imagePullPolicy": "IF_NOT_PRESENT",
      "volumeMounts": [
        {
          "name": "string",
          "mountPath": "string"
        }
      ]
```

(下页继续)

(续上页)

```
    }
  ],
  "appApiFileId": "string",
  "accessUrl": "string",
  "packageId": "string",
  "nextStage": "string"
}
```

3.10 GET one test-config

Query one test config by projectId

```
Resource URI: /mec/developer/v1/projects/{projectId}/test-config
```

Name	Definition	Type	Required
projectId	projectid	path param	yes

Example response

```
200 OK
{
  "testId": "string",
  "projectId": "string",
  "platform": "KUBERNETES",
  "deployFileId": "string",
  "privateHost": false,
  "pods": "string",
  "deployStatus": "NOTDEPLOY",
  "stageStatus": {
    "csar": "Success",
    "hostInfo": "Success",
    "instantiateInfo": "Success",
    "workStatus": "Success"
  },
  "hosts": [
    {
      "hostId": "string",
      "name": "string",
      "address": "string",
      "architecture": "string",
      "status": "NORMAL",
      "lcmIp": "string",
```

(下页继续)

(续上页)

```
      "mecHost": "string",
      "protocol": "string",
      "port": 0,
      "os": "string",
      "portRangeMin": 0,
      "portRangeMax": 0,
      "userId": "string",
      "configId": "string",
      "userName": "string",
      "password": "string",
      "vncPort": 0,
      "parameter": "string",
      "resource": "string"
    }
  ],
  "errorLog": "string",
  "workLoadId": "string",
  "appInstanceId": "string",
  "deployDate": "2021-10-12T08:56:43.051Z",
  "lcmToken": "string",
  "agentConfig": {
    "serviceName": "string",
    "href": "string",
    "port": 0
  },
  "imageFileIds": [
    "string"
  ],
  "appImages": [
    {
      "imageId": "string",
      "serviceName": "string",
      "imageName": "string",
      "version": "string",
      "env": [
        {
          "name": "string",
          "value": "string"
        }
      ],
      "ports": [
        {
          "containerPort": 0,
```

(下页继续)

(续上页)

```
          "protocol": "string",
          "nodePort": 0
        }
      ],
      "imagePullPolicy": "IF_NOT_PRESENT",
      "volumeMounts": [
        {
          "name": "string",
          "mountPath": "string"
        }
      ]
    }
  ],
  "otherImages": [
    {
      "imageId": "string",
      "serviceName": "string",
      "imageName": "string",
      "version": "string",
      "env": [
        {
          "name": "string",
          "value": "string"
        }
      ],
      "ports": [
        {
          "containerPort": 0,
          "protocol": "string",
          "nodePort": 0
        }
      ],
      "imagePullPolicy": "IF_NOT_PRESENT",
      "volumeMounts": [
        {
          "name": "string",
          "mountPath": "string"
        }
      ]
    }
  ],
  "appApiFileId": "string",
  "accessUrl": "string",
```

(下页继续)

(续上页)

```
  "packageId": "string",
  "nextStage": "string"
}
```

3.11 POST upload to store

Upload app project to developer

```
Resource URI: /mec/developer/v1/projects/{projectId}/action/upload
```

Name	Definition	Type	Required
projectId	projectid	path param	yes
userId	userId	request param	yes
userName	userName	request param	yes

Example response

```
200 OK
"csarId"
```

3.12 GET project atp task

get project atp task

```
Resource URI: /mec/developer/v1/projects/{projectId}/action/atp
```

Name	Definition	Type	Required
projectId	projectid	path param	yes

Example response

```
200 OK
true
```

3.13 GET project workstatus

GET project works tatus

```
Resource URI: /mec/developer/v1/projects/{projectId}/container/workStatus
```

Name	Definition	Type	Required
projectId	projectid	path param	yes

Example response

```
200 OK
true
```

3.14 POST terminate project

terminate project

```
Resource URI: /mec/developer/v1/projects/{projectId}/action/terminate
```

Name	Definition	Type	Required
projectId	projectid	path param	yes
userId	userId	query param	yes

Example response

```
200 OK
{
  "id": "string",
  "projectType": "CREATE_NEW",
  "name": "string",
  "version": "string",
  "provider": "string",
  "platform": [
    "string"
  ],
  "deployPlatform": "KUBERNETES",
  "type": "string",
  "industry": [
    "string"
  ],
  "description": "string",
```

(下页继续)

(续上页)

```
  "iconFileId": "string",
  "status": "ONLINE",
  "capabilityList": [
    "string"
  ],
  "lastTestId": "string",
  "userId": "string",
  "createDate": "string",
  "openCapabilityId": "string",
  "applicationProjectCapabilities": [
    {
      "projectId": "string",
      "capabilityId": "string"
    }
  ]
}
```

8.3.4 4. Host

Server of build and test or deploy app

4.1 GET all host

Query all host.

```
Resource URI: /mec/developer/v1/hosts
```

Name	Definition	Type	Required
userId	userId	query param	yes
name	name	query param	yes
ip	ip	query param	yes
limit	limit	query param	yes
offset	offset	query param	yes

Example response

```
200 OK
[
  {
    "hostId": "string",
```

(下页继续)

(续上页)

```
    "name": "string",
    "address": "string",
    "architecture": "string",
    "status": "NORMAL",
    "lcmIp": "string",
    "mecHost": "string",
    "protocol": "string",
    "port": 0,
    "os": "string",
    "portRangeMin": 0,
    "portRangeMax": 0,
    "userId": "string",
    "configId": "string",
    "userName": "string",
    "password": "string",
    "vncPort": 0,
    "parameter": "string",
    "resource": "string"
  }
]
```

4.2 GET one host

Query host by id.

```
Resource URI: /mec/developer/v1/hosts/{hostId}
```

Name	Definition	Type	Required
hostId	hostid	path param	yes

Example response

```
200 OK
{
  "hostId": "string",
  "name": "string",
  "address": "string",
  "architecture": "string",
  "status": "NORMAL",
  "lcmIp": "string",
  "mecHost": "string",
  "protocol": "string",
```

(下页继续)

(续上页)

```
  "port": 0,
  "os": "string",
  "portRangeMin": 0,
  "portRangeMax": 0,
  "userId": "string",
  "configId": "string",
  "userName": "string",
  "password": "string",
  "vncPort": 0,
  "parameter": "string",
  "resource": "string"
}
```

4.3 POST create one host

Create one host.

```
Resource URI: /mec/developer/v1/hosts/
```

Name	Definition	Type	Required
MepHost	entityclass	bodyparam	yes

```
MepHost
{
  "hostId": "string",
  "name": "string",
  "address": "string",
  "architecture": "string",
  "status": "NORMAL",
  "lcmIp": "string",
  "mecHost": "string",
  "protocol": "string",
  "port": 0,
  "os": "string",
  "portRangeMin": 0,
  "portRangeMax": 0,
  "userId": "string",
  "configId": "string",
  "userName": "string",
  "password": "string",
  "vncPort": 0,
```

(下页继续)

(续上页)

```
  "parameter": "string",
  "resource": "string"
}
```

Example response

```
200 OK
true
```

4.4 DELETE one host

Delete one host.

```
Resource URI: /mec/developer/v1/hosts/{hostId}
```

Name	Definition	Type	Required
hostId	hostid	pathparam	yes

Example response

```
200 OK
true
```

4.5 PUT modify one host

modify one host.

```
Resource URI: /mec/developer/v1/hosts/{hostId}
```

Name	Definition	Type	Required
hostId	hostid	pathparam	yes
MepHost	entityclass	bodyparam	yes

```
MepHost
{
  "hostId": "string",
  "name": "string",
  "address": "string",
  "architecture": "string",
```

(下页继续)

(续上页)

```
  "status": "NORMAL",
  "lcmIp": "string",
  "mecHost": "string",
  "protocol": "string",
  "port": 0,
  "os": "string",
  "portRangeMin": 0,
  "portRangeMax": 0,
  "userId": "string",
  "configId": "string",
  "userName": "string",
  "password": "string",
  "vncPort": 0,
  "parameter": "string",
  "resource": "string"
}
```

Example response

```
200 OK
{
  "hostId": "string",
  "name": "string",
  "address": "string",
  "architecture": "string",
  "status": "NORMAL",
  "lcmIp": "string",
  "mecHost": "string",
  "protocol": "string",
  "port": 0,
  "os": "string",
  "portRangeMin": 0,
  "portRangeMax": 0,
  "userId": "string",
  "configId": "string",
  "userName": "string",
  "password": "string",
  "vncPort": 0,
  "parameter": "string",
  "resource": "string"
}
```

4.6 GET host list

get host list

```
Resource URI: /mec/developer/v1/hosts/list
```

Name	Definition	Type	Required
os	os	query param	yes
architecture	architecture	query param	yes
limit	limit	query param	yes
offset	offset	query param	yes

Example response

```
200 OK
[
  {
    "hostId": "string",
    "name": "string",
    "address": "string",
    "architecture": "string",
    "status": "NORMAL",
    "lcmIp": "string",
    "mecHost": "string",
    "protocol": "string",
    "port": 0,
    "os": "string",
    "portRangeMin": 0,
    "portRangeMax": 0,
    "userId": "string",
    "configId": "string",
    "userName": "string",
    "password": "string",
    "vncPort": 0,
    "parameter": "string",
    "resource": "string"
  }
]
```

4.7 GET host logs

get host logs.

```
Resource URI: /mec/developer/v1/hosts/{hostId}/logs
```

Name	Definition	Type	Required
hostId	hostid	pathparam	yes

Example response

```
200 OK
[
  {
    "hostId": "string",
    "name": "string",
    "address": "string",
    "architecture": "string",
    "status": "NORMAL",
    "lcmIp": "string",
    "mecHost": "string",
    "protocol": "string",
    "port": 0,
    "os": "string",
    "portRangeMin": 0,
    "portRangeMax": 0,
    "userId": "string",
    "configId": "string",
    "userName": "string",
    "password": "string",
    "vncPort": 0,
    "parameter": "string",
    "resource": "string"
  }
]
```

8.3.5 5. Capability-groups

capability group of edge applications

5.1 POST create group

Create one Capability group.

```
Resource URI: /mec/developer/v2/capability-groups
```

Name	Definition	Type	Required
CapabilityGroup	entityclass	body param	yes

```
{
  "id": "string",
  "name": "string",
  "description": "string",
  "nameEn": "string",
  "descriptionEn": "string",
  "type": "string",
  "iconFileId": "string",
  "author": "string",
  "createTime": 0,
  "updateTime": 0,
  "parent": {},
  "parentId": "string"
}
```

Example response

```
200 OK
{
  "id": "string",
  "name": "string",
  "description": "string",
  "nameEn": "string",
  "descriptionEn": "string",
  "type": "string",
  "iconFileId": "string",
  "author": "string",
  "createTime": 0,
  "updateTime": 0,
  "parent": {},
```

(下页继续)

(续上页)

```
  "parentId": "string"
}
```

5.2 GET all group

get all group.

```
Resource URI: /mec/developer/v2/capability-groups/
```

Example response

```
200 OK
[
  {
    "id": "string",
    "name": "string",
    "description": "string",
    "nameEn": "string",
    "descriptionEn": "string",
    "type": "string",
    "iconFileId": "string",
    "author": "string",
    "createTime": 0,
    "updateTime": 0,
    "parent": {},
    "parentId": "string"
  }
]
```

5.3 GET one group

get one group

```
Resource URI: /mec/developer/v2/capability-groups/{id}
```

Name	Definition	Type	Required
id	group id	path param	yes

Example response

```
200 OK
{
  "id": "string",
  "name": "string",
  "description": "string",
  "nameEn": "string",
  "descriptionEn": "string",
  "type": "string",
  "iconFileId": "string",
  "author": "string",
  "createTime": 0,
  "updateTime": 0,
  "parent": {},
  "parentId": "string"
}
```

5.4 DELETE one group

Delete one group by id

```
Resource URI: /mec/developer/v2/capability-groups/{id}
```

Name	Definition	Type	Required
id	group id	path param	yes

Example response

```
200 OK
string output.
```

8.3.6 6. File

This part is to configure the app, perform app deployment, and test the api for file operations

6.1 GET one file

Get one file

```
Resource URI: /mec/developer/v1/files/{fileId}
```

Name	Definition	Type	Required
fileId	fileid	path param	yes
userId	userId	request param	yes
type	file type	request param	yes

Example response

```
200 OK
binary output
```

6.2 POST upload one file

Upload file

```
Resource URI: /mec/developer/v1/files
```

Name	Definition	Type	Required
file	MultipartFileclass	request part	yes
userId	userId	request param	yes

Example response

```
200 OK
{
  "fileId": "string",
  "fileName": "string",
  "url": "string",
  "userId": "string",
  "uploadDate": "2020-09-14T09:03:17.084Z",
  "filePath": "string",
```

(下页继续)

(续上页)

```
  "temp": false
}
```

6.3 POST upload helm yaml

Upload helm template yaml.

```
Resource URI: /mec/developer/v1/files/helm-template-yaml
```

Name	Definition	Type	Required
file	MultipartFileclass	request part	yes
userId	userId	reques tparam	yes
projectId	projectid	request param	yes
configType	configType	request param	yes

Example response

```
200 OK
{
  "fileId": "string",
  "fileName": "string",
  "fileContent": "string",
  "formatSuccess": false,
  "imageSuccess": false,
  "serviceSuccess": false,
  "mepAgentSuccess": false
}
```

6.4 DELETE helm yaml

Delete helm template yaml.

```
Resource URI: /mec/developer/v1/files/helm-template-yaml
```

Name	Definition	Type	Required
fileId	fileid	request param	yes

Example response

```
200 OK
"string"
```

6.5 POST get sample code

Get sample code.

```
Resource URI: /mec/developer/v1/files/samplecode
```

Name	Definition	Type	Required
apiFileIds	apifileid list	body param	yes

```
List<String>
[
  "string"
]
```

Example response

```
200 OK
byte array output
```

6.6 GET api info

Get file return one object

```
Resource URI: /mec/developer/v1/files/api-info/{fileId}
```

Name	Definition	Type	Required
fileId	file id	path param	yes
userId	the author id of upload file	request param	yes

Example response

```
200 OK
{
    fileId:"xxx",
    fileName:"xxx",
    url:"xxx",
    userId:"xxx",
```

(下页继续)

(续上页)

```
    isTemp:false,
    uploadDate:xxxx-xx-xx,
    filePath:"xxx"
}
```

6.7 GET sdk code

Get sdk code

```
Resource URI: /mec/developer/v1/files/sdk/{fileId}/download/{lan}
```

Name	Definition	Type	Required
fileId	file id	path param	yes
lan	sdk codeLanguage	path param	yes

Example response

```
200 OK
binary output
```

6.8 GET sample content

query file content

```
Resource URI: /mec/developer/v1/files/samplecode/get-file-content
```

Name	Definition	Type	Required
fileName	file name	request param	yes

Example response

```
200 OK
String output
```

6.9 POST pkg structure

post pkg structure

```
Resource URI: /mec/developer/v1/files/samplecode/get-pkg-structure
```

Name	Definition	Type	Required
apiFileIds	file id arrays	body param	yes

Example response

```
200 OK
{
  "name": "string",
  "id": "string",
  "children": [
    {}
  ],
  "parent": false
}
```

8.3.7 7. Capabilities

capability of edge applications

7.1 POST create capability

create capability

```
Resource URI: /mec/developer/v2/capabilities/
```

Name	Definition	Type	Required
Capability	Capability	body param	yes

```
Capability:
{
  "id": "string",
  "name": "string",
  "nameEn": "string",
  "version": "string",
  "description": "string",
```

(下页继续)

(续上页)

```
  "descriptionEn": "string",
  "provider": "string",
  "apiFileId": "string",
  "guideFileId": "string",
  "guideFileIdEn": "string",
  "uploadTime": 0,
  "port": 0,
  "host": "string",
  "protocol": "string",
  "appId": "string",
  "packageId": "string",
  "userId": "string",
  "selectCount": 0,
  "iconFileId": "string",
  "author": "string",
  "experienceUrl": "string",
  "group": {
    "id": "string",
    "name": "string",
    "description": "string",
    "nameEn": "string",
    "descriptionEn": "string",
    "type": "string",
    "iconFileId": "string",
    "author": "string",
    "createTime": 0,
    "updateTime": 0,
    "parent": {},
    "parentId": "string"
  },
  "groupId": "string"
}
```

Example response

```
200 OK
{
  "id": "string",
  "name": "string",
  "nameEn": "string",
  "version": "string",
  "description": "string",
  "descriptionEn": "string",
  "provider": "string",
```

(下页继续)

(续上页)


```
  "apiFileId": "string",
  "guideFileId": "string",
  "guideFileIdEn": "string",
  "uploadTime": 0,
  "port": 0,
  "host": "string",
  "protocol": "string",
  "appId": "string",
  "packageId": "string",
  "userId": "string",
  "selectCount": 0,
  "iconFileId": "string",
  "author": "string",
  "experienceUrl": "string",
  "group": {
    "id": "string",
    "name": "string",
    "description": "string",
    "nameEn": "string",
    "descriptionEn": "string",
    "type": "string",
    "iconFileId": "string",
    "author": "string",
    "createTime": 0,
    "updateTime": 0,
    "parent": {},
    "parentId": "string"
  },
  "groupId": "string"
}
```


7.2 GET all capability

GET all capabilities

```
Resource URI: /mec/developer/v2/capabilities/
```

Name	Definition	Type	Required
filterType	filterType	query param	yes

Example response

```
200 OK
[
  {
    "id": "string",
    "name": "string",
    "nameEn": "string",
    "version": "string",
    "description": "string",
    "descriptionEn": "string",
    "provider": "string",
    "apiFileId": "string",
    "guideFileId": "string",
    "guideFileIdEn": "string",
    "uploadTime": 0,
    "port": 0,
    "host": "string",
    "protocol": "string",
    "appId": "string",
    "packageId": "string",
    "userId": "string",
    "selectCount": 0,
    "iconFileId": "string",
    "author": "string",
    "experienceUrl": "string",
    "group": {
      "id": "string",
      "name": "string",
      "description": "string",
      "nameEn": "string",
      "descriptionEn": "string",
      "type": "string",
      "iconFileId": "string",
      "author": "string",
      "createTime": 0,
      "updateTime": 0,
      "parent": {},
      "parentId": "string"
    },
    "groupId": "string"
  }
]
```

7.3 GET one capability

query one capability

```
Resource URI: /mec/developer/v2/capabilities/{id}
```

Name	Definition	Type	Required
id	id	path param	yes

Example response

```
200 OK
[
  {
    "id": "string",
    "name": "string",
    "nameEn": "string",
    "version": "string",
    "description": "string",
    "descriptionEn": "string",
    "provider": "string",
    "apiFileId": "string",
    "guideFileId": "string",
    "guideFileIdEn": "string",
    "uploadTime": 0,
    "port": 0,
    "host": "string",
    "protocol": "string",
    "appId": "string",
    "packageId": "string",
    "userId": "string",
    "selectCount": 0,
    "iconFileId": "string",
    "author": "string",
    "experienceUrl": "string",
    "group": {
      "id": "string",
      "name": "string",
      "description": "string",
      "nameEn": "string",
      "descriptionEn": "string",
      "type": "string",
      "iconFileId": "string",
      "author": "string",
```

(下页继续)

(续上页)

```
      "createTime": 0,
      "updateTime": 0,
      "parent": {},
      "parentId": "string"
    },
    "groupId": "string"
  }
]
```

7.4 DELETE one capability

Delete one capability

```
Resource URI: /mec/developer/v2/capabilities/{id}
```

Name	Definition	Type	Required
id	id	path param	yes

Example response

```
200 OK
{
    "id": "string",
    "name": "string",
    "nameEn": "string",
    "version": "string",
    "description": "string",
    "descriptionEn": "string",
    "provider": "string",
    "apiFileId": "string",
    "guideFileId": "string",
    "guideFileIdEn": "string",
    "uploadTime": 0,
    "port": 0,
    "host": "string",
    "protocol": "string",
    "appId": "string",
    "packageId": "string",
    "userId": "string",
    "selectCount": 0,
    "iconFileId": "string",
    "author": "string",
```

(下页继续)

(续上页)

```
    "experienceUrl": "string",
    "group": {
      "id": "string",
      "name": "string",
      "description": "string",
      "nameEn": "string",
      "descriptionEn": "string",
      "type": "string",
      "iconFileId": "string",
      "author": "string",
      "createTime": 0,
      "updateTime": 0,
      "parent": {},
      "parentId": "string"
    },
    "groupId": "string"
  }
```

7.5 PUT one capability

update one capability

```
Resource URI: /mec/developer/v2/capabilities/{id}
```

Name	Definition	Type	Required
id	id	path param	yes
capability	capability	body param	yes

```
capability:
{
    "id": "string",
    "name": "string",
    "nameEn": "string",
    "version": "string",
    "description": "string",
    "descriptionEn": "string",
    "provider": "string",
    "apiFileId": "string",
    "guideFileId": "string",
    "guideFileIdEn": "string",
    "uploadTime": 0,
```

(下页继续)

(续上页)

```
    "port": 0,
    "host": "string",
    "protocol": "string",
    "appId": "string",
    "packageId": "string",
    "userId": "string",
    "selectCount": 0,
    "iconFileId": "string",
    "author": "string",
    "experienceUrl": "string",
    "group": {
      "id": "string",
      "name": "string",
      "description": "string",
      "nameEn": "string",
      "descriptionEn": "string",
      "type": "string",
      "iconFileId": "string",
      "author": "string",
      "createTime": 0,
      "updateTime": 0,
      "parent": {},
      "parentId": "string"
    },
    "groupId": "string"
  }
```

Example response

```
200 OK
{
    "id": "string",
    "name": "string",
    "nameEn": "string",
    "version": "string",
    "description": "string",
    "descriptionEn": "string",
    "provider": "string",
    "apiFileId": "string",
    "guideFileId": "string",
    "guideFileIdEn": "string",
    "uploadTime": 0,
    "port": 0,
    "host": "string",
```

(下页继续)

(续上页)


```
    "protocol": "string",
    "appId": "string",
    "packageId": "string",
    "userId": "string",
    "selectCount": 0,
    "iconFileId": "string",
    "author": "string",
    "experienceUrl": "string",
    "group": {
      "id": "string",
      "name": "string",
      "description": "string",
      "nameEn": "string",
      "descriptionEn": "string",
      "type": "string",
      "iconFileId": "string",
      "author": "string",
      "createTime": 0,
      "updateTime": 0,
      "parent": {},
      "parentId": "string"
    },
    "groupId": "string"
  }
```


8.3.8 8. Capability-Groups Query

Capability-Groups query

8.1 GET group by type

query group by type

```
Resource URI: /mec/developer/v2/query/capability-groups/type/{type}
```

Name	Definition	Type	Required
type	group type	path param	yes

Example response

```
200 OK
[
  {
    "id": "string",
    "name": "string",
    "description": "string",
    "nameEn": "string",
    "descriptionEn": "string",
    "type": "string",
    "iconFileId": "string",
    "author": "string",
    "createTime": 0,
    "updateTime": 0,
    "parent": {},
    "parentId": "string"
  }
]
```

8.3.9 9. AppRelease

Get pkg structure and file content

9.1 GET pkg structure

Get pkg structure

```
Resource URI: /mec/developer/v1/apprelease/{projectId}/{csarId}/action/get-pkg-
↪structure
```

Name	Definition	Type	Required
projectId	project id	pathparam	yes
csarId	csar id	pathparam	yes

Example response

```
200 OK
{
  "name": "string",
  "id": "string",
  "children": [
    {}
```

(下页继续)

(续上页)

```
  ],
  "parent": false
}
```

9.2 GET file content

Get file content

```
Resource URI: /mec/developer/v1/apprelease/{projectId}/action/get-pkg-content
```

Name	Definition	Type	Required
projectId	project id	pathparam	yes
fileName	pkg file name	requestparam	yes

Example response

```
200 OK
string output
```

8.3.10 10. ReleaseConfig

Do some configuration before release app

10.1 GET release config

Get release config

```
Resource URI: /mec/developer/v1/releaseconfig/{projectId}/action/release-config
```

Name	Definition	Type	Required
projectId	project id	pathparam	yes

Example response

```
200 OK
{
  "releaseId": "string",
  "projectId": "string",
  "guideFileId": "string",
```

(下页继续)

(续上页)

```
"appInstanceId": "string",
"capabilitiesDetail": {
  "appTrafficRule": [
    {
      "trafficRuleId": "string",
      "action": "string",
      "priority": "string",
      "filterType": "string",
      "trafficFilter": [
        {
          "protocal": [
            "string"
          ],
          "srcAddress": [
            "string"
          ],
          "srcPort": [
            "string"
          ],
          "dstAddress": [
            "string"
          ],
          "dstPort": [
            "string"
          ],
          "tag": [
            "string"
          ],
          "srcTunnelAddress": [
            "string"
          ],
          "tgtTunnelAddress": [
            "string"
          ],
          "srcTunnelPort": [
            "string"
          ],
          "dstTunnelPort": [
            "string"
          ],
          "qci": 0,
          "dscp": 0,
          "tc": 0
```

(下页继续)

(续上页)

```
        }
      ],
      "dstInterface": [
        {
          "interfaceType": "string",
          "srcMacAddress": "string",
          "dstMacAddress": "string",
          "dstIpAddress": "string",
          "tunnelInfo": {
            "tunnelType": "string",
            "tunnelDstAddress": "string",
            "tunnelSrcAddress": "string",
            "tunnelSpecificData": "string"
          }
        }
      ]
    }
  ],
  "appDNSRule": [
    {
      "dnsRuleId": "string",
      "domainName": "string",
      "ipAddressType": "string",
      "ipAddress": "string",
      "ttl": "string"
    }
  ],
  "serviceDetails": [
    {
      "groupId": "string",
      "serviceName": "string",
      "internalPort": 0,
      "version": "string",
      "protocol": "string",
      "apiJson": "string",
      "apiMd": "string",
      "dnsRulesList": [
        "string"
      ],
      "trafficRulesList": [
        "string"
      ]
    }
```

(下页继续)

(续上页)

```
    ]
  },
  "atpTest": {
    "id": "string",
    "appName": "string",
    "status": "string",
    "createTime": "string"
  },
  "testStatus": "string",
  "createTime": "2020-12-29T03:31:13.204Z"
}
```

10.2 POST release config

create release config

```
Resource URI: /mec/developer/v1/releaseconfig/{projectId}/action/release-config
```

Name	Definition	Type	Required
projectId	project id	pathparam	yes
ReleaseConfig	entityclass	bodyparam	yes

```
ReleaseConfig:
{
  "releaseId": "string",
  "projectId": "string",
  "guideFileId": "string",
  "appInstanceId": "string",
  "capabilitiesDetail": {
    "appTrafficRule": [
      {
        "trafficRuleId": "string",
        "action": "string",
        "priority": "string",
        "filterType": "string",
        "trafficFilter": [
          {
            "protocal": [
              "string"
            ],
            "srcAddress": [
```

(下页继续)

(续上页)

```
            "string"
          ],
          "srcPort": [
            "string"
          ],
          "dstAddress": [
            "string"
          ],
          "dstPort": [
            "string"
          ],
          "tag": [
            "string"
          ],
          "srcTunnelAddress": [
            "string"
          ],
          "tgtTunnelAddress": [
            "string"
          ],
          "srcTunnelPort": [
            "string"
          ],
          "dstTunnelPort": [
            "string"
          ],
          "qci": 0,
          "dscp": 0,
          "tc": 0
        }
      ],
      "dstInterface": [
        {
          "interfaceType": "string",
          "srcMacAddress": "string",
          "dstMacAddress": "string",
          "dstIpAddress": "string",
          "tunnelInfo": {
            "tunnelType": "string",
            "tunnelDstAddress": "string",
            "tunnelSrcAddress": "string",
            "tunnelSpecificData": "string"
          }
```

(下页继续)

(续上页)


```
          }
        ]
      }
    ],
    "appDNSRule": [
      {
        "dnsRuleId": "string",
        "domainName": "string",
        "ipAddressType": "string",
        "ipAddress": "string",
        "ttl": "string"
      }
    ],
    "serviceDetails": [
      {
        "groupId": "string",
        "serviceName": "string",
        "internalPort": 0,
        "version": "string",
        "protocol": "string",
        "apiJson": "string",
        "apiMd": "string",
        "dnsRulesList": [
          "string"
        ],
        "trafficRulesList": [
          "string"
        ]
      }
    ]
  },
  "atpTest": {
    "id": "string",
    "appName": "string",
    "status": "string",
    "createTime": "string"
  },
  "testStatus": "string",
  "createTime": "2020-12-29T03:31:13.204Z"
}
```


Example response

```
200 OK
{
  "releaseId": "string",
  "projectId": "string",
  "guideFileId": "string",
  "appInstanceId": "string",
  "capabilitiesDetail": {
    "appTrafficRule": [
      {
        "trafficRuleId": "string",
        "action": "string",
        "priority": "string",
        "filterType": "string",
        "trafficFilter": [
          {
            "protocal": [
              "string"
            ],
            "srcAddress": [
              "string"
            ],
            "srcPort": [
              "string"
            ],
            "dstAddress": [
              "string"
            ],
            "dstPort": [
              "string"
            ],
            "tag": [
              "string"
            ],
            "srcTunnelAddress": [
              "string"
            ],
            "tgtTunnelAddress": [
              "string"
            ],
            "srcTunnelPort": [
              "string"
            ],
            "dstTunnelPort": [
              "string"
```

(下页继续)

(续上页)


```
            ],
            "qci": 0,
            "dscp": 0,
            "tc": 0
          }
        ],
        "dstInterface": [
          {
            "interfaceType": "string",
            "srcMacAddress": "string",
            "dstMacAddress": "string",
            "dstIpAddress": "string",
            "tunnelInfo": {
              "tunnelType": "string",
              "tunnelDstAddress": "string",
              "tunnelSrcAddress": "string",
              "tunnelSpecificData": "string"
            }
          }
        ]
      }
    ],
    "appDNSRule": [
      {
        "dnsRuleId": "string",
        "domainName": "string",
        "ipAddressType": "string",
        "ipAddress": "string",
        "ttl": "string"
      }
    ],
    "serviceDetails": [
      {
        "groupId": "string",
        "serviceName": "string",
        "internalPort": 0,
        "version": "string",
        "protocol": "string",
        "apiJson": "string",
        "apiMd": "string",
        "dnsRulesList": [
          "string"
        ],
```


(下页继续)

(续上页)

```
        "trafficRulesList": [
          "string"
        ]
      }
    ]
  },
  "atpTest": {
    "id": "string",
    "appName": "string",
    "status": "string",
    "createTime": "string"
  },
  "testStatus": "string",
  "createTime": "2020-12-29T03:31:13.204Z"
}
```

10.3 PUT release config

update release config

```
Resource URI: /mec/developer/v1/releaseconfig/{projectId}/action/release-config
```

Name	Definition	Type	Required
projectId	project id	pathparam	yes
ReleaseConfig	entityclass	bodyparam	yes

```
ReleaseConfig:
{
  "releaseId": "string",
  "projectId": "string",
  "guideFileId": "string",
  "appInstanceId": "string",
  "capabilitiesDetail": {
    "appTrafficRule": [
      {
        "trafficRuleId": "string",
        "action": "string",
        "priority": "string",
        "filterType": "string",
        "trafficFilter": [
          {
```

(下页继续)

(续上页)

```
            "protocal": [
              "string"
            ],
            "srcAddress": [
              "string"
            ],
            "srcPort": [
              "string"
            ],
            "dstAddress": [
              "string"
            ],
            "dstPort": [
              "string"
            ],
            "tag": [
              "string"
            ],
            "srcTunnelAddress": [
              "string"
            ],
            "tgtTunnelAddress": [
              "string"
            ],
            "srcTunnelPort": [
              "string"
            ],
            "dstTunnelPort": [
              "string"
            ],
            "qci": 0,
            "dscp": 0,
            "tc": 0
          }
        ],
        "dstInterface": [
          {
            "interfaceType": "string",
            "srcMacAddress": "string",
            "dstMacAddress": "string",
            "dstIpAddress": "string",
            "tunnelInfo": {
              "tunnelType": "string",
```

(下页继续)

(续上页)


```
              "tunnelDstAddress": "string",
              "tunnelSrcAddress": "string",
              "tunnelSpecificData": "string"
            }
          }
        ]
      }
    ],
    "appDNSRule": [
      {
        "dnsRuleId": "string",
        "domainName": "string",
        "ipAddressType": "string",
        "ipAddress": "string",
        "ttl": "string"
      }
    ],
    "serviceDetails": [
      {
        "groupId": "string",
        "serviceName": "string",
        "internalPort": 0,
        "version": "string",
        "protocol": "string",
        "apiJson": "string",
        "apiMd": "string",
        "dnsRulesList": [
          "string"
        ],
        "trafficRulesList": [
          "string"
        ]
      }
    ]
  },
  "atpTest": {
    "id": "string",
    "appName": "string",
    "status": "string",
    "createTime": "string"
  },
  "testStatus": "string",
  "createTime": "2020-12-29T03:31:13.204Z"
}
```

Example response

```
200 OK
{
  "releaseId": "string",
  "projectId": "string",
  "guideFileId": "string",
  "appInstanceId": "string",
  "capabilitiesDetail": {
    "appTrafficRule": [
      {
        "trafficRuleId": "string",
        "action": "string",
        "priority": "string",
        "filterType": "string",
        "trafficFilter": [
          {
            "protocal": [
              "string"
            ],
            "srcAddress": [
              "string"
            ],
            "srcPort": [
              "string"
            ],
            "dstAddress": [
              "string"
            ],
            "dstPort": [
              "string"
            ],
            "tag": [
              "string"
            ],
            "srcTunnelAddress": [
              "string"
            ],
            "tgtTunnelAddress": [
              "string"
            ],
            "srcTunnelPort": [
              "string"
            ],
            "dstTunnelPort": [
```

(下页继续)

(续上页)


```
            "string"
          ],
          "qci": 0,
          "dscp": 0,
          "tc": 0
        }
      ],
      "dstInterface": [
        {
          "interfaceType": "string",
          "srcMacAddress": "string",
          "dstMacAddress": "string",
          "dstIpAddress": "string",
          "tunnelInfo": {
            "tunnelType": "string",
            "tunnelDstAddress": "string",
            "tunnelSrcAddress": "string",
            "tunnelSpecificData": "string"
          }
        }
      ]
    }
  ],
  "appDNSRule": [
    {
      "dnsRuleId": "string",
      "domainName": "string",
      "ipAddressType": "string",
      "ipAddress": "string",
      "ttl": "string"
    }
  ],
  "serviceDetails": [
    {
      "groupId": "string",
      "serviceName": "string",
      "internalPort": 0,
      "version": "string",
      "protocol": "string",
      "apiJson": "string",
      "apiMd": "string",
      "dnsRulesList": [
        "string"
```


(下页继续)

(续上页)

```
        ],
        "trafficRulesList": [
          "string"
        ]
      }
    ]
  },
  "atpTest": {
    "id": "string",
    "appName": "string",
    "status": "string",
    "createTime": "string"
  },
  "testStatus": "string",
  "createTime": "2020-12-29T03:31:13.204Z"
}
```

8.3.11 11. Health

health Check

11.1 GET health check

Get health check

```
Resource URI: /health
```

Example response

```
200 OK
'ok'.
```

8.3.12 12. Deploy

API to manipulate deployment files

12.1 GET deploy yaml

Get deploy yaml

```
Resource URI: /mec/developer/v1/deploy/{fileId}
```

Name	Definition	Type	Required
fileId	fileId	path param	yes

Example response

```
200 OK
{
  "fileId": "string",
  "fileName": "string",
  "userId": "string",
  "projectId": "string",
  "content": "string",
  "uploadTimeStamp": 0,
  "configType": "string"
}
```

12.2 PUT deploy yaml

Modify deploy yaml by file id

```
Resource URI: /mec/developer/v1/deploy/{fileId}
```

Name	Definition	Type	Required
fileId	fileId	path param	yes
fileContent	file content	body param	yes

Example response

```
200 OK
{
  "fileId": "string",
  "fileName": "string",
  "userId": "string",
  "projectId": "string",
  "content": "string",
  "uploadTimeStamp": 0,
```

(下页继续)

(续上页)

```
  "configType": "string"
}
```

12.3 GET deploy json

Get file content with json format

```
Resource URI: /mec/developer/v1/deploy/{fileId}/action/get-json
```

Name	Definition	Type	Required
fileId	fileId	path param	yes

Example response

```
200 OK
[
  {}
]
```

12.4 POST deploy yaml

Upload deploy yaml

```
Resource URI: /mec/developer/v1/deploy/{projectId}/action/save-yaml
```

Name	Definition	Type	Required
jsonStr	deploy yaml content	body param	yes
userId	userId	request param	yes
projectId	projectId	path param	yes
configType	Fields that distinguish deployment types	request param	yes

Example response

```
200 OK
{
  "fileId": "string",
  "fileName": "string",
  "userId": "string",
  "projectId": "string",
```

(下页继续)

(续上页)

```
  "content": "string",
  "uploadTimeStamp": 0,
  "configType": "string"
}
```

8.3.13 13. Capability-group stats

query group stats

13.1 GET group stats

Get group stats

```
Resource URI: /mec/developer/v2/capability-group-stats/
```

Example response

```
200 OK
[
  {
    "id": "string",
    "name": "string",
    "description": "string",
    "nameEn": "string",
    "descriptionEn": "string",
    "type": "string",
    "iconFileId": "string",
    "author": "string",
    "createTime": 0,
    "updateTime": 0,
    "parent": {
      "id": "string",
      "name": "string",
      "description": "string",
      "nameEn": "string",
      "descriptionEn": "string",
      "type": "string",
      "iconFileId": "string",
      "author": "string",
      "createTime": 0,
      "updateTime": 0,
      "parent": {},
```

(下页继续)

(续上页)

```
      "parentId": "string"
    },
    "capabilityCount": 0,
    "parentId": "string"
  }
]
```

8.3.14 14. VirtualMachine

This part is about the api for virtual machine deployment

14.1 POST one vm

Create one vm

```
Resource URI: /mec/developer/v1/projects/{projectId}/vm-create
```

Name	Definition	Type	Required
projectId	projectId	path param	yes
userId	userId	query param	yes

Example response

```
200 OK
{
  "vmId": "string",
  "projectId": "string",
  "vmName": "string",
  "status": "NOTCREATE",
  "stageStatus": {
    "hostInfo": "Success",
    "distributeInfo": "Success",
    "instantiateInfo": "Success",
    "workStatus": "Success"
  },
  "host": {
    "hostId": "string",
    "name": "string",
    "address": "string",
    "architecture": "string",
    "status": "NORMAL",
```

(下页继续)

(续上页)


```
    "lcmIp": "string",
    "mecHost": "string",
    "protocol": "string",
    "port": 0,
    "os": "string",
    "portRangeMin": 0,
    "portRangeMax": 0,
    "userId": "string",
    "configId": "string",
    "userName": "string",
    "password": "string",
    "vncPort": 0,
    "parameter": "string",
    "resource": "string"
  },
  "lcmToken": "string",
  "vmInfo": [
    {
      "vmId": "string",
      "vncUrl": "string",
      "networks": [
        {
          "ip": "string",
          "name": "string"
        }
      ]
    }
  ],
  "appInstanceId": "string",
  "packageId": "string",
  "createTime": "2021-10-13T01:47:57.447Z",
  "log": "string",
  "nextStage": "string"
}
```


14.2 DELETE vm config

delete vm config

```
Resource URI: /mec/developer/v1/projects/{projectId}/vm/{vmId}
```

Name	Definition	Type	Required
projectId	projectId	path param	yes
vmId	vmId	path param	yes
userId	user id	query param	yes

Example response

```
200 OK
boolean output
```

14.3 GET all image config

get all image config

```
Resource URI: /mec/developer/v1/projects/{projectId}/vm/image
```

Name	Definition	Type	Required
projectId	projectId	path param	yes
userId	user id	query param	yes

Example response

```
200 OK
{
  "vmId": "string",
  "imageId": "string",
  "projectId": "string",
  "vmName": "string",
  "imageName": "string",
  "appInstanceId": "string",
  "hostIp": "string",
  "sumChunkNum": 0,
  "chunkSize": 0,
  "checksum": "string",
  "stageStatus": {
    "createImageInfo": "Success",
    "imageStatus": "Success",
    "downloadImageInfo": "Success"
  },
```

(下页继续)

(续上页)

```
  "status": "NOTCREATE",
  "lcmToken": "string",
  "createTime": "2021-10-13T01:47:57.448Z",
  "log": "string",
  "nextStage": "string"
}
```

14.4 POST image config

Create image config

```
Resource URI: /mec/developer/v1/projects/{projectId}/vm/image
```

Name	Definition	Type	Required
projectId	projectId	path param	yes
userId	user id	query param	yes

Example response

```
200 OK
boolean output
```

14.5 DELETE image config

Delete image config

```
Resource URI: /mec/developer/v1/projects/{projectId}/vm/image
```

Name	Definition	Type	Required
projectId	projectId	path param	yes
userId	user id	query param	yes

Example response

```
200 OK
boolean output
```

14.6 GET image config By ProjectId

query image config By ProjectId

```
Resource URI: /mec/developer/v1/projects/{projectId}/vm
```

Name	Definition	Type	Required
projectId	projectId	path param	yes
userId	user id	query param	yes

Example response

```
200 OK
[
  {
    "vmId": "string",
    "projectId": "string",
    "vmName": "string",
    "status": "NOTCREATE",
    "stageStatus": {
      "hostInfo": "Success",
      "distributeInfo": "Success",
      "instantiateInfo": "Success",
      "workStatus": "Success"
    },
    "host": {
      "hostId": "string",
      "name": "string",
      "address": "string",
      "architecture": "string",
      "status": "NORMAL",
      "lcmIp": "string",
      "mecHost": "string",
      "protocol": "string",
      "port": 0,
      "os": "string",
      "portRangeMin": 0,
      "portRangeMax": 0,
      "userId": "string",
      "configId": "string",
      "userName": "string",
      "password": "string",
      "vncPort": 0,
      "parameter": "string",
```

(下页继续)

(续上页)


```
      "resource": "string"
    },
    "lcmToken": "string",
    "vmInfo": [
      {
        "vmId": "string",
        "vncUrl": "string",
        "networks": [
          {
            "ip": "string",
            "name": "string"
          }
        ]
      }
    ],
    "appInstanceId": "string",
    "packageId": "string",
    "createTime": "2021-10-13T01:47:57.447Z",
    "log": "string",
    "nextStage": "string"
  }
]
```


14.7 GET csar pkg

get csar pkg

```
Resource URI: /mec/developer/v1/projects/{projectId}/vm/{vmId}/package
```

Name	Definition	Type	Required
projectId	projectId	path param	yes
userId	user id	query param	yes
vmId	vmId	path param	yes

Example response

```
200 OK
binary output
```

14.8 GET vm resources

get vm resources

```
Resource URI: /mec/developer/v1/vmconfig
```

Example response

```
200 OK
{
  "vmRegulationList": [
    {
      "regulationId": 0,
      "architecture": "string",
      "nameZh": "string",
      "nameEn": "string",
      "sceneZh": "string",
      "sceneEn": "string",
      "memory": 0,
      "cpu": 0,
      "systemDisk": 0,
      "dataDisk": 0,
      "gpu": "string",
      "otherAbility": "string"
    }
  ],
  "vmSystemList": [
    {
      "systemId": 0,
      "type": "public",
      "operateSystem": "ubuntu",
      "version": "string",
      "systemBit": "string",
      "systemDisk": 0,
      "systemName": "string",
      "fileName": "string",
      "createTime": "string",
      "modifyTime": "string",
      "systemFormat": "string",
      "systemSize": 0,
      "systemSlim": "SLIM_WAIT",
      "uploadTime": "string",
      "systemPath": "string",
      "status": "UPLOAD_WAIT",
      "userId": "string",
```

(下页继续)

(续上页)

```
      "userName": "string",
      "fileMd5": "string",
      "fileIdentifier": "string",
      "errorType": "string"
    }
  ],
  "vmNetworkList": [
    {
      "networkType": "string",
      "descriptionZh": "string",
      "descriptionEn": "string",
      "networkName": "string"
    }
  ],
  "vmUserDataList": [
    {
      "operateSystem": "string",
      "flavorExtraSpecs": "string",
      "contents": "string",
      "params": "string",
      "temp": false
    }
  ]
}
```

14.9 POST vm file

uoload vm image

```
Resource URI: /mec/developer/v1/projects/{projectId}/vm/{vmId}/files
```

Name	Definition	Type	Required
projectId	projectId	path param	yes
userId	user id	query param	yes
vmId	vmId	path param	yes
file	File	query param	yes
id	id	query param	yes
chunkNumber	chunkNumber	path param	yes
chunkSize	chunkSize	query param	yes
currentChunkSize	currentChunkSize	query param	yes
totalSize	totalSize	query param	yes
identifier	identifier	query param	yes
filename	filename	query param	yes
relativePath	relativePath	query param	yes
totalChunks	totalChunks	query param	yes
type	type	query param	yes

Example response

```
200 OK
boolean output
```

14.10 GET merge image

merge vm image

```
Resource URI: /mec/developer/v1/projects/{projectId}/vm/{vmId}/merge
```

Name	Definition	Type	Required
projectId	projectId	path param	yes
userId	user id	query param	yes
vmId	vmId	path param	yes
fileName	fileName	query param	yes
identifier	identifier	query param	yes

Example response

```
200 OK
{
  "headers": {
    "string": [
```

(下页继续)

(续上页)

```
      "string"
    ]
  },
  "body": {},
  "statusCodeValue": 0,
  "statusCode": "CONTINUE"
}
```

14.11 POST clean vm

clean vm

```
Resource URI: /mec/developer/v1/projects/{projectId}/vm/clean
```

Name	Definition	Type	Required
projectId	projectId	path param	yes

Example response

```
200 OK
true.
```

14.12 GET vm pkg

query vm package info

```
Resource URI: /mec/developer/v1/projects/{projectId}/vm-package
```

Name	Definition	Type	Required
projectId	projectId	path param	yes
userId	userId	query param	yes

Example response

```
200 OK
{
  "id": "string",
  "projectId": "string",
  "vmRegulation": {
    "regulationId": 0,
```

(下页继续)

(续上页)


```
    "architecture": "string",
    "nameZh": "string",
    "nameEn": "string",
    "sceneZh": "string",
    "sceneEn": "string",
    "memory": 0,
    "cpu": 0,
    "systemDisk": 0,
    "dataDisk": 0,
    "gpu": "string",
    "otherAbility": "string"
  },
  "vmSystem": {
    "systemId": 0,
    "type": "public",
    "operateSystem": "ubuntu",
    "version": "string",
    "systemBit": "string",
    "systemDisk": 0,
    "systemName": "string",
    "fileName": "string",
    "createTime": "string",
    "modifyTime": "string",
    "systemFormat": "string",
    "systemSize": 0,
    "systemSlim": "SLIM_WAIT",
    "uploadTime": "string",
    "systemPath": "string",
    "status": "UPLOAD_WAIT",
    "userId": "string",
    "userName": "string",
    "fileMd5": "string",
    "fileIdentifier": "string",
    "errorType": "string"
  },
  "vmNetwork": [
    "string"
  ],
  "vmUserData": {
    "operateSystem": "string",
    "flavorExtraSpecs": "string",
    "contents": "string",
    "params": "string",
```


(下页继续)

(续上页)

```
    "temp": false
  },
  "vmName": "string",
  "ak": "string",
  "sk": "string",
  "appInstanceId": "string",
  "createTime": "2021-10-13T01:47:57.448Z"
}
```

14.13 POST vm pkg

create vm package config info

```
Resource URI: /mec/developer/v1/projects/{projectId}/vm-package
```

Name	Definition	Type	Required
projectId	projectId	path param	yes
userId	userId	query param	yes
vmPackageConfig	vmPackageConfig	body param	yes

```
{
  "id": "string",
  "projectId": "string",
  "vmRegulation": {
    "regulationId": 0,
    "architecture": "string",
    "nameZh": "string",
    "nameEn": "string",
    "sceneZh": "string",
    "sceneEn": "string",
    "memory": 0,
    "cpu": 0,
    "systemDisk": 0,
    "dataDisk": 0,
    "gpu": "string",
    "otherAbility": "string"
  },
  "vmSystem": {
    "systemId": 0,
    "type": "public",
    "operateSystem": "ubuntu",
```

(下页继续)

(续上页)

```
    "version": "string",
    "systemBit": "string",
    "systemDisk": 0,
    "systemName": "string",
    "fileName": "string",
    "createTime": "string",
    "modifyTime": "string",
    "systemFormat": "string",
    "systemSize": 0,
    "systemSlim": "SLIM_WAIT",
    "uploadTime": "string",
    "systemPath": "string",
    "status": "UPLOAD_WAIT",
    "userId": "string",
    "userName": "string",
    "fileMd5": "string",
    "fileIdentifier": "string",
    "errorType": "string"
  },
  "vmNetwork": [
    "string"
  ],
  "vmUserData": {
    "operateSystem": "string",
    "flavorExtraSpecs": "string",
    "contents": "string",
    "params": "string",
    "temp": false
  },
  "vmName": "string",
  "ak": "string",
  "sk": "string",
  "appInstanceId": "string",
  "createTime": "2021-10-13T01:47:57.448Z"
}
```

Example response

```
200 OK
{
  "id": "string",
  "projectId": "string",
  "vmRegulation": {
    "regulationId": 0,
```

(下页继续)

(续上页)


```
    "architecture": "string",
    "nameZh": "string",
    "nameEn": "string",
    "sceneZh": "string",
    "sceneEn": "string",
    "memory": 0,
    "cpu": 0,
    "systemDisk": 0,
    "dataDisk": 0,
    "gpu": "string",
    "otherAbility": "string"
  },
  "vmSystem": {
    "systemId": 0,
    "type": "public",
    "operateSystem": "ubuntu",
    "version": "string",
    "systemBit": "string",
    "systemDisk": 0,
    "systemName": "string",
    "fileName": "string",
    "createTime": "string",
    "modifyTime": "string",
    "systemFormat": "string",
    "systemSize": 0,
    "systemSlim": "SLIM_WAIT",
    "uploadTime": "string",
    "systemPath": "string",
    "status": "UPLOAD_WAIT",
    "userId": "string",
    "userName": "string",
    "fileMd5": "string",
    "fileIdentifier": "string",
    "errorType": "string"
  },
  "vmNetwork": [
    "string"
  ],
  "vmUserData": {
    "operateSystem": "string",
    "flavorExtraSpecs": "string",
    "contents": "string",
    "params": "string",
```


(下页继续)

(续上页)

```
    "temp": false
  },
  "vmName": "string",
  "ak": "string",
  "sk": "string",
  "appInstanceId": "string",
  "createTime": "2021-10-13T01:47:57.448Z"
}
```

14.14 DELETE vm pkg

delete vm package config info

```
Resource URI: /mec/developer/v1/projects/{projectId}/vm-package
```

Name	Definition	Type	Required
projectId	projectId	path param	yes
userId	userId	query param	yes

Example response

```
200 OK
true.
```

8.3.15 15. Capability-group stats query

This part is about the api for Capability-group stats

15.1 GET stats by type

query group stats by type

```
Resource URI: /mec/developer/v2/query/capability-group-stats/type/{type}
```

Name	Definition	Type	Required
type	type	path param	yes

Example response

```
200 OK
 [
  {
    "id": "string",
    "name": "string",
    "description": "string",
    "nameEn": "string",
    "descriptionEn": "string",
    "type": "string",
    "iconFileId": "string",
    "author": "string",
    "createTime": 0,
    "updateTime": 0,
    "parent": {
      "id": "string",
      "name": "string",
      "description": "string",
      "nameEn": "string",
      "descriptionEn": "string",
      "type": "string",
      "iconFileId": "string",
      "author": "string",
      "createTime": 0,
      "updateTime": 0,
      "parent": {},
      "parentId": "string"
    },
    "capabilityCount": 0,
    "parentId": "string"
  }
]
```

8.3.16 16. SystemImage

This part is about the api for system image deployment

16.1 POST one system image

Create one system image

```
Resource URI: /mec/developer/v1/system/images
```

Name	Definition	Type	Required
VmImage	VmImage	body param	yes

```
VmImage:
{
  "systemId": 0,
  "type": "public",
  "operateSystem": "ubuntu",
  "version": "string",
  "systemBit": "string",
  "systemDisk": 0,
  "systemName": "string",
  "fileName": "string",
  "createTime": "string",
  "modifyTime": "string",
  "systemFormat": "string",
  "systemSize": 0,
  "systemSlim": "SLIM_WAIT",
  "uploadTime": "string",
  "systemPath": "string",
  "status": "UPLOAD_WAIT",
  "userId": "string",
  "userName": "string",
  "fileMd5": "string",
  "fileIdentifier": "string",
  "errorType": "string"
}
```

Example response

```
200 OK
boolean output
```

16.2 DELETE system image

delete system image

```
Resource URI: /mec/developer/v1/system/images/{systemId}
```

Name	Definition	Type	Required
systemId	systemId	path param	yes

Example response

```
200 OK
boolean output
```

16.3 POST upload system image

uploading system image

```
Resource URI: /mec/developer/v1/system/images/list
```

Name	Definition	Type	Required
userId	userId	query param	no
quemepGetSystemImageReqryCtrl	mepGetSystemImageReq	body param	yes

```
mepGetSystemImageReq:
{
  "systemName": "string",
  "type": "public",
  "userId": "string",
  "operateSystem": "string",
  "status": "UPLOAD_WAIT",
  "uploadTimeBegin": "string",
  "uploadTimeEnd": "string",
  "queryCtrl": {
    "offset": 0,
    "limit": 0,
    "sortBy": "userName",
    "sortOrder": "ASC"
  }
}
```

Example response

```
200 OK
{
  "totalCount": 0,
  "imageList": [
    {
      "systemId": 0,
      "type": "public",
      "operateSystem": "ubuntu",
      "version": "string",
      "systemBit": "string",
      "systemDisk": 0,
      "systemName": "string",
      "fileName": "string",
      "createTime": "string",
      "modifyTime": "string",
      "systemFormat": "string",
      "systemSize": 0,
      "systemSlim": "SLIM_WAIT",
      "uploadTime": "string",
      "systemPath": "string",
      "status": "UPLOAD_WAIT",
      "userId": "string",
      "userName": "string",
      "fileMd5": "string",
      "fileIdentifier": "string",
      "errorType": "string"
    }
  ]
}
```

16.4 PUT system image

update system image

```
Resource URI: /mec/developer/v1/system/images/{systemId}
```

Name	Definition	Type	Required
systemId	systemId	path param	yes
vmImage	vmImage	body param	yes

```
vmImage:
{
```

(下页继续)

(续上页)


```
  "systemId": 0,
  "type": "public",
  "operateSystem": "ubuntu",
  "version": "string",
  "systemBit": "string",
  "systemDisk": 0,
  "systemName": "string",
  "fileName": "string",
  "createTime": "string",
  "modifyTime": "string",
  "systemFormat": "string",
  "systemSize": 0,
  "systemSlim": "SLIM_WAIT",
  "uploadTime": "string",
  "systemPath": "string",
  "status": "UPLOAD_WAIT",
  "userId": "string",
  "userName": "string",
  "fileMd5": "string",
  "fileIdentifier": "string",
  "errorType": "string"
}
```


Example response

```
200 OK
boolean output
```

16.5 PUT publish system image

publish system image

```
Resource URI: /mec/developer/v1/system/images/{systemId}/publish
```

Name	Definition	Type	Required
systemId	systemId	path param	yes

Example response

```
200 OK
boolean output
```

16.6 POST system image to upload file

upload system image

```
Resource URI: /mec/developer/v1/system/images/{systemId}/upload
```

Name	Definition	Type	Required
systemId	systemId	path param	yes
identifier	identifier	query param	yes

Example response

```
200 OK
{
  "headers": {
    "string": [
      "string"
    ]
  },
  "body": {},
  "statusCodeValue": 0,
  "statusCode": "CONTINUE"
}
```

16.7 GET system image to merge

merge system image

```
Resource URI: /mec/developer/v1/system/images/{systemId}/merge
```

Name	Definition	Type	Required
systemId	systemId	path param	yes
fileName	fileName	request param	yes
identifier	identifier	request param	yes

Example response

```
200 OK
{
  "headers": {
    "string": [
      "string"
```

(下页继续)

(续上页)

```
    ]
  },
  "body": {},
  "statusCodeValue": 0,
  "statusCode": "CONTINUE"
}
```

16.8 GET system image to download

download system image

```
Resource URI: /mec/developer/v1/system/images/{systemId}/download
```

Name	Definition	Type	Required
systemId	systemId	path param	yes

Example response

```
200 OK
binary output.
```

16.9 POST slim image

slim system image

```
Resource URI: /mec/developer/v1/system/images/{systemId}/slim
```

Name	Definition	Type	Required
systemId	systemId	path param	yes

Example response

```
200 OK
true
```

16.10 PUT reset image status

reset system image

```
Resource URI: /mec/developer/v1/system/images/{systemId}/reset
```

Name	Definition	Type	Required
systemId	systemId	path param	yes

Example response

```
200 OK
true
```

16.11 POST upload image

upload system image

```
Resource URI: /mec/developer/v1/system/images/{systemId}/upload
```

Name	Definition	Type	Required
systemId	systemId	path param	yes
id	id	query param	yes
chunkNumber	chunkNumber	path param	yes
chunkSize	chunkSize	query param	yes
currentChunkSize	currentChunkSize	query param	yes
totalSize	totalSize	query param	yes
identifier	identifier	query param	yes
filename	filename	query param	yes
relativePath	relativePath	query param	yes
totalChunks	totalChunks	query param	yes
type	type	query param	yes

Example response

```
200 OK
{
  "headers": {
    "string": [
      "string"
    ]
```

(下页继续)

(续上页)

```
  },
  "body": {},
  "statusCodeValue": 0,
  "statusCode": "CONTINUE"
}
```

16.12 DELETE upload image

cancel upload system image

```
Resource URI: /mec/developer/v1/system/images/{systemId}/upload
```

Name	Definition	Type	Required
systemId	systemId	path param	yes
identifier	identifier	query param	yes

Example response

```
200 OK
{
  "headers": {
    "string": [
      "string"
    ]
  },
  "body": {},
  "statusCodeValue": 0,
  "statusCode": "CONTINUE"
}
```

8.3.17 17. Capability query

This part is about the api for query capability

17.1 GET capability by groupid

get capability by groupid

```
Resource URI: /mec/developer/v2/query/capabilities/group-id/{groupId}
```

Name	Definition	Type	Required
groupId	groupId	path param	yes

Example response

```
200 OK
[
  {
    "id": "string",
    "name": "string",
    "nameEn": "string",
    "version": "string",
    "description": "string",
    "descriptionEn": "string",
    "provider": "string",
    "apiFileId": "string",
    "guideFileId": "string",
    "guideFileIdEn": "string",
    "uploadTime": 0,
    "port": 0,
    "host": "string",
    "protocol": "string",
    "appId": "string",
    "packageId": "string",
    "userId": "string",
    "selectCount": 0,
    "iconFileId": "string",
    "author": "string",
    "experienceUrl": "string",
    "group": {
      "id": "string",
      "name": "string",
      "description": "string",
```

(下页继续)

(续上页)

```
      "nameEn": "string",
      "descriptionEn": "string",
      "type": "string",
      "iconFileId": "string",
      "author": "string",
      "createTime": 0,
      "updateTime": 0,
      "parent": {},
      "parentId": "string"
    },
    "groupId": "string"
  }
]
```

17.2 GET capability by en name

query capability by en name

```
Resource URI: /mec/developer/v2/query/capabilities/name-en
```

Name	Definition	Type	Required
nameEn	english name	query param	yes
limit	limit name	query param	yes
offset	offset name	query param	yes

Example response

```
200 OK
[
  {
    "id": "string",
    "name": "string",
    "nameEn": "string",
    "version": "string",
    "description": "string",
    "descriptionEn": "string",
    "provider": "string",
    "apiFileId": "string",
    "guideFileId": "string",
    "guideFileIdEn": "string",
    "uploadTime": 0,
```

(下页继续)

(续上页)


```
    "port": 0,
    "host": "string",
    "protocol": "string",
    "appId": "string",
    "packageId": "string",
    "userId": "string",
    "selectCount": 0,
    "iconFileId": "string",
    "author": "string",
    "experienceUrl": "string",
    "group": {
      "id": "string",
      "name": "string",
      "description": "string",
      "nameEn": "string",
      "descriptionEn": "string",
      "type": "string",
      "iconFileId": "string",
      "author": "string",
      "createTime": 0,
      "updateTime": 0,
      "parent": {},
      "parentId": "string"
    },
    "groupId": "string"
  }
]
```


17.3 GET capability by name

GET capability by zh name

```
Resource URI: /mec/developer/v2/query/capabilities/name
```

Name	Definition	Type	Required
name	chinese name	query param	yes
limit	limit name	query param	yes
offset	offset name	query param	yes

Example response

```
200 OK
[
  {
    "id": "string",
    "name": "string",
    "nameEn": "string",
    "version": "string",
    "description": "string",
    "descriptionEn": "string",
    "provider": "string",
    "apiFileId": "string",
    "guideFileId": "string",
    "guideFileIdEn": "string",
    "uploadTime": 0,
    "port": 0,
    "host": "string",
    "protocol": "string",
    "appId": "string",
    "packageId": "string",
    "userId": "string",
    "selectCount": 0,
    "iconFileId": "string",
    "author": "string",
    "experienceUrl": "string",
    "group": {
      "id": "string",
      "name": "string",
      "description": "string",
      "nameEn": "string",
      "descriptionEn": "string",
      "type": "string",
      "iconFileId": "string",
      "author": "string",
      "createTime": 0,
      "updateTime": 0,
      "parent": {},
      "parentId": "string"
    },
    "groupId": "string"
  }
]
```

17.4 GET capability by projectid

get capability by projectid

```
Resource URI: /mec/developer/v2/query/capabilities/project-id/{projectId}
```

Name	Definition	Type	Required
projectId	projectId	path param	yes

Example response

```
200 OK
[
  {
    "id": "string",
    "name": "string",
    "nameEn": "string",
    "version": "string",
    "description": "string",
    "descriptionEn": "string",
    "provider": "string",
    "apiFileId": "string",
    "guideFileId": "string",
    "guideFileIdEn": "string",
    "uploadTime": 0,
    "port": 0,
    "host": "string",
    "protocol": "string",
    "appId": "string",
    "packageId": "string",
    "userId": "string",
    "selectCount": 0,
    "iconFileId": "string",
    "author": "string",
    "experienceUrl": "string",
    "group": {
      "id": "string",
      "name": "string",
      "description": "string",
      "nameEn": "string",
      "descriptionEn": "string",
      "type": "string",
      "iconFileId": "string",
      "author": "string",
```

(下页继续)

(续上页)

```
      "createTime": 0,
      "updateTime": 0,
      "parent": {},
      "parentId": "string"
    },
    "groupId": "string"
  }
]
```

17.5 GET capability by type

query capability by type

```
Resource URI: /mec/developer/v2/query/capabilities/type/{type}
```

Name	Definition	Type	Required
type	type	path param	yes

Example response

```
200 OK
[
  {
    "id": "string",
    "name": "string",
    "nameEn": "string",
    "version": "string",
    "description": "string",
    "descriptionEn": "string",
    "provider": "string",
    "apiFileId": "string",
    "guideFileId": "string",
    "guideFileIdEn": "string",
    "uploadTime": 0,
    "port": 0,
    "host": "string",
    "protocol": "string",
    "appId": "string",
    "packageId": "string",
    "userId": "string",
    "selectCount": 0,
    "iconFileId": "string",
```

(下页继续)

(续上页)

```
    "author": "string",
    "experienceUrl": "string",
    "group": {
      "id": "string",
      "name": "string",
      "description": "string",
      "nameEn": "string",
      "descriptionEn": "string",
      "type": "string",
      "iconFileId": "string",
      "author": "string",
      "createTime": 0,
      "updateTime": 0,
      "parent": {},
      "parentId": "string"
    },
    "groupId": "string"
  }
]
```

8.4 Developer Dependency

8.4.1 Developer-be 三方件使用清单

GroupId	ArtifactId	Version	License
ch.ethz.ganymed	ganymed-ssh2	262	BSD
ch.qos.logback	logback-classic	1.2.3	EPL 1.0,LGPL 2.1
ch.qos.logback	logback-core	1.2.3	EPL 1.0,LGPL 2.1
com.esotericsoftware.yamlbeans	yamlbeans	1.13	BSD 2-clause
com.fasterxml.jackson.core	jackson-annotations	2.10.0	Apache 2.0
com.fasterxml.jackson.core	jackson-core	2.10.5	Apache 2.0
com.fasterxml.jackson.core	jackson-databind	2.10.5.1	Apache 2.0
com.fasterxml.jackson.dataformat	jackson-dataformat-xml	2.10.0	Apache 2.0
com.fasterxml.jackson.dataformat	jackson-dataformat-yaml	2.10.0	Apache 2.0
com.fasterxml.jackson.datatype	jackson-datatype-jdk8	2.10.0	Apache 2.0
com.fasterxml.jackson.datatype	jackson-datatype-jsr310	2.10.0	Apache 2.0
com.fasterxml.jackson.jaxrs	jackson-jaxrs-base	2.9.9	Apache
com.fasterxml.jackson.jaxrs	jackson-jaxrs-json-provider	2.9.9	Apache

表 1 – 续上页

GroupId	ArtifactId	Version	License
com.fasterxml.jackson.module	jackson-module-jaxb-annotations	2.10.0	Apache 2.0
com.fasterxml.jackson.module	jackson-module-parameter-names	2.10.0	Apache 2.0
com.fasterxml	classmate	1.4.0	Apache 2.0
com.github.docker-java	docker-java-api	3.2.7	Apache 2.0
com.github.docker-java	docker-java-core	3.2.7	Apache 2.0
com.github.docker-java	docker-java-transport-jersey	3.2.7	Apache 2.0
com.github.docker-java	docker-java-transport-netty	3.2.7	Apache 2.0
com.github.docker-java	docker-java-transport	3.2.7	Apache 2.0
com.github.docker-java	docker-java	3.2.7	Apache 2.0
com.github.jsqlparser	jsqlparser	2.0	Apache 2.0,LGPL 2.1
com.github.pagehelper	pagehelper-spring-boot-autoconfigure	1.2.13	MIT
com.github.pagehelper	pagehelper-spring-boot-starter	1.2.13	MIT
com.github.pagehelper	pagehelper	5.1.11	MIT
com.google.code.findbugs	jsr305	3.0.2	Apache 2.0
com.google.code.gson	gson	2.8.5	Apache 2.0
com.google.guava	guava	26.0-jre	Apache 2.0
com.google.inject	guice	4.2.0	Apache 2.0
com.google.inject.extensions	guice-assistedinject	4.1.0	Apache 2.0
com.google.inject.extensions	guice-multibindings	4.1.0	Apache 2.0
com.google.protobuf	protobuf-java	3.7.1	BSD 3-Clause
com.h2database	h2	1.4.199	MPL 2.0,EPL 1.0
com.ibm.icu	icu4j	58.2	ICU
com.jayway.jsonpath	json-path	2.4.0	Apache 2.0
com.jcraft	jsch	0.1.54	BSD
com.kohlschutter.junixsocket	junixsocket-common	2.3.2	Apache 2.0
com.kohlschutter.junixsocket	junixsocket-native-common	2.3.2	Apache 2.0
com.mikesamuel	json-sanitizer	1.2.2	Apache 2.0
com.netflix.archaius	archaius-core	0.7.6	Apache 2.0
com.netflix.hystrix	hystrix-core	1.5.18	Apache 2.0
com.netflix.netflix-commons	netflix-commons-util	0.3.0	Apache 2.0
com.netflix.netflix-commons	netflix-statistics	0.1.1	Apache 2.0
com.netflix.ribbon	ribbon	2.3.0	Apache 2.0
com.netflix.ribbon	ribbon-core	2.3.0	Apache 2.0
com.netflix.ribbon	ribbon-loadbalancer	2.3.0	Apache 2.0
com.netflix.ribbon	ribbon-transport	2.3.0	Apache 2.0
com.netflix.servo	servo-core	0.12.21	Apache 2.0
com.netflix.spectator	spectator-api	0.83.0	Apache 2.0

表 1 – 续上页

GroupId	ArtifactId	Version	License
com.netflix.spectator	spectator-reg-servo	0.83.0	Apache 2.0
com.spencerwi	Either.java	2.1.0	MIT
com.sun.istack	istack-commons-runtime	3.0.7	CDDL 1.1,GPL 1.1
com.sun.xml.fastinfoset	FastInfoset	1.2.15	Apache 2.0
com.vaadin.external.google	android-json	0.0.20131108.vaadin1	Apache 2.0
com.zaxxer	HikariCP	3.2.0	Apache 2.0
commons-codec	commons-codec	1.11	Apache 2.0
commons-configuration	commons-configuration	1.10	Apache 2.0
commons-fileupload	commons-fileupload	1.3.3	Apache 2.0
commons-io	commons-io	2.6	Apache 2.0
commons-lang	commons-lang	2.6	Apache 2.0
commons-net	commons-net	3.3	Apache 2.0
io.lettuce	lettuce-core	5.1.8.RELEASE	Apache 2.0
io.netty	netty-buffer	4.1.36.Final	Apache 2.0
io.netty	netty-codec	4.1.36.Final	Apache 2.0
io.netty	netty-codec-dns	4.1.36.Final	Apache 2.0
io.netty	netty-codec-http	4.1.36.Final	Apache 2.0
io.netty	netty-codec-http2	4.1.36.Final	Apache 2.0
io.netty	netty-codec-socks	4.1.36.Final	Apache 2.0
io.netty	netty-common	4.1.36.Final	Apache 2.0
io.netty	netty-handler	4.1.36.Final	Apache 2.0
io.netty	netty-handler-proxy	4.1.36.Final	Apache 2.0
io.netty	netty-resolver	4.1.36.Final	Apache 2.0
io.netty	netty-resolver-dns	4.1.36.Final	Apache 2.0
io.netty	netty-tcnative-boringssl-static	2.0.25.Final	Apache 2.0
io.netty	netty-transport	4.1.36.Final	Apache 2.0
io.netty	netty-transport-native-epoll	l4.1.36.Final	Apache 2.0
io.netty	netty-transport-native-kqueue	4.1.36.Final	Apache 2.0
io.netty	netty-transport-native-unix-common	4.1.36.Final	Apache 2.0
io.projectreactor	reactor-core	3.2.8.RELEASE	Apache 2.0
io.protostuff	protostuff-api	1.5.9	Apache 2.0
io.protostuff	protostuff-collectionschema	1.5.9	Apache 2.0
io.protostuff	protostuff-core	1.5.9	Apache 2.0
io.protostuff	protostuff-parser	2.2.25	Apache 2.0
io.protostuff	protostuff-runtime	1.5.9	Apache 2.0
io.reactivex	rxjava	1.2.0	Apache 2.0
io.reactivex	rxnetty	0.4.9	Apache 2.0

表 1 – 续上页

GroupId	ArtifactId	Version	License
io.reactivex	rxnetty-contexts	0.4.9	Apache 2.0
io.reactivex	rxnetty-servo	0.4.9	Apache 2.0
io.swagger	swagger-annotations	1.5.22	Apache 2.0
io.swagger	swagger-core	1.5.22	Apache 2.0
io.swagger	swagger-models	1.5.22	Apache 2.0
io.vertx	vertx-auth-common	3.9.4	Apache 2.0
io.vertx	vertx-bridge-common	3.9.4	Apache 2.0
io.vertx	vertx-core	3.9.4	Apache 2.0
io.vertx	vertx-web	3.9.4	Apache 2.0
io.vertx	vertx-web-common	3.9.4	Apache 2.0
io.zipkin.brave	brave	5.6.0	Apache 2.0
io.zipkin.reporter2	zipkin-reporter	2.7.13	Apache 2.0
io.zipkin.zipkin2	zipkin	2.19.1	Apache 2.0
jakarta.activation	jakarta.activation-api	1.2.2	EDL 1.0
jakarta.xml.bind	jakarta.xml.bind-api	2.3.3	EDL 1.0
javax.activation	javax.activation-api	1.2.0	CDDL,GPL2.0
javax.annotation	javax.annotation-api	1.3.2	CDDL,GPLv2.0
javax.inject	javax.inject	1	Apache 2.0
javax.servlet	javax.servlet-api	4.0.1	CDDL,GPLv2.0
javax.validation	validation-api	2.0.1.Final	Apache 2.0
javax.ws.rs	javax.ws.rs-api	2.1	CDDL 1.1,GPL 1.1
javax.xml.bind	jaxb-api	2.3.1	CDDL 1.1
junit	junit	4.13	EPL 1.0
net.bytebuddy	byte-buddy	1.9.13	Apache 2.0
net.bytebuddy	byte-buddy-agent	1.9.13	Apache 2.0
net.lingala.zip4j	zip4j	2.4.0	Apache 2.0
net.minidev	accessors-smart	1.2	Apache 2.0
net.minidev	json-smart	2.3	Apache 2.0
org.antlr	ST4	4.0.8	BSD
org.antlr	antlr-runtime	3.5.2	BSD
org.antlr	antlr4	4.7	The BSD
org.antlr	antlr4-runtime	4.7	BSD
org.apache.ant	ant	1.8.2	Apache 2.0
org.apache.ant	ant-launcher	1.8.2	Apache 2.0
org.apache.commons	commons-compress	1.19	Apache 2.0
org.apache.commons	commons-lang3	3.8.1	Apache 2.0
org.apache.httpcomponents	httpclient	4.5.9	Apache 2.0

表 1 – 续上页

GroupId	ArtifactId	Version	License
org.apache.httpcomponents	httpcore	4.4.11	Apache 2.0
org.apache.httpcomponents	httpmime	4.3.1	Apache 2.0
org.apache.logging.log4j	log4j-api	2.13.3	Apache 2.0
org.apache.logging.log4j	log4j-core	2.13.3	Apache 2.0
org.apache.logging.log4j	log4j-slf4j-impl	2.13.3	Apache 2.0
org.apache.logging.log4j	log4j-to-slf4j	2.11.2	Apache 2.0
org.apache.servicecomb	common-javassist	1.3.2	Apache 2.0
org.apache.servicecomb	common-protobuf	1.3.2	Apache 2.0
org.apache.servicecomb	common-rest	1.3.2	Apache 2.0
org.apache.servicecomb	deployment	1.3.2	Apache 2.0
org.apache.servicecomb	foundation-common	1.3.2	Apache 2.0
org.apache.servicecomb	foundation-config	1.3.2	Apache 2.0
org.apache.servicecomb	foundation-metrics	1.3.2	Apache 2.0
org.apache.servicecomb	foundation-protobuf	1.3.2	Apache 2.0
org.apache.servicecomb	foundation-ssl	1.3.2	Apache 2.0
org.apache.servicecomb	foundation-vertx	1.3.2	Apache 2.0
org.apache.servicecomb	handler-bizkeeper	1.3.2	Apache 2.0
org.apache.servicecomb	handler-flowcontrol-qps	1.3.2	Apache 2.0
org.apache.servicecomb	handler-loadbalance	1.3.2	Apache 2.0
org.apache.servicecomb	java-chassis-core	1.3.2	Apache 2.0
org.apache.servicecomb	provider-jaxrs	1.3.2	Apache 2.0
org.apache.servicecomb	provider-pojo	1.3.2	Apache 2.0
org.apache.servicecomb	provider-rest-common	1.3.2	Apache 2.0
org.apache.servicecomb	provider-springmvc	1.3.2	Apache 2.0
org.apache.servicecomb	service-registry	1.3.2	Apache 2.0
org.apache.servicecomb	spring-boot2-starter	1.3.2	Apache 2.0
org.apache.servicecomb	spring-boot2-starter-servlet	1.3.2	Apache 2.0
org.apache.servicecomb	swagger-generator-core	1.3.2	Apache 2.0
org.apache.servicecomb	swagger-generator-jaxrs	1.3.2	Apache 2.0
org.apache.servicecomb	swagger-generator-springmvc	1.3.2	Apache 2.0
org.apache.servicecomb	swagger-invocation-core	1.3.2	Apache 2.0
org.apache.servicecomb	swagger-invocation-jaxrs	1.3.2	Apache 2.0
org.apache.servicecomb	swagger-invocation-springmvc	1.3.2	Apache 2.0
org.apache.servicecomb	transport-common	1.3.2	Apache 2.0
org.apache.servicecomb	transport-highway	1.3.2	Apache 2.0
org.apache.servicecomb	transport-rest-client	1.3.2	Apache 2.0
org.apache.servicecomb	transport-rest-servlet	1.3.2	Apache 2.0

表 1 – 续上页

GroupId	ArtifactId	Version	License
org.apache.tomcat	tomcat-annotations-api	9.0.21	Apache 2.0
org.apache.tomcat.embed	tomcat-embed-core	9.0.44	Apache 2.0
org.apache.tomcat.embed	tomcat-embed-el	9.0.21	Apache 2.0
org.apache.tomcat.embed	tomcat-embed-websocket	9.0.44	Apache 2.0
org.apiguardian	apiguardian-api	1.0.0	Apache 2.0
org.assertj	assertj-core	3.11.1	Apache 2.0
org.awaitility	awaitility	3.0.0	Apache 2.0
org.bouncycastle	bcpkix-jdk15on	1.56	BouncyCastle
org.bouncycastle	bcprov-jdk15on	1.68	BouncyCastle
org.checkerframework	checker-qual	3.5.0	MIT
org.codehaus.woodstox	stax2-api	4.2	BSD
org.eclipse.persistence	javax.persistence	2.1.0	EPL 1.0
org.glassfish.hk2.external	aopalliance-repackaged	2.5.0-b42	EPL 2.0
org.glassfish.hk2.external	javax.inject	2.5.0-b42	EPL 2.0
org.glassfish.hk2	hk2-api	2.5.0-b42	EPL 2.0
org.glassfish.hk2	hk2-locator	2.5.0-b42	EPL 2.0
org.glassfish.hk2	hk2-utils	2.5.0-b42	EPL 2.0
org.glassfish.hk2	osgi-resource-locator	1.0.1	EPL 2.0
org.glassfish.jaxb	jaxb-runtime	2.3.1	CDDL,GPL
org.glassfish.jaxb	txw2	2.3.1	CDDL,GPL
org.glassfish.jersey.connectors	jersey-apache-connector	2.27	Apache 2.0,BSD 2-clause,
org.glassfish.jersey.core	jersey-client	2.27	Apache 2.0,BSD 2-clause,
org.glassfish.jersey.core	jersey-common	2.27	Apache 2.0,BSD 2-clause,
org.glassfish.jersey.inject	jersey-hk2	2.27	Apache 2.0,BSD 2-clause,
org.glassfish	jakarta.el	3.0.3	EPL 2.0
org.glassfish	javax.json	1.0.4	CDDL 1.1,GPL 2.0
org.hamcrest	hamcrest-core	1.3	BSD 3-clause
org.hamcrest	hamcrest-library	1.3	BSD 3-clause
org.hamcrest	hamcrest	1.3	BSD 3-clause
org.hdrhistogram	HdrHistogram	2.1.10	BSD 2-clause,CC0 1.0
org.hibernate.validator	hibernate-validator	6.0.17.Final	Apache 2.0
org.javassist	javassist	3.24.0-GA	Apache 2.0
org.jboss.logging	jboss-logging	3.3.2.Final	Apache 2.0
org.jmockit	jmockit	1.19	MIT
org.jmockit	jmockit-coverage	1.19	MIT
org.junit.jupiter	junit-jupiter-api	5.3.2	EPL 2.0
org.junit.jupiter	junit-jupiter-engine	5.3.2	EPL 2.0

表 1 – 续上页

GroupId	ArtifactId	Version	License
org.junit.jupiter	junit-jupiter-params	5.3.2	EPL 2.0
org.junit.jupiter	junit-jupiter	5.6.2	EPL 2.0
org.junit.platform	junit-platform-commons	1.3.2	EPL 2.0
org.junit.platform	junit-platform-engine	1.3.2	EPL 2.0
org.junit.vintage	junit-vintage-engine	5.3.2	EPL 2.0
org.jvnet.staxex	stax-ex	1.8	EDL 1.0
org.mockito	mockito-core	2.23.4	MIT
org.mockito	mockito-junit-jupiter	2.23.4	MIT
org.modelmapper	modelmapper	1.1.0	Apache 2.0
org.mybatis	mybatis	3.5.6	Apache 2.0
org.mybatis	mybatis-spring	2.0.6	Apache 2.0
org.mybatis.spring.boot	mybatis-spring-boot-autoconfigure	2.1.4	Apache 2.0
org.mybatis.spring.boot	mybatis-spring-boot-starter	2.1.4	Apache 2.0
org.objenesis	objenesis	2.6	Apache 2
org.opentest4j	opentest4j	1.1.1	Apache 2
org.ow2.asm	asm	5.0.4	BSD
org.postgresql	postgresql	42.2.16	BSD 2-Clause
org.projectlombok	lombok	1.18.10	The MIT
org.reactivestreams	reactive-streams	1.0.0	CC0
org.skyscreamer	jsonassert	1.5.0	Apache 2.0
org.slf4j	jcl-over-slf4j	1.7.26	MIT
org.slf4j	jcl-to-slf4j	1.7.26	MIT
org.slf4j	slf4j-api	1.7.30	MIT
org.springframework.boot	spring-boot-autoconfigure	2.1.6.RELEASE	Apache 2.0
org.springframework.boot	spring-boot-configuration-processor	2.1.6.RELEASE	Apache 2.0
org.springframework.boot	spring-boot-starter-jdbc	2.3.3.RELEASE	Apache 2.0
org.springframework.boot	spring-boot-starter-json	2.1.6.RELEASE	Apache 2.0
org.springframework.boot	spring-boot-starter-logging	2.1.6.RELEASE	Apache 2.0
org.springframework.boot	spring-boot-starter-test	2.3.3.RELEASE	Apache 2.0
org.springframework.boot	spring-boot-starter-tomcat	2.1.6.RELEASE	Apache 2.0
org.springframework.boot	spring-boot-starter-validation	2.3.3.RELEASE	Apache 2.0
org.springframework.boot	spring-boot-starter-web	2.3.3.RELEASE	Apache 2.0
org.springframework.boot	spring-boot-starter-websocket	2.1.18.RELEASE	Apache 2.0
org.springframework.boot	spring-boot-starter	2.1.6.RELEASE	Apache 2.0
org.springframework.boot	spring-boot-test-autoconfigure	2.1.6.RELEASE	Apache 2.0
org.springframework.boot	spring-boot-test	2.1.6.RELEASE	Apache 2.0
org.springframework.boot	spring-boot	2.1.6.RELEASE	Apache 2.0

表 1 – 续上页

GroupId	ArtifactId	Version	License
org.springframework	spring-aop	5.1.13.RELEASE	Apache 2.0
org.springframework	spring-beans	5.1.13.RELEASE	Apache 2.0
org.springframework	spring-context	5.1.13.RELEASE	Apache 2.0
org.springframework	spring-core	5.1.13.RELEASE	Apache 2.0
org.springframework	spring-expression	5.1.8.RELEASE	Apache 2.0
org.springframework	spring-jcl	5.1.13.RELEASE	Apache 2.0
org.springframework	spring-jdbc	5.2.8.RELEASE	Apache 2.0
org.springframework	spring-test	5.1.13.RELEASE	Apache 2.0
org.springframework	spring-tx	5.1.13.RELEASE	Apache 2.0
org.springframework	spring-web	5.1.13.RELEASE	Apache 2.0
org.springframework	spring-webmvc	5.1.13.RELEASE	Apache 2.0
org.springframework	spring-websocket	5.1.13.RELEASE	Apache 2.0
org.springframework.security	spring-security-config	5.3.8.RELEASE	Apache 2.0
org.springframework.security	spring-security-core	5.3.8.RELEASE	Apache 2.0
org.springframework.security	spring-security-jwt	1.0.9.RELEASE	Apache 2.0
org.springframework.security	spring-security-test	5.3.4.RELEASE	Apache 2.0
org.springframework.security	spring-security-web	5.3.8.RELEASE	Apache 2.0
org.springframework.security.oauth	spring-security-oauth2	2.3.8.RELEASE	Apache 2.0
org.springframework.security.oauth.boot	spring-security-oauth2-autoconfigure	2.2.2	Apache 2.0
org.xmlunit	xmlunit-core	2.6.2	Apache 2.0
org.yaml	snakeyaml	1.26	Apache 2.0

8.4.2 Developer-fe 三方件使用清单

Type	Name	Version	License	Direct Import (Y/N)
dependencies	acorn	7.2.0	MIT	Y
dependencies	auth0-js	9.13.2	MIT	Y
dependencies	axios	0.21.1	MIT	Y
dependencies	core-js	3.3.2	MIT	Y
dependencies	dot-prop	4.2.1	MIT	Y
dependencies	echarts	4.4.0	Apache-2.0	Y
dependencies	eg-view	0.0.12	Apache-2.0	Y
dependencies	el-table-pagination	0.4.38	MIT	Y
dependencies	element-china-area-data	4.1.2	MIT	Y
dependencies	element-ui	2.12.0	MIT	Y
dependencies	elliptic	6.5.4	MIT	Y

下页继续

表 2 – 续上页

Type	Name	Version	License	Direct Import (Y/N)
dependencies	http-proxy	1.18.1	MIT	Y
dependencies	http-proxy-agent	4.0.1	MIT	Y
dependencies	kity	2.0.4		Y
dependencies	kityminder-core	1.4.50	BSD-3-Clause	Y
dependencies	less-loader	5.0.0	MIT	Y
dependencies	local-storage	2.0.0	MIT	Y
dependencies	lodash	4.17.21	MIT	Y
dependencies	mavon-editor	2.9.1	MIT	Y
dependencies	serialize-javascript	4.0.0	BSD-3-Clause	Y
dependencies	swagger-ui	3.39.0	Apache-2.0	Y
dependencies	v-charts	1.19.0	MIT	Y
dependencies	vue	2.6.11	MIT	Y
dependencies	vue-cookies	1.7.0	MIT	Y
dependencies	vue-i18n	8.15.3	MIT	Y
dependencies	vue-router	3.1.3	MIT	Y
dependencies	vue-simple-uploader	0.7.6	MIT	Y
dependencies	vuex	3.0.1	MIT	Y
devDependencies	@vue/cli-plugin-babel	4.0.0	MIT	Y
devDependencies	@vue/cli-plugin-eslint	4.0.5	MIT	Y
devDependencies	@vue/cli-service	4.0.0	MIT	Y
devDependencies	@vue/eslint-config-standard	4.0.0	MIT	Y
devDependencies	babel-eslint	10.0.3	MIT	Y
devDependencies	eslint	5.16.0	MIT	Y
devDependencies	eslint-plugin-vue	5.2.3	MIT	Y
devDependencies	less	3.10.3	Apache-2.0	Y
devDependencies	mockjs	1.1.0		Y
devDependencies	node-sass	5.5.0	MIT	Y
devDependencies	postcss-import	11.0.0	MIT	Y
devDependencies	postcss-loader	2.0.8	MIT	Y
devDependencies	postcss-url	7.2.1	MIT	Y
devDependencies	sass-loader	10.1.1	MIT	Y
devDependencies	vue-template-compiler	2.6.11	MIT	Y

8.5 Developer DataBase Design

Developer 数据库使用了开源的 PostgreSQL 数据库，推荐版本为 12.2。**PostgreSQL** 是开源的对象-关系数据库服务器 (ORDBMS)，在灵活的 BSD 许可证下发行。进一步了解 PostgreSQL 可以点击这里。

- **tbl_plugin** 存储开发者上传的插件信息，如名字、满意度、插件保存路径等

```
 CREATE TABLE IF NOT EXISTS tbl_plugin (
       pluginid              varchar(255)       NOT NULL,     --Plugin ID
       pluginname            varchar(255)       NOT NULL,     --Plugin name
       introduction          varchar(500)       NULL,         --Introduction to the␣
↪plugin
       satisfaction          float4             NOT NULL,     --Satisfaction (rating)
       codelanguage          varchar(255)       NOT NULL,     --The programming␣
↪language represented by the plugin
       plugintype            int4               NOT NULL,     --Plugin type (1:␣
↪plugin 2: sdk)
       version               varchar(255)       NOT NULL,     --Plugin version
       downloadcount         int4               NOT NULL,     --download times
       logofile              varchar(500)       NOT NULL,     --Saved path of the␣
↪plugin icon
       pluginfile            varchar(500)       NOT NULL,     --Save path of the␣
↪plugin
       userid                varchar(255)       NOT NULL,     --ID of the user who␣
↪uploaded the plugin
       uploadtime            timestamptz(6)     NOT NULL,     --Upload time
       username              varchar(255)       NOT NULL,     --The name of the user␣
↪who uploaded the plugin
       pluginsize            int4               NOT NULL,     --Plugin size (bytes)
       apifile               varchar(500)       NOT NULL,     --Saved path of the␣
↪plugin API file
       scorecount            int4               NOT NULL,     --Number of ratings for␣
↪the plugin
       pluginfilehashcode    varchar(50)        DEFAULT NULL --Plugin hash value,␣
↪verification file
       CONSTRAINT tbl_plugin_pkey PRIMARY KEY (pluginid)
 );
```

- **tbl_downloadrecord** 存储插件的下载记录，一个插件对应对应多条下载记录

```
 CREATE TABLE IF NOT EXISTS tbl_downloadrecord (
      recordid          varchar(255)     NOT NULL,     --Download record ID
      pluginid          varchar(255)     NOT NULL,     --Plugin ID (corresponding to␣
↪pluginid in tbl_plugin)
      downloaduserid    varchar(255)     NOT NULL,     --Download user ID
```

(下页继续)

(续上页)

```
      downloadusername  varchar(255)     NOT NULL,     --Download user name
      score             float4           NOT NULL,     --Rating score (1-5)
      scoretype         int4             NOT NULL,     --Rating type
      downloadtime      timestamptz(0)   NOT NULL,     --Download time
      CONSTRAINT  tbl_downloadrecord_pkey  PRIMARY KEY (recordid)
 );
```

- **tbl_app_project** 存储开发者在开发者平台自构建的应用项目的信息

```
 CREATE TABLE IF NOT EXISTS  tbl_app_project  (
      id                  varchar(50)   NOT NULL,      --Project ID
      name                varchar(100)  NOT NULL,      --project name
      provider            varchar(100)  NOT NULL,      --provider
      platform            varchar(100)  NOT NULL,      --Project structure
      industries          varchar(100)  NOT NULL,      --Project Industry
      type                varchar(50)   NOT NULL,      --Project type (video/
↪security/game, etc.)
      description         text          DEFAULT NULL,  --project description
      status              varchar(20)   NOT NULL,      --Project status (tested,␣
↪medium deployment)
      user_id             varchar(50)   NOT NULL,      --ID of the user who created␣
↪the project
      create_date         timestamptz(6) DEFAULT NULL, --Creation time
      last_test_id        varchar(50)   DEFAULT NULL,  --ID of the last test
      version             varchar(50)   DEFAULT NULL,  --Project version
      capabilities        text          DEFAULT NULL,  --The ability of the project␣
↪(the ability checked when creating the project)
      project_type        varchar(10)   DEFAULT NULL,  --Project type (new/migrated)
      icon_file_id        varchar(50)   DEFAULT NULL,  --ID of the project icon file
      open_capability_id  varchar(50)   DEFAULT NULL,  --Ability ID provided by the␣
↪developer platform itself and shared by other developers
      deploy_platform     varchar(100)  DEFAULT NULL,  --Deployment method: virtual␣
↪machine, container deployment
      CONSTRAINT  tbl_app_project_pkey  PRIMARY KEY ( id )
 );
```

- **tbl_project_image** 存储开发者在开发者平台对自构建的应用项目进行部署测试时，开发者上传或者填写的镜像信息，一个项目的可以利用不同的镜像构建多次

```
 CREATE TABLE IF NOT EXISTS  tbl_project_image  (
      id                varchar(255)       NOT NULL,          --image ID
      image_info        text               DEFAULT NULL,      --Store image␣
↪information
      project_id        varchar(255)       DEFAULT NULL,      --Project ID␣
↪(corresponding to the id in tbl_app_project)
```

(下页继续)

(续上页)

```
      helm_chart_file_id varchar(255)       DEFAULT NULL,       --id of uoloaded␣
↪deploy yaml
     CONSTRAINT  tbl_project_image_pkey  PRIMARY KEY ( id )
 );
```

- **tbl_container_image** 存储开发者在开发者平台上传容器镜像包时，生成的容器镜像信息

```
 CREATE TABLE IF NOT EXISTS  tbl_container_image  (
      image_id           varchar(255)       NOT NULL,           --image ID
      image_name         varchar(255)       NOT NULL,           --image name
      image_version      varchar(255)       NOT NULL,           --image version
      user_id            varchar(255)       NOT NULL,           --the author id of␣
↪upload container image
      user_name          varchar(255)       NOT NULL,           --the author name of␣
↪upload container image
      upload_time        timestamptz(0)     DEFAULT NULL,       --upload image time
      create_time        timestamptz(0)     DEFAULT NULL,       --create db record␣
↪time
      image_status       varchar(255)       DEFAULT NULL,       --image status
      image_type         varchar(255)       DEFAULT NULL,       --image type(private␣
↪or public)
      image_path         text               DEFAULT NULL,       --image␣
↪info(xxxx:xxx)
      file_name         varchar(255)       DEFAULT NULL,       --image type(private␣
↪or public)
      CONSTRAINT tbl_container_image_uniqueName UNIQUE (image_name,image_version,
↪user_name),
      CONSTRAINT tbl_container_image_pkey PRIMARY KEY (image_id)
 );
```

- **tbl_app_project_capability** 开发者新建容器应用项目时，选择平台能力（此表是项目和能力之间的关联表）

```
 CREATE TABLE IF NOT EXISTS  tbl_app_project_capability  (
      project_id         varchar(50)        NOT NULL,           --project_id
      capability_id      varchar(50)        NOT NULL,           --capability id
      CONSTRAINT tbl_app_project_capability_pkey PRIMARY KEY (project_id,capability_
↪id)
 );
```

- **tbl_capability_group** 开发者新建容器应用项目时，选择平台能力（能力分类表）

```
 CREATE TABLE IF NOT EXISTS  tbl_capability_group  (
      id             varchar(50)         NOT NULL,          --group id
```

(下页继续)

(续上页)

```
      name            varchar(255)        DEFAULT NULL,     --capability group zh name
      name_en         varchar(255)        DEFAULT NULL,     --capability group en name
      type            varchar(20)         DEFAULT NULL,     --capability group␣
↪type(openmep or openmep_eco)
      description     text                DEFAULT NULL,     --capability group desc
      description_en text                 DEFAULT NULL,     --capability group en desc
      icon_file_id    varchar(50)         DEFAULT NULL,     --group icon
      author          varchar(50)         DEFAULT NULL,     --the author of create␣
↪group
      update_time     bigint              NOT NULL DEFAULT 0,       --update db␣
↪record time
      create_time     bigint              NOT NULL DEFAULT 0,       --create db␣
↪record time
      CONSTRAINT tbl_capability_group_pkey PRIMARY KEY (id)
 );
```

- **tbl_capability** 开发者新建容器应用项目时，选择平台能力（能力详情表）

```
 CREATE TABLE IF NOT EXISTS  tbl_capability  (
      id                varchar(50)         NOT NULL,         --capability id
      name              varchar(100)        DEFAULT NULL,     --capability  zh name
      name_en           varchar(100)        DEFAULT NULL,     --capability  en name
      version           varchar(100)        DEFAULT NULL,     --capability  version
      description       text                DEFAULT NULL,     --capability  desc
      description_en    text                DEFAULT NULL,     --capability  en desc
      provider          varchar(100)        DEFAULT NULL,     --capability  provider
      api_file_id       varchar(255)        DEFAULT NULL,     --capability api file
      guide_file_id     varchar(255)        DEFAULT NULL,     --capability guide file
      guide_file_id_en  varchar(255)        DEFAULT NULL,     --capability guide␣
↪file(en version)
      upload_time       varchar(50)         NOT NULL,         --create capability␣
↪time
      host              varchar(50)         DEFAULT NULL,     --registered service␣
↪name
      port              int4                DEFAULT NULL,     --service port
      protocol          varchar(20)         DEFAULT NULL,     --http,https
      app_id            varchar(255)        DEFAULT NULL,     --id posted to appstore
      package_id        varchar(255)        DEFAULT NULL,     --the packageId␣
↪published to the appstore
      user_id           varchar(255)        DEFAULT NULL,     --the author id of␣
↪create this capability
      select_count      integer             NOT NULL DEFAULT 0, --the number of␣
↪select this capability
      icon_file_id      varchar(50)         DEFAULT NULL,     --capability icon file
```

(下页继续)

(续上页)

```
      author            varchar(50)         DEFAULT NULL,     --the author name of␣
↪create this capability
      experience_url    text                DEFAULT NULL,     --the access url of␣
↪capability
      CONSTRAINT tbl_capability_pkey PRIMARY KEY (id)
 );
```

- **tbl_project_test_config** 存储开发者在开发者平台对自构建的应用项目进行部署测试的配置信息，包括镜像、服务器等，一个项目可以多次配置测试项进行部署测试

```
 CREATE TABLE IF NOT EXISTS  tbl_project_test_config  (
      test_id             varchar(50)      NOT NULL,        --Test configuration ID
      project_id          varchar(50)      NOT NULL,        --Project ID␣
↪(corresponding to the id in tbl_app_project)
      agent_config        text             DEFAULT NULL,    --MEP AGENT␣
↪configuration item
      image_file_id       varchar(255)     NOT NULL,        --Image file ID
      app_api_file_id     varchar(50)      DEFAULT NULL,    --Project API document ID
      deploy_file_id      varchar(50)      DEFAULT NULL,    --Deployment file ID
      private_host        bool             DEFAULT FALSE,   --Whether the node is␣
↪private
      platform            varchar(100)     DEFAULT NULL,    --Deployment method
      access_url          varchar(200)     DEFAULT NULL,    --Project accessible URl
      error_log           text             DEFAULT NULL,    --Project deployment␣
↪test failed log
      deploy_date         timestamptz(6)   DEFAULT NULL,    --Project deployment time
      hosts               varchar(255)     DEFAULT NULL,    --Server information␣
↪selected for project deployment
      app_instance_id     varchar(50)      DEFAULT NULL,    --The project deployment␣
↪is successfully uploaded to the instance ID generated by the app store
      work_load_id        varchar(255)     DEFAULT NULL,    --Deploy the project to␣
↪applcm
      pods                text             DEFAULT NULL,    --Deploy pods information
      deploy_status       varchar(255)     DEFAULT NULL,    --Deployment status
      stage_status        varchar(255)     DEFAULT NULL,    --Deployment steps
      lcm_token           varchar(1000)    DEFAULT NULL,    --lcm token
      package_id          varchar(255)     DEFAULT NULL,    --Generated csar package␣
↪ID
      CONSTRAINT  tbl_project_test_config_pkey  PRIMARY KEY ( test_id )
 );
```

- **tbl_service_host** 存储开发者在开发者平台对自构建的应用项目进行部署测试时，可供选择的服务器信息

```
CREATE TABLE IF NOT EXISTS  tbl_service_host  (
      host_id           varchar(50)    NOT NULL,        --Server id
      user_id           varchar(50)    DEFAULT NULL,    --UserId
      name              varchar(100)   DEFAULT NULL,    --server nickname
      address           varchar(255)   DEFAULT NULL,    --address
      architecture      varchar(100)   DEFAULT NULL,    --Architecture
      status            varchar(20)    DEFAULT NULL,    --status
      protocol          varchar(20)    DEFAULT NULL,    --protocol
      lcm_ip            varchar(20)    DEFAULT NULL,    --IP address
      mec_host          varchar(20)    DEFAULT NULL,    --mec host address
      os                varchar(255)   DEFAULT NULL,    --Operating system name
      port_range_min    int            DEFAULT '-1',    --Port minimum
      port_range_max    int            DEFAULT '-1',    --Maximum port
      port              int4           DEFAULT '-1',    --Port value
      user_name         varchar(50)    DEFAULT NULL,    --User name required to log␣
↪in to the host
      password          varchar(50)    DEFAULT NULL,    --Password required to log␣
↪in to the host
      vnc_port          int4           DEFAULT 22,      --The exposed connection␣
↪port of the server deploying the edge application
      parameter         text           DEFAULT NULL,    --sandbox parameter
      delete            bool           DEFAULT NULL,    --delete or not
      resource          text           DEFAULT NULL     --Sandbox hardware␣
↪information
 );
```

- **tbl_uploaded_file** 存储开发者在开发者平台对自构建的应用项目进行部署测试时，上传的图标、镜像、api 等文件信息

```
CREATE TABLE IF NOT EXISTS  tbl_uploaded_file  (
      file_id         varchar(50)      NOT NULL,       --File ID
      file_name       varchar(255)     DEFAULT NULL,   --file name
      is_temp         bool             DEFAULT NULL,   --Is it a temporary file
      user_id         varchar(50)      DEFAULT NULL,   --ID of the user who uploaded␣
↪the file
      upload_date     timestamptz(6)   DEFAULT NULL,   --Upload time
      file_path       varchar(255)     DEFAULT NULL,   --file path
     CONSTRAINT  tbl_uploaded_file_pkey  PRIMARY KEY ( file_id )
 );
```

- **tbl_helm_template_yaml** 存储开发者在开发者平台对自构建的应用项目进行部署测试时，上传的 Helm yaml 文件信息 (目前版本通过 yaml 部署)，一个项目可以通过不同的 yaml 文件进行多次部署测试

```
CREATE TABLE IF NOT EXISTS  tbl_helm_template_yaml  (
```

(下页继续)

(续上页)

```
      file_id            varchar(50)  NOT NULL,      --File ID
      file_name          varchar(255) DEFAULT NULL,  --file name
      user_id            varchar(50)  DEFAULT NULL,  --ID of the user who uploaded␣
↪the file
      project_id         varchar(50)  DEFAULT NULL,  --Project ID (corresponding to␣
↪the id in tbl_app_project)
      content            text         DEFAULT NULL,  --document content
      upload_time_stamp  bigint       DEFAULT NULL,  --Upload time
      config_type        varchar(50)  DEFAULT NULL,  --Distinguish the deployment␣
↪type (deployment file/visual configuration)
     CONSTRAINT  tbl_helm_template_yaml_pkey  PRIMARY KEY ( file_id )
 );
```

- **tbl_host_log** 记录 app 项目在边缘节点上部署的日志

```
  CREATE TABLE IF NOT EXISTS  tbl_host_log  (
      log_id  varchar(50) NOT NULL,               --Log ID
      host_ip  varchar(50) NOT NULL,              --host IP
      user_name  varchar(50) DEFAULT NULL,        --username
      user_id  varchar(50) DEFAULT NULL,          --User ID
      project_id  varchar(50) DEFAULT NULL,       --Project ID
      project_name  varchar(50) DEFAULT NULL,     --project name
      app_instances_id  varchar(50) DEFAULT NULL, --The instance ID of the app␣
↪project
      deploy_time  varchar(50) DEFAULT NULL,      --Project deployment time
      status  varchar(50) DEFAULT NULL,           --host status
      operation  varchar(50) DEFAULT NULL,        --operating
      host_id  varchar(50) DEFAULT NULL           --host ID
   );
```

- **tbl_release_config** 存储应用发布对应的配置信息

```
 CREATE TABLE IF NOT EXISTS tbl_release_config (
        release_id            varchar(255)     NOT NULL,             --Application␣
↪release id
        project_id            varchar(255)     NOT NULL,             --Project id
        guide_file_id         varchar(255)     DEFAULT NULL,         --Guidance␣
↪document ID
        appinstance_id        varchar(255)     DEFAULT NULL,         --Instantiated␣
↪ID
        capabilities_detail   text             DEFAULT NULL,         --Application␣
↪configuration information
        atp_test              text             DEFAULT NULL,         --atp test␣
↪result
```

(下页继续)

(续上页)

```
        test_status           varchar(255)     DEFAULT NULL,          --Test status
        create_time           timestamptz(0)   NOT NULL DEFAULT NULL,--Creation time
        CONSTRAINT tbl_release_config_pkey PRIMARY KEY (release_id)
);
```

- **tbl_vm_regulation** 虚机规则，虚机属性说明（虚机部署时提供选择）

```
CREATE TABLE IF NOT EXISTS  tbl_vm_regulation  (
     regulation_id  SERIAL,                          --Rule id
     architecture  varchar(50) DEFAULT NULL,         --Virtual machine architecture
     name_zh  varchar(50) NOT NULL DEFAULT NULL,     --Chinese name of the virtual␣
↪machine
     name_en  varchar(50) NOT NULL DEFAULT NULL,     --English name of virtual␣
↪machine
     scene_zh  varchar(255) DEFAULT NULL,            --Virtual machine usage␣
↪scenarios
     scene_en  varchar(255) DEFAULT NULL,            --English description of␣
↪virtual machine usage scenarios
     memory  int4  DEFAULT NULL,                     --Virtual Memory
     cpu  int4  DEFAULT NULL,                        --Virtual machine cpu
     system_disk  int4  DEFAULT NULL,                --Virtual machine system disk␣
↪size
     data_disk  int4  DEFAULT NULL,                  --Virtual machine data disk size
     gpu  varchar(50)  DEFAULT NULL,                 --Virtual machine gpu
     other_ability  varchar(255)  DEFAULT NULL       --Other capabilities of the␣
↪virtual machine
   );
```

- **tbl_vm_network** 虚机网络配置

```
CREATE TABLE IF NOT EXISTS  tbl_vm_network  (
     network_type  varchar(50) DEFAULT NULL,    --Network Type
     description_zh  varchar(255) DEFAULT NULL, --Network Chinese description
     description_en  varchar(255) DEFAULT NULL, --English description of the network
     network_name  varchar(50) DEFAULT NULL     --Network name
  );
```

- **tbl_vm_system** 虚机系统属性说明

```
CREATE TABLE IF NOT EXISTS  tbl_vm_system  (
     system_id  SERIAL,                          --System id
     system_name varchar(128) DEFAULT NULL,      --system vm name
     type  varchar(50) DEFAULT NULL,             --System type
     operate_system  varchar(50) DEFAULT NULL,   --system name
```

(下页继续)

(续上页)

```
      version  varchar(50) NOT NULL DEFAULT NULL, --system version
      system_bit  varchar(50) DEFAULT NULL,       --System size
      system_disk  int4  DEFAULT NULL             --System disk description
      user_id varchar(50) DEFAULT NULL,           --author id or create db record
      user_name varchar(50) DEFAULT NULL,         --author name or create db record
      create_time timestamptz(6)  DEFAULT NULL,   --create time
      modify_time timestamptz(6)  DEFAULT NULL,   --update time
      system_format varchar(50) DEFAULT NULL,     --vm image format
      system_size bigint DEFAULT NULL,            --vm image size
      system_slim varchar(50) DEFAULT NULL,       --vm image size then slim
      upload_time timestamptz(6)  DEFAULT NULL,   --vm image upload time
      system_path varchar(128) DEFAULT NULL,      --uploaded vm image path
      file_name varchar(128) DEFAULT NULL,        --uploaded vm image file name
      file_md5 varchar(128) DEFAULT NULL,         --uploaded vm image file md5 value
      status varchar(50) DEFAULT NULL,            --vm image status
      file_identifier varchar(128) DEFAULT NULL,  --uploaded vm image file identifier
      error_type varchar(32) DEFAULT NULL,        -- the type when upload vm image␣
↪fail
      CONSTRAINT tbl_vm_system_uniqueName UNIQUE (system_name,user_id),
      CONSTRAINT tbl_vm_system_pkey PRIMARY KEY (system_id)
   );
```

- **tbl_project_vm_create_config** app 项目虚机部署，创建虚机的配置说明

```
   CREATE TABLE IF NOT EXISTS  tbl_project_vm_create_config  (
      vm_id   varchar(255) NOT NULL DEFAULT NULL,                --Virtual machine␣
↪id
      project_id  varchar(50) DEFAULT NULL,                      --Project id
      vm_name  varchar(50)  DEFAULT NULL,                        --Virtual machine␣
↪name
      host  varchar(512)  DEFAULT NULL,                          --Virtual machine␣
↪host description
      status  varchar(50)  DEFAULT NULL,                         --Virtual machine␣
↪state
      stage_status  varchar(500)  DEFAULT NULL,                  --Virtual machine␣
↪deployment step status
      lcm_token  varchar(1024)  DEFAULT NULL,                    --lcm token
      vm_info  varchar(512)  DEFAULT NULL,                       --Virtual machine␣
↪information
      app_instance_id  varchar(50)  DEFAULT NULL,                --The id of the␣
↪deployed app instance
      package_id  varchar(100)  DEFAULT NULL,                    --Deploy the␣
↪generated csar package id
      create_time   timestamptz(6)  DEFAULT NULL,                --Creation time
```

(下页继续)

(续上页)

```
    log  text  DEFAULT NULL,                                    --Operation log
    CONSTRAINT  tbl_project_vm_create_config_pkey  PRIMARY KEY ( vm_id )
);
```

- **tbl_project_vm_image_config** app 项目虚机部署，镜像配置说明

```
CREATE TABLE IF NOT EXISTS  tbl_project_vm_image_config  (
    vm_id   varchar(255) NOT NULL DEFAULT NULL,              --Virtual machine id
    image_id  varchar(50) DEFAULT NULL,                      --Mirror id
    project_id  varchar(50) DEFAULT NULL,                    --Project id
    vm_name  varchar(50) NOT NULL DEFAULT NULL,              --Virtual machine name
    image_name  varchar(50) DEFAULT NULL,                    --Mirror name
    app_instance_id  varchar(50)  DEFAULT NULL,              --Instance id
    host_ip  varchar(50)  DEFAULT NULL,                      --Host ip
    sum_chunk_num  varchar(50)  DEFAULT NULL,                --Number of mirrored␣
↪blocks
    chunk_size  varchar(50)  DEFAULT NULL,                   --Mirror block size
    stage_status  varchar(500)  DEFAULT NULL,                --Deployment status
    status  varchar(512)  DEFAULT NULL,                      --status
    lcm_token  varchar(1024)  DEFAULT NULL,                  --lcm token
    create_time   timestamptz(6)  DEFAULT NULL,              --Creation time
    log  text  DEFAULT NULL,                                 --Log
  CONSTRAINT  tbl_project_vm_image_config_pkey  PRIMARY KEY ( vm_id )
);
```

- **tbl_vm_user_data** app 项目虚机部署，存储的部署用户的信息

```
CREATE TABLE IF NOT EXISTS  tbl_vm_user_data  (
    operate_system   varchar(50)  DEFAULT NULL,              --vm system type
    flavor_extra_specs  text DEFAULT NULL,                   --user config flavor
    is_temp  bool DEFAULT NULL,                              --Determine whether␣
↪the user is a temporary configuration
    contents  text  DEFAULT NULL,                            --user config content
    params  text  DEFAULT NULL,                              --user config param
  CONSTRAINT  tbl_vm_user_data_pkey  PRIMARY KEY ( operate_system )
);
```

- **tbl_project_vm_package_config** app 项目虚机部署，生成虚机包所需的配置信息

```
CREATE TABLE IF NOT EXISTS tbl_project_vm_package_config (
  id varchar(50) DEFAULT NULL,                               --id
  project_id varchar(50) DEFAULT NULL,                       --project id
  vm_regulation_desc text DEFAULT NULL,                      --Rule information␣
↪required to generate sequel package
```

(下页继续)

(续上页)

```
    vm_system_desc text DEFAULT NULL,                          --vm image config info
    vm_network_desc text DEFAULT NULL,                         --vm network config␣
↪info
    vm_user_data text DEFAULT NULL,                            --user config
    vm_name varchar(500) DEFAULT NULL,                         --vm name
    ak text DEFAULT NULL,                                      --user config param
    sk text DEFAULT NULL,                                      --user config param
    app_instance_id varchar(50) DEFAULT NULL,                  --instance id of␣
↪generate vm pkg
    create_time timestamptz(6) DEFAULT NULL,                   --create time
    CONSTRAINT tbl_project_vm_package__uniqueProjectId UNIQUE (project_id),
    CONSTRAINT tbl_project_vm_package_config_pkey PRIMARY KEY (id)
  );
```

8.6 Developer Contribution

8.6.1 本地开发环境搭建

本文将指导用户如何在本地搭建 developer-be 和 developer-fe 模块，从而帮助开发者快速理解、开发代码。

其中 developer-be 的本地启动依赖于：

```
- Service Center
- DataBase
- UserManagement
```

其中 developer-fe 的本地启动依赖于:

```
- WebsiteGateway
- UserManagement
- developer-be
```

所以为了搭建本地开发环境，最终的启动顺序为 Service Center -> DataBase -> UserManagement -> developer-be -> WebsiteGateway(developer-fe)

8.6.2 流程详解

Service Center

1、下载Service Center

ServiceComb Service-Center下载

发布包

版本号	源码	Windows版本	Linux版本	Darwin版本	与之兼容的Java-Chassis版本
Apache ServiceComb Service-Center 1.3.0 (LATEST)	[src] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	Java-Chassis 1.0.0 or above
Apache ServiceComb Service-Center 1.2.0	[src] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	Java-Chassis 1.0.0 or above
Apache ServiceComb Service-Center 1.1.0	[src] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	Java-Chassis 1.0.0 or above
Apache ServiceComb (incubating) Service-Center 1.0.0	[src] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	Java-Chassis 1.0.0 or above
Apache ServiceComb (incubating) Service-Center 1.0.0-m2	[src] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	NA	Java-Chassis 1.0.0-m2 or above
Apache ServiceComb (incubating) Service-Center 1.0.0-m1	[src] [asc] [sha512]	[Binary] [asc] [sha512]	[Binary] [asc] [sha512]	NA	Java-Chassis 1.0.0-m1 or above

2、解压后运行 start-service-center.bat 和 start-frontend.bat

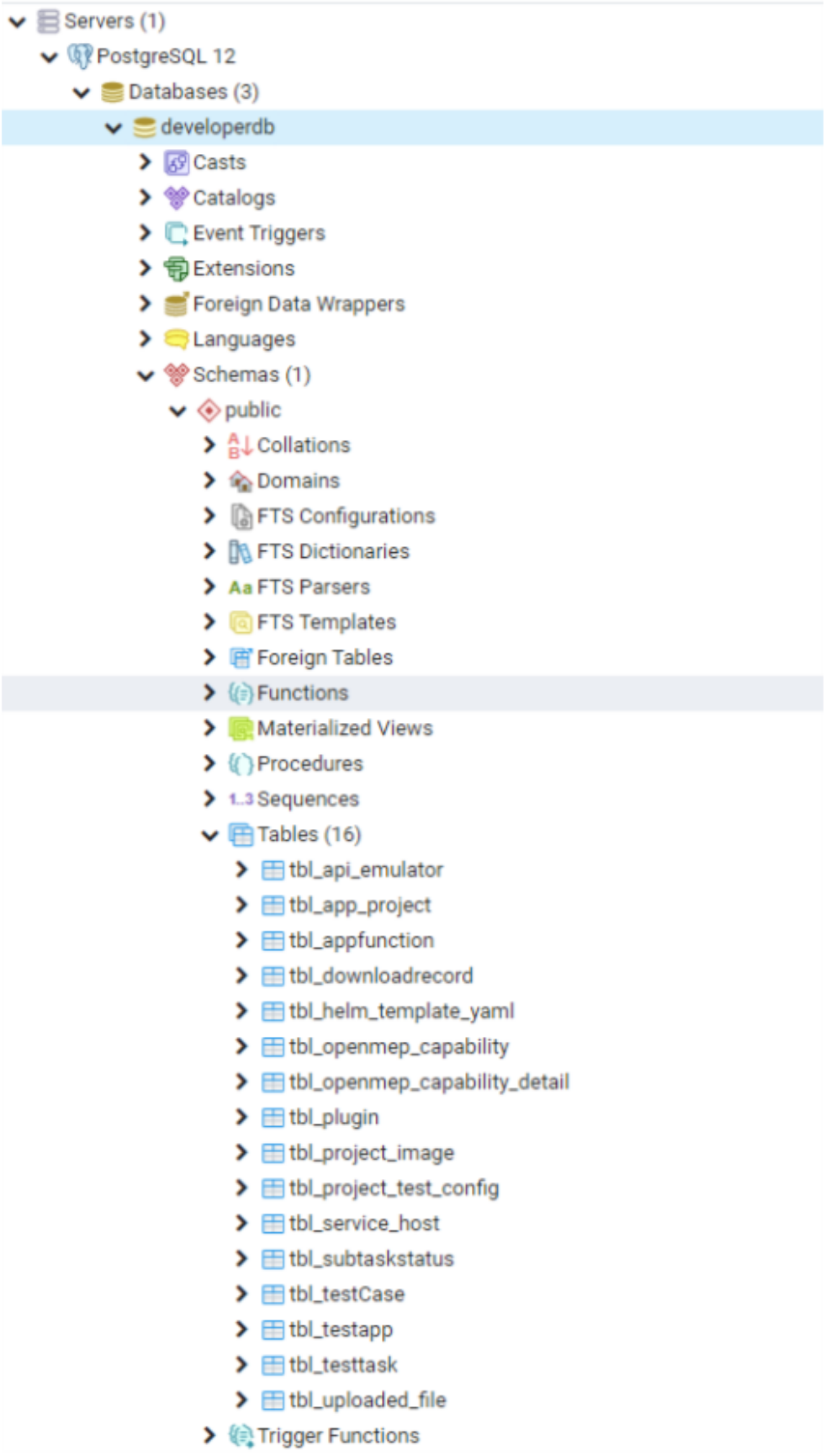
名称	修改日期	类型	大小
app	2019/12/18 14:15	文件夹	
conf	2019/12/18 14:15	文件夹	
data	2020/7/28 12:07	文件夹	
licenses	2019/12/18 14:15	文件夹	
frontend.exe	2019/10/27 16:55	应用程序	9,519 KB
LICENSE	2019/10/27 16:55	文件	46 KB
NOTICE	2019/10/27 16:55	文件	3 KB
README.md	2019/10/27 16:55	MD 文件	9 KB
scctl.exe	2019/10/27 16:55	应用程序	15,394 KB
service-center.exe	2019/10/27 16:55	应用程序	43,543 KB
start-frontend.bat	2019/10/27 16:55	Windows 批处理...	1 KB
start-service-center.bat	2019/10/27 16:55	Windows 批处理...	1 KB
syncer.exe	2019/10/27 16:55	应用程序	26,730 KB

DataBase

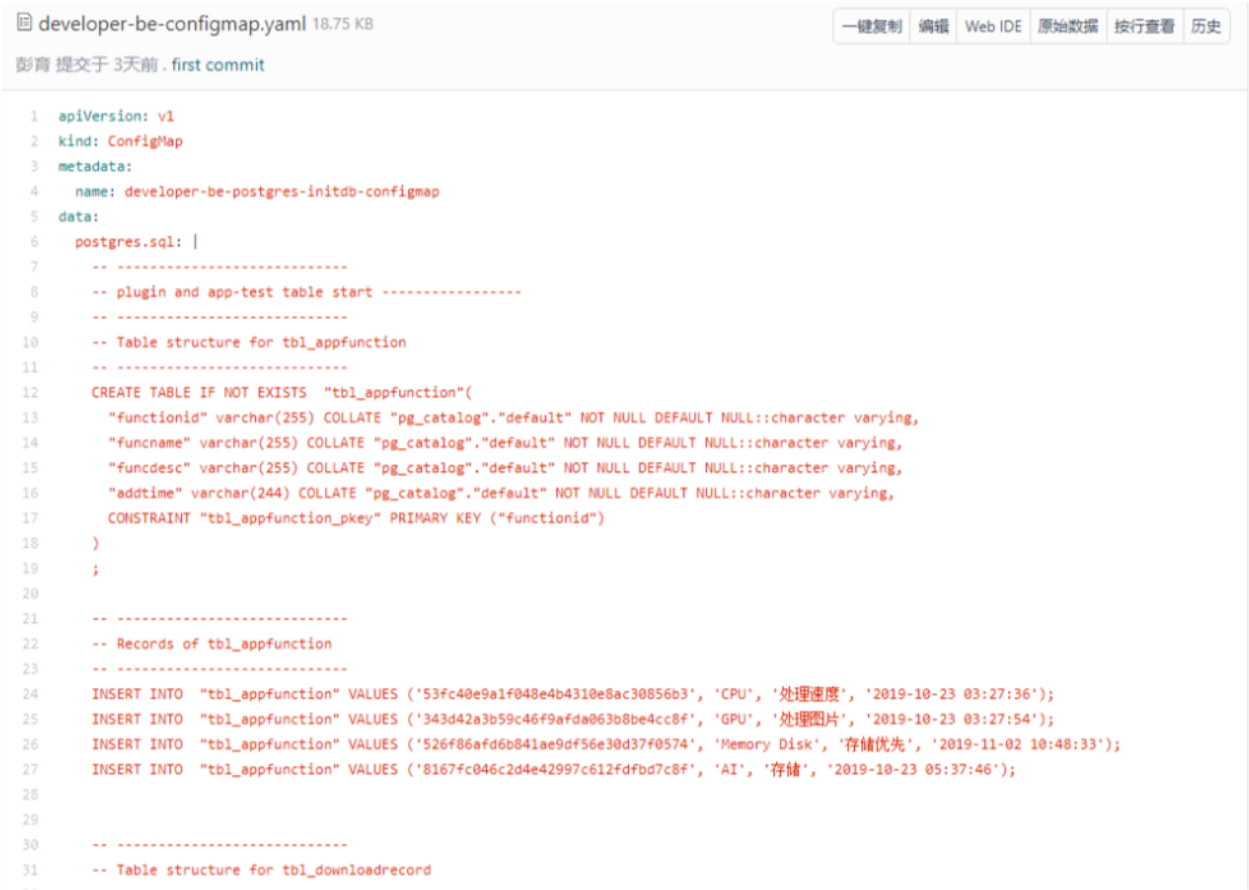
1、下载、安装、启动 Postgresql 最新版本

2、设置登录名和密码，例如 postgresql/root

3、创建名为 developerdb 的数据库


Servers (1)
PostgreSQL 12
Databases (3)
developerdb
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Schemas (1)
public
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Procedures
Sequences
Tables (16)
tbl_api_emulator
tbl_app_project
tbl_appfunction
tbl_downloadrecord
tbl_helm_template_yaml
tbl_openmep_capability
tbl_openmep_capability_detail
tbl_plugin
tbl_project_image
tbl_project_test_config
tbl_service_host
tbl_subtaskstatus
tbl_testCase
tbl_testapp
tbl_testtask
tbl_uploaded_file
Trigger Functions

4、根据developer 部署文件 中的数据库语句创建表结构



5、创建名为 usermgmtdb 的数据库

6、根据usermgmt 部署文件中的数据库语句创建表结构

User Management

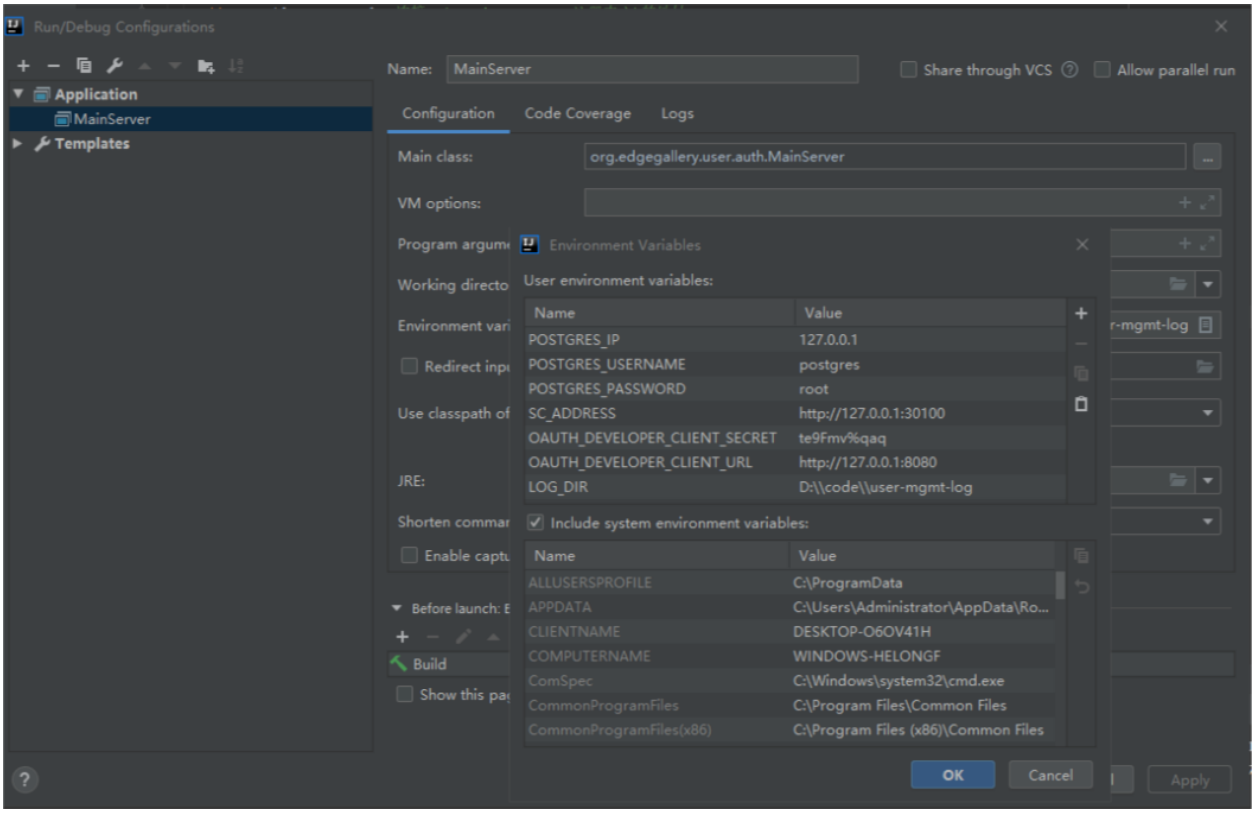
1、下载 user-mgmt-be 代码和 user-mgmt-fe 代码

2、安装 node.js 和 npm，进入 user-mgmt-fe 文件夹，运行 `npm install` 和 `npm run build`

3、在 user-mgmt-be 的 src/main/resources/文件夹中创建 static 文件夹

4、将 user-mgmt-fe 编译出的 dist 文件夹中的内容拷贝到 user-mgmt-be 的 static 文件夹中

5、在编译器（此处以 IDEA 为例）中配置 MainServer 的运行时环境变量

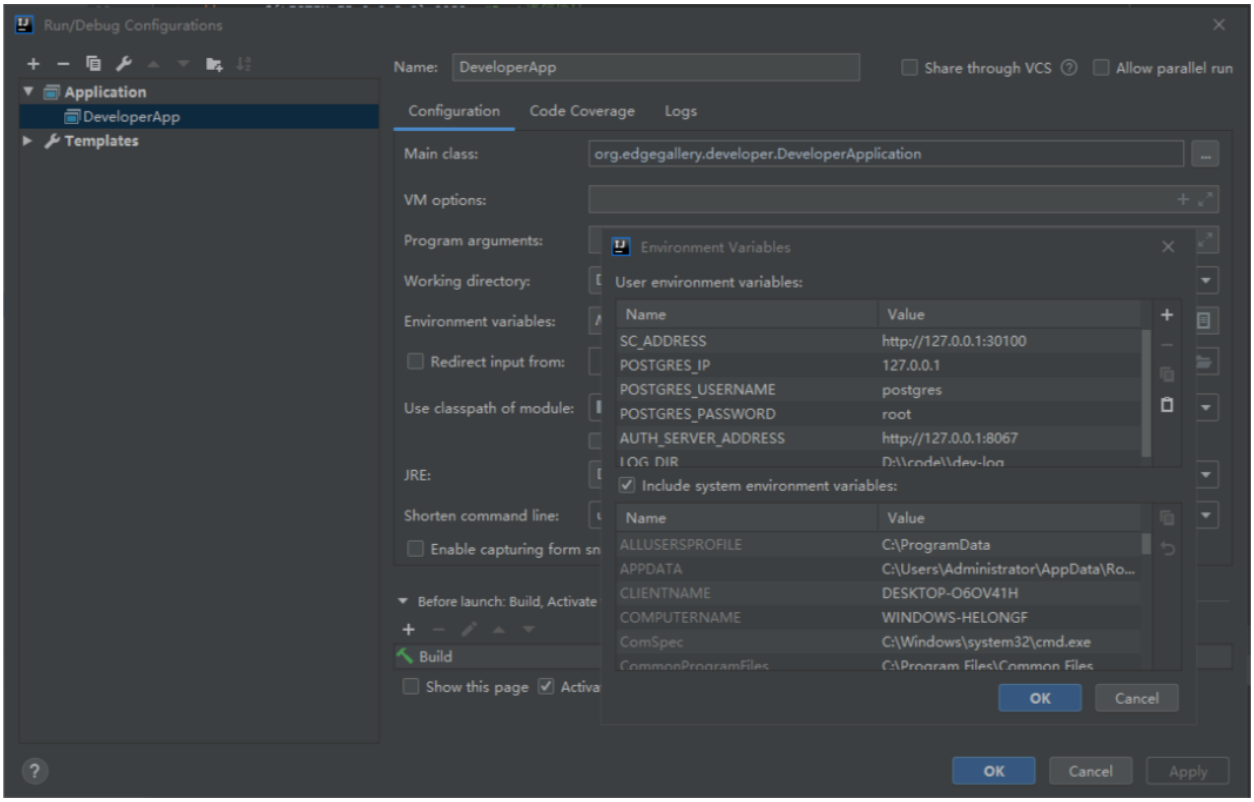


6、运行 MainServer 函数，启动 UserManagement

Developer-be

1、下载 developer-be 代码

2、在编译器（此处以 IDEA 为例）中配置 DeveloperApp 的运行时环境变量



3、运行 DeveloperApp 函数，启动 developer-be

Developer-fe

1、下载 developer-fe 代码

2、进入 developer-fe 文件夹，运行 `npm install` 和 `npm run build`

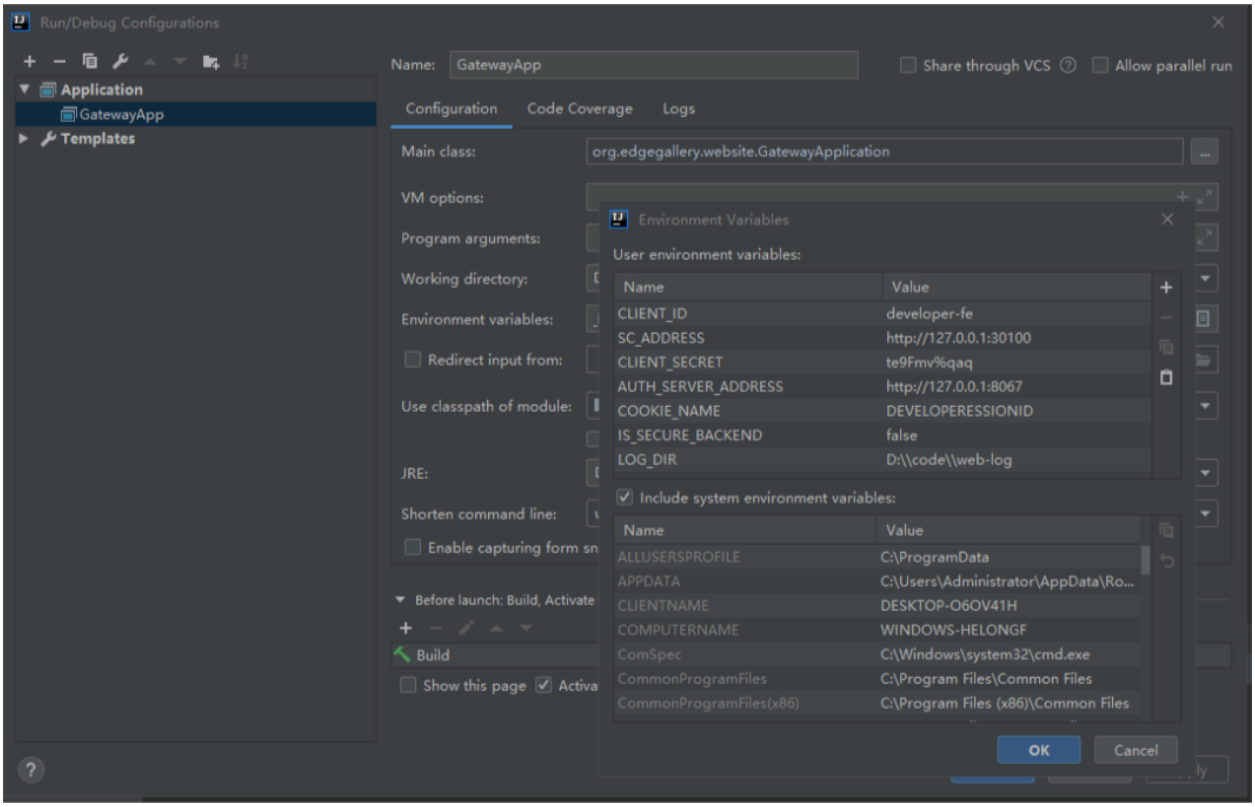
3、下载 website-gateway 代码

4、在 website-gateway 的 src/main/resources/文件夹中创建 static 文件夹

5、将 developer-fe 编译出的 dist 文件夹中的内容拷贝到 website-gateway 的 static 文件夹中

6、修改 website-gateway 中/src/main/resources/application.yaml 文件内的 `server.port` 为 `8080`

7、在编译器（此处以 IDEA 为例）中配置 GatewayApp 的运行时环境变量



8、运行 GatewayApp 启动 WebsiteGateway 与 developer-fe，浏览器访问 http://127.0.0.1:8080/ 即可访问到 developer 首页

CHAPTER 9

MECM

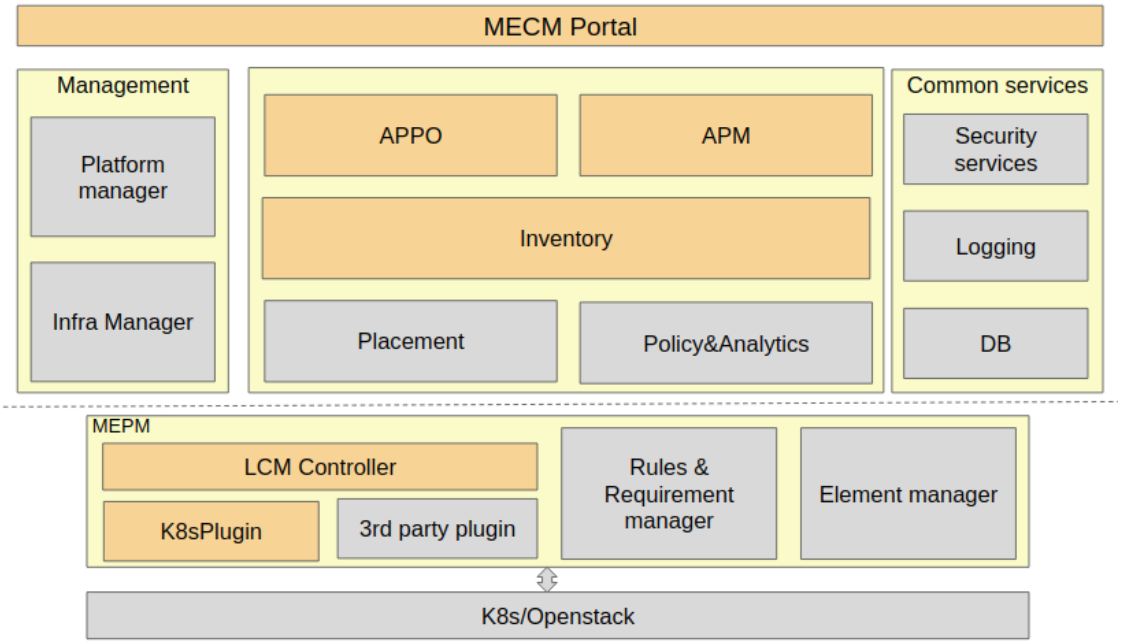
9.1 Overview 概述

9.1.1 Overview

MECM (multi access edge compute manager) provides orchestration and life cycle management of application in edgegallery architecture. MECM provides various features including application on-boarding, application orchestration by selecting appropriate edge based on deployment strategy, application life cycle management, homing and placement of application based on analytics and policies, application/edge resource monitoring and provides unified topology view.

9.2 Architecture

9.2.1 Architecture



Application Package Manager

Application Package Manager(APM) enables edgegallery to distribute/on-board applications to edge repositories by downloading application packages from appstore.

Application Orchestrator

Application Orchestrator(APPO) is mainly responsible for MEC application orchestration by requesting for life cycle management of MEC application on the edge host by executing specified workflow while maintaining the life cycle state. Orchestration workflows are modeled using Camunda modeler tool and workflows are executed by camunda engine.

Inventory

Inventory provides realtime view of deployed applications on the edge hosts and applications configurations. Inventory is also used for external system registrations.

LCM Controller

LCM Controller is responsible MEC application life cycle management operations by sending requests to appropriate pluggable plugin adapters based on the infrastructure.

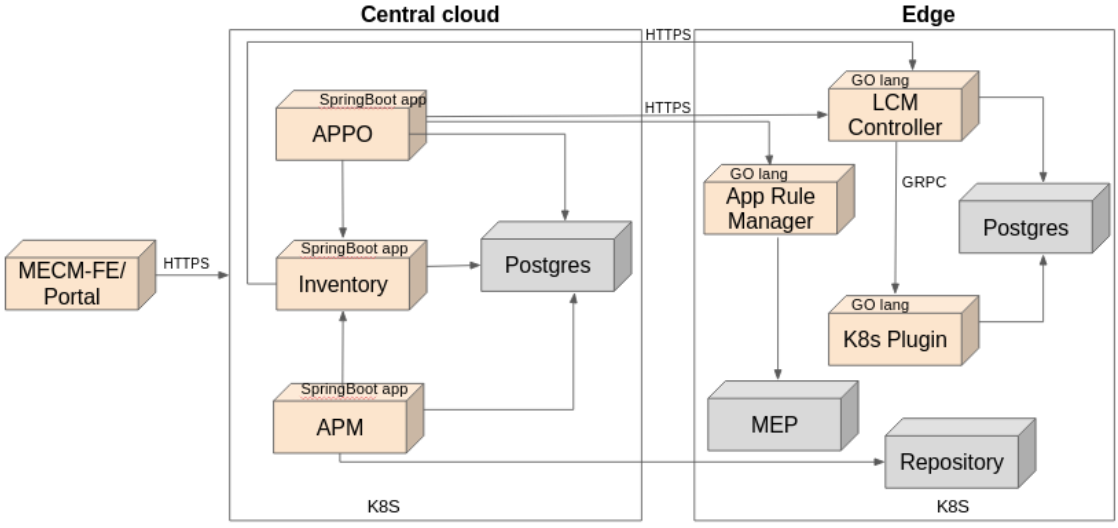
K8s Plugin

K8s Plugin is responsible for MEC application life cycle management operations on kubernetes infrastructure. K8s plugin use helm client to perform application life cycle management operations.

App Rule Manager

APP Rule Manager is responsible for configuring application rules by sending application rule configurations to MEP. Application rule configurations includes the traffic rules and DNS to be configured.

Deployment view



The deployment view shown is based on k8s deployment. MECM modules supports deployment on virtual machine or docker container to start respective service.

9.3 Interface Designs 界面设计

9.3.1 Application Package Manager Interface

The document is for the Application Package Manager project, there is one part of interfaces in the project.

Get All Application Packages

Retrieves all application packages

```
Resource URI: /apm/v1/tenants/{tenant_id}/packages
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
tenant_id	tenant id	path	yes	Valid UUID	64

Example response:

```
200 OK
[
  {
      "appIconUrl": "string",
      "appId": "string",
      "appPkgAffinity": "string",
      "appPkgDesc": "string",
      "appPkgId": "string",
      "appPkgName": "string",
      "appPkgPath": "string",
      "appPkgVersion": "string",
      "appProvider": "string",
      "createdTime": "string",
      "mecHostInfo": [
        {
          "error": "string",
          "hostIp": "string",
          "status": "string"
        }
      ],
      "modifiedTime": "string"
```

(下页继续)

(续上页)

```
    }
]
```

Onboard Application Package

Onboard application package

```
Resource URI: /apm/v1/tenants/{tenant_id}/packages
Method: POST
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access_token	header	yes	Jwt token	
tenant_id	tenant id	path	yes	Valid UUID	64
appIconUrl	appIconUrl	body	yes	Valid URL	1024
appId	appId	body	yes	AlphaNumeric with smaller case alphabets	32
appPkgAffinity	appPkgAffinity	body	yes	AlhpaNumeric	255
appPkgDesc	appPkgDesc	body	yes	AlhpaNumeric	1024
appPkgId	appPkgId	body	yes	AlphaNumeric with smaller case alphabets	32
appPkgName	appPkgName	body	yes	AlphaNumeric and allowed special characters hypen and underscore	255
appPkgPath	appPkgPath	body	yes	Valid URL	1024
appPkgVersion	appPkgVersion	body	yes	Valid URL	1024
appProvider	appProvider	body	yes	AlphaNumeric	1024
createdTime	createdTime	body	yes	Valid time	255
mecHostInfo	mecHostInfo	body	yes	Valid URL	15

Example request body:

```
{
  "appIconUrl": "string",
  "appId": "string",
  "appPkgAffinity": "string",
  "appPkgDesc": "string",
  "appPkgId": "string",
  "appPkgName": "string",
  "appPkgPath": "string",
  "appPkgVersion": "string",
  "appProvider": "string",
  "createdTime": "string",
  "mecHostInfo": [
    {
      "error": "string",
      "hostIp": "string",
      "status": "string"
    }
  ],
  "modifiedTime": "string"
}
```

Example response:

```
200 OK
[
  {
     "packageId": "string"
  }
]
```

Get Application Package Information

Retrieves application package information.

```
Resource URI: /apm/v1/tenants/{tenant_id}/packages/{app_package_id}
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
app_package_id	app package id	path	yes	Alphanumeric and must be smaller case	32
tenant_id	tenant id	path	yes	Valid UUID	64

Example response:

```
200 OK
  {
    "appIconUrl": "string",
    "appId": "string",
    "appPkgAffinity": "string",
    "appPkgDesc": "string",
    "appPkgId": "string",
    "appPkgName": "string",
    "appPkgPath": "string",
    "appPkgVersion": "string",
    "appProvider": "string",
    "createdTime": "string",
    "mecHostInfo": [
      {
        "error": "string",
        "hostIp": "string",
        "status": "string"
      }
    ],
    "modifiedTime": "string"
  }
```

Get Application template

Retrieves application template.

```
Resource URI: /apm/v1/tenants/{tenant_id}/packages/{app_package_id}/
↪apptemplate
Method: GET
```

Name	Definition	Type	Re-quired	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
app_package_id	app package id	path	yes	Alphanumeric and must be smaller case	32
tenant_id	tenant id	path	yes	Valid UUID	64

Example response:

```
200 OK
  {
    "appPkgName": "positioning_service",
    "version": "1.0",
    "appId": "efa14bd4327c43098343402246619ff3",
    "appPackageId": "9022daf9de964391ae29907221ef95a1",
    "inputs": [
        {
            "name": "APP_Name",
            "type": "string",
            "defaultValue": null,
            "description": "APP_Name"
        },
        {
            "name": "network_name_1",
            "type": "string",
            "defaultValue": "fst01_OM_Plane",
            "description": "APP_NETWORK"
        }
    ]
}
```

Delete Application Packages

Deletes application package

```
Resource URI: /apm/v1/tenants/{tenant_id}/packages/{app_package_id}
Method: DELETE
```

Name	Definition	Type	Re-quired	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
app_package_id	app package id	path	yes	Alphanumeric and must be smaller case	32
tenant_id	tenant id	path	yes	Valid UUID	64
appId	app id	path	yes	Alphanumeric and must be smaller case	32

Example response:

```
200 OK
{
 "success"
}
```

Download Csar

Download application package CSAR

```
Resource URI: /apm/v1/tenants/{tenant_id}/packages/{app_package_id}/download
Method: GET
```

Name	Definition	Type	Re-quired	Allowed	Max Length
app_package_id	app package id	path	yes	Alphanumeric and must be smaller case	32
tenant_id	tenant id	path	yes	Valid UUID	64

Example response:

```
200 OK
  {
```

(下页继续)

(续上页)


```
  "description": "string",
  "file": {
    "absolute": true,
    "absolutePath": "string",
    "canonicalPath": "string",
    "directory": true,
    "file": true,
    "freeSpace": 0,
    "hidden": true,
    "name": "string",
    "parent": "string",
    "path": "string",
    "totalSpace": 0,
    "usableSpace": 0
  },
  "filename": "string",
  "inputStream": {},
  "open": true,
  "readable": true,
  "uri": {
    "absolute": true,
    "authority": "string",
    "fragment": "string",
    "host": "string",
    "opaque": true,
    "path": "string",
    "port": 0,
    "query": "string",
    "rawAuthority": "string",
    "rawFragment": "string",
    "rawPath": "string",
    "rawQuery": "string",
    "rawSchemeSpecificPart": "string",
    "rawUserInfo": "string",
    "scheme": "string",
    "schemeSpecificPart": "string",
    "userInfo": "string"
  },
  "url": {
    "authority": "string",
    "content": {},
    "defaultPort": 0,
    "file": "string",
```


(下页继续)

(续上页)

```
      "host": "string",
      "path": "string",
      "port": 0,
      "protocol": "string",
      "query": "string",
      "ref": "string",
      "userInfo": "string"
    }
  }
```

Delete Application Package

Deletes an application packages

```
Resource URI: /apm/v1/tenants/{tenant_id}/packages/{app_package_id}/hosts/
↪{host_ip}
Method: DELETE
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
app_package_id	app package id	path	yes	Alphanumeric and must be smaller case	32
host_ip	host ip	path	yes	Valid IP address	15
tenant_id	tenant id	path	yes	Valid UUID	64

Example response:

```
200 OK
{
 "success"
}
```

On-boards application with package.

On-boards application with package provided.

```
Resource URI: /apm/v1/tenants/{tenant_id}/packages/upload
Method: POST
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access_token	header	yes	Jwt token	
appPkgName	appPkgName	body	yes	AlphaNumeric and allowed special characters hypen and underscore	255
appPkgVersion	appPkgVersion	body	yes	Valid URL	1024
File	file	formData	No		
hostList	host list	formData	yes		
tenant_id	tenant id	path	no	Valid UUID	64

Example request body:

```
{
  "access_token": "string",
  "appPackageName": "string",
  "appPackageVersion": "string",
  "file": "file",
  "hostList": "string",
  "tenant_id": "string",
}
```

Example response:

```
202 Accepted
{
  "appId": "string",
  "appPackageId": "string"
}
```

Queries liveness & readiness

Retrieves data for liveness & readiness

```
Resource URI: /apm/v1/health
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	

Example response:

```
200 OK
```

Retrieves all application packages info from app store.

Retrieves all application packages info from app store.

```
Resource URI: /apm/v1/apps/info/appstores/{appstore_ip}
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
appstore_ip	appstore_ip	body	yes	AlphaNumeric with smaller case alphabets	32

Example response:

```
200 OK
```

Sync application packages

Sync application package by downloading package from appstore.

```
Resource URI: /apm/v1/apps/sync
Method: POST
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access_token	header	yes	Jwt token	
appId	appId	body	yes	AlphaNumeric with smaller case alphabets	32
packageId	packageId	body	yes	AlphaNumeric with smaller case alphabets	32

Example request body:

```
[
    {
        "appId": "string",
        "appstoreIp": "string",
        "packageId": "string"
    },
    {
        "appId": "string",
        "appstoreIp": "string",
        "packageId": "string"
    }
]
```

Example response:

```
200 OK
[
  {
     "packageId": "string"
  }
]
```

Retrieves all application packages sync status

Retrieves all application packages sync status.

```
Resource URI: /apm/v1/apps/syncstatus
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	

Example response:

```
200 OK
```

Get Application Package Information

Retrieve application packages sync status.

```
Resource URI: /apm/v1/apps/{app_id}/packages/{package_id}/syncstatus
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
appId	appId	body	yes	AlphaNumeric with smaller case alphabets	32
packageId	packageId	body	yes	AlphaNumeric with smaller case alpha	

Example response:

```
200 OK
{
 "appId": "ad dolore laborum",
 "name": "Excep",
 "operationalInfo": "est magna",
 "packageId": "sit n",
 "syncStatus": "cillum exercitation quis"
}
```

Get Application Package Information

Retrieve application packages sync status.

```
Resource URI: /apm/v1/tenants/{tenant_id}/app_package_infos/sync
Method: POST
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
tenant_id	tenant id	path	yes	Valid UUID	64

Example response:

```
200 OK
{
    "appPackageRecord": [
        {
            "appPkgId":
↪"8e3d6f90bad74cecb7544fc5ebf56782a5df67ab47ba40e09015136573d7bce439937079-
↪99fe-4cd8-881f-04ca8c4fe09d",
            "appPkgName": "Video_Surveillance_ap",
            "appPkgVersion": "2",
            "appPkgPath": "abc",
            "appProvider": "",
            "appPkgDesc": "abc",
            "appPkgAffinity": "bangalore",
            "appIconUrl": "",
            "createdTime": "2021-03-12 07:59:55.695956+00",
            "modifiedTime": "",
            "appId": "1",
            "tenantId": "e921ce54-82c8-4532-b5c6-8516cf75f7a7",
            "packageId": "1",
            "origin": "MEPM",
            "syncStatus": false,
            "mecHostInfo": [
                {
                    "pkgHostKey":
↪"8e3d6f90bad74cecb7544fc5ebf56782a5df67ab47ba40e09015136573d7bce439937079-
↪99fe-4cd8-881f-04ca8c4fe09d119.8.47.5",
                    "hostIp": "119.8.47.5",
                    "appPkgId":
↪"8e3d6f90bad74cecb7544fc5ebf56782a5df67ab47ba40e09015136573d7bce4",
                    "status": "",
```

(下页继续)

(续上页)

```
                    "tenantId": "e921ce54-82c8-4532-b5c6-8516cf75f7a7",
                    "error": "success",
                    "origin": "MEPM",
                    "syncStatus": false
                }
            ]
        }
    ]
}
```

APM Open Api Swagger

- Please refer APM_OPEN_API

9.3.2 Applcm Interface

The document is for the Applcm project, there is one part of interfaces in the project.

Upload Config File

Upload Config File

```
Resource URI: /lcmcontroller/v2/configuration
Method: POST
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt Token	
hostIp	host Ip	string	yes	Valid IP Address	15
configFile	configFile	formData	yes		

Example response:

```
200 OK
[
  "response": {}
]
```

```
201 Created
[
  "response": {}
]
```

Delete Config File

Removes the config file

```
Resource URI: /lcmcontroller/v2/configuration
Method: DELETE
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access_token	header	yes	Jwt Token	
hostIp	host Ip	string	yes	Valid IP Address	15

Example response:

```
200 OK
```

Instantiate Application

Application instantiated

```
Resource URI: /lcmcontroller/v2/app_instance/{appInstanceId}/instantiate
Method: POST
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt Token	
appInstanceId	appInstance Id	path	yes	Valid UUID	64
tenantId	tenant Id	path	yes	Valid UUID	64
hostIp	host Ip	string	yes	Valid IP	15
appName	appName	string	yes	AlphaNumeric	128
parameters	key value pair	map	no	valid string	

Example request body:

```
{
    "hostIp": "119.8.47.5",
    "packageId":
↪"ea3b8191f8a84565bd8eb5933c8bbcd560e4f65665a84cba851c93ece4fd825d",
    "appName": "testApplication",
    "parameters": {
            "mepIp": "MEP IP",
            "mepPort": "MEP port",
            "ak": "access key",
            "sk": "secure key",
            "...": "..."
    }
}
```

Example response:

```
200 OK
  {
    {}
  }
```

Terminates Application

Terminates application instance id

```
Resource URI: /lcmcontroller/v2/tenants/{tenantId}/app_instances/
↪{appInstanceId}/terminate
Method: POST
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt Token	
appInstanceId	appInstance Id	path	yes	Valid UUID	64
tenantId	tenant Id	path	yes	Valid UUID	64

Example response:

```
200 OK
```

Query

Get the statistics information

```
Resource URI: /lcmcontroller/v2/tenants/{tenantId}/app_instances/
↪{appInstanceId}
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt Token	
appInstanceId	appInstance Id	path	yes	Valid UUID	64
tenantId	tenant Id	path	yes	Valid UUID	64

Example response:

```
200 OK
  {
    {}
  }
```

Query Kpi

Get cpu , memory and file system size

```
Resource URI: /lcmcontroller/v2/tenants/{tenantId}/hosts/{hostIp}/kpi
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt Token	
hostIp	host Ip	path	yes	Valid IP Address	15
tenantId	tenant Id	path	yes	Valid UUID	64

Example response:

```
200 OK
  {
    {}
  }
```

Query MepCapabilities

Get Mep Capabilities

```
Resource URI: /lcmcontroller/v2/tenants/{tenantId}/hosts/{hostIp}/mep_
↪capabilities
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt Token	
hostIp	host Ip	path	yes	Valid IP Address	15
tenantId	tenant Id	path	yes	Valid UUID	64

Example response:

```
200 OK
    [
     {
       "capabilityId": "string",
       "capabilityName": "string",
       "status": "string",
       "version": "string",
        "consumers": [
          {
            "applicationInstanceId": "string"
          },
          {
            "applicationInstanceId": "string"
          },
        ]
     },
     {
       "capabilityId": "string",
       "capabilityName": "string",
       "status": "string",
       "version": "string",
        "consumers": [
          {
            "applicationInstanceId": "string"
          }
        ]
     }
    ]
```

Get Mep Capability

Retrieves edge host performance statistics based on capability id

```
Resource URI: /lcmcontroller/v2/tenants/{tenantId}/hosts/{hostIp}/mep_
↪capabilities/{capability_id}
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
host_ip	edge host ip	path	yes	Valid IP	15
tenant_id	tenant_id	path	yes	Valid UUID	64
capability_id	capability_id	path	yes	Alphanumeric characters,special characters are hypen and underscore	128

Example response:

```
200 OK
  [
   {
     "capabilityId": "string",
     "capabilityName": "string",
     "status": "string",
     "version": "string",
      "consumers": [
        {
          "applicationInstanceId": "string"
        },
        {
          "applicationInstanceId": "string"
        },
      ]
   }
  ]
```

Queries liveness & readiness

Retrieves data for liveness & readiness

```
Resource URI: /lcmcontroller/v1/health
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	

Example response:

```
200 OK
```

App Deployment Status

Retrieves status of the application deployment

```
Resource URI: /lcmcontroller/v1/hosts/{hostIp}/packages/{packageId}/status
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
host_ip	edge host ip	path	yes	Valid IP	15
packageId	package Id	path	yes	Valid UUID without hypen	64

Example response:

```
200 OK
{
  true
}
```

Query workload

GetWorkloadDescription

```
Resource URI: /lcmcontroller/v2/tenants/:tenantId/app_instances/
↪:appInstanceId/workload/events
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt Token	
appInstanceId	appInstance Id	path	yes	Valid UUID	64
tenantId	tenant Id	path	yes	Valid UUID	64

Example response:

```
200 OK
  {
    {}
  }
```

Query AppInstance information

AppInstance information

```
Resource URI: /lcmcontroller/v1/tenants/:tenantId/app_instances
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt Token	
tenantId	tenant Id	path	yes	Valid UUID	64

Example response:

```
200 OK
  {
    {}
  }
```

Query app instances records

Sync app instances records

```
Resource URI: /lcmcontroller/v2/tenants/:tenantId/app_instances/sync_updated
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt Token	
tenantId	tenant Id	path	yes	Valid UUID	64

Example response:

```
200 OK
  {
    {}
  }
```

Query stale records

Sync app instances stale records

```
Resource URI: /lcmcontroller/v2/tenants/:tenantId/app_instances/sync_deleted
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt Token	
tenantId	tenant Id	path	yes	Valid UUID	64

Example response:

```
200 OK
  {
    {}
  }
```

Upload package

Upload package

```
Resource URI: /lcmcontroller/v2/tenants/:tenantId/packages
Method: POST
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt Token	
packageId	package Id	header	yes	Valid UUID without hypen	64
appId	app Id	header	yes	Valid UUID without hypen	64
tenantId	tenant Id	path	yes	Valid UUID	64
packages	package file	formData	yes	Valid UUID	64

Example response:

```
200 OK
```

Delete package

Delete package

```
Resource URI: /lcmcontroller/v2/tenants/:tenantId/packages/:packageId
Method: DELETE
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access_token	header	yes	Jwt Token	
tenantId	tenant Id	path	yes	Valid UUID	64
packageId	package Id	path	yes	Valid UUID without hypen	64

Example response:

```
200 OK
```

Delete application package on host

Delete application package on host

```
Resource URI: /lcmcontroller/v2/tenants/:tenantId/packages/:packageId/hosts/
↪:hostIp
Method: DELETE
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access_token	header	yes	Jwt Token	
tenantId	tenant Id	path	yes	Valid UUID	64
packageId	package Id	path	yes	Valid UUID without hypen	64
host_ip	host ip	path	yes	Valid IP	15

Example response:

```
200 OK
```

Distribute package

Distribute package

```
Resource URI: /lcmcontroller/v2/tenants/:tenantId/packages/:packageId
Method: POST
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt Token	
packageId	package Id	header	yes	Valid UUID without hypen	64
packages	package file	formData	yes	Valid UUID	64

Example response:

```
200 OK
```

Query

Distribution status

```
Resource URI: /lcmcontroller/v2/tenants/:tenantId/packages/:packageId
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt Token	
tenantId	tenant Id	path	yes	Valid UUID	64
packageId	package Id	header	yes	Valid UUID without hypen	64

Example response:

```
200 OK
  {
    {}
  }
```

Distribution status

Distribution status

```
Resource URI: /lcmcontroller/v1/tenants/:tenantId/packages
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt Token	
tenantId	tenant Id	path	yes	Valid UUID	64

Example response:

```
200 OK
  {
    {}
  }
```

Sync app package records

Sync app package records

```
Resource URI: /lcmcontroller/v2/tenants/:tenantId/packages/sync_updated
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt Token	
tenantId	tenant Id	path	yes	Valid UUID	64

Example response:

```
200 OK
  {
    {}
  }
```

Sync app package stale records

Sync app package stale records

```
Resource URI: /lcmcontroller/v2/tenants/:tenantId/packages/sync_deleted
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt Token	
tenantId	tenant Id	path	yes	Valid UUID	64

Example response:

```
200 OK
  {
    {}
  }
```

Add MEC host

```
Resource URI: /lcmcontroller/v1//hosts
Method: POST
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt Token	
mecHostInfo	MecHostInfo	body	yes	Valid UUID	64

Example response:

```
200 OK
```

Update MEC host

```
Resource URI: /lcmcontroller/v1//hosts
Method: PUT
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt Token	
mecHostInfo	MecHostInfo	body	yes	Valid UUID	64

Example response:

```
200 OK
```

Query MEC hosts

```
Resource URI: /lcmcontroller/v1/hosts
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt Token	

Example response:

```
200 OK
  {
    {}
  }
```

Delete MEC host

```
Resource URI: /lcmcontroller/v1/hosts:hostIp
Method: DELETE
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access_token	header	yes	Jwt Token	
host_ip	host ip	path	yes	Valid IP	15

Example response:

```
200 OK
```

Batch terminate application

```
Resource URI: /lcmcontroller/v1/tenants/:tenantId/app_instances/
↪batchTerminate
Method: DELETE
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access_token	header	yes	Jwt Token	
tenantId	tenant Id	path	yes	Valid UUID	64

Example response:

```
200 OK
```

Sync mec host records

```
Resource URI: /lcmcontroller/v1/hosts/sync_updated
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt Token	

Example response:

```
200 OK
  {
    {}
  }
```

Sync mec host stale records

```
Resource URI: /lcmcontroller/v1/hosts/sync_deleted
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt Token	

Example response:

```
200 OK
  {
    {}
  }
```

Create Image

```
Resource URI: /lcmcontroller/v1/tenants/:tenantId/app_instances/
↪:appInstanceId/images
Method: POST
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt Token	
appInstanceId	appInstance Id	path	yes	Valid UUID	64
tenantId	tenant Id	path	yes	Valid UUID	64
vmId	vm Id	body	yes	Valid UUID	64

Example response:

```
200 OK
  {
    {}
  }
```

Delete Image

```
Resource URI: /lcmcontroller/v1/tenants/:tenantId/app_instances/
↪:appInstanceId/images/:imageId
Method: DELETE
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt Token	
appInstanceId	appInstance Id	path	yes	Valid UUID	64
tenantId	tenant Id	path	yes	Valid UUID	64
imageId	image Id	path	yes	Valid UUID	64

Example response:

```
200 OK
  {
    {}
  }
```

Get Image

```
Resource URI: /lcmcontroller/v1/tenants/:tenantId/app_instances/
↪:appInstanceId/images/:imageId
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt Token	
appInstanceId	appInstance Id	path	yes	Valid UUID	64
tenantId	tenant Id	path	yes	Valid UUID	64
imageId	image Id	path	yes	Valid UUID	64

Example response:

```
200 OK
  {
    {}
  }
```

Get Image file

```
Resource URI: /lcmcontroller/v1/tenants/:tenantId/app_instances/
↪:appInstanceId/images/:imageId/file
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt Token	
appInstanceId	appInstance Id	path	yes	Valid UUID	64
tenantId	tenant Id	path	yes	Valid UUID	64
imageId	image Id	path	yes	Valid UUID	64
chunk_num	chunk num	header	yes	Valid int	64

Example response:

```
200 OK
  {
    {}
  }
```

Get Services

```
Resource URI: /lcmcontroller/v1/mep/subscribe_statistic
Method: GET
```

Example response:

```
200 OK
  [{
        "serInstanceId": "xxx",
        "serName": "xxx",
        "serCategory": {
                "href": "/example/catalogue1",
                "id": "id12345",
                "name": "RNI",
                "version": "1.2.3"
        },
        "version": "4.5.8",
        "state": "INACTIVE",
        "transportId": "Rest1",
        "transportInfo": {
                "id": "TransId12345",
                "name": "REST",
                "description": "REST API",
                "type": "REST_HTTP",
                "protocol": "HTTP",
                "version": "2.0",
                "endpoint": {
                        "uris": null,
                        "addresses": null,
                        "alternative": null
                },
                "security": {
                        "oAuth2Info": {
                                "grantTypes": ["OAUTH2_CLIENT_CREDENTIALS"],
                                "tokenEndpoint": "/mecSerMgmtApi/security/
↪TokenEndPoint"
                        }
                }
        },
        "serializer": "JSON",
        "scopeOfLocality": "MEC_SYSTEM",
        "isLocal": true,
        "livenessInterval": 60,
```

(下页继续)

(续上页)

```
        "_links": {
                "self": {
                        "liveness": "/mepserver/mec_service_mgmt/v1/
↪applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/services/
↪31fe525ee3dbccbf25a234f8e81d696c/liveness"
                },
                "appInstanceId": "5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f"
        }
}]
```

Get Kong Logs

```
Resource URI: /lcmcontroller/v1/mep/kong_log
Method: GET
```

```
200 OK
{
        "appServices": [{
                "callTimes": [0, 0, 0, 0, 0, 0, 0],
                "desc": "",
                "name": "hehe5"
        }],
        "mepServices": [{
                "callTimes": [0, 0, 0, 3, 0, 0, 0],
                "desc": "",
                "name": "serviceRegister"
        }]
}
```

Get Subscribe Statistic

```
Resource URI: /lcmcontroller/v1/mep/subscribe_statistic
Method: GET
```

```
200 OK
{
        "subscribeNum": {
                "appSubscribeNum": 0,
                "serviceSubscribedNum": 0
        },
```

(下页继续)

(续上页)

```
        "subscribeRelations": []
}
```

9.3.3 Application Orchestrator Interfaces

The document is for the Application Orchestrator project, there are two parts of interfaces in the project.

Get All Application Instances

Retrieves application instances information

```
Resource URI: /appo/v1/tenants/{tenant_id}/app_instance_infos
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access_token	header	yes	Jwt token	
tenant_id	tenant id	path	yes	Valid UUID	64

Example response:

```
200 OK
[
  {
     "appInstanceId": "string",
     "appPackageId": "string",
     "appName": "string",
     "appId": "string",
     "appDescriptor": "string",
     "mecHost": "string",
     "applcmHost": "string",
     "operationalStatus": "string",
     "operationInfo":"string"
  }
]
```

Get Application Instance

Retrieves application instance information

```
Resource URI: /appo/v1/tenants/{tenant_id}/app_instance_infos/{appInstance_
↪id}
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access_token	header	yes	Jwt token	
appInstance_id	application instance id	path	yes	Valid UUID	64
tenant_id	tenant id	path	yes	Valid UUID	64

Example response:

```
200 OK
  {
     "appInstanceId": "string",
     "appPackageId": "string",
     "appName": "string",
     "appId": "string",
     "appDescriptor": "string",
     "mecHost": "string",
     "applcmHost": "string",
     "operationalStatus": "string",
     "operationInfo":"string"
  }
```

Create AppInstance

Creates App Instance

```
Resource URI: /appo/v1/tenants/{tenant_id}/app_instances
Method: POST
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
appInstanceDescription	appInstanceDescription	RequestPart	yes	Valid UUID	64
appName	appName	RequestPart	yes	Alphanumeric characters,special characters are hypen and underscore	128
appPackageId	appPackageId	RequestPart	yes	Alphanumeric in lower case	64
appId	appId	RequestPart	yes	Alphanumeric in lower case	64
mecHost	mecHost	RequestPart	yes	Valid IP Address	15
hwcapabilities	hardwareCapabilities	body	No	AlphaNumeric characters allowed	128
tenant_id	tenant_id	path	yes	Valid UUID	64

Example request body:

```
{
  "appInstanceDescription": "string",
  "appName": "string",
  "appPackageId": "string",
  "appId": "string",
  "hwCapabilities": [
      "string"
    ],
  "mecHost": "string"
}
```

Example response:

```
202 Request Accepted
[
  {
```

(下页继续)

(续上页)

```
    "app_instance_id":"string"
  }
]
```

Get App Instance Information

Retrieves application instance information

```
Resource URI: /appo/v1/tenants/{tenant_id}/app_instances/{app_instance_id}
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
app_instance_id	application instance id	header	yes	Valid UUID	64
tenant_id	tenant id	path	yes	Valid UUID	64

Example response:

```
200 OK
  {
     "appInstanceId": "string",
     "appPackageId": "string",
     "appName": "string",
     "appId": "string",
     "appDescriptor": "string",
     "mecHost": "string",
     "applcmHost": "string",
     "operationalStatus": "string",
     "operationInfo":"string"
  }
```

Instantiate Application

Instantiate application instance.

```
Resource URI: /appo/v1/tenants/{tenant_id}/app_instances/{app_instance_id}
Method: POST
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
app_instance_id	application instance id	path	yes	Valid UUID	64
tenant_id	tenant id	path	yes	Valid UUID	64
parameters	key value pair	body	No	Valid String	

Example request body:

```
{
    "parameters": {
        "mepIp"           : "mep ip",
        "mepPort"         : "mep port",
        "ak"              : "ak",
        "sk"              : "sk" ,
        "..."             : "..."
    }
}
```

Example response:

```
202 Request Accepted
  {
    "response": {}
  }
```

Delete Application Instance

Terminates application instance.

```
Resource URI: /appo/v1/tenants/{tenant_id}/app_instances/{app_instance_id}
Method: DELETE
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access_token	header	yes	Jwt token	
app_instance_id	application instance id	path	yes	Valid UUID	64
tenant_id	tenant id	path	yes	Valid UUID	64

Example response:

```
202 Request Accepted
  {
    "response": {}
  }
```

Get Query Kpi

Retrieves edge host performance statistics

```
Resource URI: /appo/v1/tenants/{tenant_id}/hosts/{host_ip}/kpi
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
host_ip	edge host ip	path	yes	Valid IP	15
tenant_id	tenant_id	path	yes	Valid UUID	64

Example response:

```
200 OK
  [
    "response": "{
    "pods ":[{
    "podstatus":"String","
    "podname":"String",
    "containers":[{
    "containername": "String",
    "metricsusage":{
    "cpuusage" : "String",
    "memusage" : "String",
    "diskusage" : "String"}}]}"
  ]
```

Get Mep Capabilities

Retrieves edge host performance statistics

```
Resource URI: /appo/v1/tenants/{tenant_id}/hosts/{host_ip}/mep_capabilities
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
host_ip	edge host ip	path	yes	Valid IP	15
tenant_id	tenant_id	path	yes	Valid UUID	64

Example response:

```
200 OK
  [
   {
     "capabilityId": "string",
     "capabilityName": "string",
     "status": "string",
     "version": "string",
      "consumers": [
        {
          "applicationInstanceId": "string"
        },
        {
          "applicationInstanceId": "string"
        },
      ]
   },
   {
     "capabilityId": "string",
     "capabilityName": "string",
     "status": "string",
     "version": "string",
      "consumers": [
        {
          "applicationInstanceId": "string"
        }
      ]
   }
  ]
```

Get Mep Capability

Retrieves edge host performance statistics based on capability id

```
Resource URI: /appo/v1/tenants/{tenant_id}/hosts/{host_ip}/mep_capabilities/
↪{capability_id}
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
host_ip	edge host ip	path	yes	Valid IP	15
tenant_id	tenant_id	path	yes	Valid UUID	64
capability_id	capability_id	path	yes	Alphanumeric characters,special characters are hypen and underscore	128

Example response:

```
200 OK
   {
     "capabilityId": "string",
     "capabilityName": "string",
     "status": "string",
     "version": "string",
      "consumers": [
        {
          "applicationInstanceId": "string"
        },
        {
          "applicationInstanceId": "string"
        },
      ]
   }
```

Create Batch

Batch create application instances.

```
Resource URI: /appo/v1/tenants/{tenant_id}/app_instances/batch_create
Method: POST
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
tenant_id	tenant id	path	yes	Valid UUID	64
appName	appName	Request-Part	yes	Alphanumeric characters,special characters are hypen and underscore	128
appInstanceDescription	appInstanceDescription	Request-Part	yes	Valid UUID	64
appPackageId	appPackageId	Request-Part	yes	Alphanumeric in lower case	64
appId	appId	Request-Part	yes	Alphanumeric in lower case	64
mecHost	mecHost	Request-Part	yes	Valid IP Address	15
hwcapabilities	hardwareCapabilities	body	No	AlphaNumeric characters allowed	128

Example request body:

```
{
  "appPackageId": "string",
  "appName": "string",
  "appInstanceDescription": "string",
  "appId": "string",
  "hwCapabilities": [
     "string"
   ],
   "mecHost": [
     "string"
```

(下页继续)

(续上页)

```
    ]
}
```

Example response:

```
202 Accepted
  {
    "response": [
            {
              "applicationInstanceId": "string",
              "host": "string",
              "status": "string"
            },
            {
              "applicationInstanceId": "string",
              "host": "string",
              "status": "string"
            }
          ]
  }
```

Batch instantiate

Batch instantiate application instances.

```
Resource URI: /appo/v1/tenants/{tenant_id}/app_instances/batch_instantiate
Method: POST
```

Name	Definition	Type	Re-quired	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
tenant_id	tenant id	path	yes	Valid UUID	64
instantiationParameters	array of key value pair	body	No	Valid string	

Example request body:

```
{
    "instantiationParameters": [
```

(下页继续)

(续上页)

```
        {
          "appInstanceId": "d2134a3b-85e7-4bf6-b030-14c71219188a",
          "parameters": {
                          "mepIp"          : "mep ip",
                          "mepPort"        : "mep port",
                          "ak"             : "ak",
                          "sk"             : "sk",
                          "..."            : "..."
          }
        }
    ]
}
```

Example response:

```
202 Accepted
  {
    "response": [
            {
              "applicationInstanceId": "string",
              "host": "string",
              "status": "string"
            },
            {
              "applicationInstanceId": "string",
              "host": "string",
              "status": "string"
            }
          ]
  }
```

Batch terminate

Batch terminates application instances.

```
Resource URI: /appo/v1/tenants/{tenant_id}/app_instances/batch_terminate
Method: POST
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
tenant_id	tenant id	path	yes	Valid UUID	64
appInstanceIds	application instance id	body	yes	Valid UUID	64

Example request body:

```
{
 "appInstanceIds" : ["string","string"]
}
```

Example response:

```
202 Accepted
  {
    "response": [
            {
              "applicationInstanceId": "string",
              "host": "string",
              "status": "string"
            },
            {
              "applicationInstanceId": "string",
              "host": "string",
              "status": "string"
            }
          ]
  }
```

Batch Query

Batch Query information

```
Resource URI: /appo/v1/tenants/{tenant_id}/app_instance_infos/{appInstance_
↪ids}
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access_token	header	yes	Jwt token	
appInstance_ids	application instance id	header	yes	Valid UUID	64
tenant_id	tenant id	path	yes	Valid UUID	64

Example response:

```
200 OK
  {
     "appInstanceId": "string",
     "appPackageId": "string",
     "appName": "string",
     "appId": "string",
     "appDescriptor": "string",
     "mecHost": "string",
     "applcmHost": "string",
     "operationalStatus": "string",
     "operationInfo":"string"
  }
```

Queries liveness & readiness

Retrieves data for liveness & readiness

```
Resource URI: /appo/v1/health
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	

Example response:

```
200 OK
```

Create AppRule Config

Configures the app rule

```
Resource URI:  /appo/v1/tenants/{tenantId}/app_instances/{appInstanceId}/
↪appd_configuration
Method: POST
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access_token	header	yes	Jwt token	
appInstance_id	application instance id	path	yes	Valid UUID	64
tenant_id	tenant id	path	yes	Valid UUID	64
appInstanceIds	application instance id	path	yes	Valid UUID	64

Example request body:

```
"appTrafficRule": [
   {
     "trafficRuleId": "string",
     "filterType": "string",
     "priority": int,
     "action": "DROP",
     "trafficFilter": [
       {
         "srcAddress": [
           "string"
         ],
         "dstAddress": [
           "string"
         ],
         "tag": [
            "string"
         ],
         "srcTunnelAddress": [
            "string"
         ],
         "dstTunnelAddress": [
            "string"
         ],
```

(下页继续)

(续上页)

```
          "srcTunnelPort": [
             "string"
          ],
           "dstTunnelPort": [
              "string"
          ],
          "dstPort": [
            "string"
          ],
          "protocol": [
            "string"
          ],
          "qCI": int,
          "dSCP": int,
          "tC": int
        }
      ],
      "action": "string",
      "state": "string"
    }
  ],
    "dstInterface": [
      {
       "interfaceType": "string",
       "tunnelInfo":
         {
           "tunnelType": "string",
           "tunnelDstAddress": "string",
           "tunnelSrcAddress": "string",
           "tunnelSpecificData": "string"
         },
           "srcMacAddress": "string",
           "dstMacAddress": "string",
           "dstIpAddress": "string"
           }
     ],
    "appDNSRule": [
     {
       "dnsRuleId": "string",
       "domainName": "string",
       "ipAddressType": "string",
       "ipAddress": "string",
       "ttl": int,
```

(下页继续)

(续上页)

```
    }
  ],
  "appSupportMp1": bool,
  "appName": "string"
```

Example response:

```
200 OK
{
  "response":
    {
       "apprule_task_id": "string"
    }
}
```

Update AppRule Config

Update the app rule

```
Resource URI:  /appo/v1/tenants/{tenantId}/app_instances/{appInstanceId}/
↪appd_configuration
Method: PUT
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access_token	header	yes	Jwt token	
appInstance_id	application instance id	path	yes	Valid UUID	64
tenant_id	tenant id	path	yes	Valid UUID	64
appInstanceIds	application instance id	path	yes	Valid UUID	64

Example request body:

```
 "appTrafficRule": [
    {
      "trafficRuleId": "string",
      "filterType": "string",
      "priority": int,
      "action": "DROP",
      "trafficFilter": [
        {
```

(下页继续)

(续上页)

```
          "srcAddress": [
            "string"
          ],
          "dstAddress": [
            "string"
          ],
          "tag": [
             "string"
          ],
          "srcTunnelAddress": [
             "string"
          ],
          "dstTunnelAddress": [
             "string"
          ],
          "srcTunnelPort": [
             "string"
          ],
           "dstTunnelPort": [
              "string"
          ],
          "dstPort": [
            "string"
          ],
          "protocol": [
            "string"
          ],
          "qCI": int,
          "dSCP": int,
          "tC": int
        }
      ],
      "action": "string",
      "state": "string"
    }
  ],
    "dstInterface": [
      {
       "interfaceType": "string",
       "tunnelInfo":
         {
           "tunnelType": "string",
           "tunnelDstAddress": "string",
```

(下页继续)

(续上页)

```
            "tunnelSrcAddress": "string",
            "tunnelSpecificData": "string"
          },
            "srcMacAddress": "string",
            "dstMacAddress": "string",
            "dstIpAddress": "string"
            }
      ],
     "appDNSRule": [
      {
        "dnsRuleId": "string",
        "domainName": "string",
        "ipAddressType": "string",
        "ipAddress": "string",
        "ttl": int,
      }
  ],
  "appSupportMp1": bool,
  "appName": "string"
```

Example response:

```
202 OK
{
  "response":
    {
       "apprule_task_id": "string"
    }
}
```

Delete AppRule Config

Deletes the existing app rule

```
Resource URI:  /appo/v1/tenants/{tenantId}/app_instances/{appInstanceId}/
↪appd_configuration
Method: DELETE
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access_token	header	yes	Jwt token	
appInstance_id	application instance id	path	yes	Valid UUID	64
tenant_id	tenant id	path	yes	Valid UUID	64
appInstanceIds	application instance id	path	yes	Valid UUID	64

Example request body:

```
{
  "appTrafficRule": ["string", "string"],
  "appDNSRule": ["string", "string"]
}
```

Example response:

```
202 Accepted
{
  "response":
    {
       "apprule_task_id": "string"
    }
}
```

Get AppRule Config Status

Queries app rule config status

```
Resource URI: /appo/v1/tenants/{{tenant-id}}/apprule_task_infos/{apprule_
↪task_id}
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access_token	header	yes	Jwt token	
appInstance_id	application instance id	path	yes	Valid UUID	64
tenant_id	tenant id	path	yes	Valid UUID	64
appInstanceIds	application instance id	path	yes	Valid UUID	64

Example Response body:

```
200 OK

{
  "response":
    {
        "taskId": "string",
        "appInstanceId": "string",
        "detailed": "string",
        "configResult": "string"
    }
}
```

Synchronizes application instance info from edge

Synchronizes application instance info from all edge

```
Resource URI: /appo/v1/tenants/{{tenant-id}}/app_instance_infos/sync
Method: POST
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access_token	header	yes	Jwt token	
tenant_id	tenant id	path	yes	Valid UUID	64

Example Response body:

```
200 OK

{
    "appInstanceUpdatedRecs": [
        {
            "appInstanceId": "e921ce54-82c8-4532-b5c6-8516cf75f7a6",
            "createTime": "2021-03-12T07:59:55.695956Z",
            "mecHost": "119.8.53.3",
            "deployType": "helm",
            "tenantId": "e921ce54-82c8-4532-b5c6-8516cf75f7a6",
            "appPackageId": "",
            "appName": "",
            "origin": "mepm",
            "syncStatus": false
        }
```

(下页继续)

(续上页)

```
    ]
}
```

APPO Open Api Swagger

- Please refer APPO_OPEN_API

9.3.4 Inventory Interfaces

The document is for the Inventory project, there are three parts of interfaces in the project.

Get All Application Records

Retrieves all application store records

```
Resource URI: /inventory/v1/appstores
Method: GET
```

Example response:

```
200 OK
  {
    {
      "appstoreIp": "string",
      "appstoreName": "string",
      "appstorePort": "string",
      "producer": "string",
      "uri": "string",
      "userName": "string"
    }
  }
```

Add Application Store Record

Adds new application store record

```
Resource URI: /inventory/v1/appstores
Method: POST
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
appstoreIp	appstoreIp	body	yes	Valid IP Address	15
appstoreName	appstoreName	body	yes	AlphaNumeric, special characters allowed are hypen and underscore	128
appstorePort	appstorePort	body	yes	Valid Port	5
producer	producer	body	yes	AlphaNumeric, special characters allowed are hypen and underscore	128
uri	uri	body	yes	Valid UUID	128
userName	userName	body	yes	AlphaNumeric, special characters allowed are hypen and underscore	128

Example request body:

```
{
  "appstoreIp": "string",
  "appstoreName": "string",
  "appstorePort": "string",
  "producer": "string",
  "uri": "string",
  "userName": "string"
}
```

Example response:

```
200 OK
[
  {
    "Saved"
  }
]
```

Delete All Application Records

Deletes all application stores records

```
Resource URI: /inventory/v1/appstores
Method: DELETE
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	

Example response:

```
200 OK
[
  {
    "Deleted"
  }
]
```

Get Application Stores Record

Retrieves application store record

```
Resource URI: /inventory/v1/appstores/{appstore_ip}
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access_token	header	yes	Jwt token	
appstore_ip	appstore IP	path	yes	Valid IP Address	15

Example response:

```
200 OK
  {
    "appstoreIp" : "string"
    "appstorePort" : "string"
    "uri" : "string"
    "userName" : "string"
    "appstoreName" : "string"
```

(下页继续)

(续上页)

```
  "producer" : "string"
}
```

Add Application record

Adds a new application store record entry into the Inventory.

```
Resource URI: /inventory/v1/appstores
Method: POST
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
appstoreIp	appstoreIp	body	yes	Valid IP Address	15
appstoreName	appstoreName	body	yes	AlphaNumeric, special characters allowed are hypen and underscore	128
appstorePort	appstorePort	body	yes	Valid Port	5
appstoreRepoName	appstoreRepoName	body	yes	Valid appstoreRepoName	128
appstoreRepo	appstoreRepo	body	yes	Valid appstoreRepo	128
appstoreRepoUserName	appstoreRepoUserName	body	yes	Valid appstoreRepoUserName	128
appstoreRepoPassword	appstoreRepoPassword	body	yes	Valid appstoreRepoPassword	15
producer	producer	body	yes	AlphaNumeric, special characters allowed are hypen and underscore	128

Example request body:

```
{
  "appstoreIp" : "string"
  "appstorePort" : "string"
  "appstoreName" : "string"
  "appstoreRepoName" : "string"
  "appstoreRepo" : "string"
  "appstoreRepoUserName" : "string"
  "appstoreRepoPassword" : "string"
  "producer" : "string"
}
```

Example response:

```
200 OK
[
  {
    "Saved"
  }
]
```

Update Application record

Updates an exiting application store record in the Inventory matching the given tenant ID & application store IP

```
Resource URI: /inventory/v1/appstores/{appstore_ip}
Method: PUT
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
appstoreIp	appstoreIp	body	yes	Valid IP Address	15
appstoreName	appstoreName	body	yes	AlphaNumeric, special characters allowed are hypen and underscore	128
appstorePort	appstorePort	body	yes	Valid Port	5
appstoreRepoName	appstoreRepoName	body	yes	Valid appstoreRepoName	128
appstoreRepo	appstoreRepo	body	yes	Valid appstoreRepo	128
appstoreRepoUserName	appstoreRepoUserName	body	yes	Valid appstoreRepoUserName	128
appstoreRepoPassword	appstoreRepoPassword	body	yes	Valid appstoreRepoPassword	15
producer	producer	body	yes	AlphaNumeric, special characters allowed are hypen and underscore	128

Example request body:

```
  {
    "appstoreIp" : "string"
    "appstorePort" : "string"
    "appstoreName" : "string"
```

(下页继续)

(续上页)

```
    "appstoreRepoName" : "string"
    "appstoreRepo" : "string"
    "appstoreRepoUserName" : "string"
    "appstoreRepoPassword" : "string"
    "producer" : "string"
  }
```

Example response:

```
200 OK
[
  {
    "Updated"
  }
]
```

Get App Stores Record

Retrieves all application store record

```
Resource URI: /inventory/v1/appstores
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access_token	header	yes	Jwt token	

Example response:

```
200 OK
[
  {
    "appstoreIp" : "string"
    "appstorePort" : "string"
    "appstoreName" : "string"
    "appstoreRepoName" : "string"
    "appstoreRepo" : "string"
    "appstoreRepoUserName" : "string"
    "appstoreRepoPassword" : "string"
    "producer" : "string"
  },
  {
```

(下页继续)

(续上页)

```
    "appstoreIp" : "string"
    "appstorePort" : "string"
    "appstoreName" : "string"
    "appstoreRepoName" : "string"
    "appstoreRepo" : "string"
    "appstoreRepoUserName" : "string"
    "appstoreRepoPassword" : "string"
    "producer" : "string"
  }
]
```

Delete App Stores Record

Deletes all application store records for a given tenant.

```
Resource URI: /inventory/v1/appstores/
Method: DELETE
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	

Example response:

```
200 OK
[
  {
    "Deleted"
  }
]
```

Get App Stores Record

Retrieves a specific application store record in the Inventory matching the given tenant ID & application store IP

```
Resource URI: /inventory/v1/appstores/{appstore_ip}
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access_token	header	yes	Jwt token	
appstore_ip	appstore IP	path	yes	Valid IP Address	15

Example response:

```
200 OK
  {
    "appstoreIp" : "string"
    "appstorePort" : "string"
    "appstoreName" : "string"
    "appstoreRepoName" : "string"
    "appstoreRepo" : "string"
    "appstoreRepoUserName" : "string"
    "appstoreRepoPassword" : "string"
    "producer" : "string"
  }
```

Delete App Stores Record

Deletes a specific application store record in the Inventory matching the given tenant ID & application store IP

```
Resource URI: /inventory/v1/appstores/{appstore_ip}
Method: DELETE
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
appstore_ip	appstore IP	path	yes	Valid IP Address	15

Example response:

```
200 OK
[
  {
    "Deleted"
  }
]
```

Update App Store Record

Updates existing application store record

```
Resource URI: /inventory/v1/appstores/{appstore_ip}
Method: PUT
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
appstore_ip	appstore IP	path	yes	Valid IP Address	15
appstorIp	appstorIp	body	yes	Valid IP Address	15
appstoreName	appstoreName	body	yes	AlphaNumeric, special characters allowed are hypen and underscore	128
appstorePort	appstorePort	body	yes	Valid Port	5
producer	producer	body	yes	AlphaNumeric, special characters allowed are hypen and underscore	128
uri	uri	body	yes	Valid UUID	128
userName	userName	body	yes	AlphaNumeric, special characters allowed are hypen and underscore	128

Example request body:

```
{
  "appstoreIp": "string",
  "appstoreName": "string",
  "appstorePort": "string",
  "producer": "string",
  "uri": "string",
  "userName": "string"
}
```

Example response:

```
200 OK
[
  {
    "Updated"
```

(下页继续)

(续上页)

```
  }
]
```

Deletes Application Record

Deletes application store record

```
Resource URI: /inventory/v1/appstores/{appstore_ip}
Method: DELETE
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
appstore_ip	appstore IP	path	yes	Valid IP address	15

Example response:

```
200 OK
[
  {
    "Deleted"
  }
]
```

Get All MEC Host Records

Retrieves all MEC host records

```
Resource URI: /inventory/v1/mechosts
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	

Example response:

```
200 OK
[
```

(下页继续)

(续上页)


```
{
        "mechostIp": "string",
        "mechostName": "string",
        "zipCode": "string",
        "city": "string",
        "address": "string",
        "affinity": "string",
        "userName": "string",
        "edgerepoName": null,
        "edgerepoIp": "string",
        "edgerepoPort": "string",
        "edgerepoUsername": "string",
        "applcmIp": "string",
        "hwcapabilities": [
            {
                "hwType": "string",
                "hwVendor": "string",
                "hwModel": "string"
            },
            {
                "hwType": "string",
                "hwVendor": "string",
                "hwModel": "string"
            }
        ]
    }
]
```


Add MEC Host Record

Adds new MEC host record

```
Resource URI: /inventory/v1/mechosts
Method: POST
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
mechost_ip	mechost IP	path	yes	Valid IP address	15
address	address	body	yes	Valid IP address	255
affinity	affinity	body	yes	AlphaNumeric, special characters allowed are hypen,underscore and comma	128
applcmIp	applcmIp	body	yes	Valid IP address	15
city	city	body	yes	AlphaNumeric, special characters allowed are slash and space	128
edgeName	edgeName	body	yes	AlphaNumeric, special characters allowed are slash and space	128
edgerepoIp	edgerepoIp	body	yes	Valid IP address	255
edgerepoUsername	edgerepoUsername	body	yes	AlphaNumeric, special characters allowed are hypen and underscore	128
mechostIp	mechostIp	body	yes	Valid IP address	15
mechostName	mechostName	body	yes	AlphaNumeric, special characters allowed are hypen and underscore	128
userName	userName	body	yes	AlphaNumeric, special characters allowed are hypen and underscore	128
zipCode	zipCode	body	yes	Alpha, special characters allowed are hypen and underscore	128
hwcapabilities	hardwareCapabilities	body	No	AlphaNumeric characters allowed	128
hwType	hardwareType	body	No	AlphaNumeric characters allowed	128
hwVendor	hardwareVendor	body	No	AlphaNumeric characters allowed	128
hwModel	hardwareModel	body	No	AlphaNumeric characters allowed	128

Example request body:

```
[
  {
      "mechostIp": "string",
      "mechostName": "string",
      "zipCode": "string",
      "city": "string",
```

(下页继续)

(续上页)

```
      "address": "string",
      "affinity": "string",
      "userName": "string",
      "edgerepoName": null,
      "edgerepoIp": "string",
      "edgerepoPort": "string",
      "edgerepoUsername": "string",
      "applcmIp": "string",
      "hwcapabilities": [
          {
              "hwType": "string",
              "hwVendor": "string",
              "hwModel": "string"
          },
          {
              "hwType": "string",
              "hwVendor": "string",
              "hwModel": "string"
          }
      ]
  }
]
```

Example response:

```
200 OK
[
  {
    "Saved"
  }
]
```

Delete All MEC Host Records

Deletes all application stores records

```
Resource URI: /inventory/v1/mechosts
Method: DELETE
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	

Example response:

```
200 OK
[
  {
    "Deleted"
  }
]
```

Get MEC Host Record

Retrieves MEC host records

```
Resource URI: /inventory/v1/mechosts/{mechost_ip}
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
mechost_ip	mechost_ip	path	yes	Valid IP address	15

```
200 OK
{
      "mechostIp": "string",
      "mechostName": "string",
      "zipCode": "string",
      "city": "string",
      "address": "string",
      "affinity": "string",
      "userName": "string",
      "edgerepoName": null,
      "edgerepoIp": "string",
      "edgerepoPort": "string",
      "edgerepoUsername": "string",
      "applcmIp": "string",
      "hwcapabilities": [
          {
              "hwType": "string",
              "hwVendor": "string",
              "hwModel": "string"
          },
          {
```

(下页继续)

(续上页)

```
            "hwType": "string",
            "hwVendor": "string",
            "hwModel": "string"
        }
    ]
}
```

Update MEC Host Record

Updates existing MEC host record

```
Resource URI: /inventory/v1/mechosts/{mechost_ip}
Method: PUT
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
mechost_ip	mechost IP	path	yes	Valid IP address	15
address	address	body	yes	Valid IP address	255
affinity	affinity	body	yes	AlphaNumeric, special characters allowed are hypen,underscore and comma	128
applcmIp	applcmIp	body	yes	Valid IP address	15
city	city	body	yes	AlphaNumeric, special characters allowed are slash and space	128
edgeName	edgeName	body	yes	AlphaNumeric, special characters allowed are slash and space	128
edgerepoIp	edgerepoIp	body	yes	Valid IP address	255
edgerepoUsername	edgerepoUsername	body	yes	AlphaNumeric, special characters allowed are hypen and underscore	128
mechostIp	mechostIp	body	yes	Valid IP address	15
mechostName	mechostName	body	yes	AlphaNumeric, special characters allowed are hypen and underscore	128
userName	userName	body	yes	AlphaNumeric, special characters allowed are hypen and underscore	128
zipCode	zipCode	body	yes	Alpha, special characters allowed are hypen and underscore	128
hwcapabilities	hardwareCapabilities	body	No	AlphaNumeric characters allowed	128
hwType	hardwareType	body	No	AlphaNumeric characters allowed	128
hwVendor	hardwareVendor	body	No	AlphaNumeric characters allowed	128
hwModel	hardwareModel	body	No	AlphaNumeric characters allowed	128

Example request body:

```
{
      "mechostIp": "string",
      "mechostName": "string",
      "zipCode": "string",
      "city": "string",
      "address": "string",
```

(下页继续)

(续上页)

```
      "affinity": "string",
      "userName": "string",
      "edgerepoName": null,
      "edgerepoIp": "string",
      "edgerepoPort": "string",
      "edgerepoUsername": "string",
      "applcmIp": "string",
      "hwcapabilities": [
          {
              "hwType": "string",
              "hwVendor": "string",
              "hwModel": "string"
          },
          {
              "hwType": "string",
              "hwVendor": "string",
              "hwModel": "string"
          }
      ]
  }
```

Example response:

```
200 OK
[
  {
    "Updated"
  }
]
```

Delete MEC Host Record

Deletes application stores record

```
Resource URI: /inventory/v1/mechosts/{mechost_ip}
Method: DELETE
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
mechost_ip	mechost IP	path	yes	Valid IP Address	15

Example response:

```
200 OK
[
  {
    "Deleted"
  }
]
```

Add Application record

Adds new application record.

```
Resource URI: /inventory/v1/tenants/{tenant_id}/mechosts/{mechost_ip}/apps
Method: POST
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
appInstanceId	appInstance id	body	yes	Valid UUID	64
mechost_ip	mechost ip	path	yes	Valid IP Address	15
appName	appName	body	yes	AlphaNumeric,special characters allowed are hypen and underscore	128
capabilities	capabilities	body	yes	capabilities	10
packageId	packageId	body	yes	AlphaNumeric,special characters allowed are hypen and underscore	64
status	status	body	yes	AlphaNumeric,special characters allowed are hypen and underscore	128

Example request body:

```
{
  "appInstanceId": "string",
  "appName": "string",
  "capabilities": [
    "string"
```

(下页继续)

(续上页)

```
  ],
  "packageId": "string",
  "status": "string"
}
```

Example response:

```
200 OK
[
  {
    "Saved"
  }
]
```

Update Application record

Updates Application record.

```
Resource URI: /inventory/v1/tenants/{tenant_id}/mechosts/{mechost_ip}/apps/
↪{app_id}
Method: PUT
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
appInstanceId	appInstance id	body	yes	Valid UUID	64
tenant_id	tenant identifier	path	yes	Valid UUID	64
mechost_ip	mechost ip	path	yes	Valid IP Address	15
appName	appName	body	yes	AlphaNumeric,special characters allowed are hypen and underscore	128
capabilities	capabilities	body	yes	capabilities	10
packageId	packageId	body	yes	AlphaNumeric,special characters allowed are hypen and underscore	64
status	status	body	yes	AlphaNumeric,special characters allowed are hypen and underscore	128

Example request body:

```
{
  "appInstanceId": "string",
  "appName": "string",
  "capabilities": [
    "string"
  ],
  "packageId": "string",
  "status": "string"
}
```

Example response:

```
200 OK
[
  {
    "Updated"
  }
]
```

Delete Application record

Delete application record

```
Resource URI: /inventory/v1/tenants/{tenant_id}/mechosts/{mechost_ip}/apps/
↪{app_id}
Method: DELETE
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
mechost_ip	mechost_ip	path	yes	Valid IP address	15
tenant_id	tenant identifier	path	yes	Valid UUID	64
app_id	application id	path	yes	Valid UUID	64

Example response:

```
200 OK
[
  {
    "Deleted"
```

(下页继续)

(续上页)

```
  }
]
```

Retrieves MEC host record

Retrieves MEC host record for specific capabilities

```
Resource URI: /inventory/v1/tenants/{tenant_id}/mechosts/{mechost_ip}/
↪capabilities
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
mechost_ip	mechost_ip	path	yes	Valid IP address	15
tenant_id	tenant identifier	path	yes	Valid UUID	64
capabilities	capabilities	path	yes	capabilities	10

Example response:

```
200 OK
{
      "hwcapabilities": [
          {
              "hwType": "string",
              "hwVendor": "string",
              "hwModel": "string"
          },
          {
              "hwType": "string",
              "hwVendor": "string",
              "hwModel": "string"
          }
      ]
  }
```

Retrieves MEC application record

Retrieves applications matching capability in a specific MEC host record.

```
Resource URI: /inventory/v1/tenants/{tenant_id}/mechosts/{mechost_ip}/
↪capabilities/{capability_type}/applications
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
mechost_ip	mechost_ip	path	yes	Valid IP address	15
tenant_id	tenant identifier	path	yes	Valid UUID	64
capabilities	capabilities	path	yes	capabilities	10
capabilityType	capability type	path	yes	AlphaNumeric,special characters allowed are hypen and underscore	128

Example response:

```
200 OK
{
      "apps": [
          {
              "appInstanceId": "string",
              "appName": "string",
              "packageId": "string"
              "capabilities": [
                  "string"
              ],
              "status": "Created"
      ]
  }
```

Upload K8s Configuration File

Upload K8s configuration file to applcm

```
Resource URI: /inventory/v1/mechosts/{mechost_ip}/k8sconfig
Method: POST
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
file	file	formData	yes		
mechost_ip	mechost IP	path	yes	Valid IP Address	15

Example response:

```
200 OK
  [
    "response": {}
  ]
```

Delete K8s Configuration File

Deletes K8s configuration file from applcm

```
Resource URI: /inventory/v1/mechosts/{mechost_ip}/k8sconfig
Method: DELETE
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
mechost_ip	mechost IP	path	yes	Valid IP Address	15

Example response:

```
200 OK
[
  {
    "Deleted"
  }
]
```

Queries liveness & readiness

Retrieves data for liveness & readiness

```
Resource URI: /inventory/v1/health
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	

Example response:

```
200 OK
```

Add APPDRule record to Inventory.

Adds a new APPDRule record entry into the Inventory.

```
Resource URI: /inventory/v1/tenants/{tenant_id}/app_instances/{app_instance_
↪id}/appd_configuration
Method: POST
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
tenant_id	tenant identifier	path	yes	Valid UUID	64
appInstanceIds	application instance id	path	yes	Valid UUID	64

Example request body:

```
{
  "appTrafficRule": [
    {
      "trafficRuleId": "string",
      "filterType": "string",
      "priority": int,
      "trafficFilter": [
        {
          "srcAddress": [
            "string"
```

(下页继续)

(续上页)

```
          ],
          "dstAddress": [
            "string"
          ],
          "srcPort": [
            "string"
          ],
          "dstPort": [
            "string"
          ],
          "protocol": [
            "string"
          ],
          "qCI": int,
          "dSCP": int,
          "tC": int
        }
      ],
      "action": "string",
      "state": "string"
    }
  ],
  "appDNSRule": [
    {
      "dnsRuleId": "string",
      "domainName": "string",
      "ipAddressType": "string",
      "ipAddress": "string",
      "ttl": int,
      "state": "string"
    }
  ],
  "appSupportMp1": true,
  "appName": "abcd"
}
```

Example response:

```
200 OK
{
    "Saved"
}
```

Update APPDRule record to Inventory.

Updates a existing APPDRule record entry into the Inventory matching the given tenant ID & app instance ID.

```
Resource URI: /inventory/v1/tenants/{tenant_id}/app_instances/{app_instance_
↪id}/appd_configuration
Method: PUT
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
tenant_id	tenant identifier	path	yes	Valid UUID	64
appInstanceIds	application instance id	path	yes	Valid UUID	64

Example request body:

```
{
  "appTrafficRule": [
    {
      "trafficRuleId": "string",
      "filterType": "string",
      "priority": int,
      "trafficFilter": [
        {
          "srcAddress": [
            "string"
          ],
          "dstAddress": [
            "string"
          ],
          "srcPort": [
            "string"
          ],
          "dstPort": [
            "string"
          ],
          "protocol": [
            "string"
          ],
          "qCI": int,
          "dSCP": int,
          "tC": int
```

(下页继续)

(续上页)

```
        }
      ],
      "action": "string",
      "state": "string"
    }
  ],
  "appDNSRule": [
    {
      "dnsRuleId": "string",
      "domainName": "string",
      "ipAddressType": "string",
      "ipAddress": "string",
      "ttl": int,
      "state": "string"
    }
  ],
  "appSupportMp1": true,
  "appName": "abcd"
}
```

Example response:

```
200 OK
{
    "Updated"
}
```

Get APPDRule record to Inventory.

Retrieves appDRule records for given tenant ID and app instance ID.

```
Resource URI: /inventory/v1/tenants/{tenant_id}/app_instances/{app_instance_
↪id}/appd_configuration
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
tenant_id	tenant identifier	path	yes	Valid UUID	64
appInstanceIds	application instance id	path	yes	Valid UUID	64

Example response body:

```
200 OK
{
  "appTrafficRule": [
    {
      "trafficRuleId": "string",
      "filterType": "string",
      "priority": int,
      "trafficFilter": [
        {
          "srcAddress": [
            "string"
          ],
          "dstAddress": [
            "string"
          ],
          "srcPort": [
            "string"
          ],
          "dstPort": [
            "string"
          ],
          "protocol": [
            "string"
          ],
          "qCI": int,
          "dSCP": int,
          "tC": int
        }
      ],
      "action": "string",
      "state": "string"
    }
  ],
  "appDNSRule": [
    {
      "dnsRuleId": "string",
      "domainName": "string",
      "ipAddressType": "string",
      "ipAddress": "string",
      "ttl": int,
      "state": "string"
    }
  ],
  "appSupportMp1": true,
```

(下页继续)

(续上页)

```
  "appName": "abcd"
}
```

Delete APPDRule record .

Deletes appDRule records for a given tenant and app instance.

```
Resource URI: /inventory/v1/tenants/{tenant_id}/app_instances/{app_instance_
↪id}/appd_configuration
Method: DELETE
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
tenant_id	tenant identifier	path	yes	Valid UUID	64
appInstanceIds	application instance id	path	yes	Valid UUID	64

Example response body:

```
200 OK
{
"DELETED"
}
```

Synchronizes mec host info from edge

Synchronizes mec host info from edge.

```
Resource URI: /mepms/{mepm_ip}/mechost/sync
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
mepm_ip	mepm_ip	path	yes	Valid IP Address	15

Example response body:

```
200 OK
{
    "mecHostUpdatedRecs": [
        {
            "mechostIp": "127.0.0.2",
            "mechostName": "edgegallery3",
            "zipCode": "560038",
            "city": "bangalore",
            "address": "anadapura",
            "affinity": "karanataka",
            "userName": "ramasubbareddy",
            "configUploadStatus": "false",
            "coordinates": "1,2",
            "vim": "openstack",
            "origin": "MEPM",
            "hwcapabilities": [
                {
                    "hwType": "type6",
                    "hwVendor": "vendor1",
                    "hwModel": "model"
                }
            ]
        }
    ]
}
```

Synchronizes mec host info from edge

Synchronizes mec host info from edge.

```
Resource URI: /tenants/{tenant_id}/mepms/{mepm_ip}/apprule/sync
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
tenant_id	tenant identifier	path	yes	Valid UUID	64
mepm_ip	mepm_ip	path	yes	Valid IP Address	15

Example response body:

```
200 OK
{
    "appdRuleUpdatedRecs": [
        {
            "appdRuleId": "e921ce54-82c8-4532-b5c6-8516cf75f7a771ea862b-5806-
↪4196-bce3-434bf9c95b18",
            "tenantId": "e921ce54-82c8-4532-b5c6-8516cf75f7a7",
            "appInstanceId": "71ea862b-5806-4196-bce3-434bf9c95b18",
            "appName": "abcd",
            "appSupportMp1": true,
            "appTrafficRule": [
                {
                    "trafficRuleId": "TrafficRule1",
                    "filterType": "FLOW",
                    "priority": 1,
                    "action": "DROP",
                    "trafficFilter": [
                        {
                            "trafficFilterId": "7e302be6-253b-4d09-8d1d-
↪ef6630db7b88",
                            "srcAddress": [
                                "192.168.1.1/28",
                                "192.168.1.2/28"
                            ],
                            "srcPort": null,
                            "dstAddress": [
                                "192.168.1.1/28"
                            ],
                            "dstPort": [
                                "6666666666"
                            ],
                            "protocol": [
                                "TCP"
                            ],
                            "qCI": 1,
                            "dSCP": 0,
                            "tC": 1,
                            "tag": [
                                "1"
                            ],
                            "srcTunnelAddress": [
                                "1.1.1.1/24"
                            ],
                            "dstTunnelAddress": null,
```

(下页继续)

(续上页)


```
                            "srcTunnelPort": [
                                "65536"
                            ],
                            "dstTunnelPort": [
                                "65537"
                            ]
                        }
                    ],
                    "dstInterface": [
                        {
                            "dstInterfaceId": "cf053d4c-250b-450f-a657-
↪a30328a3b27a",
                            "interfaceType": "",
                            "srcMacAddress": "",
                            "dstMacAddress": "",
                            "dstIpAddress": "",
                            "TunnelInfo": {
                                "tunnelInfoId": "be36d5e2-41db-4270-b6cf-
↪ba9e7b7a89aa",
                                "tunnelType": "",
                                "tunnelDstAddress": "",
                                "tunnelSrcAddress": "",
                                "tunnelSpecificData": ""
                            }
                        }
                    ]
                }
            ],
            "appDnsRule": [
                {
                    "dnsRuleId": "dnsRule4",
                    "domainName": "www.example.com",
                    "ipAddressType": "IP_V4",
                    "ipAddress": "192.0.2.0",
                    "ttl": 30
                }
            ],
            "Origin": "MEPM"
        }
    ]
}
```

Get MEPM Record

Retrieves all MEPM records

```
Resource URI: /inventory/v1/mepms
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	

```
200 OK
[
    {
      "mepmIp": "string",
      "mepmName": "string",
      "mepmPort": "string",
      "userName": "string"
    }
]
```

Add MEPM Record

Adds new MEPM record

```
Resource URI: /inventory/v1/mepms
Method: POST
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
mepmIp	mepm Ip	body	yes	Valid IP address	15
mepmName	mepm Name	body	yes	AlphaNumeric, special characters allowed are hypen and underscore	128
mepmPort	mepm Port	body	yes	Vaild Port	5
userName	user Name	body	yes	AlphaNumeric, special characters allowed are hypen and underscore	128

Example request body:

```
{
    "mepmIp": "string",
    "mepmName": "string",
    "mepmPort": "string",
    "userName": "string"
}
```

Example response:

```
200 OK
[
  {
    "Saved"
  }
]
```

Delete All MEPM Records

Deletes all mepm records

```
Resource URI: /inventory/v1/mepms
Method: DELETE
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	

Example response:

```
200 OK
[
  {
    "Deleted"
  }
]
```

Get MEPM Record

Retrieves mepm records

```
Resource URI: /inventory/v1/mepms/{mepm_ip}
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
mepm_ip	mepm IP	path	yes	Valid IP address	15

```
200 OK
{
    "mepmIp": "string",
    "mepmName": "string",
    "mepmPort": "string",
    "userName": "string"
}
```

Update MEPM Record

Updates existing mepm record

```
Resource URI: /inventory/v1/mepms/{mepm_ip}
Method: PUT
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
mepm_ip	mepm IP	path	yes	Valid IP address	15
mepmIp	mepm Ip	body	yes	Vaild IP address	15
mepmName	mepm Name	body	yes	AlphaNumeric, special characters allowed are hypen and underscore	128
mepmPort	mepm Port	body	yes	Vaild Port	5
userName	user Name	body	yes	AlphaNumeric, special characters allowed are hypen and underscore	128

Example request body:

```
{
    "mepmIp": "string",
    "mepmName": "string",
    "mepmPort": "string",
    "userName": "string"
}
```

Example response:

```
200 OK
[
  {
    "Updated"
  }
]
```

Delete MEPM Record

Deletes mepm record

```
Resource URI: /inventory/v1/mepms/{mepm_ip}
Method: DELETE
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	
mepm_ip	mepm IP	path	yes	Valid IP Address	15

Example response:

```
200 OK
[
  {
    "Deleted"
  }
]
```

Inventory Open Api Swagger

- Please refer INVENTORY_OPEN_API

9.3.5 MECM App Rule Manager Interfaces

The document is for the MECM App Rule Manager Interfaces project .

Queries liveness & readiness

Retrieves data for liveness & readiness

```
Resource URI: /apprulemgr/v1/health
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access token	header	yes	Jwt token	

Example response:

```
200 OK
```

Create AppRule Config

Configures the app rule

```
Resource URI:  /apprulemgr/v1/tenants/{tenantId}/app_instances/
↪{appInstanceId}/appd_configuration
Method: POST
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access_token	header	yes	Jwt token	
appInstance_id	application instance id	path	yes	Valid UUID	64
tenant_id	tenant id	path	yes	Valid UUID	64
appInstanceIds	application instance id	path	yes	Valid UUID	64

Example request body:

```
"appTrafficRule": [
   {
     "trafficRuleId": "string",
     "filterType": "string",
     "priority": int,
     "trafficFilter": [
       {
         "srcAddress": [
           "string"
         ],
         "dstAddress": [
           "string"
         ],
         "srcPort": [
           "string"
         ],
         "dstPort": [
           "string"
         ],
         "protocol": [
           "string"
         ],
         "qCI": int,
         "dSCP": int,
         "tC": int
       }
     ],
     "action": "string",
     "state": "string"
   }
 ],
 "appDNSRule": [
   {
     "dnsRuleId": "string",
     "domainName": "string",
     "ipAddressType": "string",
     "ipAddress": "string",
     "ttl": int,
     "state": "string"
   }
 ],
 "appSupportMp1": bool,
 "appName": "string"
```

Example response:

```
200 OK
{
  "taskId":"string",
  "appInstanceId":"string",
  "configResult":"string",
  "configPhase":"string",
  "detailed":"string"
}
```

Update AppRule Config

Updates the existing app rule

```
Resource URI: /apprulemgr/v1/tenants/{tenantId}/app_instances/{appInstanceId}
↪/appd_configuration
Method: PUT
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access_token	header	yes	Jwt token	
appInstance_id	application instance id	path	yes	Valid UUID	64
tenant_id	tenant id	path	yes	Valid UUID	64
appInstanceIds	application instance id	path	yes	Valid UUID	64

Example request body:

```
 "appTrafficRule": [
    {
      "trafficRuleId": "string",
      "filterType": "string",
      "priority": int,
      "trafficFilter": [
        {
          "srcAddress": [
            "string"
          ],
          "dstAddress": [
            "string"
          ],
          "srcPort": [
            "string"
```

(下页继续)

(续上页)

```
          ],
          "dstPort": [
            "string"
          ],
          "protocol": [
            "string"
          ],
          "qCI": int,
          "dSCP": int,
          "tC": int
        }
      ],
      "action": "string",
      "state": "string"
    }
  ],
  "appDNSRule": [
    {
      "dnsRuleId": "string",
      "domainName": "string",
      "ipAddressType": "string",
      "ipAddress": "string",
      "ttl": int,
      "state": "string"
    }
  ],
  "appSupportMp1": bool,
  "appName": "string"
```

Example response:

```
200 OK
{
  "taskId":"string",
  "appInstanceId":"string",
  "configResult":"string",
  "configPhase":"string",
  "detailed":"string"
}
```

Get AppRule Config

Queries the existing app rule

```
Resource URI: /apprulemgr/v1/tenants/{tenantId}/app_instances/{appInstanceId}
↪/appd_configuration
Method: GET
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access_token	header	yes	Jwt token	
appInstance_id	application instance id	path	yes	Valid UUID	64
tenant_id	tenant id	path	yes	Valid UUID	64
appInstanceIds	application instance id	path	yes	Valid UUID	64

Example Response body:

```
200 OK

 "appTrafficRule": [
    {
      "trafficRuleId": "string",
      "filterType": "string",
      "priority": int,
      "trafficFilter": [
        {
          "srcAddress": [
            "string"
          ],
          "dstAddress": [
            "string"
          ],
          "srcPort": [
            "string"
          ],
          "dstPort": [
            "string"
          ],
          "protocol": [
            "string"
          ],
          "qCI": int,
          "dSCP": int,
```

(下页继续)

(续上页)

```
          "tC": int
        }
      ],
      "action": "string",
      "state": "string"
    }
  ],
  "appDNSRule": [
    {
      "dnsRuleId": "string",
      "domainName": "string",
      "ipAddressType": "string",
      "ipAddress": "string",
      "ttl": int,
      "state": "string"
    }
  ],
  "appSupportMp1": bool,
  "appName": "string"
```

Delete AppRule Config

Delete the existing app rule

```
Resource URI: /apprulemgr/v1/tenants/{tenantId}/app_instances/{appInstanceId}
↪/appd_configuration
Method: DELETE
```

Name	Definition	Type	Required	Allowed	Max Length
access_token	access_token	header	yes	Jwt token	
appInstance_id	application instance id	path	yes	Valid UUID	64
tenant_id	tenant id	path	yes	Valid UUID	64
appInstanceIds	application instance id	path	yes	Valid UUID	64

Example Response body:

```
200 OK
{
 "response" :
```

(下页继续)

(续上页)

```
  {
   "apprule_task_id" : "string"
  }
}
```

9.3.6 K8s plugin Interfaces

The document is for the K8s plugin project.

InstantiateRequest

Application instantiated

Field	Type	Label	Description
accessToken	string	required	Provide the required access token
appInstanceId	string	required	Valid UUID
hostIp	string	required	Valid IP
package	bytes	required	Valid UUID without hypen
parameters	key value pair	No	Instantiation parameters, key-value pairs

InstantiateResponse

Represents the InstantiateRequest status.

Field	Type	Label	Description
status	string	required	Status of the instantiate request

TerminateRequest

Terminates application instance id

Field	Type	Label	Description
accessToken	string	required	Provide the required access token
appInstanceId	string	required	Valid UUID
hostIp	string	required	Valid IP

TerminateResponse

Represents the TerminateRequest status.

Field	Type	Label	Description
status	string	required	Status of the terminate request

QueryRequest

Get statistics information

Field	Type	Label	Description
accessToken	string	required	Provide the required access token
appInstanceId	string	required	Valid UUID
hostIp	string	required	Valid IP

QueryResponse

Represents the QueryRequest status.

Field	Type	Label	Description
status	string	required	Status of the Query request

UploadCfgRequest

Upload Config File

Field	Type	Label	Description
accessToken	string	required	Provide the required access token
hostIp	string	required	Valid IP
configFile	formData	required	Provide valid config file

UploadCfgResponse

Represents the UploadCfgRequest status.

Field	Type	Label	Description
status	string	required	Status of the UploadCfgRequest

RemoveCfgRequest

Removes the config file

Field	Type	Label	Description
accessToken	string	required	Provide the required access token
hostIp	string	required	Valid IP

RemoveCfgResponse

Field	Type	Label	Description
status	string	required	Status of the RemoveCfgRequest

CreateVmImageRequest

Create VM image request

Field	Type	Label	Description
accessToken	string	required	Provide the required access token
appInstanceId	string	required	Valid UUID
hostIp	string	required	Valid IP
vmId	string	required	Valid UUID

CreateVmImageResponse

Field	Type	Label	Description
response	string	required	response of the CreateVmImageResponse

QueryVmImageRequest

Query VM image request

Field	Type	Label	Description
accessToken	string	required	Provide the required access token
appInstanceId	string	required	Valid UUID
hostIp	string	required	Valid IP
imageId	string	required	Valid UUID

QueryVmImageResponse

Field	Type	Label	Description
response	string	required	response of the QueryVmImageResponse

DeleteVmImageRequest

Delete VM image request

Field	Type	Label	Description
accessToken	string	required	Provide the required access token
appInstanceId	string	required	Valid UUID
hostIp	string	required	Valid IP
imageId	string	required	Valid UUID

DeleteVmImageResponse

Field	Type	Label	Description
response	string	required	response of the DeleteVmImageResponse

DownloadVmImageRequest

Download VM image request

Field	Type	Label	Description
accessToken	string	required	Provide the required access token
appInstanceId	string	required	Valid UUID
hostIp	string	required	Valid IP
imageId	string	required	Valid UUID
chunkNum	int32	required	Valid integer

DownloadVmImageResponse

Field	Type	Label	Description
content	byte	required	response of the DownloadVmImageResponse

AppLCM Service

Service for handling requests.

Method Name	Request Type	Response Type	Description
instantiate	InstantiateRequest	InstantiateResponse	Instantiate application
terminate	TerminateRequest	TerminateResponse	Terminate application
query	QueryRequest	QueryResponse	Query application
uploadConfig	UploadCfgRequest	UploadCfgResponse	Upload file configuration
removeConfig	RemoveCfgRequest	RemoveCfgResponse	Remove file configuration

9.4 Database Designs 数据库设计

9.4.1 Apm Database Design

The APM database uses an open source PostgreSQL database, and the recommended version is 12.3.

- apppackage: Store application package information, such as app_pkg_id, app_pkg_path and tenant_id.

```
create table apppackage (
    id varchar(255) not null,
    app_pkg_id  varchar(255) not null,
    app_pkg_name varchar(255) not null,
    app_pkg_version varchar(255) not null,
    app_pkg_path varchar(255) not null,
    app_provider varchar(200),
```

(下页继续)

(续上页)

```
    app_pkg_desc varchar(500),
    app_pkg_affinity varchar(200),
    app_icon_url varchar(255),
    app_id varchar(255) not null,
    tenant_id varchar(255) not null,
    local_file_path varchar(2000),
    created_time timestamp default current_timestamp,
    modified_time timestamp default current_timestamp,
    primary key (id)
);
```

- apppackagehost: Store application package distribution information, such as app_pkg_id, distribution_status , tenant_id.

```
create table apppackagehost (
    id INT GENERATED BY DEFAULT AS IDENTITY,
    pkg_host_key varchar(255) not null,
    host_ip varchar(255) not null,
    app_pkg_id varchar(255) not null,
    distribution_status varchar(200) not null,
    tenant_id varchar(200) not null,
    error varchar(2000),
    primary key (id)
);
```

- apmtenant: Store tenant information, such as tenant.

```
create table apmtenant (
    tenant  varchar(255) not null,
    primary key (tenant)
);
```

9.4.2 Lcm Controller Database Design

The LcmController database uses an open source PostgreSQL database, and the recommended version is 12.2. Two database tables currently designed and applied:

- app_info_record: Store app information, such as appInsId, hostIp, deployType, appPackageId, appName, Origin, SyncStatus and tenantId.

```
CREATE TABLE app_info_record (
        APPINSID            VARCHAR(200)       NOT NULL,
        MECHOST             VARCHAR(200)       NULL,
```

(下页继续)

(续上页)

```
            DEPLOYTYPE              VARCHAR(200)        NULL,
            TENANTID                VARCHAR(200)        NULL,
            CONSTRAINT app_info_record_pkey PRIMARY KEY (APPINSID),
        APPPACKAGEID            VARCHAR(200)        NULL,
        APPNAME                 VARCHAR(200)        NULL,
        ORIGIN                  VARCHAR(200)        NULL,
        SYNCSTATUS              VARCHAR(200)        NULL
    );
```

- tenant_info_record: Store tenant information, such as tenantId.

```
CREATE TABLE tenant_info_record (
            TENANTID                VARCHAR(200)        NOT NULL,
            CONSTRAINT tenant_info_record_pkey PRIMARY KEY (TENANTID)
    );
```

- mec_host: Store mec host information, such as mecHostId, mecHostIp, mecHostName, zipCode, city, address, affinity UserName, configUploadStatus, coordinates, vim, origin and syncStatus

```
CREATE TABLE mec_host (
        MECHOSTID               VARCHAR(200)        NOT NULL,
        MECHOSTIP               VARCHAR(200)        NULL,
        MECHOSTNAME             VARCHAR(200)        NULL,
        ZIPCODE                 VARCHAR(200)        NULL,
        CITY                    VARCHAR(200)        NULL,
        ADDRESS                 VARCHAR(200)        NULL,
        AFFINITY                VARCHAR(200)        NULL,
        USERNAME                VARCHAR(200)        NULL,
        CONFIGUPLOADSTATUS      VARCHAR(200)        NULL,
        COORDINATES             VARCHAR(200)        NULL,
        VIM                     VARCHAR(200)        NULL,
        ORIGIN                  VARCHAR(200)        NULL,
        SYNCSTATUS              VARCHAR(200)        NULL,
    CONSTRAINT mec_host_pkey PRIMARY KEY (MECHOSTID)
);
```

- mec_hw_capability: Store mec hardware capability information, such as MecCapabilityId, hwType, hwVendor and hwModel

```
CREATE TABLE mec_hw_capability (
        MECCAPABILITYID VARCHAR(200)        NOT NULL,
        HWTYPE          VARCHAR(200)        NULL,
        HWVENDOR        VARCHAR(200)        NULL,
        HWMODEL         VARCHAR(200)        NULL
```

(下页继续)

(续上页)

```
);
```

- app_instance_stale_rec: Store app instance stale record information, such as appInstanceId and tenantId

```
CREATE TABLE app_instance_stale_rec (
        APPINSTANCEID VARCHAR(200)       NOT NULL,
        TENANTID      VARCHAR(200)       NULL
);
```

- mec_host_stale_rec: Store mec host stale record information, such as mecHostId

```
CREATE TABLE mec_host_stale_rec (
        MECHOSTID VARCHAR(200)       NOT NULL,
);
```

- app_package_record: Store app package record information, such as appPkgId, appPkgName, appPkgVersion, appPkgPath, appPkgProvider, appPkgDesc, appPkgAffinity, appIconUrl, appId, tenantId, packageId, origin and syncStatus

```
CREATE TABLE app_package_record (
        APPPKGID       VARCHAR(200)       NOT NULL,
        APPPKGNAME     VARCHAR(200)       NULL,
        APPPKGVERSION  VARCHAR(200)       NULL,
        APPPKGPATH     VARCHAR(200)       NULL,
        APPPROVIDER    VARCHAR(200)       NULL,
        APPPKGDESC     VARCHAR(200)       NULL,
        APPPKGAFFINITY VARCHAR(200)       NULL,
        APPICONURL     VARCHAR(200)       NULL,
        APPID          VARCHAR(200)       NULL,
        TENANTID       VARCHAR(200)       NULL,
        PACKAGEID      VARCHAR(200)       NULL,
        ORIGIN         VARCHAR(200)       NULL,
        SYNCSTATUS     VARCHAR(200)       NULL
        );
```

- app_package_host_record: Store app package host record information, such as pkgHostKey, hostIp, appPkgId, status, tenantId, error, origin and syncStatus

```
CREATE TABLE app_package_host_record  {
        PKGHOSTKEY VARCHAR(200)       NOT NULL,
        HOSTIP     VARCHAR(200)       NULL,
        APPPKGID   VARCHAR(200)       NULL,
        STATUS     VARCHAR(200)       NULL,
```

(下页继续)

(续上页)

```
        TENANTID    VARCHAR(200)        NULL,
        ERROR       VARCHAR(200)        NULL,
        ORIGIN      VARCHAR(200)        NULL,
        SYNCSTATUS VARCHAR(200)         NULL
    );
```

- app_package_stale_rec: Store app package stale record information, such as appPkgId and tenantId

```
CREATE TABLE app_package_stale_rec (
        APPKGID       VARCHAR(200)        NOT NULL,
        TENANTID      VARCHAR(200)        NULL
   );
```

- app_package_host_stale_rec: Store app package stale record information, such as pacakgeId, tenantId and hostIp

```
CREATE TABLE app_package_host_stale_rec (
        PACKAGEID       VARCHAR(200)        NOT NULL,
        TENANTID        VARCHAR(200)        NULL,
        HOSTIP          VARCHAR(200)        NULL
    );
```

- edge_authenticate_rec: Store edge authenticate record information, such as authenticateId, name and key

```
CREATE TABLE edge_authenticate_rec (
        AUTHENTICATEID  VARCHAR(200)        NOT NULL,
        NAME            VARCHAR(200)        NULL,
        KEY             VARCHAR(200)        NULL
   );
```

9.4.3 K8s Plugin Database Design

The K8sPlugin database uses an open source PostegreSQL database, and the recommended version is 12.2. One database table currently designed and applied:

- app_instance_info_record: MECM LcmController Database Design.mdStore app information, such as appInsId, hostIp and workloadId.

```
CREATE TABLE app_instance_info_record (
            APPINSID                VARCHAR(200)        NOT NULL,
            HOSTIP                  VARCHAR(200)        NULL,
            WORKLOADID              VARCHAR(200)        NULL,
```

(下页继续)

(续上页)

```
        CONSTRAINT app_info_record_pkey PRIMARY KEY (APPINSID)
);
```

9.4.4 Appo Database Design

The APPO database uses an open source PostgreSQL database, and the recommended version is 12.3.

- appinstanceinfo: Store app instance information, such as app_instance_id, applcm_host and tenant.

```
create table appinstanceinfo (
   app_instance_id  varchar(64) not null,
    app_package_id varchar(64) not null,
    app_id varchar(64) not null,
    tenant varchar(64) not null,
    app_name varchar(128) not null,
    app_descriptor varchar(256) not null,
    mec_host varchar(15) not null,
    applcm_host varchar(15),
    operational_status varchar(128) not null,
    operation_info varchar(256),
    create_time timestamp default current_timestamp,
    update_time timestamp default current_timestamp,
    primary key (app_instance_id)
);
```

- appotenant: Store tenant information, such as tenant.

```
create table appotenant (
    tenant  varchar(255) not null,
    primary key (tenant)
);
```

9.4.5 Inventory Database Design

The Inventory database uses an open source PostgreSQL database, and the recommended version is 12.3.

- applcminventory: Store applcm information, such as applcm_id, tenant_id, applcm_ip and applcm_port.

```
create table applcminventory (
    applcm_id varchar(255) not null,
    tenant_id  varchar(255) not null,
```

(下页继续)

(续上页)

```
    applcm_ip varchar(255) not null,
    applcm_port varchar(255) not null,
    user_name varchar(255),
    created_time varchar(200),
    modified_time varchar(200),
    primary key (applcm_id)
);
```

- appstoreinventory: Store app store distribution information, such as appstore_id, appstore_ip , appstore_port.

```
create table appstoreinventory (
    appstore_id varchar(255) not null,
    appstore_ip varchar(255) not null,
    appstore_port varchar(255) not null,
    uri varchar(255) not null,
    tenant_id varchar(200) not null,
    user_name varchar(255),
    appstore_name varchar(255),
    producer varchar(255),
    created_time varchar(200),
    modified_time varchar(200),
    primary key (appstore_id)
);
```

- mechostinventory: Store MEC host information, such as mechost_id, mechost_ip, applcm_ip, zip_code, city.

```
create table mechostinventory (
    mechost_id varchar(255) not null,
    tenant_id varchar(255) not null,
    mechost_ip varchar(255) not null,
    applcm_ip varchar(255) not null,
    mechost_name varchar(255) not null,
    zip_code varchar(200),
    city varchar(255) not null,
    address varchar(255) not null,
    affinity varchar(255),
    user_name varchar(255),
    edge_name varchar(255),
    edgerepo_ip varchar(255),
    edgerepo_port varchar(255),
    edgerepo_username varchar(255),
    config_upload_status varchar(255),
```

(下页继续)

(续上页)

```
    -- config_file_path varchar(2000),
    created_time varchar(200),
    modified_time varchar(200),
    primary key (mechost_id)
);
```

- tenantinventory: Store tenant information, such as tenant_id.

```
create table tenantinventory (
    tenant_id  varchar(255) not null,
    applcm_count int,
    appstore_count int,
    mechost_count int,
    primary key (tenant_id)
);
```

9.5 User Guides 使用指导

9.5.1 MECM Product Document

MECM 管理面

MECM 管理面主要提供边缘节点、APPLCM 注册以及应用的分发部署功能。

V1.1 版本新增特性如下:

1. 新增游客、租户及管理员角色，管理员可以进行注册和删除操作，租户可以进行应用同步，分发，部署操作，游客只有浏览权限 2. 新增应用商店注册功能，支持管理员新增注册应用商店 3. 新增应用同步功能，支持应用从应用商店到 APM 同步

用户指南

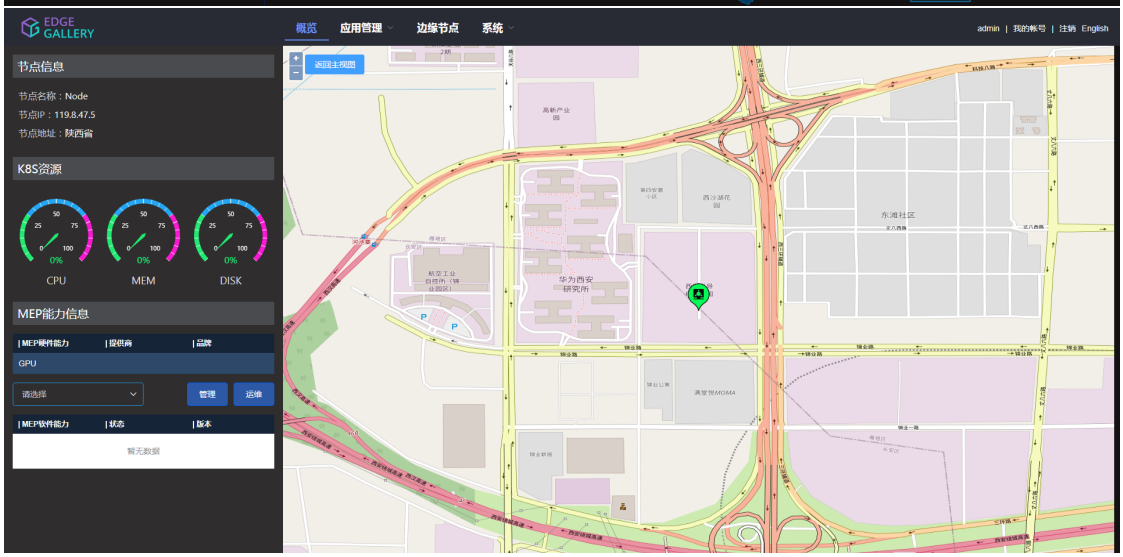
！ ！ ！注意事项: 目前由于客户端和服务端协议问题，用户访问可能存在证书异常，如存在数据获取异常问题，建议先在浏览器访问https://mecm.edgegallery.org:30202，然后同意继续访问，然后回到 MECM 界面刷新即可，此问题会在下个版本解决。

您的连接不是私密连接
攻击者可能会试图从 窃取您的信息（例如：密码、通讯内容或信用卡信息）。了解详情
NET::ERR_CERT_AUTHORITY_INVALID
如果您想获得 Chrome 最高级别的安全保护，请开启增强型保护
高级
返回安全连接

您的连接不是私密连接
攻击者可能会试图从 窃取您的信息（例如：密码、通讯内容或信用卡信息）。了解详情
NET::ERR_CERT_AUTHORITY_INVALID
如果您想获得 Chrome 最高级别的安全保护，请开启增强型保护
隐藏详情
返回安全连接
此服务器无法证明它是 您计算机的操作系统不信任其安全证书。出现此问题的原因可能是配置有误或您的连接被拦截了。
继续前往 （不安全）

1. 概述：地图区域 & 详情展示（节点以及 KPI、MEP 能力以及硬件能力），MEP 详情

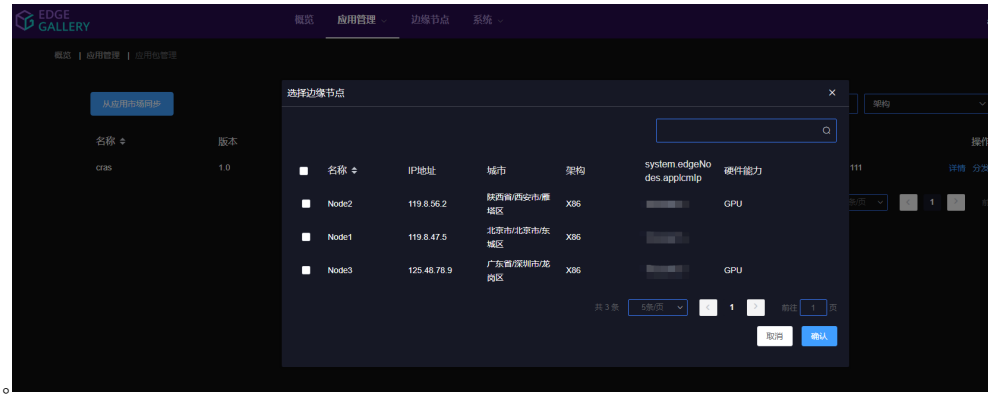
1.1 地图 & 详情：点击地图上的节点，可显示节点以及 KPI、MEP 能力以及硬件能力详情



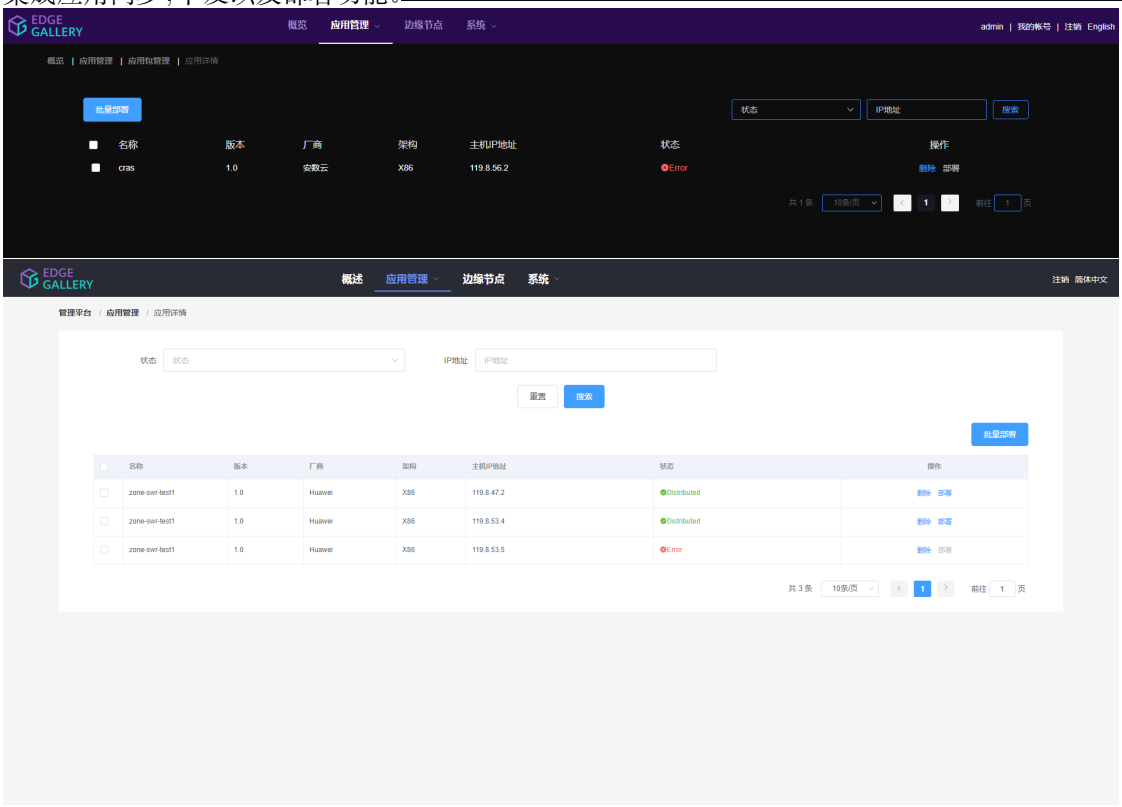
1.2 MEP 详情：点击管理按钮，跳转至 MEP 详情界面

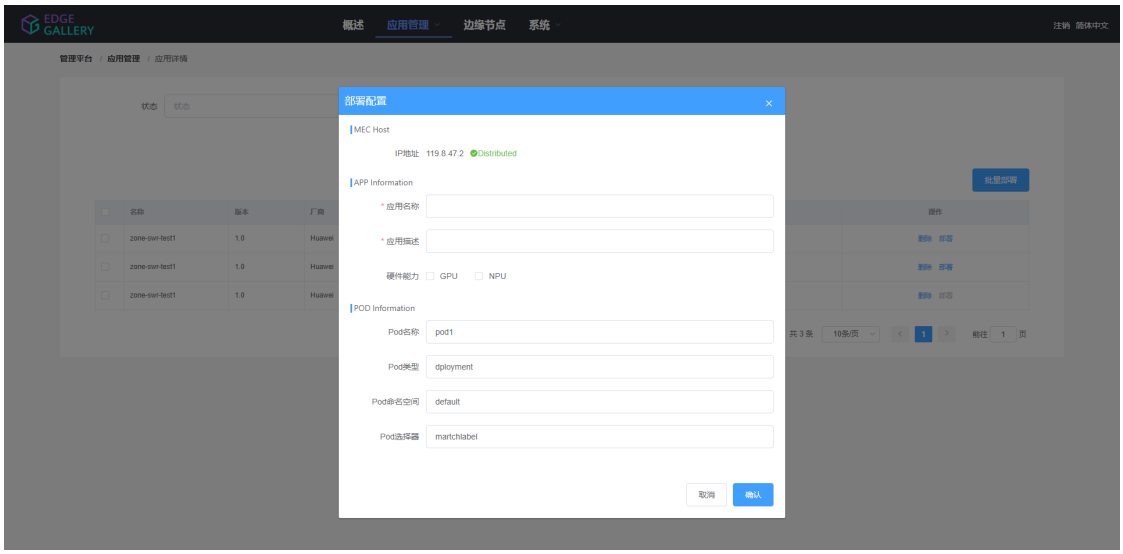
2. 应用管理：应用包管理，应用实例管理

2.1 应用包管理



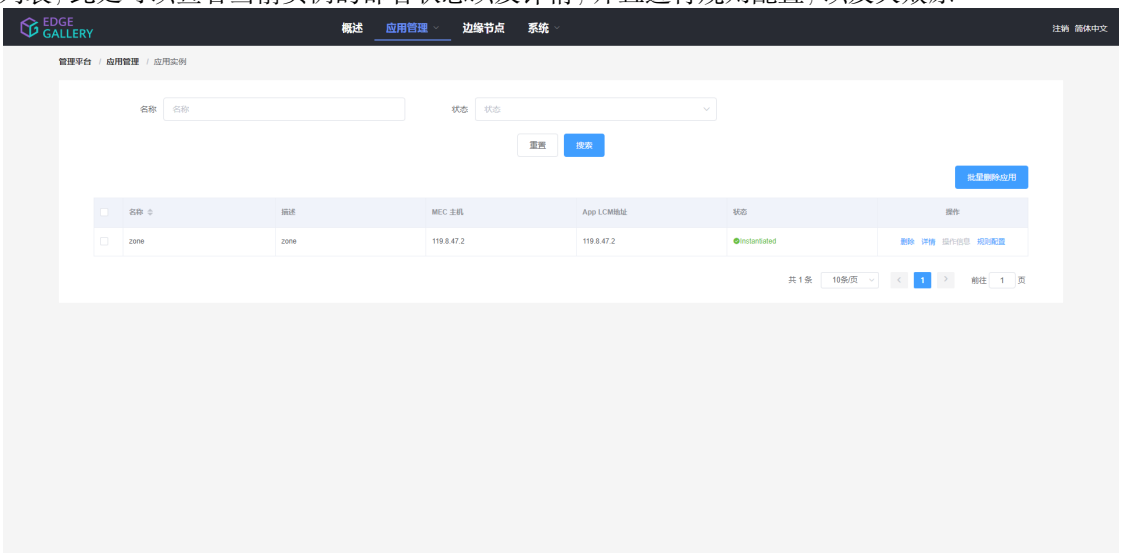
集成应用同步,下发以及部署功能。

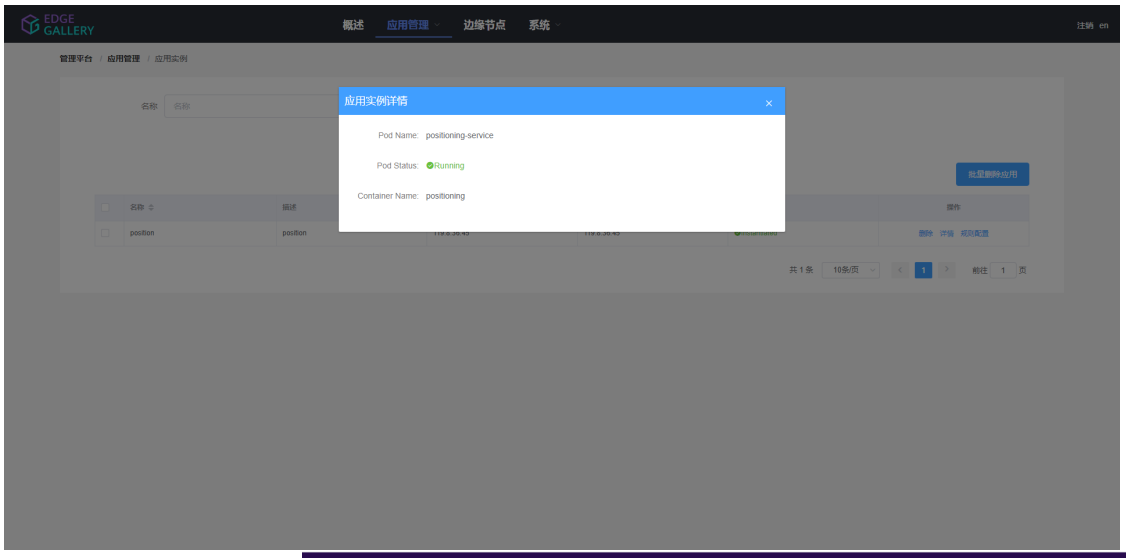


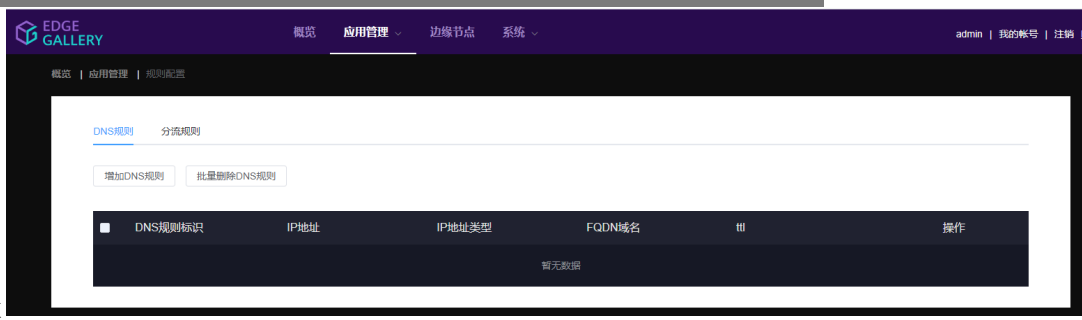


2.2 应用实例管理

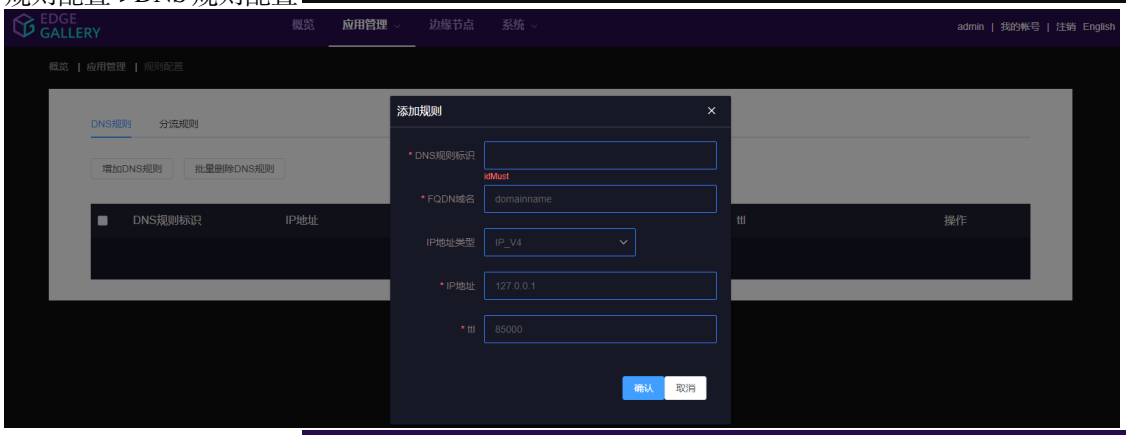
集成应用实例列表，此处可以查看当前实例的部署状态以及详情，并且进行规则配置，以及失败原因查看等功能

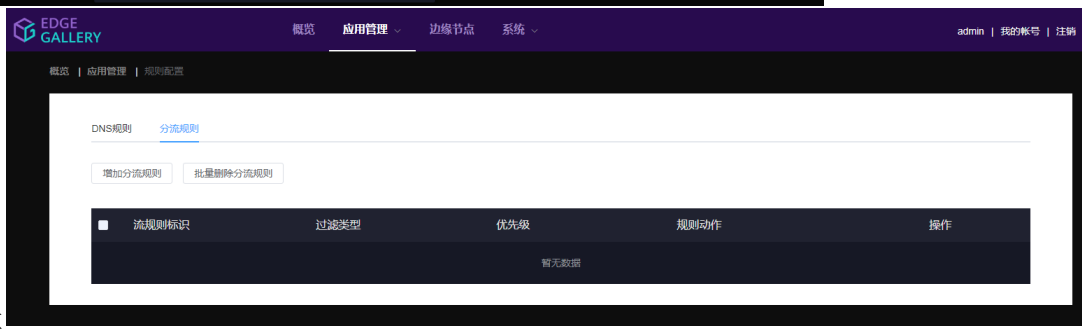




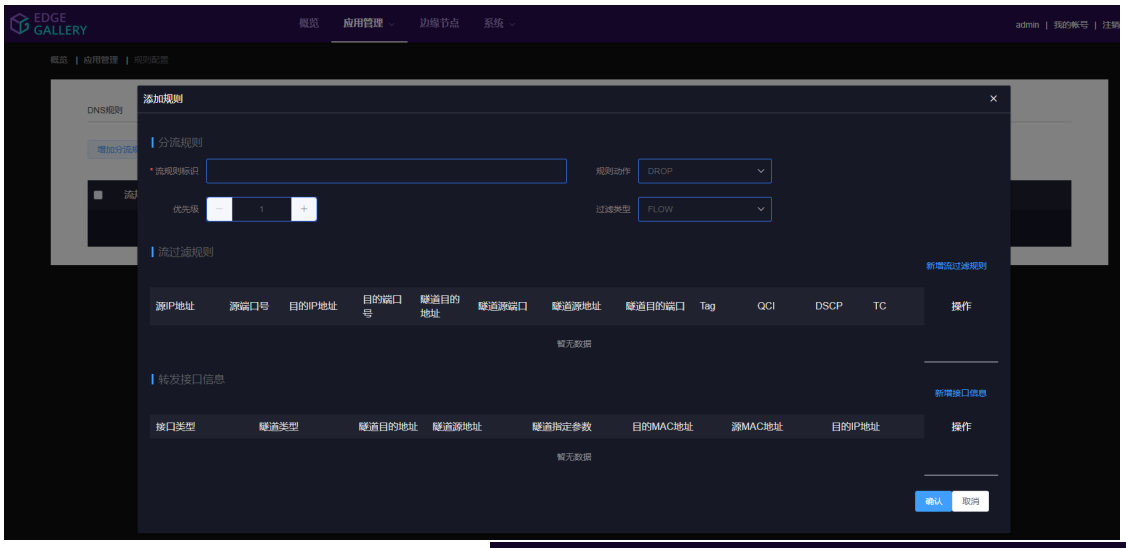


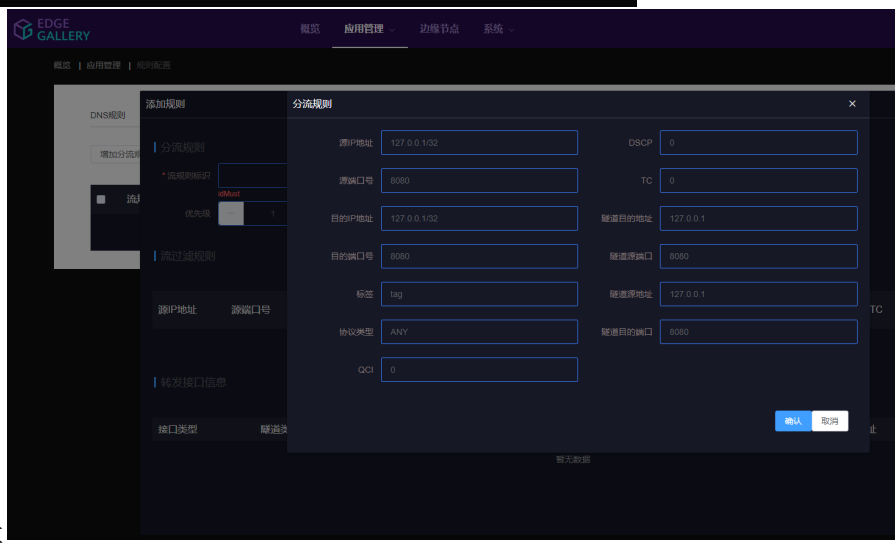
规则配置->DNS 规则配置



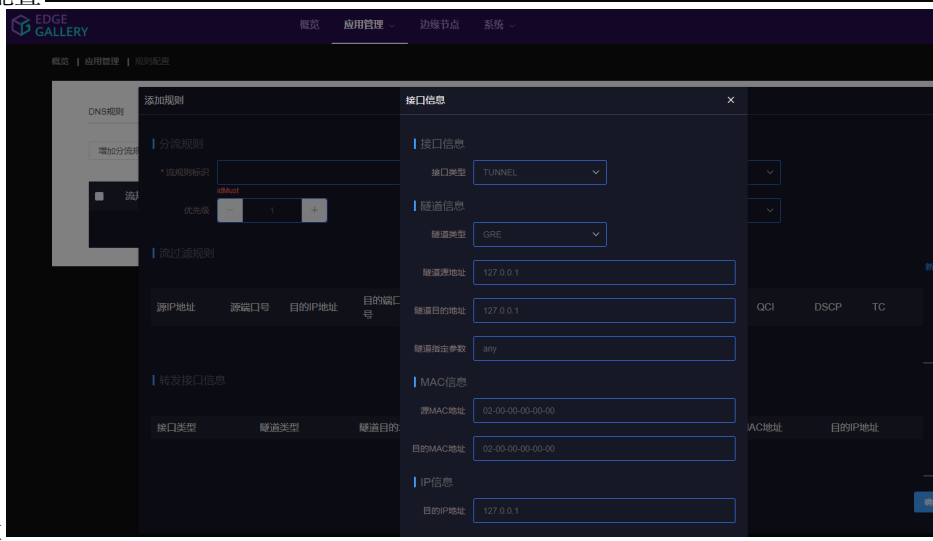


规则配置-> 分流规则配置



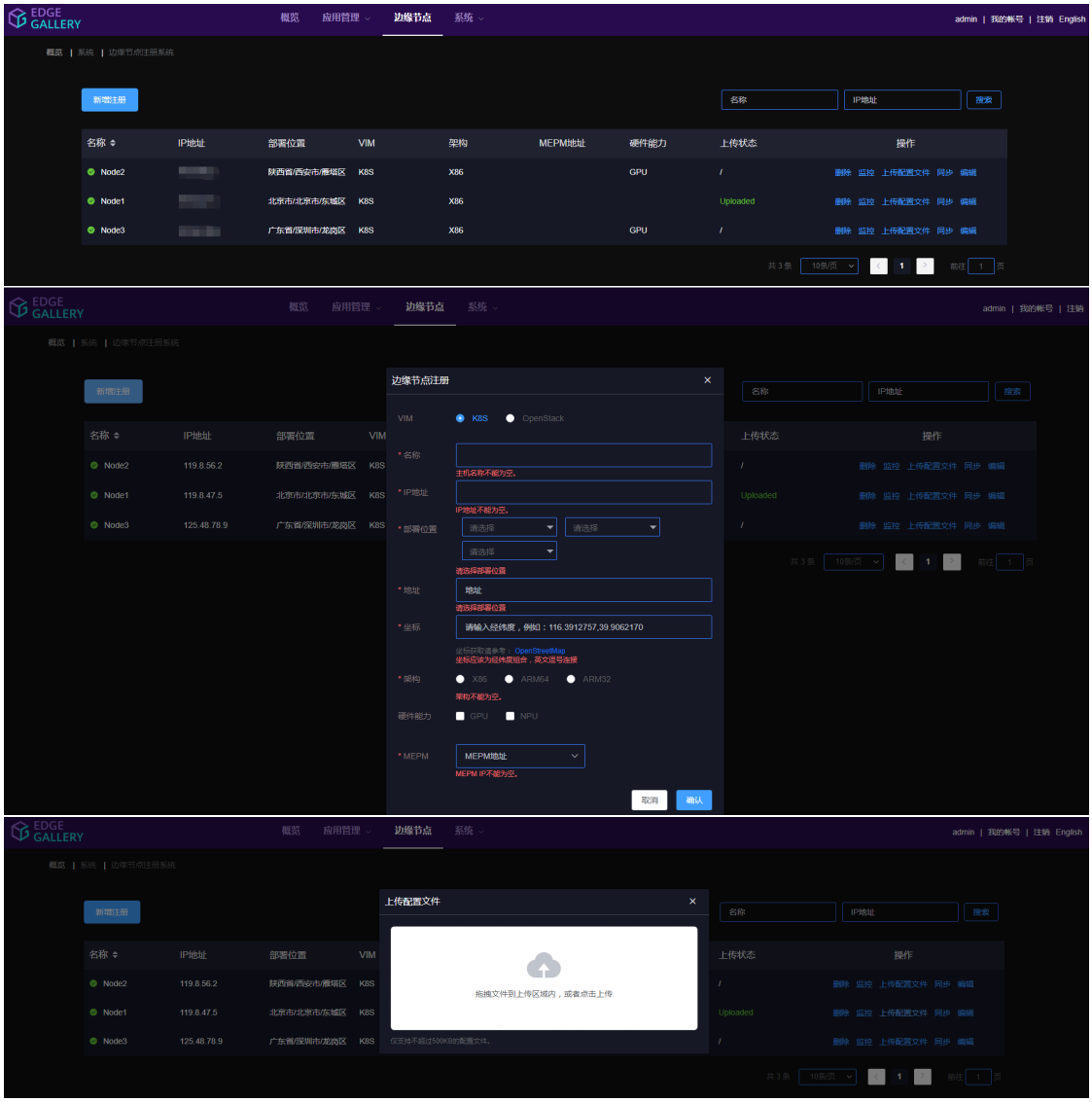


规则配置-> 分流规则配置-> 过滤规则配置



规则配置-分流规则配置-接口信息配置

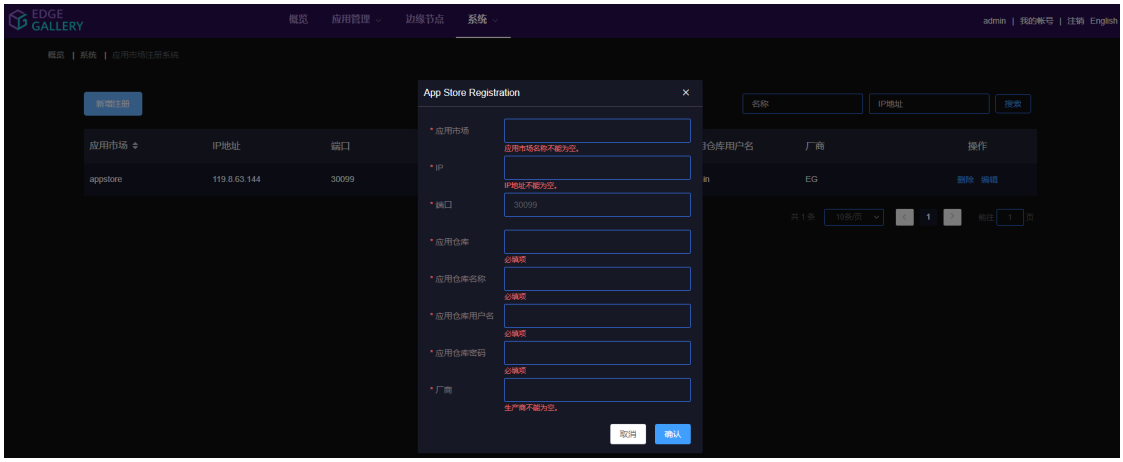
3. 边缘节点：边缘节点列表，边缘节点注册，上传 k8s config 文件



4. 系统管理：APPLCM 注册、APP Rule MGR 注册、应用仓库注册

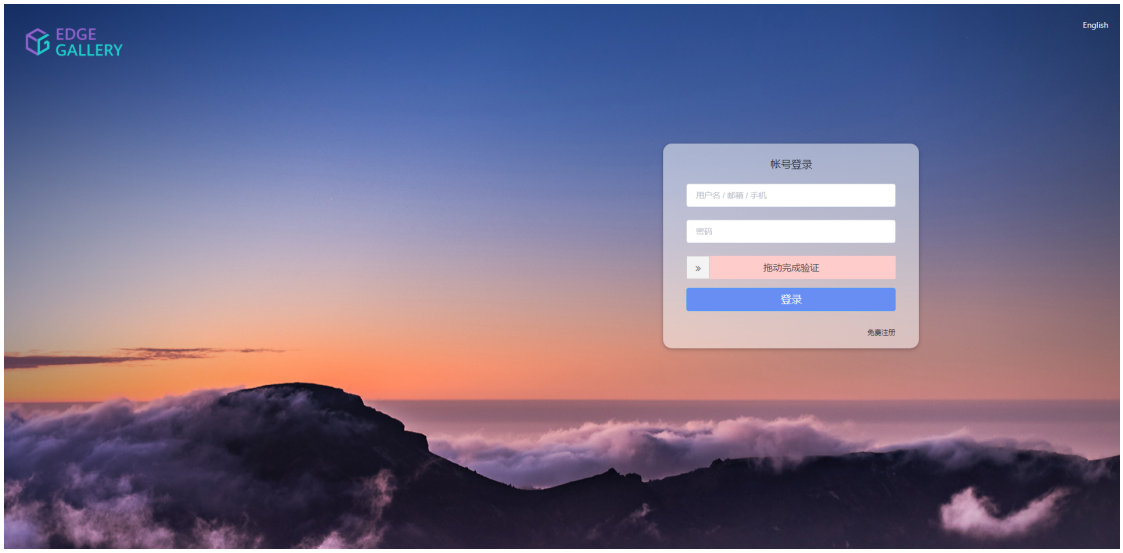
4.1 MEPM 注册

4.4 应用仓库注册

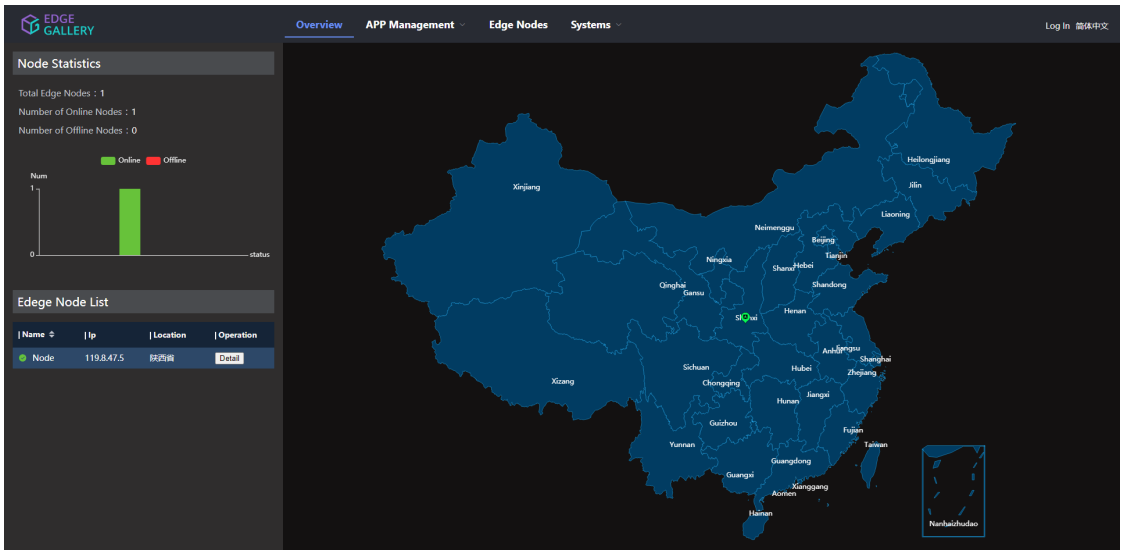


9.5.2 MECM User Interface Document

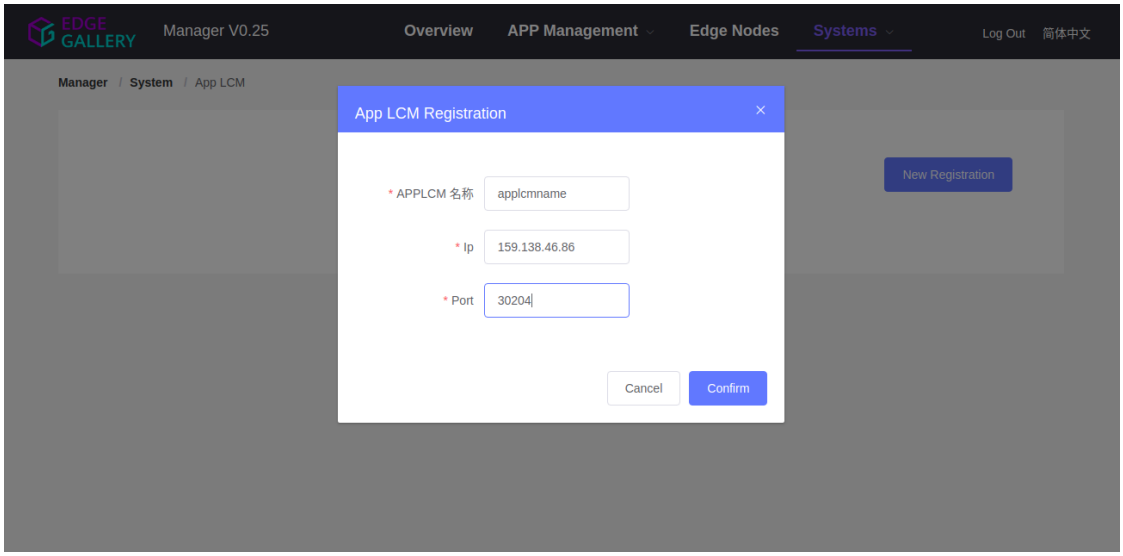
1. Register new account for the EdgeGallery in case of first time user, or else you can login with existing account, below is the example.



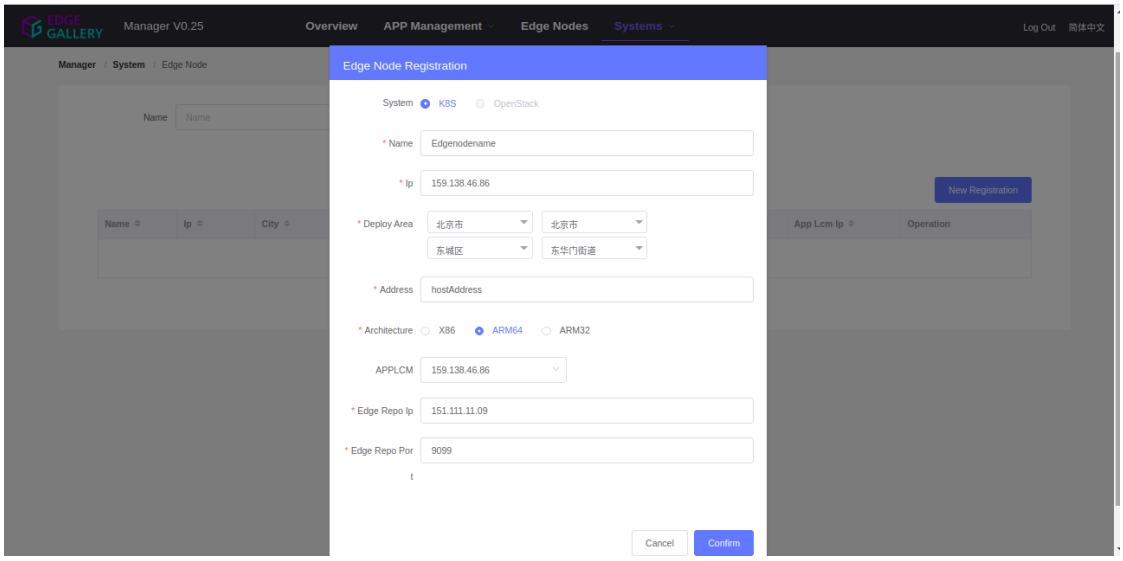
1. Upon successful login the home page is visible with map and nodes in the region.



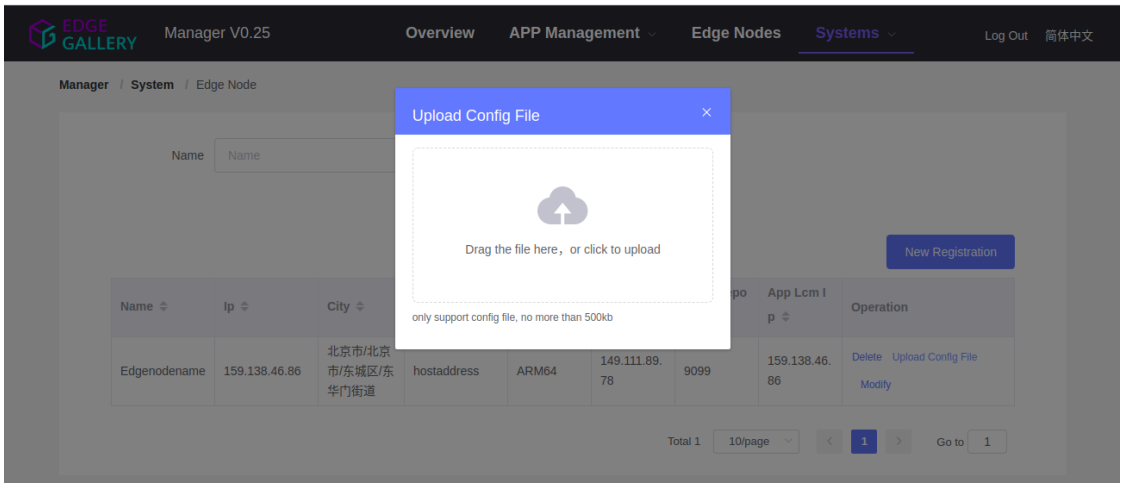
1. For new applcm registration, click on the tab ie. Systems -> External System Manager -> Applcm. Then select New Registration. Enter the Applcm, IP, Port number.



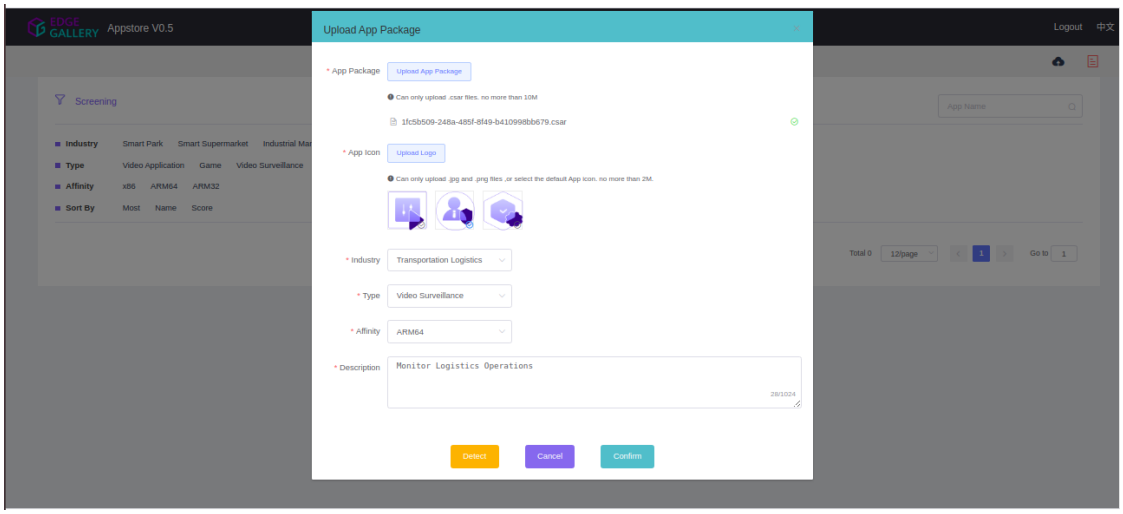
1. Need to add Edge node deatils by navigating to Systems -> External System Manager -> EdgeNode. Then click on New Registration.



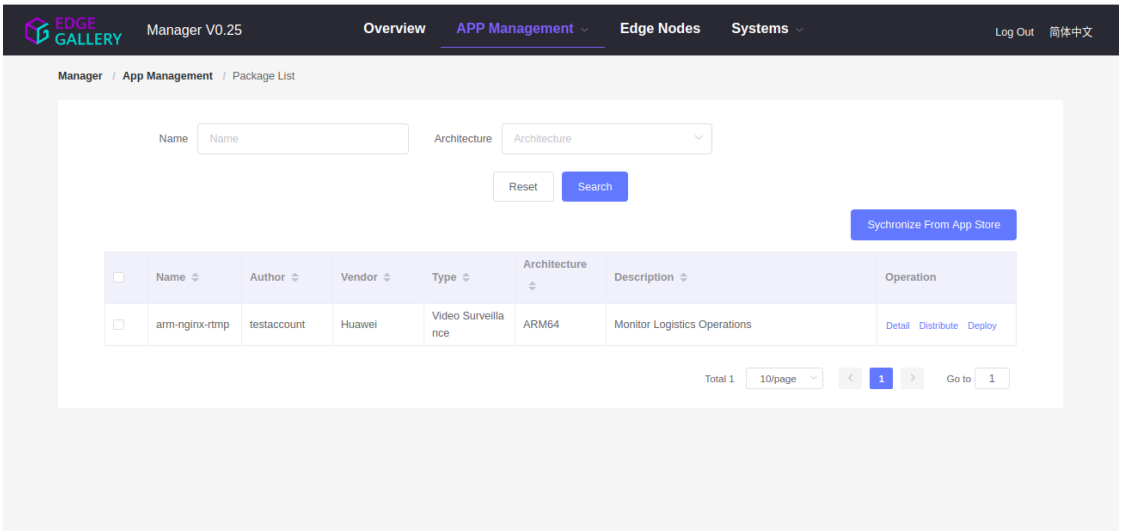
1. Config file of the edge node should be uploaded, by selecting the upload config file under Operation.



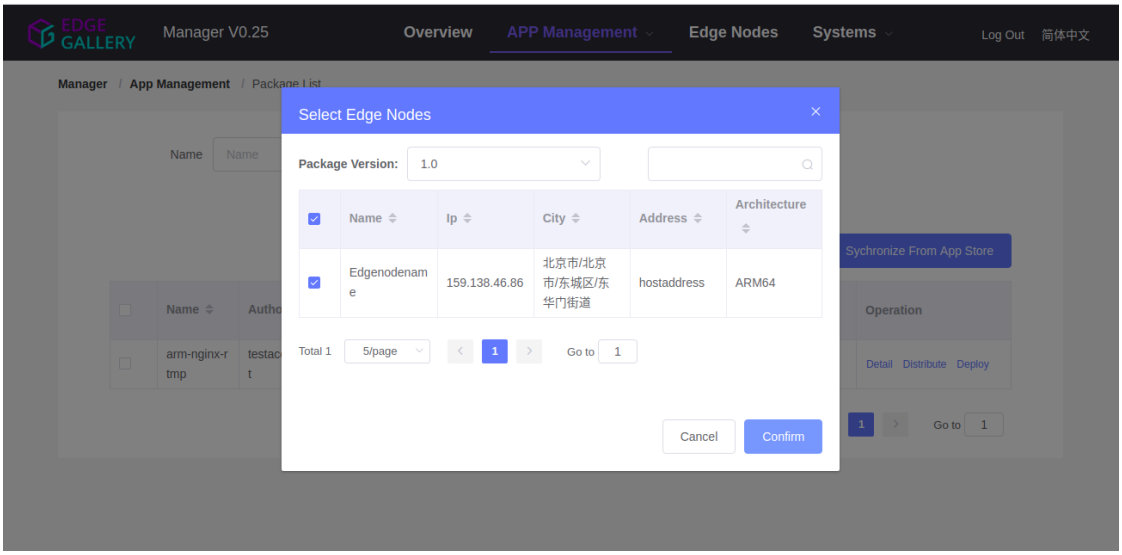
1. Now, need to login to the appstore portal and upload the package



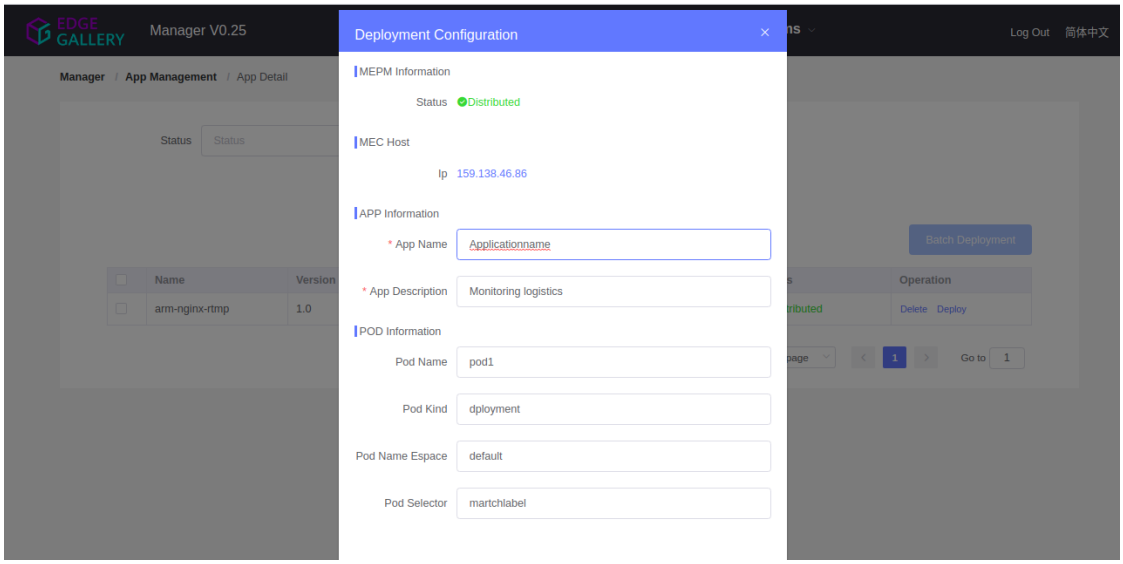
1. Nextly, Redirecting to the edge gallery mecm from appstore navigate to AppManagement -> Package Management



1. Choose the appropriate package and select the Distribute and select required Edge Node.



1. Once Distribution is successfull, then the deploy the application.



9.6 How to Contribute 如何贡献

9.6.1 Contribution to APM

This section provides details on how to participate/contribute to the APM(Application package manager) project.

Build local development environment

This section will guide developers to build APM module locally and help developers to quickly understand and develop code

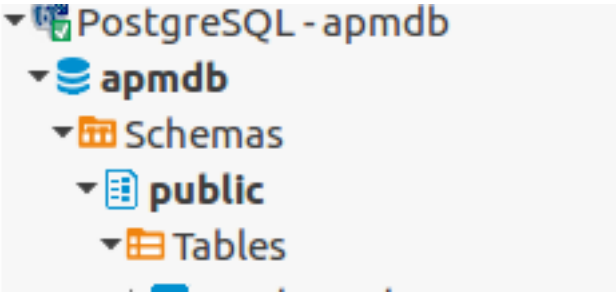
The local startup of APM depends on：

```
- DataBase
```

In order to build a local environment, the final startup sequence is Mecm-Prerequisite-> DataBase -> APM

DataBase

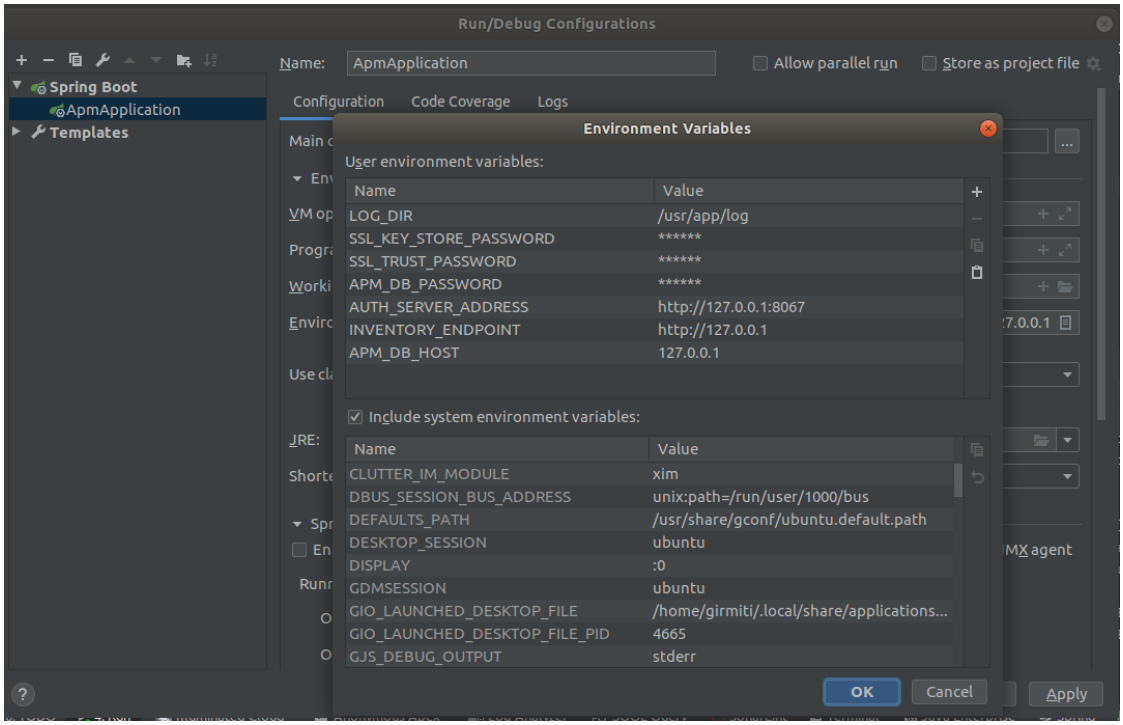
1. Download and install postgres 12.3 database
2. Set login name and password, such as apm
3. Create a database named apmdb



Application package manager

1. Download APM Code

2. Configure the runtime environment variables (using IDEA as an example)



1. Update key-store and trust-store path in application.yaml

```
# Keystore
key-store: /usr/app/ssl/keystore.p12
key-store-password: ${SSL_KEY_STORE_PASSWORD:}
key-store-type: ${SSL_KEY_STORE_TYPE:PKCS12}
key-alias: ${SSL_KEY_ALIAS:edgegallery}
trust-store: /usr/app/ssl/keystore.jks
trust-store-password: ${SSL_TRUST_PASSWORD:}
```

1. Run ApmApplication to start APM

9.6.2 Contribution to Lcm Controller and K8s Plugin (Applcm)

This section provides details on how to participate/contribute to the LCM controller project.

Build local development environment

This section will guide developers to build LCM controller module locally and help developers to quickly understand and develop code

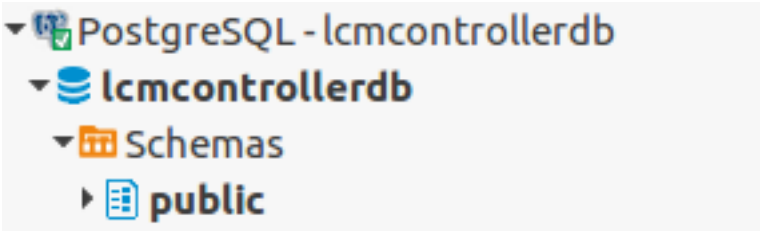
The local startup of LCM controller depends on：

```
- DataBase
```

In order to build a local environment, the final startup sequence is DataBase -> LCMController

DataBase

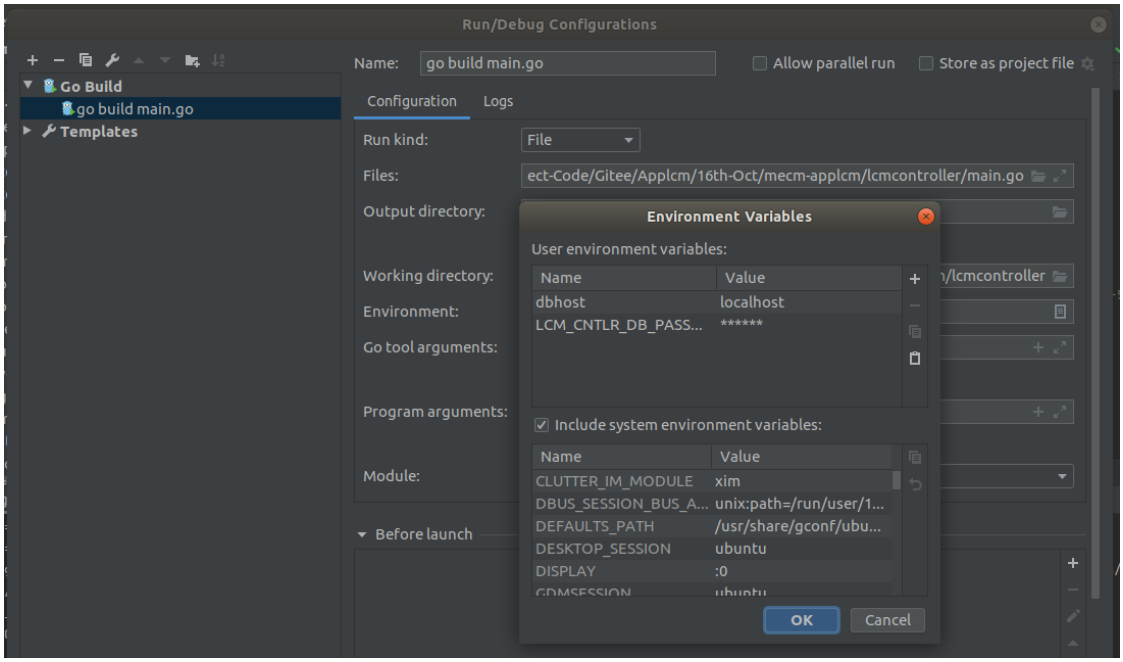
1. Download and install postgres 12.3 database
2. Set login name and password, such as lcmcontroller
3. Create a database named lcmcontrollerdb



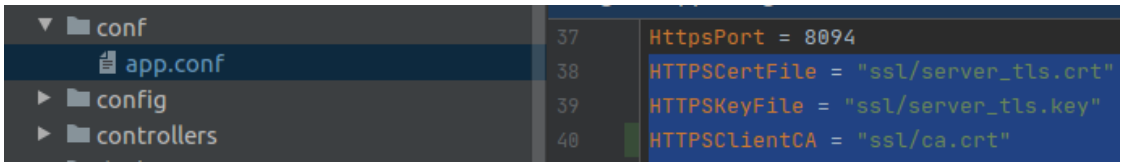
LCM contorller

1. Download LCM Controller Code

2. Configure the runtime environment variables (using GO Land as an example)



1. Update certificates path in app.conf

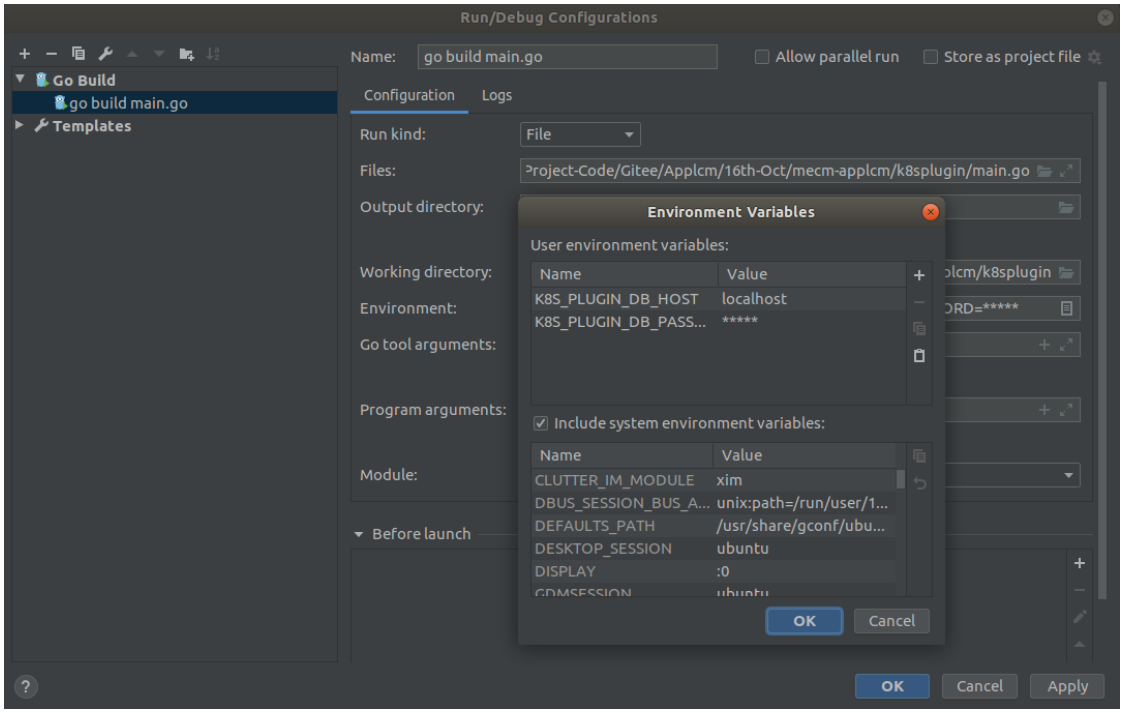


1. Run main to start LCM controller

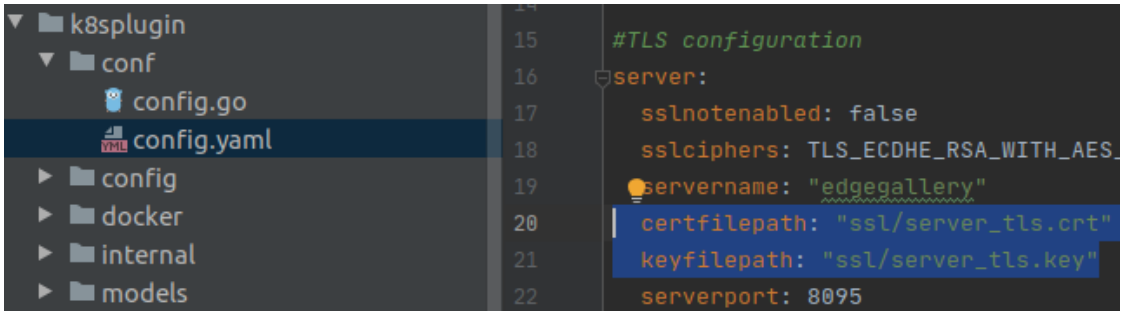
K8s plugin

1. Download K8s plugin code

2. Configure the runtime environment variables (using GO Land as an example)



1. Update certificates path in config.yaml



1. Run main to start K8s plugin.

How to generate swagger documentation for Lcm Controller

Download LCM Controller Code

Prerequisites

Step 1 - Install go lang

```
cd ~

curl -O https://dl.google.com/go/go1.10.3.linux-amd64.tar.gz
```

(下页继续)

(续上页)

```
tar xvf go1.10.3.linux-amd64.tar.gz

sudo chown -R root:root ./go

sudo mv go /usr/local

sudo nano ~/.profile

export GOPATH=$HOME/mecm-applcm

export PATH=$PATH:/usr/local/go/bin:$GOPATH/bin

export GOROOT=$HOME/go

export GOPATH=$HOME/mecm-applcm

export PATH=$PATH:$GOROOT/bin:$GOPATH/bin

source ~/.profile
```

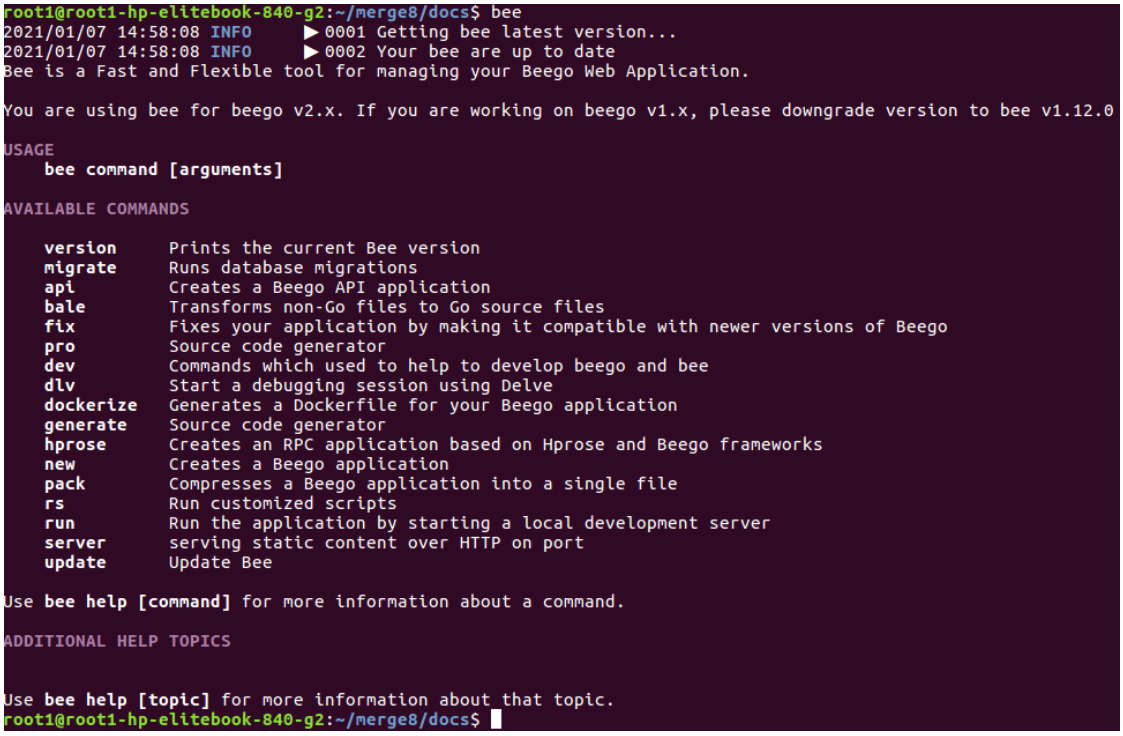
Step 2 - Install bee

```
You will need to install or upgrade Beego and the Bee dev tool:

go get -u github.com/beego/beego/v2

go get -u github.com/beego/bee/v2

bee
```




```
Change runmode configuration to dev from prod in app.conf file

Now coming to the auto-generation of documentation, end the server' s session␣
↪and use this command:

bee run -gendoc=true -downdoc=true

Now, go to the browser and type:

https://localhost:8094/swagger/
```

9.6.3 Contribution to Appo

This section provides details on how to participate/contribute to the application orchestrator project.

Build local development environment

This section will guide developers to build Appo module locally and help developers to quickly understand and develop code

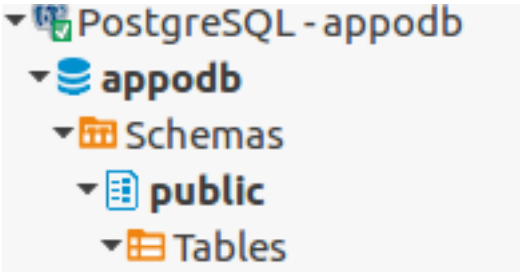
The local startup of Appo depends on：

```
- DataBase
```

In order to build a local environment, the final startup sequence is Mecm-Prerequisite -> DataBase -> Appo

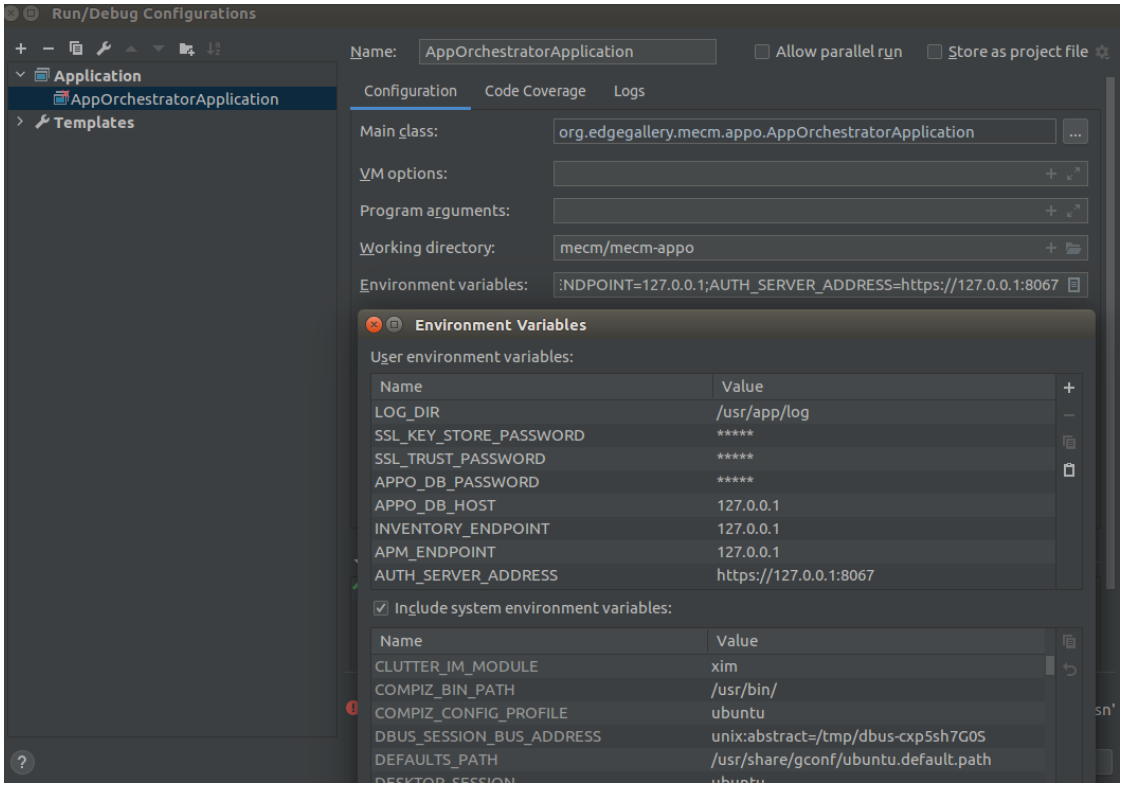
DataBase

1. Download and install postgres 12.3 database
2. Set login name and password, such as appo
3. Create a database named appodb



Application orchestrator

1. Download Appo(application orchestrator) code
2. Configure the runtime environment variables (using IDEA as an example)



1. Update key-store and trust-store path in application.yaml

```
# Keystore
key-store: /usr/app/ssl/keystore.p12
key-store-password: ${SSL_KEY_STORE_PASSWORD:}
key-store-type: ${SSL_KEY_STORE_TYPE:PKCS12}
key-alias: ${SSL_KEY_ALIAS:edgegallery}
trust-store: /usr/app/ssl/keystore.jks
trust-store-password: ${SSL_TRUST_PASSWORD:}
```

1. Run AppOrchestratorApplication to start application orchestrator

9.6.4 Contribution to Inventory

This section provides details on how to participate/contribute to the inventory project.

Build local development environment

This section will guide developers to build Inventory module locally and help developers to quickly understand and develop code

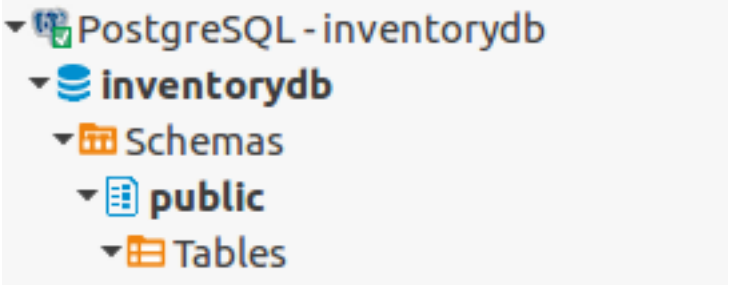
The local startup of Inventory depends on：

```
- DataBase
```

In order to build a local environment, the final startup sequence is Mecm-Prerequisite -> DataBase -> Inventory

DataBase

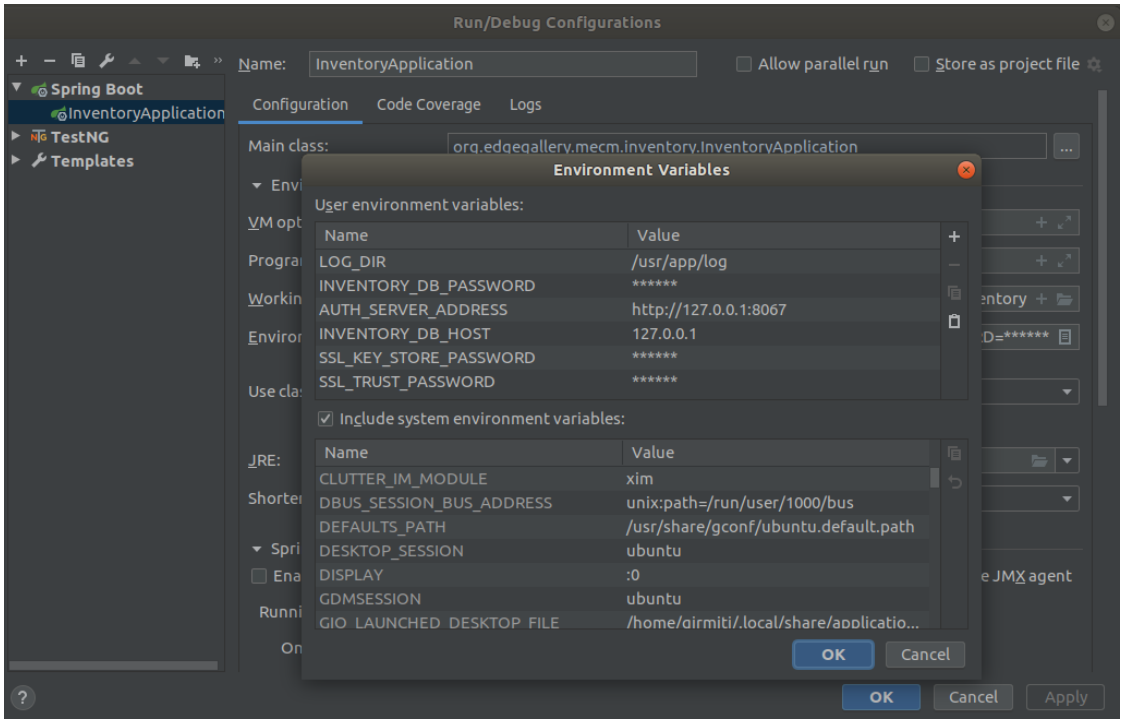
1. Download and install postgres 12.3 database
2. Set login name and password, such as inventory
3. Create a database named inventorydb



Inventory

1. Download Inventory Code

2. Configure the runtime environment variables (using IDEA as an example)



1. Update key-store and trust-store path in application.yaml

```
# Keystore
key-store: /usr/app/ssl/keystore.p12
key-store-password: ${SSL_KEY_STORE_PASSWORD:}
key-store-type: ${SSL_KEY_STORE_TYPE:PKCS12}
key-alias: ${SSL_KEY_ALIAS:edgegallery}
trust-store: /usr/app/ssl/keystore.jks
trust-store-password: ${SSL_TRUST_PASSWORD:}
```

1. Run InventoryApplication to start inventory

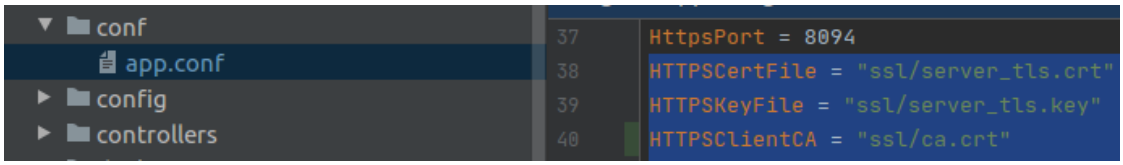
9.6.5 Contribution to App Rule Manager

This section provides details on how to participate/contribute to the App Rule Manager project.

Build local development environment

This section will guide developers to build App Rule Manager module locally and help developers to quickly understand and develop code

1. Download App Rule Manager Code
2. Update certificates path in app.conf



1. Run main to start App Rule Manager

9.7 Port Matrix 端口矩阵

9.7.1 MECM 端口矩阵

模块	端口号	外部端口号	作用描述
APPO	8091	30201	对外提供应用生命周期管理的服务端口
APM	8092	30202	对外提供应用包管理的服务端口
Inventory	8093	30203	对外提供存量管理的服务端口
applcm controller	8094	30204	applcm 中对外提供应用生命周期管理的服务端口
aplcm k8splugin	8095	30205	applcm 中的 k8s 插件对外提供 k8s 的服务端口
FE	8443	30093	前台模块的对外开放端口
AppRuleManager	8096	30206	提供用于应用程序规则管理的外部服务端口

CHAPTER 10

MEP

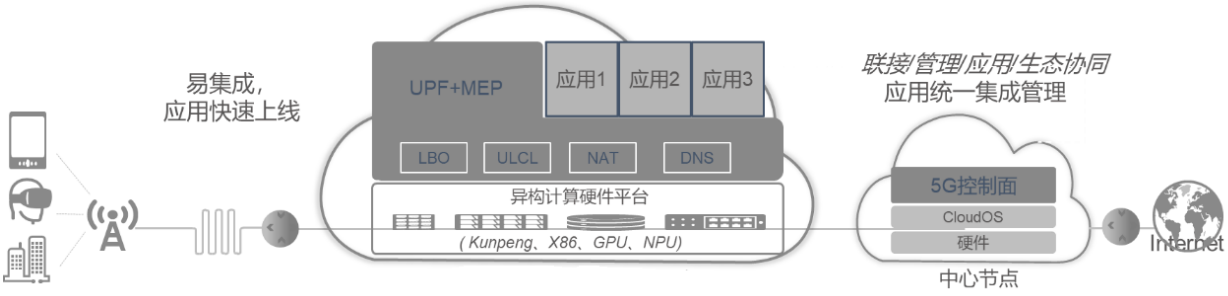
10.1 概述

10.1.1 MEP 项目整体介绍

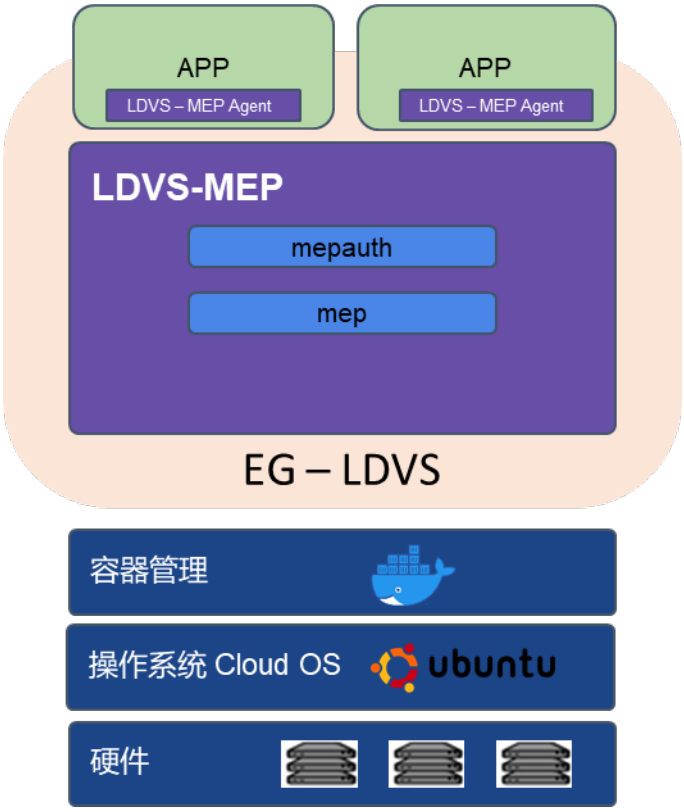
随着物联网、人工智能、云计算、移动互联网、大数据和大视频等产业技术的蓬勃发展，以及围绕 ICT 开放生态的成熟，网络资源和计算能力逐步朝着资源集中化和边缘化方向演进。

多接入边缘计算 MEC（Multi-access Edge Computing）为典型的资源边缘化模式，在移动网络边缘提供 IT 服务环境和云计算能力，实时完成移动网络边缘的业务处理。MEC 将随着 CT 和 IT 深度融合趋势，物联网的兴起、人工智能技术的发展，以及企业对生产数据的安全性、实时性的诉求，持续快速的发展。

在 MEC 场景下，海量的应用将运行在网络边缘进行业务处理，并且应用能够使用网络的开放能力，应用之间也能够互相进行能力提供和消费。



EdgeGallery MEP 项目旨在打造一个边缘侧的开源、开放的参考 MEP 平台，MEP 项目当前包含 MEP 和 MEP-Agent 两个组件。



MEP（MEC Platform, MEC 平台）是 ETSI MEC 标准定义的 MEC 平台网元，部署位置在边缘侧，主要涵盖如下功能：

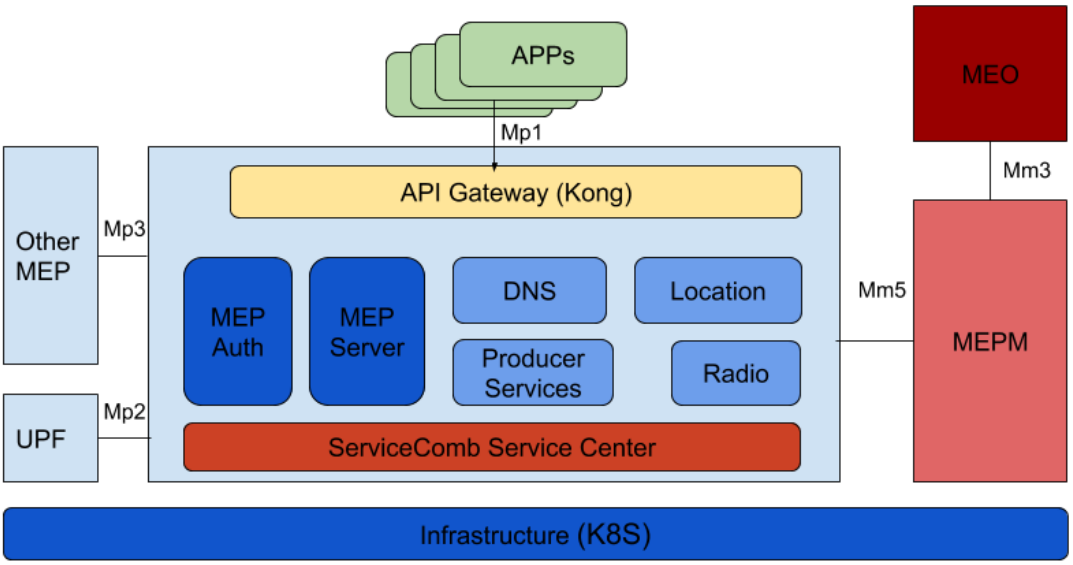
- 为 ME APP 提供服务治理能力（注册、发现、订阅、通知）。
- 接收 MEPM 下发的流规则（Traffice Rules）配置。
- 接收 MEPM 下发的 DNS 规则（DNS Rules）配置。
- 为 MEC APP 提供 DNS 等 MEC 公共服务。

MEP 组件运行时包含如下微服务：

- **mepauth** 为应用提供认证鉴权，同时承担部分配置管理的能力。
- **mep** 提供服务治理，app 订阅通知管理的相关功能。
- **kong** 采用了开源网关软件 kong，负责相关路由转发和插件配置等功能。
- **postgres** postgres 为 kong 和 mepauth 提供数据存储能力。

MEP-Agent 组件应用集成适配器，这个组件在实际使用时，应用开发者可以将其作为应用的一部分。它实现了实例配置信息导入，服务注册等相关基础功能，可以简化应用与平台的集成。

10.1.2 MEP 整体架构



图中涉及的 MEP 关联主要接口有：

- **Mp1** APP 与 MEP 之间，提供 APP 服务注册发现，APP 状态通知订阅等能力。
- **Mp2** MEP 与 UPF 之间，提供数据面的配置能力。
- **Mm3** MEO 与 MEPM 之间，提供包管理，APP 生命周期管理能力
- **Mm5** MEPM 与 MEP 之间，执行 MEP 平台配置管理，配置 APP 规则等能力。

对于应用 App 来说，Mp1 是 APP 与 MEP 交互最重要的接口，APP 能通过 Mp1 将自身服务注册到 MEP 平台，同时也能够通过服务发现调用 MEP 对外提供的服务。

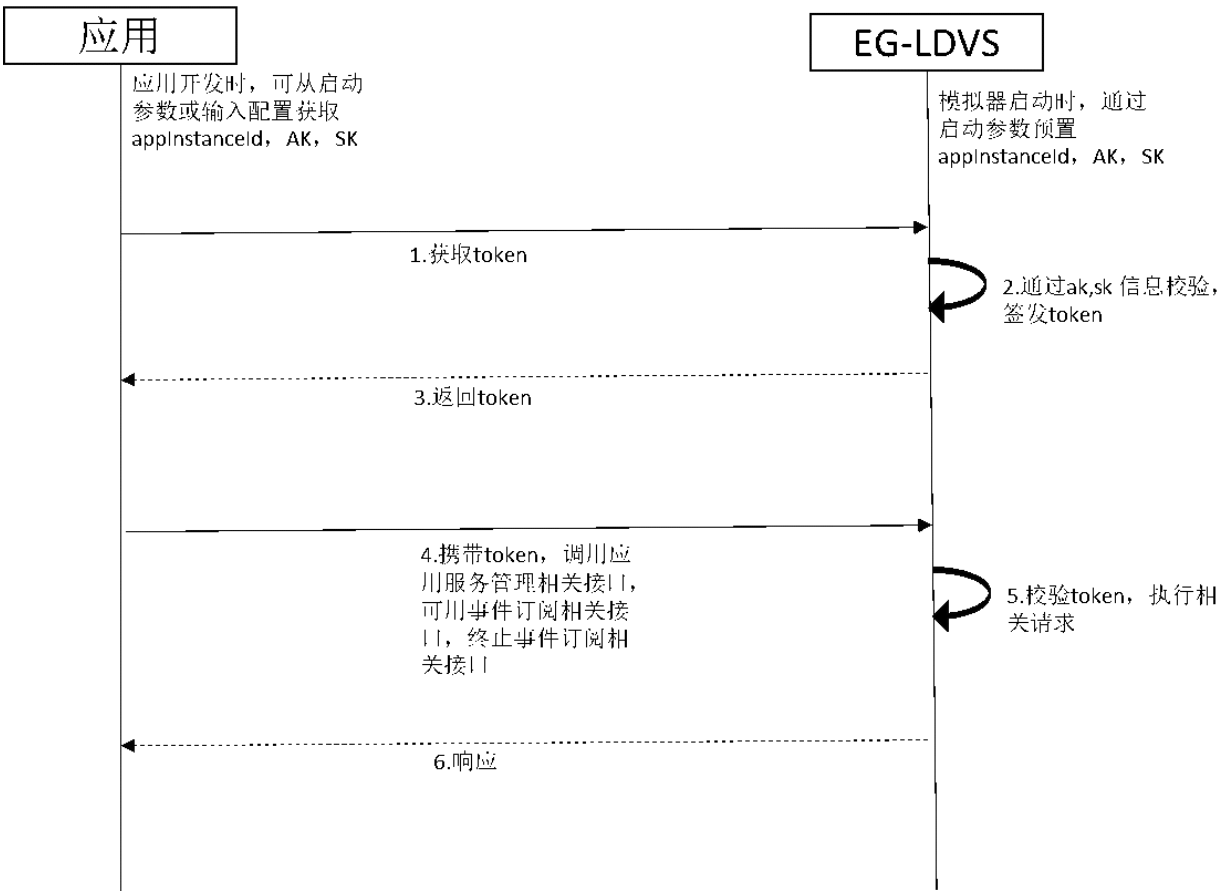
10.2 特性设计

10.2.1 MEP-auth 模块相关特性

MEP-auth 为 APP 提供认证鉴权功能

MEP-auth 为 APP 提供认证鉴权功能，提供 token 申请接口，APP 可以基于 AK/SK 签名算法，向 MEP-auth 提供正确的签名，获得 token，然后通过该 token 访问 MEP-server 相关接口。

应用 APP 通过 MEP-auth 申请 token 的流程如下图所示：



1. 应用使用 服务认证接口，获取 token
2. MEP-auth 对 ak/sk 进行签名校验并签发 token
3. MEP-auth 返回 token
4. 应用携带 token 进行服务接口调用。可调用 应用服务管理相关接口，可用事件订阅相关接口，终止事件订阅相关接口。
5. MEP 基于 kong 的 jwt 插件校验 token 并路由相关请求至 MEP-server。

token 申请接口详细描述请参看“MEP features”章节。

MEP-auth 为 API 网关提供配置功能

MEP-auth 模块在初始化，会首先对 API 网关（kong）进行初始化:

1. 在 kong 添加一个 consumer
2. 为 MEP-auth 自身配置 service 和 route
3. 为 MEP-server 配置 servcie 和 route
4. 为 MEP-auth 和 MEP-server 配置并启用 kong 插件

MEP-auth 在初始化 kong 过程中开启的插件包括：

- **JWT 插件**为相应接口提供 token 校验能力
- **Appid-header 插件**在接口 request 中插入 X-AppinstanceID 头，以供 MEP-server 校验；校验申请 token 的 client ip 与调用接口的 client ip 一致
- **Rate Limiting 插件**为 MEP-auth 和 MEP-server 接口提供流量控制能力
- **IP Restriction 插件**为 MEP-auth 提供客户端 IP 白名单功能支持
- **response-transformer 插件**提供清除 response 中 server header 能力
- **pre-function 插件**提供修改接口请求 x_forwarded_for 能力

10.2.2 MEP sever 特性

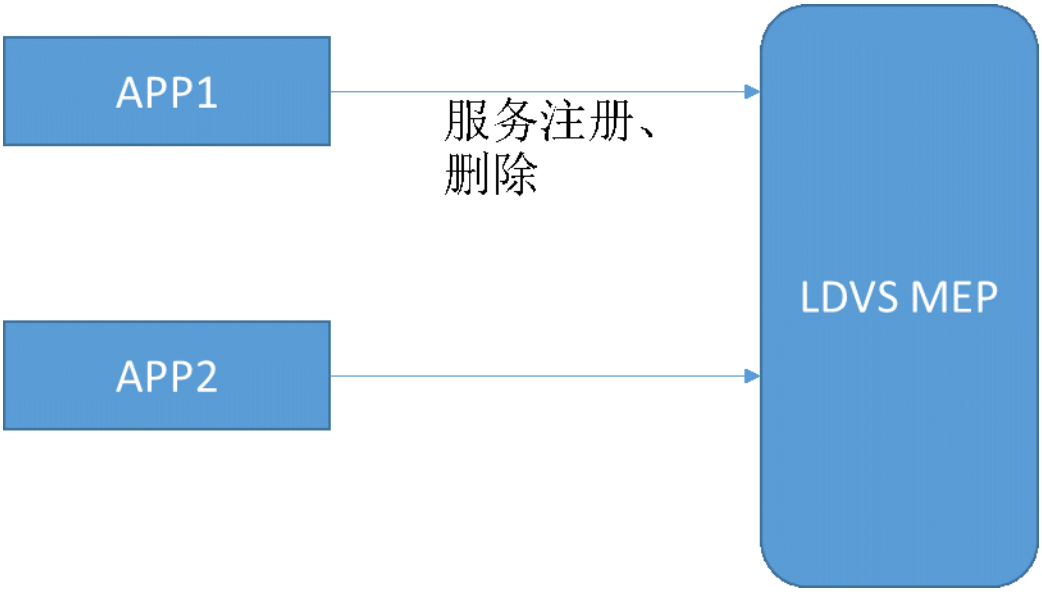
MEP 服务治理功能

定义

服务管理功能是指 MEP 能够进行服务的注册，更新，删除，查询的能力。

设计原理

MEP 提供服务注册，更新，删除，查询相关 API 接口。应用能够通过 LDVS-MEP 进行服务的注册，更新，删除，查询。



10.2.3 应用服务状态订阅/通知功能

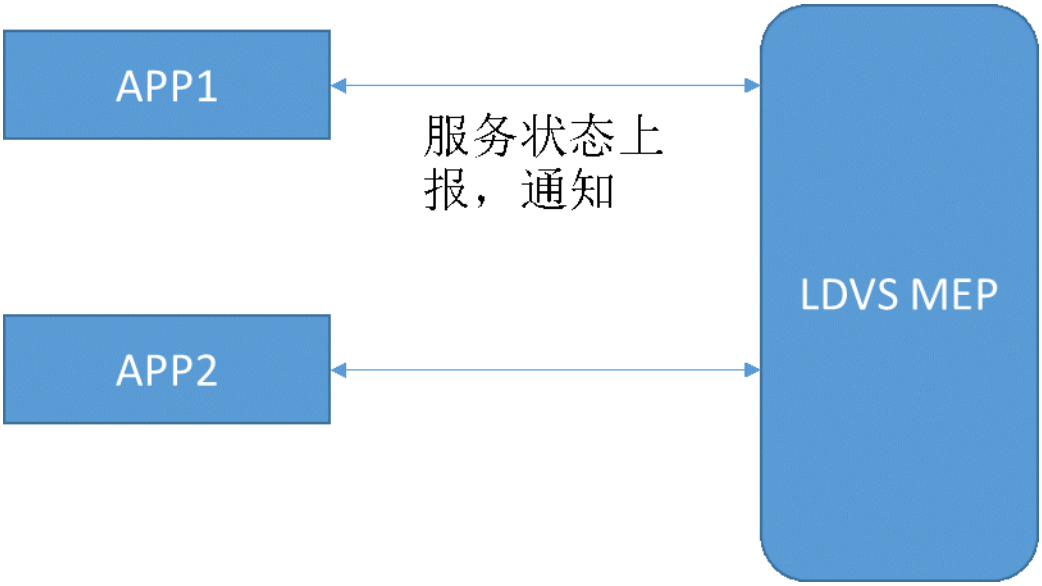
定义

应用服务状态订阅/通知功能是指 LDVS 能够进行服务状态的订阅，更新，删除，通知的能力。

设计原理

MEP 管理应用服务的状态，应用通过 MEP 进行服务的注册，当服务消费者期望使用服务供应者的服务时，需要通过 MEP 订阅服务供应者的服务状态。通过订阅接口，MEP 能够识别服务消费者关注的服务，在服务状态发生变化时，能够将相关的服务状态通知到服务消费者。

应用能够通过 LDVS-MEP 进行服务状态的订阅，更新，删除，查询。



MEP Service hearbeat

Each MEC service instance that has previously registered in MEC platform and is configured for heartbeat shall send heartbeat message to the MEC platform periodically in order to show that the MEC service instance is still operational. When the MEC platform has not received the heartbeat for a configurable amount of time, the MEC platform considers that the service instance can no longer be discovered and The service state will change to SUSPEND if it was ACTIVE before.

How to configure hearbeat for a service

Service register message will carry the livenessinterval field in it. It is an option field. If it is configured that indicate this service want to send heartbeat message. MEP platform support livenessinterval value is 60s. It will overwrite any value set by the service. Note: For details of service register message please check the interface doc.

How to send heartbeat message

Services which already has set livenessinterval in its register message should send hearbeat message to MEP. When service registers, its response message will carry the URL (liveness), using which service should send the heartbeat to MEP. Note: For details of hearbeat message please check the interface document.

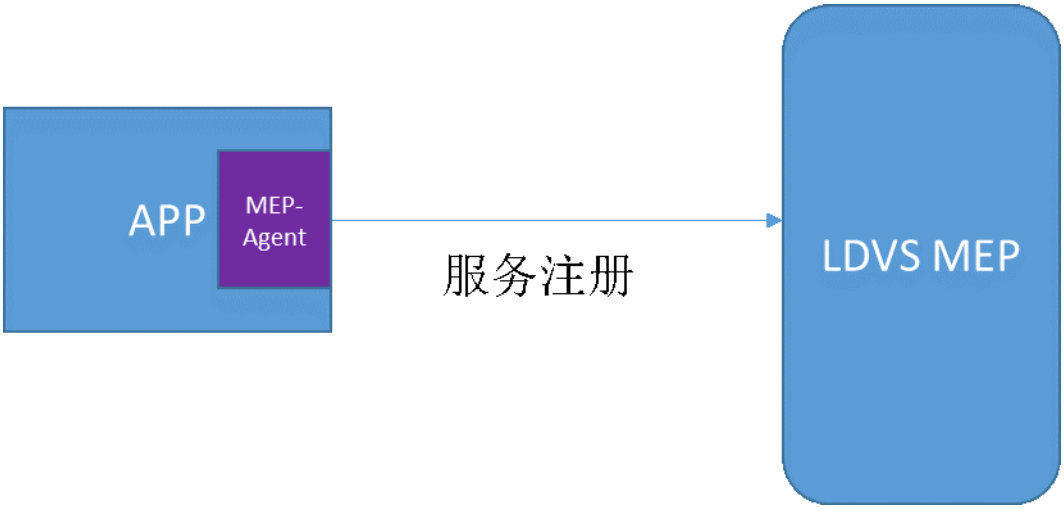
10.2.4 EG-LDVS 应用集成插件 mep-agent

定义

LDVS MEP-Agent 作为应用快速集成适配器，实现服务配置导入和服务注册的功能。

设计原理

LDVS MEP 管理应用的服务，应用需要将其服务注册到 MEP 中，MEP-Agent 作为适配器，将服务信息（包括应用实例 ID）导入给应用，同时提供配置的方式将应用的服务注册到 MEP 中。



MEP-Agent Get token API

MEP-Agent will run as sidecar along with application. If Apllication need to obtain token from mep for communication with services, it can query mep-agent api.

```
GET /mep-agent/v1/token


{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    }
  ]
}
```

Return example

```
HTTP/1.1 200 OK
{
        "access_token":"xxxx",
        "token_type":"Bearer",
        "expires_in":"3600"
}
```

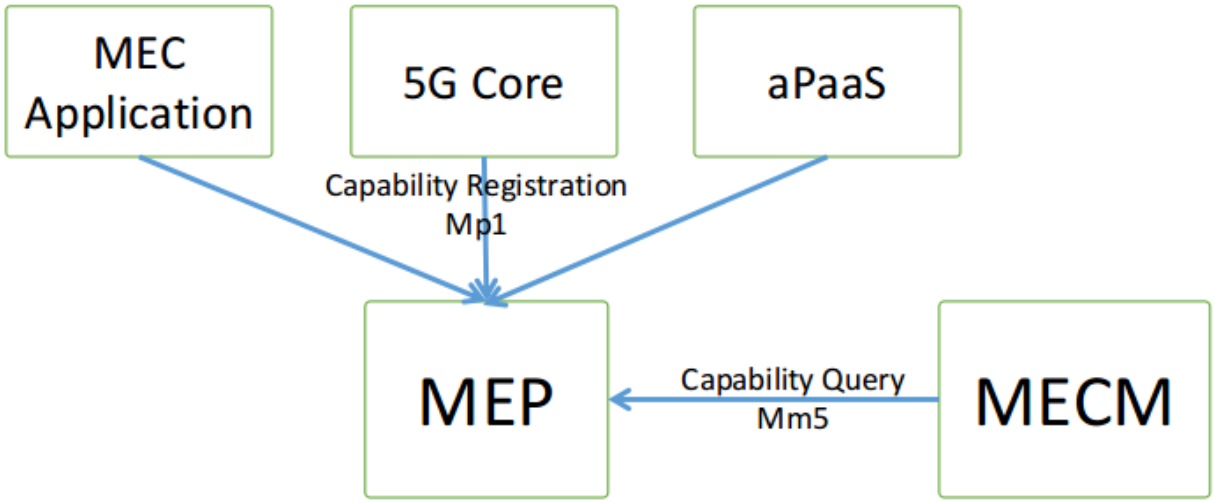
10.2.5 Capabilities(Service) query support

定义

Once capabilities(services) registered to the mep, other applications can consume it. MEP support the listing of these capabilities(services) over Mm5 interface.

设计原理

The capability query interface is exposed over Mm5 interface to MECM module and it can be queried together or independently. Each capability consists of a list of the present consumers to it.



10.2.6 DNS server 特性

The MEP platform provides the domain name resolution services to the applications deployed in the EdgeGallery MEC, which can be utilized by the device applications in UE. MEC platform receives the DNS configurations from the MEC management, which includes the FQDN(Fully Qualified Domain Name), IP address and related entries. MEC applications can later query or modify the state of these configurations.

The three major operations for DNS are:

1. DNS management operations.
2. Query configuration and Activation/Deactivation by the MEC applications.
3. DNS query by the device applications.

We will see each of these features in the below section.

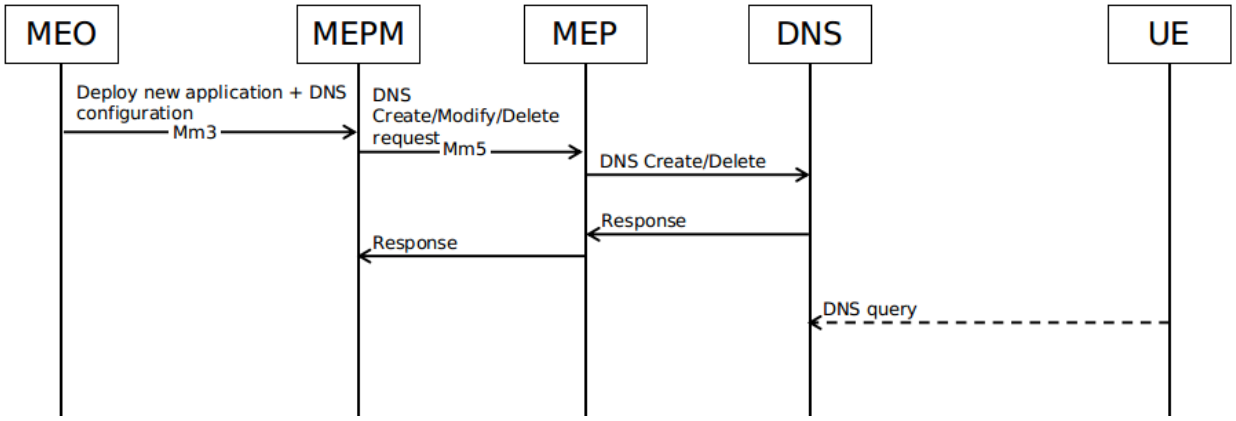
DNS management operations

DNS configurations are created either by MECM modules during start-up or from the OSS directly. DNS management supports both create, update, query and delete operations.

Rule creation at start-up

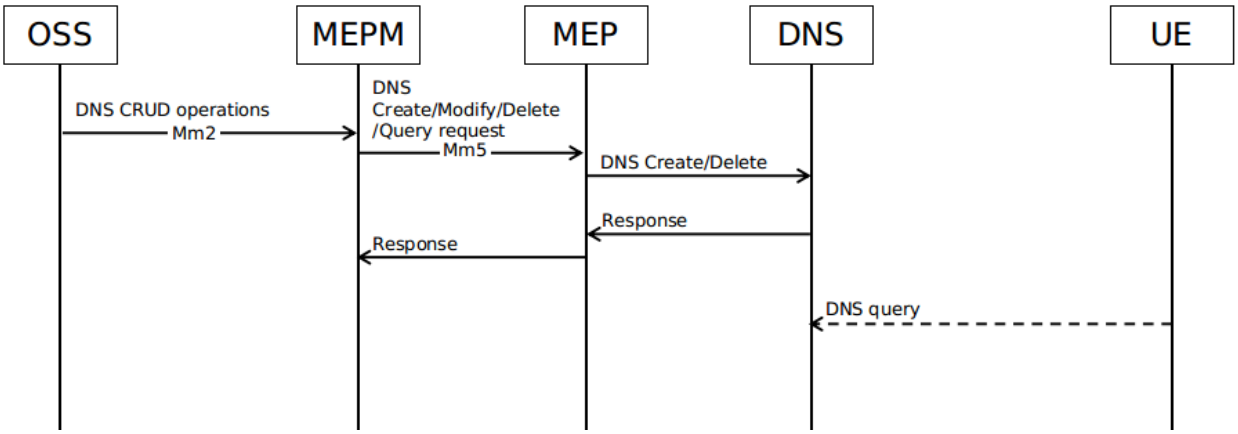
DNS configurations are embedded in the application descriptor and MEO(Multi-access Edge Orchestrator) will get these information during the start-up of the application. Once MEO receives the application descriptor, it searches for the DNS related configurations and pass this information to the MEPM(MEC Platform Manager) module in the edge along with the start-up request on Mm3 interface.

Upon receiving the DNS configurations from MEO, MEPM send this request to the MEP server on Mm5 interface after deploying the MEC application.



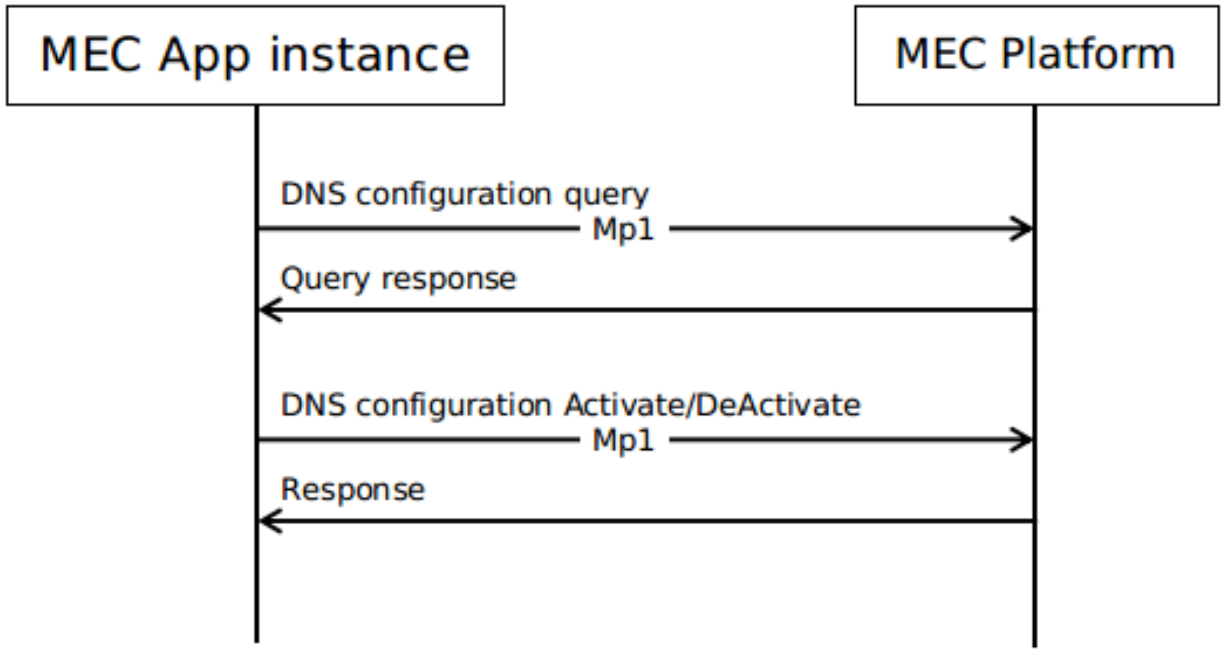
Rule creation from OSS

DNS configurations can be updated from OSS directly over Mm2 interface through MEPM.



DNS configurations from MEC applications

MEC applications can query the DNS configurations created for it and can activate or deactivate the same. Activation or deactivation can be performed by modifying the state of the DNS configuration.



DNS query

Device application in the UE can query the DNS server for the domain name resolution. By default the DNS server will listen on the 53 port.

How to configure the DNS server?

The dns server in MEP runs inside a container and can be configured using few command line parameters.

Command

```
dnsserver [OPTION]...
```

Detailed list of the optional parameters are stated below.

Parameter	Type	Range/Length	Default	Description
-db	string	1~256	dbEgDns	Backend store db name
-port	number	1~65535	53	DNS server listening port
-managementPort	number	1~65535	8080	Management interface listening port
-connectionTimeout	number	2~50	2	Connection timeout
-ipAdd	string	Ipv4/Ipv6	0.0.0.0	DNS listening ip address
-managementIpAdd	string	Ipv4/Ipv6	0.0.0.0	Management interface listening ip address
-forwarder	string	Ipv4/Ipv6		DNS proxy server IP
-loadBalance	bool	true/false	false	For domain names with multiple ip address, enabling this option will perform loadbalancing by shuffling the response IP list

Example

```
$ dnsserver -port=8053 -managementPort=8080 -loadBalance -forwarder <some-dns-server>
```

10.2.7 HealthCheck

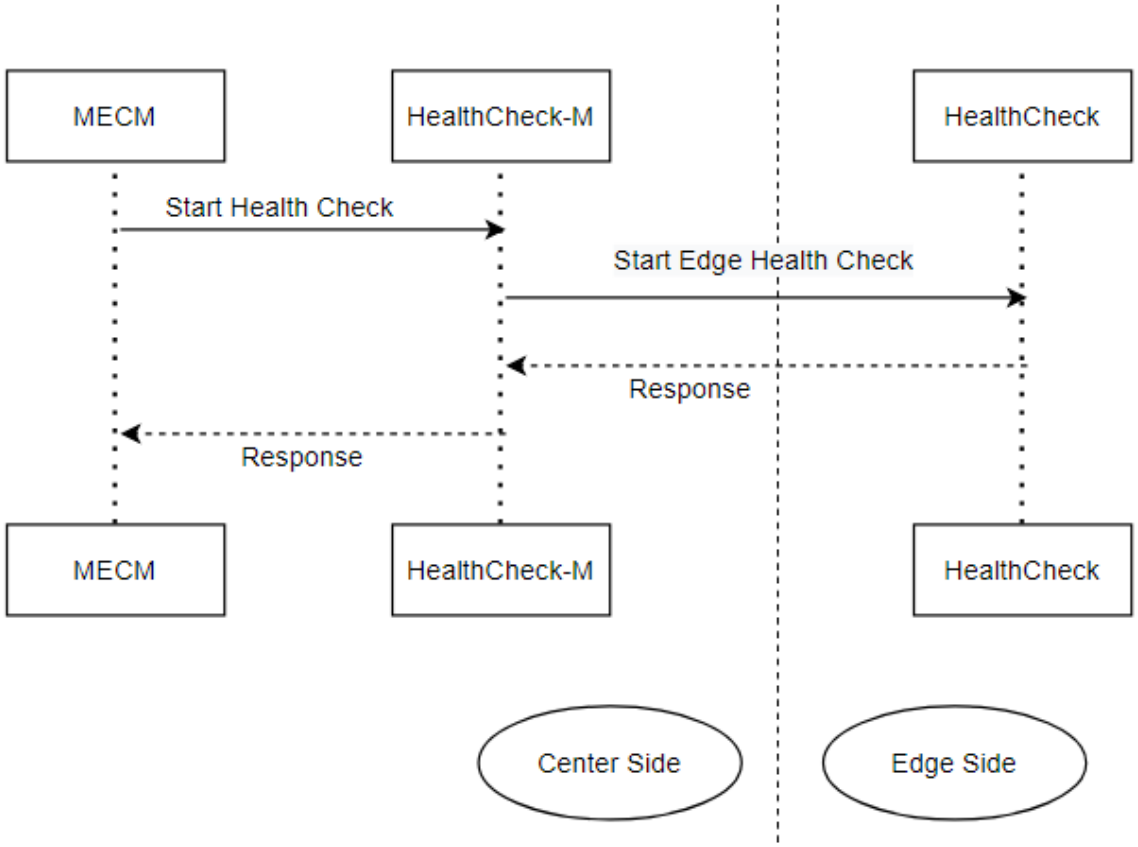
需求来源

*EdgeGallery 版本规划

需求描述（Requirement Description）

*Support mec-m check its edge healthy condition

Usecase



1. MECM 调用中心侧服务 HealthCheck-M，启动整体健康检查服务
2. 中心侧 HealthCheck-M 根据 MECM 调用时 body 内的 mec list 检查所辖的边缘节点
3. 各边缘节点 HealthCheck 收到中心侧 HealthCheck-M 的检查信息，先自查内部 mep/lcm 的健康情况，当边缘侧的 mep/lcm 均健康时，此边缘节点则认为本边缘节点健康
4. 完成自查后，边缘节点间启动互查机制，被访问的节点 mep/lcm 均健康则认为节点健康
5. 各边缘节点把通讯后互查结果反馈回 HealthCheck-M，HealthCheck-M 启动投票机制
6. 中心侧 HealthCheck-M 投票则是，当某节点收到其他节点的通讯成功的结果超过 mec list 总数的一半时中心侧认为此节点健康
7. HealthCheck-M 把投票结果反馈给 MECM 或调用方

涉及模块 &&EPIC&&Story

*MECM

* 调用方启动 HealthCheck-M，HealthCheck-M 访问 MECM-inventory 获取 mec list

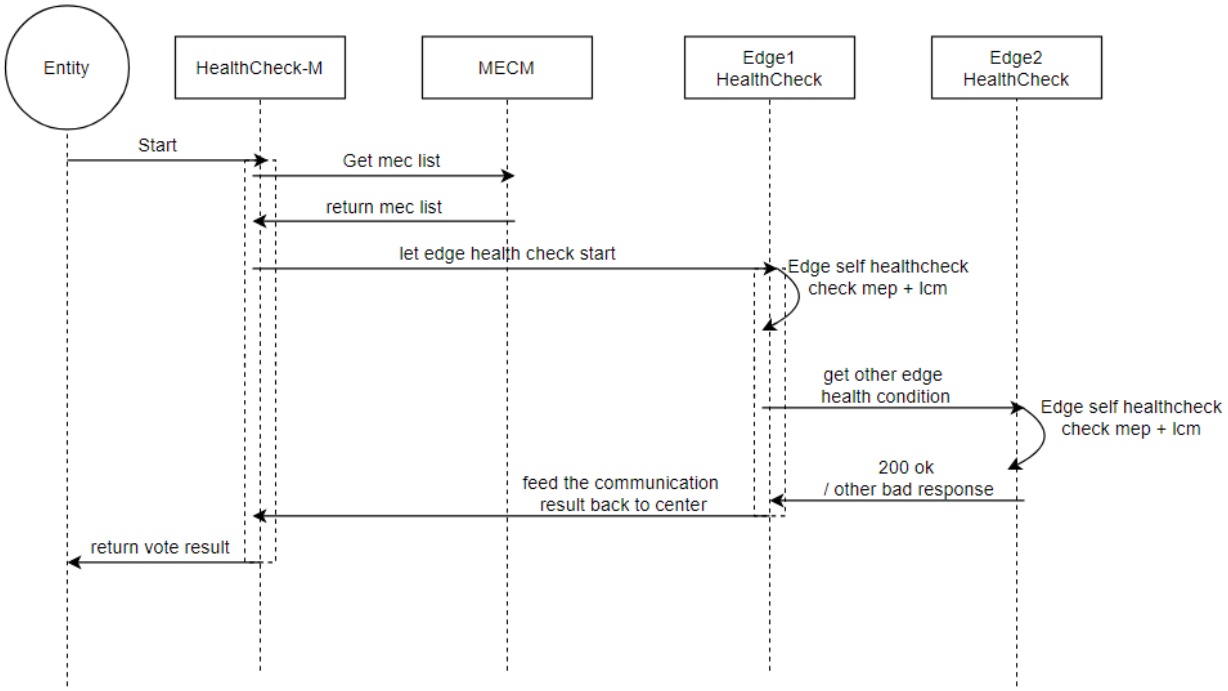
*MEP

* 边缘侧健康检查本节点 MEP 情况，MEP 和 LCM 均健康时才认为此节点健康

*LCM

* 边缘侧健康检查本节点 MEP 情况，MEP 和 LCM 均健康时才认为此节点健康

业务流程



接口定义

10.3 接口设计

- *Interface design*
  - *Introduction to MEP interface*
  - *MEP-auth module interface*

    * *Service Authentication Interface*
    * *AK/SK configuration interface*
- *MEP-server interface*
    * *Application Service Management Related Interface*
        1. *Query application services*
        2. *Query application specific services*
        3. *Register application service*
        4. *Update application service*
        5. *Delete application service*
        6. *Confirm ready service*
    * *Available event subscription related interface*
        1. *Query available event subscriptions*
        2. *Register for available event subscription*
        3. *Delete available event subscription*
        4. *Query specified available event subscription*
    * *Termination event subscription related interface*
        1. *Query termination event subscription*
        2. *Registration termination event subscription*
        3. *Delete termination event subscription*
        4. *Query specified termination event subscription*
    * *Services interfaces*
        1. *Query availability services list*
        2. *Query availability individual service*
        3. *Query transport capabilities*
    * *Heartbeat related interfaces*
        1. *Query service liveness info*
        2. *Update liveness info*
    * *DNS rule configuration interfaces*
        1. *Query all dns rules*
        2. *Query a specific dns rule*

            - 3. Update a specific dns rule
        * Traffic rule configuration interfaces
            - 1. Query all traffic rules
            - 2. Query a specific traffic rule
            - 3. Update a specific traffic rule
        * Mm5 Interface for appd configurations
            - 1. Create a new appd configurations
            - 2. Query appd configuration
            - 3. Modify appd configuration
            - 4. Delete appd configuration
            - 5. Query task status
        * Application Termination
            - 1. App Instance Termination
            - 2. App Instance Termination Confirmation
        * Query Platform Capabilities(Services)
            - 1. Query all capabilities
            - 2. Query individual capability
        * Timing interfaces
            - 1. Query platform time
            - 2. Query timing capabilities
        * Abnormal status code
    - Dns-Server
        * 1. Create new entry
        * 2. Modify the entry
        * 3. Delete an entry
    - MEP Agent
        * 1. Get token
        * 2. Get producer endpoint by service name

10.3.1 Introduction to MEP interface

MEP 主要包含 MEP-server 和 MEP-auth 两个主要功能模块。MEP server 接口分为两类，一类为遵循 ETSI MEC 011 v2.1.1 标准的 Mp1 接口，主要为 App 提供服务注册发现，App 状态通知订阅，Dns 规则获取等功能；另一类为 Mm5 接口，主要为 MECM/MEPM 提供配置管理功能。MEP auth 目前主要作为鉴权模块，为 App 提供 token 申请发放功能。URL 为服务自己的 URL，PORT 为服务自己的 PORT。如果经过 KONG，PORT 变成 KONG 的 PORT，URL 需要添加对应的路由。mepauth 直接添加 {KONG_MEPAUTH_ROUTE}（当前值为/mepauth），mepserver 把/mep 用 {KONG_MEPSERVER_ROUTE}（当前值为/mepserver）替换。

10.3.2 MEP-auth module interface

Service Authentication Interface

应用通过 AK/SK 向 LDVS 进行鉴权认证，认证通过后返回 token，每个 token 具有有效期（1 小时），token 分配方在有效期结束后将收回 token，当携带过期的 token 的 HTTP 请求到达的时候，将被以状态码 401 返回；此时 token 携带方需要重新进行 token 获取。当 token 还在有效期内，token 携带方再次使用相同的用户获取 token 时返回当前仍然有效的 token。

URL

```
POST /mep/token
```

请求参数

名称	类型	描述	IN	必选
Content-Type	String	MIME 类型，填”application/json”	header	是
Authorization	String	认证信息	header	是
x-sdk-date	String	签名时间（当前时间戳，格式：YYYYMMDDTHHMMSSZ）	header	是
Host	String	与生成认证信息签名用到的 host 字段保持一致	header	是

Body 参数

无

请求示例

```
POST /mep/token

{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json"
    },
```

(下页继续)

(续上页)

```
    {
      "key": "Authorization",
      "value": "SDK-HMAC-SHA256 Access= QVUJMSUMgS0VZLS0tLS0=,
      SignedHeaders=content-type;host;x-sdk-date,
      Signature=142b0dc3feaeb3662b2033a8e6425596546e08a231aa39179b4060867dd15d3d"
    },
    {
      "key": "x-sdk-date",
      "value": "20060102T150405Z"
    },
    {
      "key":"Host",
      "value":"xxx"
    }
  ]
}
```

返回参数

返回码：200

OK

名称	类型	描述	必选
access_token	String	Token	是
token_type	String	Token 类型（Bearer）	是
expires_in	int	失效时间	是

返回示例

```
HTTP/1.1 200 OK
{
        "access_token":"xxxx",
        "token_type":"Bearer",
        "expires_in":"3600"
}
```

注：在测试时，请确保启动 EG-LDVS 时使用的 AK/SK 和获取 token 时使用的 AK/SK 相同。基于 AK/SK 的认证，EG-LDVS 对于相同的 AK 若 5 分钟内三次认证失败，对应 AK 将被锁定 15 分钟。

AK/SK configuration interface

URL

```
PUT /mep/appMng/v1/applications/{appInstanceId}/confs
```

请求参数

名称	类型	描述	IN	必选
Content-Type	String	MIME 类型，填”application/json”	header	是
appInstanceId	String	APP 实例 ID（UUID）	path	是

Body 参数

名称	类型	描述	必选
authInfo	Object	用户信息	是
>credentials	Object	证书信息	是
»accessKeyId	String	AK	是
»secretKey	String	SK	是
appInfo	Object	app 信息	是
>appName	String	app 名称	是
>requiredServices	Array[String]	app 依赖的服务列表	是

请求示例

```
PUT /mep/appMng/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/confs
{
    "header": [
        {
            "key": "Content-Type",
            "value": "application/json"
        }
    ],
    "body": {
        "authInfo": {
            "credentials": {
                "accessKeyId": "QVUJMSUMgS0VZLS0tLS0",
                "secretKey":
↪"DXPb4sqElKhcHe07Kw5uorayETwId1JOjjOIRomRs5wyszoCR5R7AtVa28KT3lSc"
            }
        },
        "appInfo": {
            "appName": "name1",
```

(下页继续)

(续上页)

```
            "requiredServices": [
                "serviceA",
                "serviceB"
            ]
        }
    }
}
```

返回参数

返回码：200

名称	类型	描述	必选
authInfo	Object	用户信息	是
>credentials	Object	证书信息	是
»accessKeyId	String	AK	是
»secretKey	String	SK	是
appInfo	Object	app 信息	是
>appName	String	app 名称	是
>requiredServices	Array[String]	app 依赖的服务	是

返回示例

```
HTTP/1.1 200 OK
{
    "header": [
        {
            "key": "Content-Type",
            "value": "application/json"
        }
    ],
    "body": {
        "authInfo": {
            "credentials": {
                "accessKeyId": "QVUJMSUMgS0VZLS0tLS0",
                "secretKey":
↪"DXPb4sqElKhcHe07Kw5uorayETwId1JOjjOIRomRs5wyszoCR5R7AtVa28KT3lSc"
            }
        },
        "appInfo": {
            "appName": "name1",
            "requiredServices": [
                "serviceA",
```

(下页继续)

(续上页)

```
                "serviceB"
            ]
        }
    }
}
```

10.3.3 MEP-server interface

Application Service Management Related Interface

1. Query application services

查询指定 APP 的服务列表

URL

```
GET /mep/mec_service_mgmt/v1/applications/{appInstanceId}/services
```

请求参数

名称	类型	描述	IN	必选
Authorization	String	Token 信息，格式：Bearer token 信息	header	是
appInstanceId	String	APP 实例 ID（UUID）	path	是

Body 参数

无

请求示例

```
GET /mep/mec_service_mgmt/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/
↪services
{
  "header": [
    {
      "key": "Authorization",
      "value": "Bearer xxx"
    }
  ]
}
```

返回参数

返回码：200

OK

名称	类型	描述
serInstanceId	String	服务实例 ID
serName	String	服务名称
serCategory	Object	服务类别
>href	String	目录参考
>id	String	类别 ID
>name	String	类别名称
>version	String	类别版本号
version	String	服务版本号
state	String	服务状态（ACTIVE/INACTIVE）
transportInfo	Object	服务信息
>id	String	ID
>name	String	名称
> description	String	描述
> type	String	类型（REST_HTTP/MB_TOPIC_BASED/MB_ROUTING/MB_PUBSUB/RPC/RPC_STRE
> protocol	String	协议名称
> version	String	版本号
> endpoint	Ojbect	Endpoint，三种信息其中的一个（uris/addresses/alternative）
»uris	Array[String]	URI 格式的服务入口信息列表
»addresses	Array[Object]	服务入口信息地址列表（IP 地址 + 端口号）
»>host	String	IP 地址
»>port	String	端口号
»alternative	String	已实现格式或者外部规范定义的服务入口信息
> security	Object	安全信息
» oAuth2Info	Object	OAuth2.0 参数
»> grantTypes	Array[String]	准予类型（OAUTH2_AUTHORIZATION_CODE/OAUTH2_IMPLICIT_GRANT/OAUTH
»> tokenEndpoint	String	Token endpoint
> implSpecificInfo	String	其他实现信息
serializer	String	序列化类型（JSON/XML/PROTOBUF3）
scopeOfLocality	String	地域范围（MEC_SYSTEM/MEC_HOST/NFVI_POP/ZONE/ZONE_GROUP/NFVI_NODE
consumedLocalOnly	boolean	是否只被本地域 APP 消费（true/false）
isLocal	boolean	是否在本地域（true/false）
livenessInterval	integer	Interval (in seconds) between two consecutive heartbeat messages
_links	Object	Links to resources related to this resource. Shall be absent in HTTP requests.
>self	object	Link to this resource. Shall be present in HTTP responses.
»liveness	string	Link to the resource where the MEC platform expects the service instance to send the liveness i

返回示例

```
HTTP/1.1 200 OK
[
  {
    "serInstanceId": "0bc92b06cc213d2ad8beda71bd0e1460",
    "serName": "ExampleService",
    "serCategory": {
      "href": "/example/catalogue1",
      "id": "id12345",
      "name": "RNI",
      "version": "version1"
    },
    "version": "ServiceVersion1",
    "state": "ACTIVE",
    "transportInfo": {
      "id": "TransId12345",
      "name": "REST",
      "description": "REST API",
      "type": "REST_HTTP",
      "protocol": "HTTP",
      "version": "2.0",
      "endpoint": {},
      "security": {
        "oAuth2Info": {
          "grantTypes": [
            "OAUTH2_CLIENT_CREDENTIALS"
          ],
          "tokenEndpoint": "/mecSerMgmtApi/security/TokenEndPoint"
        }
      },
      "implSpecificInfo": {}
    },
    "serializer": "JSON",
    "scopeOfLocality": "MEC_SYSTEM",
    "consumedLocalOnly": false,
    "isLocal": true,
    "livenessInterval": 60,
    "_links": {
      "self": {
        "liveness" : "/mec_service_mgmt/v1/applications/5abe4782-2c70-4e47-9a4e-
↪0ee3a1a0fd1f/service/0bc92b06cc213d2ad8beda71bd0e1460/liveness"
      }
    }
  }
```

(下页继续)

(续上页)

```
]
```

2. Query application specific services

查询对应 APP 的指定服务

URL

```
GET /mep/mec_service_mgmt/v1/applications/{appInstanceId}/services/{serviceId}
```

请求参数

名称	类型	描述	IN	必选
Authorization	String	Token 信息，格式：Bearer token 信息	header	是
appInstanceId	String	APP 实例 ID（UUID）	path	是
serviceId	String	APP 服务实例 ID	path	是

Body 参数

无

请求示例

```
GET /mep/mec_service_mgmt/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/
↪services/0bc92b06cc213d2ad8beda71bd0e1460
{
  "header": [
    {
      "key": "Authorization",
      "value": "Bearer xxx"
    }
  ]
}
```

返回参数

返回码：200

OK

名称	类型
serInstanceId	String
serName	String
serCategory	Object
>href	String
>id	String
>name	String
>version	String
version	String
state	String
transportInfo	Object
>id	String
>name	String
> description	String
> type	String
> protocol	String
> version	String
> endpoint	Ojbect
»uris	Array[String]
»addresses	Array[Object]
»>host	String
»>port	String
»alternative	String
> security	Object
» oAuth2Info	Object
»> grantTypes	Array[String]
»> tokenEndpoint	String
> implSpecificInfo	String
serializer	String
scopeOfLocality String	地域范围（MEC_SYSTEM/MEC_HOST/NFVI_POP/ZONE/ZONE_GROUP/NFVI_NODE），默认值 M
consumedLocalOnly	boolean
isLocal	boolean
livenessInterval	integer
_links	Object
>self	object
»liveness	string

返回示例

```
HTTP/1.1 200 OK
```

(下页继续)

(续上页)


```
{
 "serInstanceId": "0bc92b06cc213d2ad8beda71bd0e1460",
 "serName": "ExampleService",
 "serCategory": {
      "href": "/example/catalogue1",
      "id": "id12345",
      "name": "RNI",
      "version": "version1"
 },
 "version": "ServiceVersion1",
 "state": "ACTIVE",
 "transportInfo": {
   "id": "TransId12345",
   "name": "REST",
   "description": "REST API",
   "type": "REST_HTTP",
   "protocol": "HTTP",
   "version": "2.0",
   "endpoint": {},
   "security": {
        "oAuth2Info": {
          "grantTypes": [
            "OAUTH2_CLIENT_CREDENTIALS"
          ],
          "tokenEndpoint": "/mecSerMgmtApi/security/TokenEndPoint"
        }
      },
      "implSpecificInfo": {}
    },
    "serializer": "JSON",
    "scopeOfLocality": "MEC_SYSTEM",
    "consumedLocalOnly": false,
    "isLocal": true,
    "livenessInterval": 60,
    "_links": {
      "self": {
        "liveness" : "/mec_service_mgmt/v1/applications/5abe4782-2c70-4e47-9a4e-
↪0ee3a1a0fd1f/service/0bc92b06cc213d2ad8beda71bd0e1460/liveness"
      }
    }
 }
```

3. Register application service

为指定 APP 注册服务。

URL

```
POST /mep/mec_service_mgmt/v1/applications/{appInstanceId}/services
```

请求参数:

名称	类型	描述	IN	必选
Content-Type	String	MIME 类型，填”application/json”	header	是
Authorization	String	Token 信息，格式：Bearer token 信息	header	是
appInstanceId	String	APP 实例 ID（UUID）	path	是

Body 参数

名称	类型	描述
serName	String	服务名称（字母/数字/_/-的组合，但不能以 _ 和-开头或结尾），最大长度 128
serCategory	Object	服务类别
>href	String	目录参考
>id	String	类别 ID，字母与数字组合，最大长度 64
>name	String	类别名称
>version	String	类别版本号
version	String	服务版本号（格式为 x[.y[.z]]，xyz 取值范围 0~32767，例如：1；1.2；1.2.3）
state	String	服务状态（ACTIVE/INACTIVE）
transportId	String	服务信息 ID，字母与数字组合，最大长度 64
transportInfo	Object	服务信息
>id	String	ID，字母与数字组合，最大长度 64
>name	String	名称（字母/数字/_/-的组合，但不能以 _ 和-开头或结尾），最大长度 128
> description	String	描述，最大长度 128
> type	String	类型（REST_HTTP/MB_TOPIC_BASED/MB_ROUTING/MB_PUBSUB/RPC/RPC_STRE
> protocol	String	协议名称，最大长度 32
> version	String	版本号
> endpoint	Object	Endpoint，三种信息其中的一个（uris/addresses/alternative）
»uris	Array[String]	URI 格式的服务入口信息列表
»addresses	Array[Object]	服务入口信息地址列表（IP 地址 + 端口号）
»>host	String	IP 地址。例如：192.168.0.1
»>port	String	端口号（<=65535）。例如：8080
»alternative	String	已实现格式或者外部规范定义的服务入口信息
> security	Object	安全信息

表 3 - 续上页

名称	类型	描述
» oAuth2Info	Object	OAuth2.0 参数
»> grantTypes	String	准予类型（OAUTH2_AUTHORIZATION_CODE/OAUTH2_IMPLICIT_GRANT/OAUTH
»> tokenEndpoint	String	Token endpoint
> implSpecificInfo	String	其他实现信息
serializer	String	序列化类型（JSON/XML/PROTOBUF3）
scopeOfLocality	String	地域范围（MEC_SYSTEM/MEC_HOST/NFVI_POP/ZONE/ZONE_GROUP/NFVI_NODE
consumedLocalOnly	boolean	是否只被本地域 APP 消费（true/false）
isLocal	boolean	是否在本地域（true/false）
livenessInterval	integer	Interval (in seconds) between two consecutive heartbeat messages

请求示例

```
POST /mep/mec_service_mgmt/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/
↪services
{
  "header": [
    {
      "key": "Content-Type",
      "value": "application/json",
      "key": "Authorization",
      "value": "Bearer xxx"
    }
  ],
  "body": {
    "serName": "FaceRegService",
    "serCategory": {
    "href": "/example/catalogue1",
    "id": "id12345",
    "name": "RNI",
    "version": "1.2.4"
    },
    "version": "4.5.7",
    "state": "ACTIVE",
    "transportId": "Rest1",
    "transportInfo": {
    "id": "TransId12345",
    "name": "REST",
    "description": "REST API",
    "type": "REST_HTTP",
    "protocol": "HTTP",
    "version": "2.0",
```

(下页继续)

(续上页)

```
    "endpoint": {},
    "security": {
      "oAuth2Info": {
            "grantTypes": [
            "OAUTH2_CLIENT_CREDENTIALS"
            ],
            "tokenEndpoint": "/mecSerMgmtApi/security/TokenEndPoint"
      }
    },
    "implSpecificInfo": {}
     },
     "serializer": "JSON",
     "scopeOfLocality": "MEC_SYSTEM",
     "consumedLocalOnly": false,
     "isLocal": true,
     "livenessInterval": 60
  }
}
```

返回参数

返回码：201

OK

名称	类型	描述
serInstanceId	String	服务实例 ID
serName	String	服务名称
serCategory	Object	服务类别
>href	String	目录参考
>id	String	类别 ID
>name	String	类别名称
>version	String	类别版本号
version	String	服务版本号
state	String	服务状态（ACTIVE/INACTIVE）
transportInfo	Object	
>id	String	ID
>name	String	名称
> description	String	描述
> type	String	类型（REST_HTTP/MB_TOPIC_BASED/MB_ROUTING/MB_PUBSUB/RPC/RPC_STRE
> protocol	String	协议名称
> version	String	版本号

表 4 – 续上页

名称	类型	描述
> endpoint	Ojbect	endpoint，三种信息其中的一个（uris/addresses/alternative）
»uris	Array[String]	URI 格式的服务入口信息列表
»addresses	Array[Object]	服务入口信息地址列表（IP 地址 + 端口号）
»>host	String	IP 地址
»>port	String	端口号
»alternative	String	已实现格式或者外部规范定义的服务入口信息
> security	Object	安全信息
» oAuth2Info	Object	OAuth2.0 参数
»> grantTypes	Array[String]	准予类型（OAUTH2_AUTHORIZATION_CODE/OAUTH2_IMPLICIT_GRANT/OAUTH
»> tokenEndpoint	String	Token endpoint
> implSpecificInfo	String	其他实现信息
serializer	String	序列化类型（JSON/XML/PROTOBUF3）
scopeOfLocality	String	地域范围（MEC_SYSTEM/MEC_HOST/NFVI_POP/ZONE/ZONE_GROUP/NFVI_NODE
consumedLocalOnly	boolean	是否只被本地域 APP 消费（true/false）
isLocal	boolean	是否在本地域（true/false）
livenessInterval	integer	Interval (in seconds) between two consecutive heartbeat messages
_links	Object	Links to resources related to this resource. Shall be absent in HTTP requests.
>self	object	Link to this resource. Shall be present in HTTP responses.
»liveness	string	Link to the resource where the MEC platform expects the service instance to send the liveness i

返回示例

```
HTTP/1.1 201 OK
{
    "serInstanceId": "0bc92b06cc213d2ad8beda71bd0e1460",
    "serName": "FaceRegService",
    "serCategory": {
        "href": "/example/catalogue1",
        "id": "id12345",
        "name": "RNI",
        "version": "1.2.4"
    },
    "version": "4.5.7",
    "state": "ACTIVE",
    "transportId": "Rest1",
    "transportInfo": {
        "id": "TransId12345",
        "name": "REST",
        "description": "REST API",
        "type": "REST_HTTP",
```

(下页继续)

(续上页)

```
        "protocol": "HTTP",
        "version": "2.0",
        "endpoint": {},
        "security": {
            "oAuth2Info": {
                    "grantTypes": [
                    "OAUTH2_CLIENT_CREDENTIALS"
                    ],
                    "tokenEndpoint": "/mecSerMgmtApi/security/TokenEndPoint"
            }
        },
        "implSpecificInfo": {}
    },
    "serializer": "JSON",
    "scopeOfLocality": "MEC_SYSTEM",
    "consumedLocalOnly": false,
    "isLocal": true,
    "livenessInterval": 60,
    "_links": {
      "self": {
        "liveness" : "/mec_service_mgmt/v1/applications/5abe4782-2c70-4e47-9a4e-
↪0ee3a1a0fd1f/service/0bc92b06cc213d2ad8beda71bd0e1460/liveness"
      }
    }
}
```

注：单个应用注册的服务数量应不超过 50 个，超出后服务将报错。

4. Update application service

更新 APP 指定服务信息。

URL

```
PUT /mep/mec_service_mgmt/v1/applications/{appInstanceId}/services/{serviceId}
```

请求参数

名称	类型	描述	IN	必选
Content-Type	String	MIME 类型，填”application/json”	header	是
Authorization	String	Token 信息，格式：Bearer token 信息	header	是
appInstanceId	String	APP 实例 ID（UUID）	path	是
serviceId	String	APP 服务实例 ID	path	是

Body 参数

名称	类型	描述
serName	String	服务名称（字母/数字/_/-的组合，但不能以 _ 和-开头或结尾），最大长度 128
serCategory	Object	服务类别
>href	String	目录参考
>id	String	类别 ID，字母与数字组合，最大长度 64
>name	String	类别名称
>version	String	类别版本号
version	String	服务版本号（例如：1；1.0；1.0.2）
state	String	服务状态（ACTIVE/INACTIVE）
transportId	String	服务信息 ID，字母与数字组合，最大长度 64
transportInfo	Object	服务信息
>id	String	ID，字母与数字组合，最大长度 64
>name	String	名称（字母/数字/_/-的组合，但不能以 _ 和-开头或结尾），最大长度 128
> description	String	描述，最大长度 128
> type	String	类型（REST_HTTP/MB_TOPIC_BASED/MB_ROUTING/MB_PUBSUB/RPC/RPC_STRE
> protocol	String	协议名称，最大长度 32
> version	String	版本号
> endpoint	Object	Endpoint，三种信息其中的一个（uris/addresses/alternative）
»uris	Array[String]	URI 格式的服务入口信息列表
»addresses	Array[Object]	服务入口信息地址列表（IP 地址 + 端口号）
»>host	String	IP 地址。例如：192.168.0.1
»>port	String	端口号（<=65535）。例如：8080
»alternative	String	已实现格式或者外部规范定义的服务入口信息
> security	Object	安全信息
» oAuth2Info	Object	OAuth2.0 参数
»> grantTypes	String	准予类型（OAUTH2_AUTHORIZATION_CODE/OAUTH2_IMPLICIT_GRANT/OAUTH
»> tokenEndpoint	String	Token endpoint
> implSpecificInfo	String	其他实现信息
serializer	String	序列化类型（JSON/XML/PROTOBUF3）
scopeOfLocality	String	地域范围（MEC_SYSTEM/MEC_HOST/NFVI_POP/ZONE/ZONE_GROUP/NFVI_NODE
consumedLocalOnly	boolean	是否只被本地域 APP 消费（true/false）
isLocal	boolean	是否在本地域（true/false）
livenessInterval	integer	Interval (in seconds) between two consecutive heartbeat messages

请求示例

```
PUT /mep/mec_service_mgmt/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/
↪services/0bc92b06cc213d2ad8beda71bd0e1460
```

(下页继续)

(续上页)

```
{
        "header": [
            {
                "key": "Content-Type",
                "value": "application/json",
                "key": "Authorization",
                "value": "Bearer xxx"
            }
        ],
        "body": {
            "serName": "FaceRegService",
            "serCategory": {
                "href": "/example/catalogue1",
                "id": "id12345",
                "name": "RNI",
                "version": "1.2.4"
            },
            "version": "4.5.7",
            "state": "ACTIVE",
            "transportId": "Rest1",
            "transportInfo": {
                "id": "TransId12345",
                "name": "REST",
                "description": "REST API",
                "type": "REST_HTTP",
                "protocol": "HTTP",
                "version": "2.0",
                "endpoint": {},
                "security": {
                    "oAuth2Info": {
                        "grantTypes": [
                            "OAUTH2_CLIENT_CREDENTIALS"
                        ],
                        "tokenEndpoint": "/mecSerMgmtApi/security/TokenEndPoint"
                    }
                },
                "implSpecificInfo": {}
            },
            "serializer": "JSON",
            "scopeOfLocality": "MEC_SYSTEM",
            "consumedLocalOnly": false,
            "isLocal": true,
            "livenessInterval": 60
```

(下页继续)

(续上页)

```
        }
}
```

返回参数

返回码：200

OK

名称	类型	描述
serInstanceId	String	服务实例 ID
serName	String	服务名称
serCategory	Object	服务类别
>href	String	目录参考
>id	String	类别 ID
>name	String	类别名称
>version	String	类别版本号
version	String	服务版本号
state	String	服务状态（ACTIVE/INACTIVE）
transportInfo	Object	服务信息
>id	String	ID
>name	String	名称
> description	String	描述
> type	String	类型（REST_HTTP/MB_TOPIC_BASED/MB_ROUTING/MB_PUBSUB/RPC/RPC_STRE
> protocol	String	协议名称
> version	String	版本号
> endpoint	Object	endpoint，三种信息其中的一个（uris/addresses/alternative）
»uris	Array[String]	URI 格式的服务入口信息列表
»addresses	Array[Object]	服务入口信息地址列表（IP 地址 + 端口号）
»>host	String	IP 地址
»>port	String	端口号
»alternative	String	已实现格式或者外部规范定义的服务入口信息
> security	Object	安全信息
» oAuth2Info	Object	OAuth2.0 参数
»> grantTypes	String	准予类型（OAUTH2_AUTHORIZATION_CODE/OAUTH2_IMPLICIT_GRANT/OAUTH
»> tokenEndpoint	String	Token endpoint
> implSpecificInfo	String	其他实现信息
serializer	String	序列化类型（JSON/XML/PROTOBUF3）
scopeOfLocality	String	地域范围（MEC_SYSTEM/MEC_HOST/NFVI_POP/ZONE/ZONE_GROUP/NFVI_NODE

表 6 – 续上页

名称	类型	描述
consumedLocalOnly	boolean	是否只被本地域 APP 消费（true/false）
isLocal	boolean	是否在本地域（true/false）
livenessInterval	integer	Interval (in seconds) between two consecutive heartbeat messages
_links	Object	Links to resources related to this resource. Shall be absent in HTTP requests.
>self	object	Link to this resource. Shall be present in HTTP responses.
»liveness	string	Link to the resource where the MEC platform expects the service instance to send the liveness i

返回示例

```
HTTP/1.1 200 OK
{
    "serInstanceId": "0bc92b06cc213d2ad8beda71bd0e1460",
    "serName": "FaceRegService",
    "serCategory": {
        "href": "/example/catalogue1",
        "id": "id12345",
        "name": "RNI",
        "version": "1.2.4"
    },
    "version": "4.5.7",
    "state": "ACTIVE",
    "transportId": "Rest1",
    "transportInfo": {
        "id": "TransId12345",
        "name": "REST",
        "description": "REST API",
        "type": "REST_HTTP",
        "protocol": "HTTP",
        "version": "2.0",
        "endpoint": {},
        "security": {
            "oAuth2Info": {
                "grantTypes": [
                    "OAUTH2_CLIENT_CREDENTIALS"
                ],
                "tokenEndpoint": "/mecSerMgmtApi/security/TokenEndPoint"
            }
        },
        "implSpecificInfo": {}
    },
    "serializer": "JSON",
    "scopeOfLocality": "MEC_SYSTEM",
```

(下页继续)

(续上页)

```
    "consumedLocalOnly": false,
    "isLocal": true,
    "livenessInterval": 60,
    "_links": {
      "self": {
        "liveness" : "/mec_service_mgmt/v1/applications/5abe4782-2c70-4e47-9a4e-
↪0ee3a1a0fd1f/service/0bc92b06cc213d2ad8beda71bd0e1460/liveness"
      }
    }
}
```

5. Delete application service

删除 APP 指定服务。

URL

```
DELETE /mep/mec_service_mgmt/v1/applications/{appInstanceId}/services/{serviceId}
```

请求参数

名称	类型	描述	IN	必选
Authorization	String	Token 信息，格式：Bearer token 信息	header	是
appInstanceId	String	APP 实例 ID（UUID）	path	是
serviceId	String	APP 服务实例 ID	path	是

Body 参数

无

请求示例

```
DELETE /mep/mec_service_mgmt/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/
↪services/0bc92b06cc213d2ad8beda71bd0e1460
{
    "header": [
        {
            "key": "Authorization",
            "value": "Bearer xxx"
        }
    ],
    "body": {
```

(下页继续)

(续上页)

```
    }
}
```

返回参数

返回码：204

OK

返回示例

```
HTTP/1.1 204 OK
{
}
```

6. Confirm ready service

The MEC application that intends to communicate with MEC sends a "MEC App is running" message towards the MEC platform to confirm that the instantiation and the start-up phase have been successfully completed.

URL

```
POST /mep/mec_app_support/v1/applications/{appInstanceId}/confirm_ready
```

Request parameters

Name	Type	Description	IN	Required
appInstanceId	String	APP 实例 ID(UUID)	path	Yes

None

Body Parameters

Name	Type	Description	Required
indication	string	Indication about the MEC application instance: - READY	Yes

Example Request

```
POST /mep/mec_app_support/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/
↪confirm_ready
{
```

(下页继续)

(续上页)

```
  "indication": "READY"
}
```

Return Parameters

None

Return Code: 204 No Content

Example Response

```
HTTP/1.1 204 No Content
```

Exception status code

Available event subscription related interface

1. Query available event subscriptions

查询指定 APP 的可用事件订阅信息

URL

```
GET /mep/mec_service_mgmt/v1/applications/{appInstanceId}/subscriptions
```

请求参数

名称	类型	描述	IN	必选
Authorization	String	Token 信息，格式：Bearer token 信息	header	是
appInstanceId	String	APP 实例 ID（UUID）	path	是

Body 参数

无

请求示例

```
GET /mep/mec_service_mgmt/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/
↪subscriptions
{
    "header": [
        {
            "key": "Authorization",
            "value": "Bearer xxx"
        }
```

(下页继续)

(续上页)

```
    ]
}
```

返回参数

返回码：200

OK

名称	类型	描述	必选
_links	Object	结果信息	是
>self	Object	调用者信息	是
»href	String	调用者 URL	是
>subscriptions	Array[Object]	订阅者列表	是
»href	String	订阅者 URI	是
»rel	String	固定值 SerAvailabilityNotificationSubscription	是

返回示例

```
HTTP/1.1 200 OK
{
    "_links": {
        "self": {
            "href":"/mec_service_mgmt/v1/applications/5abe4782-2c70-4e47-9a4e-
↪0ee3a1a0fd1f /subscriptions"
        },
        "subscriptions": [
            {
                "href":"/mec_service_mgmt/v1/applications/5abe4782-2c70-4e47-9a4e-
↪0ee3a1a0fd1f/subscriptions/826a3890-8b05-416f-8d24-7a87e9eca731",
                "rel": "SerAvailabilityNotificationSubscription"
            }
        ]
    }
}
```

2. Register for available event subscription

为指定 APP 注册可用事件订阅

URL

```
POST /mep/mec_service_mgmt/v1/applications/{appInstanceId}/subscriptions
```

请求参数

名称	类型	描述	IN	必选
Content-Type	String	MIME 类型，填”application/json”	header	是
Authorization	String	Token 信息，格式：Bearer token 信息	header	是
appInstanceId	String	APP 实例 ID（UUID）	path	是

Body 参数

名称	类型	描述	必选
subscriptionType	String	订阅类型（SerAvailabilityNotificationSubscription）	是
callbackReference	String	用于接收通知的回调 URI，不支持域名格式，仅支持 IP 地址 +URL，且不能包含 localhost、127.x.x.x、mepserver 容器 IP 网段，例如：https://159.138.1.2:8080/xxx	是
filteringCriteria	Object	过滤条件列表	否
>serInstanceIds	Array[String]	上报事件的应用服务实例 ID 列表	否
>serNames	Array[String]	上报事件的应用服务实例名称列表（每个服务实例名称都是字母/数字/_/-的组合，但不能以 _ 和-开头或结尾，最大长度 128）	否
>serCategories	Array[Object]	上报事件的应用服务类别列表	否
»href	String	类别参考	否
»id	String	类别 ID，最大长度 64	否
»name	String	类别名称（字母/数字/_/-的组合，但不能以 _ 和-开头或结尾），最大长度 128	否
»version	String	类别版本号	否
>states	Array[String]	上报事件的应用服务状态（ACTIVE/INACTIVE）	否
>isLocal	boolean	是否在本地域（true/false）	否

请求示例

```
POST /mep/mec_service_mgmt/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/
↪subscriptions
{
    "header": [
        {
            "key": "Content-Type",
            "value": "application/json",
            "key": "Authorization",
            "value": "Bearer xxx"
        }
    ]
    "body": {
```

(下页继续)

(续上页)


```
        "subscriptionType": "SerAvailabilityNotificationSubscription",
        "callbackReference": "https://159.138.1.2:8080/callback",
        "filteringCriteria": {
            "serInstanceIds": [
                "0bc92b06cc213d2ad8beda71bd0e1460"
            ],
            "serNames": [
                "FaceRegService5"
            ],
            "serCategories": [
                {
                    "href": "/example/catalogue1",
                    "id": "id12345",
                    "name": "RNI",
                    "version": "1.2.2"
                }
            ],
            "states": [
                "ACTIVE"
            ],
            "isLocal": true
        }
    }
}
```


返回参数

返回码：201

OK

名称	类型	描述	必选
subscriptionId	String	订阅 Id	是
subscriptionType	String	订阅类型（SerAvailabilityNotificationSubscription）	是
callbackReference	String	用于接收通知的回调 URI，不支持域名格式，仅支持 IP 地址 +URL，且不能包含 localhost、127.x.x.x、mepserver 容器 IP 网段，例如：https://159.138.1.2:8080/xxx	是
_links	Object	调用者自身信息	是
>self	Object	调用者自身信息	是
»href	String	调用者自身 URI	是
filteringCriteria	Object	过滤条件列表	否
>serInstanceIds	Array[String]	上报事件的应用服务实例 ID 列表	否
>serNames	Array[String]	上报事件的应用服务实例名称列表	否
>serCategories	Array[Object]	上报事件的应用服务类别列表	否
»href	String	类别参考	否
»id	String	类别 ID	否
»name	String	类别名称	否
»version	String	类别版本号	否
>states	Array[String]	上报事件的应用服务状态（ACTIVE/INACTIVE）	否
>isLocal	boolean	是否在本地域（true/false）	否

返回示例

```
HTTP/1.1 201 OK
{
    "subscriptionId": "826a3890-8b05-416f-8d24-7a87e9eca731",
    "subscriptionType": "SerAvailabilityNotificationSubscription",
    "callbackReference": "https://159.138.1.2:8080/callback",
    "_links": {
```

(下页继续)

(续上页)

```
        "self": {
            "href":"/mec_service_mgmt/v1/applications/5abe4782-2c70-4e47-9a4e-
↪0ee3a1a0fd1f/subscriptions/826a3890-8b05-416f-8d24-7a87e9eca731"
        }
    },
    "filteringCriteria": {
        "serInstanceIds": [
            "0bc92b06cc213d2ad8beda71bd0e1460"
        ],
        "serNames": [
            "FaceRegService5"
        ],
        "serCategories": [
            {
                "href": "/example/catalogue1",
                "id": "id12345",
                "name": "RNI",
                "version": "1.2.2"
            }
        ],
        "states": [
            "ACTIVE"
        ],
        "isLocal": true
    }
}
```

注：单个应用注册的可用事件订阅数量应不超过 50 个，超出后服务将报错。

3. Delete available event subscription

删除指定可用事件订阅信息

URL

```
DELETE /mep/mec_service_mgmt/v1/applications/{appInstanceId}/subscriptions/
↪{subscriptionId}
```

请求参数

名称	类型	描述	IN	必选
Authorization	String	Token 信息，格式：Bearer token 信息	header	是
appInstanceId	String	APP 实例 ID（UUID）	path	是
subscriptionId	String	事件订阅 ID	path	是

Body 参数

无

请求示例

```
DELETE /mep/mec_service_mgmt/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/
↪subscriptions/826a3890-8b05-416f-8d24-7a87e9eca731
{
    "header": [
        {
            "key": "Authorization",
            "value": "Bearer xxx"
        }
    ]
}
```

返回参数

返回码：204

OK

返回示例

```
HTTP/1.1 204 OK
{
}
```

4. Query specified available event subscription

查询指定 ID 的可用事件订阅信息

URL

```
GET /mep/mec_service_mgmt/v1/applications/{appInstanceId}/subscriptions/
↪{subscriptionId}
```

请求参数

名称	类型	描述	IN	必选
Authorization	String	Token 信息，格式：Bearer token 信息	header	是
appInstanceId	String	APP 实例 ID（UUID）	path	是
subscriptionId	String	事件订阅 ID	path	是

Body 参数

无

请求示例

```
GET /mep/mec_service_mgmt/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/
↪subscriptions/826a3890-8b05-416f-8d24-7a87e9eca731
{
    "header": [
        {
            "key": "Authorization",
            "value": "Bearer xxx"
        }
    ]
}
```

返回参数

返回码：200

OK

名称	类型	描述	必选
subscriptionType	String	订阅类型（SerAvailabilityNotificationSubscription）	是
callbackReference	String	用于接收通知的回调 URI，不支持域名格式，仅支持 IP 地址 +URL，且不能包含 localhost、127.x.x.x、mepserver 容器 IP 网段，例如：https://159.138.1.2:8080/xxx	是
_links	Object	调用者自身信息	是
>self	Object	调用者自身信息	是
»href	String	调用者自身 URI	是
filteringCriteria	Object	过滤条件列表	否
>serInstanceIds	Array[String]	上报事件的应用服务实例 ID 列表	否
>serNames	Array[String]	上报事件的应用服务实例名称列表	否
>serCategories	Array[Object]	上报事件的应用服务类别列表	否
»href	String	类别参考	否
»id	String	类别 ID	否
»name	String	类别名称	否
»version	String	类别版本号	否
>states	Array[String]	上报事件的应用服务状态（ACTIVE/INACTIVE）	否
>isLocal	boolean	是否在本地域（true/false）	否

返回示例

```
HTTP/1.1 200 OK
{
    "subscriptionType": "SerAvailabilityNotificationSubscription",
    "callbackReference": "https://159.138.1.2:8080/callback",
    "_links": {
        "self": {
            "href":"/mec_service_mgmt/v1/applications/5abe4782-2c70-4e47-9a4e-
↪0ee3a1a0fd1f/subscriptions/826a3890-8b05-416f-8d24-7a87e9eca731"
        }
```

(下页继续)

(续上页)

```
    },
    "filteringCriteria": {
        "serInstanceIds": [
            "0bc92b06cc213d2ad8beda71bd0e1460"
        ],
        "serNames": [
            "FaceRegService5"
        ],
        "serCategories": [
            {
                "href": "/example/catalogue1",
                "id": "id12345",
                "name": "RNI",
                "version": "1.2.2"
            }
        ],
        "states": [
            "ACTIVE"
        ],
        "isLocal": true
    }
}
```

Termination event subscription related interface

1. Query termination event subscription

查询指定 APP 的终止事件订阅信息

URL

```
GET /mep/mec_app_support/v1/applications/{appInstanceId}/subscriptions
```

请求参数

名称	类型	描述	IN	必选
Authorization	String	Token 信息，格式：Bearer token 信息	header	是
appInstanceId	String	APP 实例 ID（UUID）	path	是

Body 参数

无

请求示例

```
GET /mep/mec_app_support/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/
↪subscriptions
{
    "header": [
        {
            "key": "Authorization",
            "value": "Bearer xxx"
        }
    ]
}
```

返回参数

返回码：200

OK

名称	类型	描述	必选
_links	Object	结果信息	是
>self	Object	调用者信息	是
»href	String	调用者 URL	是
>subscriptions	Array[Object]	订阅者列表	是
»href	String	订阅者 URI	是
»rel	String	固定值 AppTerminationNotificationSubscription	是

返回示例

```
HTTP/1.1 200 OK
{
    "_links": {
        "self": {
            "href":"/mec_app_support/v1/applications/5abe4782-2c70-4e47-9a4e-
↪0ee3a1a0fd1f /subscriptions"
        },
        "subscriptions": [
            {
                "href":"/mec_app_support/v1/applications/5abe4782-2c70-4e47-9a4e-
↪0ee3a1a0fd1f/subscriptions/826a3890-8b05-416f-8d24-7a87e9eca731",
                "rel": "AppTerminationNotificationSubscription"
            }
        ]
    }
}
```

(下页继续)

(续上页)

2. Registration termination event subscription

为指定 APP 注册终止事件订阅

URL

```
POST /mep/mec_app_support/v1/applications/{appInstanceId}/subscriptions
```

请求参数

名称	类型	描述	IN	必选
Content-Type	String	MIME 类型，填”application/json”	header	是
Authorization	String	Token 信息，格式：Bearer token 信息	header	是
appInstanceId	String	APP 实例 ID（UUID）	path	是

Body 参数

名称	类型	描述	必选
subscriptionType	String	订阅类型（AppTerminationNotificationSubscription）	是
callbackReference	String	用于接收通知的回调 URI，不支持域名格式，仅支持 IP 地址 +URL，且不能包含 localhost、127.x.x.x、mepserver 容器 IP 网段，例如：https://159.138.1.2:8080/xxx	是
appInstanceId	String	APP 实例 ID（UUID）	是

请求示例

```
POST /mep/mec_app_support/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/
↪subscriptions
{
    "header": [
        {
            "key": "Content-Type",
            "value": "application/json",
            "key": "Authorization",
            "value": "Bearer xxx"
        }
```

(下页继续)

(续上页)

```
    ]
    "body": {
        "subscriptionType": "AppTerminationNotificationSubscription",
        "callbackReference": "https://159.138.1.2:8080/callback",
        "appInstanceId": "5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f"
    }
}
```

返回参数

返回码：201

OK

名称	类型	描述	必选
subscriptionId	String	订阅 Id	是
subscriptionType	String	订阅类型（AppTerminationNotificationSubscription）	是
callbackReference	String	用于接收通知的回调 URI，不支持域名格式，仅支持 IP 地址 +URL，且不能包含 localhost、127.x.x.x、mepserver 容器 IP 网段，例如：https://159.138.1.2:8080/xxx	是
_links	Object	调用者自身信息	是
>self	Object	调用者自身信息	是
»href	String	调用者自身 URI	是
appInstanceId	String		是

返回示例

```
HTTP/1.1 201 OK
{
    "subscriptionId": "826a3890-8b05-416f-8d24-7a87e9eca731",
    "subscriptionType": "AppTerminationNotificationSubscription",
    "callbackReference": "https://159.138.1.2:8080/callback",
    "_links": {
        "self": {
            "href":"/mec_app_support/v1/applications/5abe4782-2c70-4e47-9a4e-
↪0ee3a1a0fd1f/subscriptions/826a3890-8b05-416f-8d24-7a87e9eca731"
```

(下页继续)

(续上页)

```
        }
    },
    "appInstanceId": "5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f"
}
```

注：单个应用注册的终止事件订阅数量应不超过 50 个，超出后服务将报错。

3. Delete termination event subscription

删除指定终止事件订阅信息

URL

```
DELETE /mep/mec_app_support/v1/applications/{appInstanceId}/subscriptions/
↪{subscriptionId}
```

请求参数

名称	类型	描述	IN	必选
Authorization	String	Token 信息，格式：Bearer token 信息	header	是
appInstanceId	String	APP 实例 ID（UUID）	path	是
subscriptionId	String	事件订阅 ID（UUID）	path	是

Body 参数

无

请求示例

```
DELETE /mep/mec_app_support/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/
↪subscriptions/826a3890-8b05-416f-8d24-7a87e9eca731
{
    "header": [
        {
            "key": "Authorization",
            "value": "Bearer xxx"
        }
    ]
}
```

返回参数

返回码：204

OK

返回示例

```
HTTP/1.1 204 OK
{

}
```

4. Query specified termination event subscription

查询指定 ID 的终止事件订阅信息

URL

```
GET /mep/mec_app_support/v1/applications/{appInstanceId}/subscriptions/
↪{subscriptionId}
```

请求参数

名称	类型	描述	IN	必选
Authorization	String	Token 信息，格式：Bearer token 信息	header	是
appInstanceId	String	APP 实例 ID（UUID）	path	是
subscriptionId	String	事件订阅 ID（UUID）	path	是

Body 参数

无

请求示例

```
GET /mep/mec_app_support/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/
↪subscriptions/826a3890-8b05-416f-8d24-7a87e9eca731
{
    "header": [
        {
            "key": "Authorization",
            "value": "Bearer xxx"
        }
    ]
}
```

返回参数

返回码：200

OK

名称	类型	描述	必选
subscriptionType	String	订阅类型（AppTerminationNotificationSubscription）	是
callbackReference	String	用于接收通知的回调 URI，不支持域名格式，仅支持 IP 地址 +URL，且不能包含 localhost、127.x.x.x、mepserver 容器 IP 网段，例如：https://159.138.1.2:8080/xxx	是
_links	Object	调用者自身信息	是
>self	Object	调用者自身信息	是
»href	String	调用者自身 URI	是
appInstanceId	String		是

返回示例

```
HTTP/1.1 200 OK
{
    "subscriptionType": "AppTerminationNotificationSubscription",
    "callbackReference": "https://159.138.1.2:8080/callback",
    "_links": {
        "self": {
            "href":"/mec_app_support/v1/applications/5abe4782-2c70-4e47-9a4e-
↪0ee3a1a0fd1f/subscriptions/826a3890-8b05-416f-8d24-7a87e9eca731"
        }
    },
    "appInstanceId": "5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f "
}
```

Service

1. Query availability services list

This method retrieves information about a list of mecService resources. This method is typically used in "service availability query" procedure

URL

```
GET /mep/mec_service_mgmt/v1/services
```

请求参数

名称	类型	描述	IN	必选
Authorization	String	Token 信息，格式：Bearer token 信息	header	是
ser_name	String	service name	query	否

Body 参数

无

请求示例

```
GET /mep/mec_service_mgmt/v1/services?ser_name=xxx
{
  "header": [
    {
      "key": "Authorization",
      "value": "Bearer xxx"
    }
  ]
}
```

返回参数

返回码：200

OK

名称	类型	描述
serInstanceId	String	服务实例 ID
serName	String	服务名称
serCategory	Object	服务类别
>href	String	目录参考
>id	String	类别 ID
>name	String	类别名称
>version	String	类别版本号
version	String	服务版本号
state	String	服务状态（ACTIVE/INACTIVE）
transportInfo	Object	服务信息
>id	String	ID

表 7 – 续上页

名称	类型	描述
>name	String	名称
> description	String	描述
> type	String	类型（REST_HTTP/MB_TOPIC_BASED/MB_ROUTING/MB_PUBSUB/RPC/RPC_STRE
> protocol	String	协议名称
> version	String	版本号
> endpoint	Ojbect	Endpoint，三种信息其中的一个（uris/addresses/alternative）
»uris	Array[String]	URI 格式的服务入口信息列表
»addresses	Array[Object]	服务入口信息地址列表（IP 地址 + 端口号）
»>host	String	IP 地址
»>port	String	端口号
»alternative	String	已实现格式或者外部规范定义的服务入口信息
> security	Object	安全信息
» oAuth2Info	Object	OAuth2.0 参数
»> grantTypes	Array[String]	准予类型（OAUTH2_AUTHORIZATION_CODE/OAUTH2_IMPLICIT_GRANT/OAUTH
»> tokenEndpoint	String	Token endpoint
> implSpecificInfo	String	其他实现信息
serializer	String	序列化类型（JSON/XML/PROTOBUF3）
scopeOfLocality	String	地域范围（MEC_SYSTEM/MEC_HOST/NFVI_POP/ZONE/ZONE_GROUP/NFVI_NODE
consumedLocalOnly	boolean	是否只被本地域 APP 消费（true/false）
isLocal	boolean	是否在本地域（true/false）
livenessInterval	integer	Interval (in seconds) between two consecutive heartbeat messages
_links	Object	Links to resources related to this resource. Shall be absent in HTTP requests.
>self	object	Link to this resource. Shall be present in HTTP responses.
»liveness	string	Link to the resource where the MEC platform expects the service instance to send the liveness i

返回示例

```
HTTP/1.1 200 OK
[
  {
    "serInstanceId": "0bc92b06cc213d2ad8beda71bd0e1460",
    "serName": "ExampleService",
    "serCategory": {
      "href": "/example/catalogue1",
      "id": "id12345",
      "name": "RNI",
      "version": "version1"
    },
    "version": "ServiceVersion1",
    "state": "ACTIVE",
```

(下页继续)

(续上页)

```
    "transportInfo": {
      "id": "TransId12345",
      "name": "REST",
      "description": "REST API",
      "type": "REST_HTTP",
      "protocol": "HTTP",
      "version": "2.0",
      "endpoint": {},
      "security": {
        "oAuth2Info": {
          "grantTypes": [
            "OAUTH2_CLIENT_CREDENTIALS"
          ],
          "tokenEndpoint": "/mecSerMgmtApi/security/TokenEndPoint"
        }
      },
      "implSpecificInfo": {}
    },
    "serializer": "JSON",
    "scopeOfLocality": "MEC_SYSTEM",
    "consumedLocalOnly": false,
    "isLocal": true,
    "livenessInterval": 60,
    "_links": {
      "self": {
        "liveness" : "/mec_service_mgmt/v1/applications/5abe4782-2c70-4e47-9a4e-
↪0ee3a1a0fd1f/service/0bc92b06cc213d2ad8beda71bd0e1460/liveness"
      }
    }
  }
]
```

2. Query availability individual service

This method retrieves information about a mecService resource. This method is typically used in "service availability query" procedure

URL

```
GET /mep/mec_service_mgmt/v1/services/{serviceId}
```

请求参数

名称	类型	描述	IN	必选
serviceId	String	service id	path	是
Authorization	String	Token 信息，格式：Bearer token 信息	header	是
ser_name	String	service name	query	否

Body 参数

无

请求示例

```
GET /mep/mec_service_mgmt/v1/services?ser_name=xxx
{
  "header": [
    {
      "key": "Authorization",
      "value": "Bearer xxx"
    }
  ]
}
```

返回参数

返回码：200

OK

名称	类型	描述
serInstanceId	String	服务实例 ID
serName	String	服务名称
serCategory	Object	服务类别
>href	String	目录参考
>id	String	类别 ID
>name	String	类别名称
>version	String	类别版本号
version	String	服务版本号
state	String	服务状态（ACTIVE/INACTIVE）
transportInfo	Object	服务信息
>id	String	ID
>name	String	名称
> description	String	描述
> type	String	类型（REST_HTTP/MB_TOPIC_BASED/MB_ROUTING/MB_PUBSUB/RPC/RPC_STRE

表 8 – 续上页

名称	类型	描述
> protocol	String	协议名称
> version	String	版本号
> endpoint	Ojbect	Endpoint，三种信息其中的一个（uris/addresses/alternative）
»uris	Array[String]	URI 格式的服务入口信息列表
»addresses	Array[Object]	服务入口信息地址列表（IP 地址 + 端口号）
»>host	String	IP 地址
»>port	String	端口号
»alternative	String	已实现格式或者外部规范定义的服务入口信息
> security	Object	安全信息
» oAuth2Info	Object	OAuth2.0 参数
»> grantTypes	Array[String]	准予类型（OAUTH2_AUTHORIZATION_CODE/OAUTH2_IMPLICIT_GRANT/OAUTH
»> tokenEndpoint	String	Token endpoint
> implSpecificInfo	String	其他实现信息
serializer	String	序列化类型（JSON/XML/PROTOBUF3）
scopeOfLocality	String	地域范围（MEC_SYSTEM/MEC_HOST/NFVI_POP/ZONE/ZONE_GROUP/NFVI_NODE
consumedLocalOnly	boolean	是否只被本地域 APP 消费（true/false）
isLocal	boolean	是否在本地域（true/false）
livenessInterval	integer	Interval (in seconds) between two consecutive heartbeat messages
_links	Object	Links to resources related to this resource. Shall be absent in HTTP requests.
>self	object	Link to this resource. Shall be present in HTTP responses.
»liveness	string	Link to the resource where the MEC platform expects the service instance to send the liveness i

返回示例

```
HTTP/1.1 200 OK
{
    "serInstanceId": "0bc92b06cc213d2ad8beda71bd0e1460",
    "serName": "ExampleService",
    "serCategory": {
      "href": "/example/catalogue1",
      "id": "id12345",
      "name": "RNI",
      "version": "version1"
    },
    "version": "ServiceVersion1",
    "state": "ACTIVE",
    "transportInfo": {
      "id": "TransId12345",
      "name": "REST",
      "description": "REST API",
```

(下页继续)

(续上页)


```
      "type": "REST_HTTP",
      "protocol": "HTTP",
      "version": "2.0",
      "endpoint": {},
      "security": {
        "oAuth2Info": {
          "grantTypes": [
            "OAUTH2_CLIENT_CREDENTIALS"
          ],
          "tokenEndpoint": "/mecSerMgmtApi/security/TokenEndPoint"
        }
      },
      "implSpecificInfo": {}
    },
    "serializer": "JSON",
    "scopeOfLocality": "MEC_SYSTEM",
    "consumedLocalOnly": false,
    "isLocal": true,
    "livenessInterval": 60,
    "_links": {
      "self": {
        "liveness" : "/mec_service_mgmt/v1/applications/5abe4782-2c70-4e47-9a4e-
↪0ee3a1a0fd1f/service/0bc92b06cc213d2ad8beda71bd0e1460/liveness"
      }
    }
  }
```


3. Query transport information

Transport information query provides a standardized means to the MEC applications to discover the available transports supported in MEP.

URL

```
GET mep/mec_service_mgmt/v1/transports
```

Request parameters

None

Body Parameters

None

Example Request

```
GET mep/mec_service_mgmt/v1/transports
```

Return Parameters

Return Code: 200 OK

Example Response

```
HTTP/1.1 200 OK
[
  {
    "id": "TransId12345",
    "name": "REST",
    "description": "REST API",
    "type": "REST_HTTP",
    "protocol": "HTTP",
    "version": "2.0",
    "endpoint": {},
    "security": {
      "oAuth2Info": {
        "grantTypes": [
          "OAUTH2_CLIENT_CREDENTIALS"
        ],
        "tokenEndpoint": "/mecSerMgmtApi/security/TokenEndPoint"
      }
    },
    "implSpecificInfo": {}
  }
]
```

Exception status code

Heartbeat related interfaces

1. Query service liveness info

This method retrieves information about an individual mec service liveness resource

URL

```
GET /mep/mec_service_mgmt/v1/applications/{appInstanceId}/services/{serviceId}/
↪liveness
```

Request parameters

Name	Type	Description	IN	Required
Authorization	String	Token 信息，格式：Bearer token 信息	header	Yes
appInstanceId	String	APP 实例 ID（UUID）	path	Yes
serviceId	String	APP 服务实例 ID	path	是

Body Parameters

None

Example Request

```
GET /mep/mec_service_mgmt/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/
↪services/0bc92b06cc213d2ad8beda71bd0e1460/liveness
```

Return Parameters

Name	Type	Description
state	enum **{ACTIVE, INACTIVE, SUSPENDED}**	State
timeStamp	object	The time when the last heartbeat message was received
>seconds	int32	The seconds partof the time
>nanoseconds	int32	The nano seconds part of the time
interval	int	Interval between two consecutive hearbeat message

Return Code: 200 OK

Example Response

```
HTTP/1.1 200 OK
{
    "state": "ACTIVE",
    "timeStamp": {
        "seconds": 1605249472,
        "nanoseconds": 436892745
      }
    "interval": 60
}
```

Exception status code

2. Update liveness info

This method also referred as "heartbeat" message to (re)confirm the active status about an individual mec service. But not to change the state from "INACTIVE" to "ACTIVE". NOTE: As per ETSI doc, this message should be a PATCH message. Since we use servicecomb rest server which doesn't support PATCH message. we kept it as PUT. But behviour will be similar to patch.

URL

```
PUT /mep/mec_service_mgmt/v1/applications/{appInstanceId}/services/{serviceId}/
↪liveness
```

Request parameters

Name	Type	Description	IN	Required
Authorization	String	Token 信息，格式：Bearer token 信息	header	Yes
appInstanceId	String	APP 实例 ID（UUID）	path	Yes
serviceId	String	APP 服务实例 ID	path	是

Body Parameters

Example Request

```
PUT /mep/mec_service_mgmt/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/
↪services/0bc92b06cc213d2ad8beda71bd0e1460/liveness
{
  state: "ACTIVE"
}
```

Return Parameters None

Return Code: 204 OK

Example Response Empty response

Exception status code

HTTP Status code	Description
400	Bad request, used to indicate that the requested parameters are incorrect.
403	The current operation is forbidden.
404	The requested resource was not found.
409	conflict state.

DNS rule configuration interfaces

Using the Mp1 interfaces mec apps can query and activate/deactivate dns rules associated to it. Implementation of this interface are as per the *ETSI GS MEC 011 V2.1.1* document.

1. Query all dns rules

Query all DNS rules associated with an application.

URL

```
GET /mep/mec_app_support/v1/applications/{appInstanceId}/dns_rules
```

Request parameters

Name	Type	Description	IN	Required
Authorization	String	Token 信息，格式：Bearer token 信息	header	Yes
appInstanceId	String	APP 实例 ID（UUID）	path	Yes

Body Parameters

None

Example Request

```
GET /mep/mec_app_support/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/dns_
↪rules
```

Return Parameters

Name	Type	Description
dnsRuleId	string	Rule Id
domainName	string	MEC app domain name
ipAddressType	enum **{IP_V4, IP_V6}**	MEC app ip address type
ipAddress	string	MEC app ip address
ttl	int (non-zero value)	TTL value
state	enum **{ACTIVE, INACTIVE}**	State

Return Code: 200 OK

Example Response

```
HTTP/1.1 200 OK
[
```

(下页继续)

(续上页)

```
    {
        "dnsRuleId": "20a19292-384c-4d44-ba64-a8e8269f0f25",
        "domainName": "voicerecgservice.com",
        "ipAddressType": "IP_V4",
        "ipAddress": "192.168.27.48",
        "ttl": 30,
        "state": "ACTIVE"
    },
    {
        "dnsRuleId": "bbc14ed1-92f4-457f-95e8-93aa723a9f12",
        "domainName": "facerecgservice.com",
        "ipAddressType": "IP_V4",
        "ipAddress": "192.168.147.240",
        "ttl": 30,
        "state": "INACTIVE"
    }
]
```

2. Query a specific dns rule

Query single DNS rule associated with an application.

URL

```
GET /mep/mec_app_support/v1/applications/{appInstanceId}/dns_rules/{dnsRuleId}
```

Request parameters

Name	Type	Description	IN	Required
Authorization	String	Token 信息，格式：Bearer token 信息	header	Yes
appInstanceId	String	APP 实例 ID(UUID)	path	Yes
dnsRuleId	String	DNS rule identifier(UUID)	path	Yes

Body Parameters

None

Example Request

```
GET /mep/mec_app_support/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/dns_
↪rules/bbc14ed1-92f4-457f-95e8-93aa723a9f12
```

Return Parameters

Name	Type	Description
dnsRuleId	string	Rule Id
domainName	string	MEC app domain name
ipAddressType	enum **{IP_V4, IP_V6}**	MEC app ip address type
ipAddress	string	MEC app ip address
ttl	int (non-zero value)	TTL value
state	enum **{ACTIVE, INACTIVE}**	State

Return Code: 200 OK

Example Response

```
HTTP/1.1 200 OK
{
    "dnsRuleId": "bbc14ed1-92f4-457f-95e8-93aa723a9f12",
    "domainName": "facerecgservice.com",
    "ipAddressType": "IP_V4",
    "ipAddress": "192.168.147.240",
    "ttl": 30,
    "state": "INACTIVE"
}
```

3. Update a specific dns rule

Modify the state of a rule associated with an application. This interface can modify the state from **ACTIVE** to **INACTIVE** or vice versa. No other field can be modified using this interface.

URL

```
PUT /mep/mec_app_support/v1/applications/{appInstanceId}/dns_rules/{dnsRuleId}
```

Request parameters

Name	Type	Description	IN	Required
Authorization	String	Token 信息，格式：Bearer token 信息	header	Yes
appInstanceId	String	APP 实例 ID(UUID)	path	Yes
dnsRuleId	String	DNS rule identifier(UUID)	path	Yes

Body Parameters

Name	Type	Description	Required
dnsRuleId	string	Rule Id	No, if present must be same with actual
domainName	string	MEC app domain name	No, if present must be same with actual
ipAddressType	enum **{IP_V4, IP_V6}**	MEC app ip address type	No, if present must be same with actual
ipAddress	string	MEC app ip address	No, if present must be same with actual
ttl	int (non-zero value)	TTL value	No, if present must be same with actual
state	enum **{ACTIVE, INACTIVE}**	State	Yes

Example Request

```
PUT /mep/mec_app_support/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/dns_
↪rules/bbc14ed1-92f4-457f-95e8-93aa723a9f12
{
    "dnsRuleId": "bbc14ed1-92f4-457f-95e8-93aa723a9f12",
    "domainName": "facerecgservice.com",
    "ipAddressType": "IP_V4",
    "ipAddress": "192.168.147.240",
    "ttl": 30,
    "state": "ACTIVE"
}
```

Return Parameters

Name	Type	Description
dnsRuleId	string	Rule Id
domainName	string	MEC app domain name
ipAddressType	enum **{IP_V4, IP_V6}**	MEC app ip address type
ipAddress	string	MEC app ip address
ttl	int (non-zero value)	TTL value
state	enum **{ACTIVE, INACTIVE}**	State

Return Code: 200 OK

Example Response

```
HTTP/1.1 200 OK
{
    "dnsRuleId": "bbc14ed1-92f4-457f-95e8-93aa723a9f12",
    "domainName": "facerecgservice.com",
    "ipAddressType": "IP_V4",
    "ipAddress": "192.168.147.240",
    "ttl": 30,
    "state": "ACTIVE"
}
```

Traffic rule configuration interfaces

Using the Mp1 interfaces mec apps can query and modify the traffic rules associated to it. Implementation of this interface are as per the *ETSI GS MEC 011 V2.1.1* document.

1. Query all traffic rules

Query all traffic rules associated with an application.

URL

```
GET /mep/mec_app_support/v1/applications/{appInstanceId}/traffic_rules
```

Request parameters

Name	Type	Description	IN	Required
Authorization	String	Token 信息，格式：Bearer token 信息	header	Yes
appInstanceId	String	APP 实例 ID（UUID）	path	Yes

Body Parameters

None

Example Request

```
GET /mep/mec_app_support/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/traffic_
↪rules
```

Return Parameters

Name	Type	Description
trafficRuleId	string	Traffic rule ID.
filterType	enum **{FLOW, PACKET}**	Filter type.
priority	number (1 ~ 256)	Priority of the traffic rule. If any conflict in the traffic rules, then higher the priority higher the precedence.
trafficFilter	list	List of traffic filters.
> srcAddress	list	List of source ip address.
> dstAddress	list	List of destination ip address.
> srcPort	list	List of source port address.
> dstPort	list	List of destination port address.
> protocol	list	List of protocols.
> tag	list	List of tags.
> srcTunnelAddress	list	List of source tunnel address.
> tgtTunnelAddress	list	List of target tunnel address.
> srcTunnelPort	list	List of source tunnel ports.
> dstTunnelPort	list	List of destination tunnel ports.
> qCI	number	Match packets with same QCI.
> dSCP	number	Match packets with same DSCP.
> tC	number	Match packets with same TC.
action	enum **{DROP, FORWARD_DECAPSULATED, FORWARD_AS_IS, PASSTHROUGH, DUPLICATE_DECAPSULATED, DUPLICATE_AS_IS}**	Action when traffic rule matches.
dstInterface	list	Destination interface parameters.

Return Code: 200 OK

Example Response

```
HTTP/1.1 200 OK
[
  {
    "trafficRuleId": "TrafficRule1",
    "filterType": "FLOW",
    "priority": 1,
    "trafficFilter": [
      {
        "srcAddress": [
          "192.158.1.1/28"
        ],
        "dstAddress": [
          "192.168.1.1/28"
        ],
        "srcPort": [
          "8080"
        ],
        "dstPort": [
          "8080"
        ],
        "protocol": [
          "TCP"
        ],
        "tag": null,
        "srcTunnelAddress": null,
        "tgtTunnelAddress": null,
        "srcTunnelPort": null,
        "dstTunnelPort": null,
        "qCI": 2,
        "dSCP": 0,
        "tC": 1
      }
    ],
    "action": "DROP",
    "dstInterface": [],
    "state": "ACTIVE"
  }
]
```

2. Query a specific traffic rule

Query single traffic rule associated with an application.

URL

```
GET /mep/mec_app_support/v1/applications/{appInstanceId}/traffic_rules/{trafficRuleId}
```

Request parameters

Name	Type	Description	IN	Required
Authorization	String	Token 信息，格式：Bearer token 信息	header	Yes
appInstanceId	String	APP 实例 ID(UUID)	path	Yes
trafficRuleId	String	Traffic rule identifier(UUID)	path	Yes

Body Parameters

None

Example Request

```
GET /mep/mec_app_support/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/traffic_
↪rules/TrafficRule1
```

Return Parameters

Name	Type	Description
trafficRuleId	string	Traffic rule ID.
filterType	enum **{FLOW, PACKET}**	Filter type.
priority	number (1 ~ 256)	Priority of the traffic rule. If any conflict in the traffic rules, then higher the priority higher the precedence.
trafficFilter	list	List of traffic filters.
> srcAddress	list	List of source ip address.
> dstAddress	list	List of destination ip address.
> srcPort	list	List of source port address.
> dstPort	list	List of destination port address.
> protocol	list	List of protocols.
> tag	list	List of tags.
> srcTunnelAddress	list	List of source tunnel address.
> tgtTunnelAddress	list	List of target tunnel address.
> srcTunnelPort	list	List of source tunnel ports.
> dstTunnelPort	list	List of destination tunnel ports.
> qCI	number	Match packets with same QCI.
> dSCP	number	Match packets with same DSCP.
> tC	number	Match packets with same TC.
action	enum **{DROP, FORWARD_DECAPSULATED, FORWARD_AS_IS, PASSTHROUGH, DUPLICATE_DECAPSULATED, DUPLICATE_AS_IS}**	Action when traffic rule matches.
dstInterface	list	Destination interface parameters.

Return Code: 200 OK

Example Response

```
HTTP/1.1 200 OK
{
  "trafficRuleId": "TrafficRule1",
  "filterType": "FLOW",
  "priority": 1,
  "trafficFilter": [
    {
      "srcAddress": [
        "192.158.1.1/28"
      ],
      "dstAddress": [
        "192.168.1.1/28"
      ],
      "srcPort": [
        "8080"
      ],
      "dstPort": [
        "8080"
      ],
      "protocol": [
        "TCP"
      ],
      "tag": null,
      "srcTunnelAddress": null,
      "tgtTunnelAddress": null,
      "srcTunnelPort": null,
      "dstTunnelPort": null,
      "qCI": 2,
      "dSCP": 0,
      "tC": 1
    }
  ],
  "action": "DROP",
  "dstInterface": [],
  "state": "ACTIVE"
}
```

3. Update a specific traffic rule

Modify the traffic rule associated with an application.

URL

```
PUT /mep/mec_app_support/v1/applications/{appInstanceId}/traffic_rules/{trafficRuleId}
```

Request parameters

Name	Type	Description	IN	Required
Authorization	String	Token 信息，格式：Bearer token 信息	header	Yes
appInstanceId	String	APP 实例 ID(UUID)	path	Yes
trafficRuleId	String	Traffic rule identifier(UUID)	path	Yes

Body Parameters

Name	Type	Description
trafficRuleId	string	Traffic rule ID.
filterType	enum **{FLOW, PACKET}**	Filter type.
priority	number (1 ~ 256)	Priority of the traffic rule. If any conflict in the traffic rules, then higher the priority higher the precedence.
trafficFilter	list	List of traffic filters.
> srcAddress	list	List of source ip address.
> dstAddress	list	List of destination ip address.
> srcPort	list	List of source port address.
> dstPort	list	List of destination port address.
> protocol	list	List of protocols.
> tag	list	List of tags.
> srcTunnelAddress	list	List of source tunnel address.
> tgtTunnelAddress	list	List of target tunnel address.
> srcTunnelPort	list	List of source tunnel ports.
> dstTunnelPort	list	List of destination tunnel ports.
> qCI	number	Match packets with same QCI.
> dSCP	number	Match packets with same DSCP.
> tC	number	Match packets with same TC.
action	enum **{DROP, FORWARD_DECAPSULATED, FORWARD_AS_IS, PASSTHROUGH, DUPLICATE_DECAPSULATED, DUPLICATE_AS_IS}**	Action when traffic rule matches.
dstInterface	list	Destination interface parameters.

Example Request

```
PUT /mep/mec_app_support/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/traffic_
↪rules/TrafficRule1
{
  "trafficRuleId": "TrafficRule1",
  "filterType": "FLOW",
  "priority": 1,
  "trafficFilter": [
    {
      "srcAddress": [
        "192.158.1.1/28"
      ],
      "dstAddress": [
        "192.168.1.1/28"
      ],
      "srcPort": [
        "8080"
      ],
      "dstPort": [
        "8080"
      ],
      "protocol": [
        "TCP"
      ],
      "tag": null,
      "srcTunnelAddress": null,
      "tgtTunnelAddress": null,
      "srcTunnelPort": null,
      "dstTunnelPort": null,
      "qCI": 2,
      "dSCP": 0,
      "tC": 1
    }
  ],
  "action": "DROP",
  "dstInterface": [],
  "state": "ACTIVE"
}
```

Return Parameters

Name	Type	Description
trafficRuleId	string	Traffic rule ID.
filterType	enum **{FLOW, PACKET}**	Filter type.
priority	number (1 ~ 256)	Priority of the traffic rule. If any conflict in the traffic rules, then higher the priority higher the precedence.
trafficFilter	list	List of traffic filters.
> srcAddress	list	List of source ip address.
> dstAddress	list	List of destination ip address.
> srcPort	list	List of source port address.
> dstPort	list	List of destination port address.
> protocol	list	List of protocols.
> tag	list	List of tags.
> srcTunnelAddress	list	List of source tunnel address.
> tgtTunnelAddress	list	List of target tunnel address.
> srcTunnelPort	list	List of source tunnel ports.
> dstTunnelPort	list	List of destination tunnel ports.
> qCI	number	Match packets with same QCI.
> dSCP	number	Match packets with same DSCP.
> tC	number	Match packets with same TC.
action	enum **{DROP, FORWARD_DECAPSULATED, FORWARD_AS_IS, PASSTHROUGH, DUPLICATE_DECAPSULATED, DUPLICATE_AS_IS}**	Action when traffic rule matches.
dstInterface	list	Destination interface parameters.

Return Code: 200 OK

Example Response

```
HTTP/1.1 200 OK
{
  "trafficRuleId": "TrafficRule1",
  "filterType": "FLOW",
  "priority": 1,
  "trafficFilter": [
    {
      "srcAddress": [
        "192.158.1.1/28"
      ],
      "dstAddress": [
        "192.168.1.1/28"
      ],
      "srcPort": [
        "8080"
      ],
      "dstPort": [
        "8080"
      ],
      "protocol": [
        "TCP"
      ],
      "tag": null,
      "srcTunnelAddress": null,
      "tgtTunnelAddress": null,
      "srcTunnelPort": null,
      "dstTunnelPort": null,
      "qCI": 2,
      "dSCP": 0,
      "tC": 1
    }
  ],
  "action": "DROP",
  "dstInterface": [],
  "state": "ACTIVE"
}
```

Mm5 Interface for appd configurations

Using the Mm5 interfaces MECM can create, query, update or delete the appd configurations which includes multiple dns and traffic rules associated to an application. Implementation of this interface are specified in *ETSI GS MEC 010-1 V1.1.1* and *ETSI GS MEC 010-2 V2.1.1* documents.

1. Create a new appd configurations

Create a new appd configuration and associate it with a MEC application. Each configuration can have thirty-two DNS rules and sixteen Traffic rules. There can be only one appd configuration per application at a time and the appd configuration is an asynchronous in nature. Once the request is received in the mep, after validation mep will start a task to handle multiple dns and traffic rules and return a task id to the caller of the API. Using the task-id the caller can check the status of the operation at any time.

URL

```
POST /mepcfg/app_lcm/v1/applications/{appInstanceId}/appd_configuration
```

Request parameters

Name	Type	Description	IN	Required
appInstanceId	String	APP 实例 ID(UUID)	path	Yes

Body Parameters

Name	Type
appTrafficRule	list
> trafficRuleId	string
> filterType	enum **{FLOW, PACKET}**
> priority	number (1 ~ 256)
> trafficFilter	list
» srcAddress	list
» dstAddress	list
» srcPort	list
» dstPort	list
» protocol	list
» tag	list
» srcTunnelAddress	list
» tgtTunnelAddress	list
» srcTunnelPort	list
» dstTunnelPort	list

Name	Type
» qCI	number
» dSCP	number
» tC	number
> action	enum **{DROP, FORWARD_DECAPSULATED, FORWARD_AS_IS, PASSTHROUGH, DUPLICATE_**
> dstInterface	list
» interfaceType	enum **{TUNNEL, MAC, IP}**
» tunnelInfo	object
»> tunnelType	enum **{GTP_U, GRE}**
»> tunnelDstAddress	string
»> tunnelSrcAddress	string
» srcMacAddress	string
» dstMacAddress	string
» dstIpAddress	string
> state	enum **{ACTIVE, INACTIVE}**
appDNSRule	list
> dnsRuleId	string
> domainName	string
> ipAddressType	enum **{IPv4, IPv6}**
> ipAddress	string
> ttl	number
> state	enum **{ACTIVE, INACTIVE}**
appSupportMp1	bool
appName	string

Example Request

```
POST /mepcfg/app_lcm/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/appd_
↪configuration
{
  "appTrafficRule": [
    {
      "trafficRuleId": "TrafficRule1",
      "filterType": "FLOW",
      "priority": 1,
      "trafficFilter": [
        {
          "srcAddress": [
            "192.158.1.1/28"
          ],
          "dstAddress": [
```

(下页继续)

(续上页)


```
            "192.168.1.1/28"
          ],
          "srcPort": [
            "8080"
          ],
          "dstPort": [
            "8080"
          ],
          "protocol": [
            "TCP"
          ],
          "qCI": 1,
          "dSCP": 0,
          "tC": 1
        }
      ],
      "action": "DROP",
      "state": "ACTIVE"
    }
  ],
  "appDNSRule": [
    {
      "dnsRuleId": "dnsRule1",
      "domainName": "facerecgservice.com",
      "ipAddressType": "IP_V4",
      "ipAddress": "192.168.147.240",
      "ttl": 30,
      "state": "ACTIVE"
    }
  ],
  "appSupportMp1": true,
  "appName": "abc1"
}
```


Return Parameters

Name	Type	Description
taskId	string	Task ID
appInstanceId	string	Application instance ID
configResult	string	Configuration result
configPhase	string	Progress percentage
Detailed	string	Detailed reason

Return Code: 200 OK

Example Response

```
HTTP/1.1 200 OK
{
  "taskId": "d464dd0a-d927-4baf-bc2c-d902fb0b2c73",
  "appInstanceId": "5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f",
  "configResult": "PROCESSING",
  "configPhase": "0",
  "Detailed": "Operation In progress"
}
```

2. Query appd configuration

Query appd configuration associated with an application.

URL

```
GET /mepcfg/app_lcm/v1/applications/{appInstanceId}/appd_configuration
```

Request parameters

Name	Type	Description	IN	Required
appInstanceId	String	APP 实例 ID(UUID)	path	Yes

Body Parameters

None

Example Request

```
GET /mepcfg/app_lcm/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/appd_
↪configuration
```

Return Parameters

Name	Type
appTrafficRule	list
> trafficRuleId	string
> filterType	enum **{FLOW, PACKET}**
> priority	number (1 ~ 256)
> trafficFilter	list
» srcAddress	list

Name	Type
» dstAddress	list
» srcPort	list
» dstPort	list
» protocol	list
» tag	list
» srcTunnelAddress	list
» tgtTunnelAddress	list
» srcTunnelPort	list
» dstTunnelPort	list
» qCI	number
» dSCP	number
» tC	number
> action	enum **{DROP, FORWARD_DECAPSULATED, FORWARD_AS_IS, PASSTHROUGH, DUPLICATE_**
> dstInterface	list
» interfaceType	enum **{TUNNEL, MAC, IP}**
» tunnelInfo	object
»> tunnelType	enum **{GTP_U, GRE}**
»> tunnelDstAddress	string
»> tunnelSrcAddress	string
» srcMacAddress	string
» dstMacAddress	string
» dstIpAddress	string
> state	enum **{ACTIVE, INACTIVE}**
appDNSRule	list
> dnsRuleId	string
> domainName	string
> ipAddressType	enum **{IPv4, IPv6}**
> ipAddress	string
> ttl	number
> state	enum **{ACTIVE, INACTIVE}**
appSupportMp1	bool
appName	string

Return Code: 200 OK

Example Response

```
HTTP/1.1 200 OK
{
```

(下页继续)

(续上页)


```
  "appTrafficRule": [
    {
      "trafficRuleId": "TrafficRule1",
      "filterType": "FLOW",
      "priority": 1,
      "trafficFilter": [
        {
          "srcAddress": [
            "192.158.1.1/28"
          ],
          "dstAddress": [
            "192.168.1.1/28"
          ],
          "srcPort": [
            "8080"
          ],
          "dstPort": [
            "8080"
          ],
          "protocol": [
            "TCP"
          ],
          "tag": null,
          "srcTunnelAddress": null,
          "tgtTunnelAddress": null,
          "srcTunnelPort": null,
          "dstTunnelPort": null,
          "qCI": 2,
          "dSCP": 0,
          "tC": 1
        }
      ],
      "action": "DROP",
      "dstInterface": [],
      "state": "ACTIVE"
    }
  ],
  "appDNSRule": [
    {
      "dnsRuleId": "dnsRule1",
      "domainName": "facerecgservice.com",
      "ipAddressType": "IP_V4",
      "ipAddress": "192.168.147.240",
```


(下页继续)

(续上页)

```
      "ttl": 30,
      "state": "ACTIVE"
    }
  ],
  "appSupportMp1": true,
  "appName": "abc1"
}
```

3. Modify appd configuration

Appd configuration for each applications can be configured using this interface. MEP will take the difference of existing configuration and the new modification input and then act on create/modify/delete of dns and/or traffic rules.

URL

```
PUT /mepcfg/app_lcm/v1/applications/{appInstanceId}/appd_configuration
```

Request parameters

Name	Type	Description	IN	Required
appInstanceId	String	APP 实例 ID(UUID)	path	Yes

Body Parameters

Name	Type
appTrafficRule	list
> trafficRuleId	string
> filterType	enum **{FLOW, PACKET}**
> priority	number (1 ~ 256)
> trafficFilter	list
» srcAddress	list
» dstAddress	list
» srcPort	list
» dstPort	list
» protocol	list
» tag	list
» srcTunnelAddress	list
» tgtTunnelAddress	list
» srcTunnelPort	list
» dstTunnelPort	list

Name	Type
» qCI	number
» dSCP	number
» tC	number
> action	enum **{DROP, FORWARD_DECAPSULATED, FORWARD_AS_IS, PASSTHROUGH, DUPLICATE_**
> dstInterface	list
» interfaceType	enum **{TUNNEL, MAC, IP}**
» tunnelInfo	object
»> tunnelType	enum **{GTP_U, GRE}**
»> tunnelDstAddress	string
»> tunnelSrcAddress	string
» srcMacAddress	string
» dstMacAddress	string
» dstIpAddress	string
> state	enum **{ACTIVE, INACTIVE}**
appDNSRule	list
> dnsRuleId	string
> domainName	string
> ipAddressType	enum **{IPv4, IPv6}**
> ipAddress	string
> ttl	number
> state	enum **{ACTIVE, INACTIVE}**
appSupportMp1	bool
appName	string

Example Request

```
PUT /mepcfg/app_lcm/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/appd_
↪configuration
{
  "appTrafficRule": [
    {
      "trafficRuleId": "TrafficRule1",
      "filterType": "FLOW",
      "priority": 1,
      "trafficFilter": [
        {
          "srcAddress": [
            "192.158.1.1/28"
          ],
          "dstAddress": [
```

(下页继续)

(续上页)


```
            "192.168.1.1/28"
          ],
          "srcPort": [
            "8080"
          ],
          "dstPort": [
            "8080"
          ],
          "protocol": [
            "TCP"
          ],
          "qCI": 1,
          "dSCP": 0,
          "tC": 1
        }
      ],
      "action": "DROP",
      "state": "ACTIVE"
    }
  ],
  "appDNSRule": [
    {
      "dnsRuleId": "dnsRule1",
      "domainName": "facerecgservice.com",
      "ipAddressType": "IP_V4",
      "ipAddress": "192.168.147.240",
      "ttl": 30,
      "state": "ACTIVE"
    }
  ],
  "appSupportMp1": true,
  "appName": "abc1"
}
```


Return Parameters

Name	Type	Description
taskId	string	Task ID
appInstanceId	string	Application instance ID
configResult	string	Configuration result
configPhase	string	Progress percentage
Detailed	string	Detailed reason

Return Code: 200 OK

Example Response

```
HTTP/1.1 200 OK
{
  "taskId": "d464dd0a-d927-4baf-bc2c-d902fb0b2c73",
  "appInstanceId": "5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f",
  "configResult": "PROCESSING",
  "configPhase": "0",
  "Detailed": "Operation In progress"
}
```

4. Delete appd configuration

Delete appd configuration request.

URL

```
DELETE /mepcfg/app_lcm/v1/applications/{appInstanceId}/appd_configuration
```

Request parameters

Name	Type	Description	IN	Required
appInstanceId	String	APP 实例 ID(UUID)	path	Yes

Body Parameters

None

Example Request

```
DELETE /mepcfg/app_lcm/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/appd_
↪configuration
```

Return Parameters

Name	Type	Description
taskId	string	Task ID
appInstanceId	string	Application instance ID
configResult	string	Configuration result
configPhase	string	Progress percentage
Detailed	string	Detailed reason

Return Code: 200 OK

Example Response

```
HTTP/1.1 200 OK
{
  "taskId": "d464dd0a-d927-4baf-bc2c-d902fb0b2c73",
  "appInstanceId": "5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f",
  "configResult": "PROCESSING",
  "configPhase": "0",
  "Detailed": "Operation In progress"
}
```

5. Query task status

Once the appd configuration create, modify or delete is submitted, mep will create a task and return a task id to the caller and the caller can periodically check the status of the task using this id. This interface calls the mep for the task status.

URL

```
GET /mepcfg/app_lcm/v1/tasks/{taskId}/appd_configuration
```

Request parameters

Name	Type	Description	IN	Required
taskId	String	Task ID(UUID)	path	Yes

Body Parameters

None

Example Request

```
GET /mepcfg/app_lcm/v1/tasks/d464dd0a-d927-4baf-bc2c-d902fb0b2c73/appd_configuration
```

Return Parameters

Name	Type	Description
taskId	string	Task ID
appInstanceId	string	Application instance ID
configResult	string	Configuration result
configPhase	string	Progress percentage
Detailed	string	Detailed reason

Return Code: 200 OK

Example Response

```
HTTP/1.1 200 OK
{
  "taskId": "d464dd0a-d927-4baf-bc2c-d902fb0b2c73",
  "appInstanceId": "5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f",
  "configResult": "SUCCESS",
  "configPhase": "100",
  "Detailed": ""
}
```

Application Termination

MEP support termination/stop of instance of an application after MEP receive the request from MECM.

1. App Instance Termination

Interface to remove the application’ s ak/sk values, unregister the services and delete the DNS and traffic rule.

URL

```
DELETE /mep/mec_app_support/v1/applications/{appInstanceId}/AppInstanceTermination
```

Request parameters

Request parameters

Name	Type	Description	IN	Required
appInstanceId	String	APP 实例 ID(UUID)	path	Yes
appInstanceId	String	APP 实例 ID(UUID)	header	Yes

Example Request

```
DELETE /mep/mec_app_support/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/
↪AppInstanceTermination
```

Body Parameters

None

Return Code: 200 OK

Example Response

```
""
```

2. App Instance Termination Confirmation

After the MEC platform receives a request to terminate or stop a MEC application instance, the MEC platform notifies the MEC application instance that it will be terminated or stopped soon incase if graceful termination/stop is subscribed.

URL

```
POST /mep/mec_app_support/v1/applications/{appInstanceId}/confirm_termination
```

Request parameters

Name	Type	Description	IN	Required
appInstanceId	String	APP 实例 ID(UUID)	path	Yes

Example Request

```
POST /mep/mec_app_support/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/
↪confirm_termination
```

Body Parameters

```
{
    "notificationType": "AppTerminationNotification",
    "operationAction": "STOPPING",
    "maxGracefulTimeout": "30",
    "_links": {
      "subscription": "mep/mec_app_support/v1/applications/6abe4782-2c70-4e47-9a4e-
↪0ee3a1a0fd1e/subscriptions/6abe4782",
      "confirmTermination": "/mep/mec_app_support/v1/applications/6abe4782-2c70-4e47-
↪9a4e-0ee3a1a0fd1e/confirm_termination",
    },
}

Return Code: 204 No Content

**Example Response**
```

POST /mep/mec_app_support/v1/applications/5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f/confirm_termination { “operationAction” : “TERMINATING” }

```

Exception status code

<table border="1" class="docutils">
<thead>
```

(下页继续)

(续上页)

```
<tr>
<th><strong>Data type</strong></th>
<th><strong>Response codes</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>204 No Content</td>
<td>The request is acknowledged. The response body shall be empty.</td>
</tr>
<tr>
<td>ProblemDetails</td>
<td>401 Unauthorized</td>
<td>It is used when the client did not submit the appropriate credentials. In the␣
↪returned ProblemDetails structure, the "detail" attribute should convey more␣
↪information about the error.</td>
</tr>
<tr>
<td>ProblemDetails</td>
<td>403 Forbidden</td>
<td>The operation is not allowed given the current status of the resource. More␣
↪information shall be provided in the "detail" attribute of the "ProblemDetails"␣
↪structure.</td>
</tr>
<tr>
<td>ProblemDetails</td>
<td>404 Not Found</td>
<td>It is used when a client provided a URI that cannot be mapped to a valid resource␣
↪URI. In the returned ProblemDetails structure, the "detail" attribute should convey␣
↪more information about the error.</td>
</tr>
<tr>
<td>ProblemDetails</td>
<td>409 Conflict</td>
<td>The operation cannot be executed currently, due to a conflict with the state of␣
↪the resource. Typically, this is because the application instance resource is in␣
↪NOT_INSTANTIATED state or because there is no termination ongoing. The response␣
↪body shall contain a ProblemDetails structure, in which the "detail" attribute␣
↪shall convey more information about the error.</td>
</tr>
<tr>
```

(下页继续)

(续上页)

```
<td>ProblemDetails</td>
<td>429 Too Many Requests</td>
<td>It is used when a rate limiter has triggered. In the returned ProblemDetails␣
↪structure, the "detail" attribute should convey more information about the error.</
↪td>
</tr>
</tbody>
</table>

### Query Platform Capabilities(Services)

MEP supports for querying the capabilities(services) registered with it. These␣
↪capability information will be used by the MECM to display it to the user on its␣
↪portal. MECM send the capability query to MEP over Mm5 interface.

#### 1. Query all capabilities

Interface to query all capabilities. This interface return the capability list␣
↪registered to the queried MEP server along with the consumers of it.

**URL**
```

GET /mepcfg/mec_platform_config/v1/capabilities

```

**Request parameters**

None

**Body Parameters**

None

**Example Request**
```

GET /mepcfg/mec_platform_config/v1/capabilities

```

**Return Parameters**
| Name | Type | Description |
| --- | --- | --- |
| capabilityId | String | Capability/Service id |
```

(下页继续)

(续上页)

```
| capabilityName | String                        | Capability/Service name  |
| status         | Enum **{ACTIVE, INACTIVE}**   | Status                   |
| version        | String                        | Version info             |
| consumers      | Array[Object]                 | Consumer object list     |
| &gt;applicationInstanceId | String             | Consumer application id  |

Return Code: 200 OK

**Example Response**
```

HTTP/1.1 200 OK [{ “capabilityId”: “16384563dca094183778a41ea7701d15”, “capabilityName”: “FaceRegService” , “status”: “ACTIVE”, “version”: “4.5.8”, “consumers”: [{ “applicationInstanceId”: “5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f” }, { “applicationInstanceId”: “f05a5591-d8f2-4f89-8c0b-8cea6d45712e” }, { “applicationInstanceId” : “86dfc97d-325e-4feb-ac4f-280a0ba42513” }] }, { “capabilityId” : “f7e898d1c9ea9edd05e1181bc09afc5e” , “capabilityName”: “Location”, “status”: “ACTIVE”, “version”: “v1.19”, “consumers”: [{ “applicationInstanceId” : “88922760-861b-4578-aae5-77b8fcb06142” }] }]

```
#### 2. Query individual capability

Interface to query an individual capability from MEP. This interface return a
↪capability along with the consumers of it.

**URL**
```

GET /mepcfg/mec_platform_config/v1/capabilities/{capabilityId}

```
**Request parameters**
| **Name** | **Type** | **Description** | **IN** | **Required** |
| --- | --- | --- | --- | --- |
| capabilityId  | String | Capability ID (UUID)   | path |  Yes   |

**Body Parameters**

None

**Example Request**
```

GET /mepcfg/mec_platform_config/v1/capabilities/16384563dca094183778a41ea7701d15

```
**Return Parameters**
| Name            | Type                          | Description              |
| --------------- | ----------------------------- | ------------------------ |
| capabilityId    | String                        | Capability/Service id    |
| capabilityName  | String                        | Capability/Service name  |
| status          | Enum **{ACTIVE, INACTIVE}**   | Status                   |
| version         | String                        | Version info             |
| consumers       | Array[Object]                 | Consumer object list     |
| &gt;applicationInstanceId | String              | Consumer application id  |

Return Code: 200 OK

**Example Response**
```

HTTP/1.1 200 OK { “capabilityId”: “16384563dca094183778a41ea7701d15”, “capabilityName”: “FaceRegService” , “status”: “ACTIVE”, “version”: “4.5.8”, “consumers”: [{ “applicationInstanceId”: “5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f” }, { “applicationInstanceId”: “f05a5591-d8f2-4f89-8c0b-8cea6d45712e” }, { “applicationInstanceId” : “86dfc97d-325e-4feb-ac4f-280a0ba42513” }] }

```
### Timing interfaces

Interface to get platform time for MEP applications.

#### 1. Query platform time

Interface to get the current MEP platform time.

**URL**
```

GET mep/mec_app_support/v1/timing/current_time

```
**Request parameters**

None

<table border="1" class="docutils">
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
```

(下页继续)

(续上页)

```
</thead>
<tbody>
<tr>
<td>seconds</td>
<td>integer</td>
<td>The seconds part of the time in unix time format.</td>
</tr>
<tr>
<td>nanoSeconds</td>
<td>integer</td>
<td>The nano-seconds part of the time in unix time format.</td>
</tr>
<tr>
<td>timeSourceStatus</td>
<td>enum{TRACEABLE, NONTRACEABLE}</td>
<td>1 = TRACEABLE - time source is locked to the UTC time source. 2 = NONTRACEABLE -␣
↪time source is not locked to the UTC time source</td>
</tr>
</tbody>
</table>

**Example Request**
```

GET mep/mec_app_support/v1/timing/current_time

```

**Body Parameters**

None

Return Code: 200 OK

**Example Response**
```

HTTP/1.1 200 OK { “seconds” : 1627971217, “nanoSeconds” : 869279458, “timeSourceStatus” : “TRACEABLE” }

```
#### 2. Query timing capability

Interface to get the timing capabilities. The information regarding available packet␣
↪timing facilities.
```

(下页继续)

(续上页)

```
**URL**
```

GET mep/mec_app_support/v1/timing/timing_caps

```
**Request parameters**

None

Return Parameters:

<table border="1" class="docutils">
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>timeStamp</td>
<td>Object</td>
<td>Time stamp information.</td>
<td>No</td>
</tr>
<tr>
<td>&gt;seconds</td>
<td>integer</td>
<td>The seconds part of the time in unix time format.</td>
<td>Yes</td>
</tr>
<tr>
<td>&gt;nanoSeconds</td>
<td>integer</td>
<td>The nano-seconds part of the time in unix time format.</td>
<td>Yes</td>
</tr>
<tr>
<td>ntpServers</td>
<td>Array[Object]</td>
<td>Details about all NTP server available in the platform.</td>
<td>No</td>
```

(下页继续)

(续上页)

```
</tr>
<tr>
<td>&gt;ntpServerAddrType</td>
<td>enum{IP_ADDRESS, DNS_NAME}</td>
<td>Address type.</td>
<td>Yes</td>
</tr>
<tr>
<td>&gt;ntpServerAddr</td>
<td>string</td>
<td>NTP server address.</td>
<td>Yes</td>
</tr>
<tr>
<td>&gt;minPollingInterval</td>
<td>integer</td>
<td>Minimum poll interval for NTP messages, in seconds as a power of two Range: 3…17</
↪td>
<td>Yes</td>
</tr>
<tr>
<td>&gt;maxPollingInterval</td>
<td>integer</td>
<td>Maximum poll interval for NTP messages, in seconds as a power of two Range: 3…17</
↪td>
<td>Yes</td>
</tr>
<tr>
<td>&gt;localPriority</td>
<td>integer</td>
<td>Local priority value.</td>
<td>Yes</td>
</tr>
<tr>
<td>&gt;authenticationOption</td>
<td>enum{NONE, SYMMETRIC_KEY, AUTO_KEY}</td>
<td>Available authentication option.</td>
<td>Yes</td>
</tr>
<tr>
<td>&gt;authenticationKeyNum</td>
<td>integer</td>
<td>Authentication key number. This configuration is valid if selected␣
↪authenticationOption is SymmetricKey</td>
```

(下页继续)

(续上页)

```
<td>Yes</td>
</tr>
<tr>
<td>ptpMasters</td>
<td>Array[Object]</td>
<td>Details about all PTP server available in the platform.</td>
<td>No</td>
</tr>
<tr>
<td>&gt;ptpMasterIpAddress</td>
<td>string</td>
<td>PTP master IP address.</td>
<td>Yes</td>
</tr>
<tr>
<td>&gt;ptpMasterLocalPriority</td>
<td>integer</td>
<td>Local priority number.</td>
<td>Yes</td>
</tr>
<tr>
<td>&gt;delayReqMaxRate</td>
<td>integer</td>
<td>Acceptable maximum rate of the Delay_Req messages in packets per second</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Example Request**
```

GET mep/mec_app_support/v1/timing/timing_caps

```
**Body Parameters**

None

Return Code: 200 OK

**Example Response**
```

HTTP/1.1 200 OK { “timeStamp” : { “seconds” : 1627971264, “nanoSeconds” : 478959804 }, “ntpServers” : [{

“ntpServerAddrType”: “DNS_NAME”, “ntpServerAddr”: “mep-ntp”, “minPollingInterval”: 4, “maxPollingInterval” : 17, “localPriority” : 1, “authenticationOption” : “NONE” , “authenticationKeyNum” : 0 }] }

```
### Abnormal status code
|**HTTP 状态码 **|** 描述 **|
|---|---|
|400| 错误的请求，用来表示请求的参数不正确。|
|401| 当前请求需要鉴权认证。|
|403| 禁止当前操作。|
|404| 请求的资源未被发现。|
|412| 请求中的先决条件验证失败。|
|414| 请求的 URI 超长，服务器拒绝处理请求。|
|500| 内部服务器错误。|
|503| 服务不可用。|

## Dns-Server

Mep has a programmable dns server which can perform the name resolution of the MEC␣
↪applications. This management interface of the dns server is handled using a rest␣
↪interface and this section cover the details of this interface.

### 1. Create new entry

DNS entry can be added using this interface. Once a record is added, the resource␣
↪will be available for the device application to query. Using this interface␣
↪multiple records for a zone can be submitted together.

**URL**
```

POST /mep/dns_server_mgmt/v1/rrecord

```
**Request parameters**

None

**Query parameters**
| Name          | Type                        | Description              | Required  ␣
↪    |
| ------------- | --------------------------- | ------------------------ | ----------
↪-- |
| zone          | string                      | Zone name. Default is "." | Yes |
```

(下页继续)

(续上页)

```
**Body Parameters**
| Name          | Type                          | Description              | Required  ␣
↪    |
| ------------- | ----------------------------- | ------------------------ | -----------
↪-- |
| name          | string                        | Domain name | Yes |
| type          | enum **{A, AAAA}**            | DNS resource type, A in case of Ipv4␣
↪and AAAA for Ipv6 | Yes |
| class         | enum **{IN}**                 | Resource record class | Yes on create␣
↪|
| ttl           | int (non-zero value)          | Record TTL value | Yes on create |
| rData         | list(string)                  | IP Address list, Ip type must match␣
↪with the **type** field | Yes on create |

**Example Request**
```

POST /mep/dns_server_mgmt/v1/rrecord?zone=" com." [{ "name" : "www.example.com." , "type" : "A" , "class" : "IN" , "ttl" : 30, "rData" : ["172.168.15.101"] }]

```
**Return Parameters**

None

Return Code: 200 Success

**Example Response**
```

HTTP/1.1 200 Success

```
### 2. Modify the entry

DNS entry can be modified using this interface. Once a record is modified, the␣
↪resource will be available for the device application to query. Using this␣
↪interface multiple records for a zone can be submitted together.

**URL**
```

PUT /mep/dns_server_mgmt/v1/rrecord/{fqdn}/{rrtype}

```
**Request parameters**
| **Name** | **Type** | **Description** | **IN** | **Required** |
| --- | --- | --- | --- | --- |
| fqdn  | String | Fully Qualified Domain Name | path |  Yes   |
| rrtype      | String | Resource record type of the entry(A or AAAA). | path | Yes |

**Query parameters**
| Name          | Type                         | Description              | Required  ␣
↪    |
| ------------- | ---------------------------- | ------------------------ | -----------
↪-- |
| zone          | string                       | Zone name. Default is "." | Yes |

**Body Parameters**
| Name          | Type                         | Description              | Required  ␣
↪    |
| ------------- | ---------------------------- | ------------------------ | -----------
↪-- |
| name          | string                       | Domain name | Yes |
| type          | enum **{A, AAAA}**           | DNS resource type, A in case of Ipv4␣
↪and AAAA for Ipv6 | Yes |
| class         | enum **{IN}**                | Resource record class | Yes on create␣
↪|
| ttl           | int (non-zero value)         | Record TTL value | Yes on create |
| rData         | list(string)                 | IP Address list, Ip type must match␣
↪with the **type** field | Yes on create |

**Example Request**
```

PUT /mep/dns_server_mgmt/v1/rrecord/www.example.com/A?zone=" com." [{ "name" : "www.example.com." , "type" : "A" , "class" : "IN" , "ttl" : 30, "rData" : ["172.168.15.201"] }]

```
**Return Parameters**

None

Return Code: 200 Success

**Example Response**
```

HTTP/1.1 200 Success

```
### 2. Delete an entry

DNS entry can be deleted from the dns-server using this interface.

**URL**
```

DELETE /mep/dns_server_mgmt/v1/rrecord/{fqdn}/{rrtype}

```
**Query parameters**A
| Name          | Type                         | Description              | Required  ␣
↪    |
| ------------- | ---------------------------- | ------------------------ | -----------
↪-- |
| zone          | string                       | Zone name. Default is "." | Yes |

**Request parameters**
| **Name** | **Type** | **Description** | **IN** | **Required** |
| --- | --- | --- | --- | --- |
| fqdn  | String | Fully Qualified Domain Name | path |  Yes   |
| rrtype      | String | Resource record type of the entry(A or AAAA). | path | Yes |

**Body Parameters**

None

**Example Request**
```

DELETE /mep/dns_server_mgmt/v1/rrecord/www.example.com./A?zone=com.

```
**Return Parameters**

None

Return Code: 200 Success

**Example Response**
```

HTTP/1.1 200 Success

```
## MEP Agent
Mep agent is a component which run as side car with application. It helps to get␣
↪authenticaion token, do service registration, send heartbeat on behalf of␣
↪application.

#### 1. Get token
Applicaiontion can call this API to get authentication token from mepauth.
**URL**
```

GET /mep-agent/v1/token

```
**Request parameters**
 | **Name** | **Type** | **Description** | **IN** | **Required** |
 | --- | --- | --- | --- | --- |
 | Content-Type |   String  | MIME type, fill in "application/json"                ␣
↪     |               header  | Yes|

**Body Parameters**

None

**Example Request**
```

GET /mep-agent/v1/token

{ “header” : [{ “key” : “Content-Type” , “value” : “application/json” }] }

```


**Return Parameters**

Return code: 200 OK
|Name     |       Type  |   Description   |               Required|
|---|---|---|---|
|access\_token |  String  | Token|                 Yes|
|token\_type     |String   |Token Type (Bearer) |   Yes|
|expires\_in    | int     | Expire time             | Yes|

**Example Response**:
```

HTTP/1.1 200 OK { “access_token” :” xxxx” , “token_type” :” Bearer” , “expires_in” :” 3600” }

```


#### 2. Get producer endpoint by service name
```

(下页继续)

(续上页)

```
Applicaiontion can call this API to get producer endpoint from mepserver.
**URL**
```

GET /mep-agent/v1/endpoint/:serName

```
**Request parameters**
 | **Name** | **Type** | **Description** | **IN** | **Required** |
 | --- | --- | --- | --- | --- |
 | Content-Type |  String  | MIME type, fill in "application/json"   | header  | Yes|
 | serName |  String  | service name   | path | Yes |

**Body Parameters**

None

**Example Request**
```

GET /mep-agent/v1/endpoint/serviceName

{ “header” : [{ “key” : “Content-Type” , “value” : “application/json” }] }

```
**Return Parameters**

Return code: 200 OK
|Name     |       Type  |   Description   |  Required|
|---|---|---|---|
| endpoint |  Object  | endpoint | Yes|
|&gt;uris  | Array\[String\]|   URI 格式的服务入口信息列表 |  Yes|
|&gt;addresses | Array\[Object\] |  服务入口信息地址列表（IP 地址 + 端口号）   | No|
|&gt;&gt;host   |         String      |      IP 地址      |  No|
|&gt;&gt;port   |         String     |       端口号             |  No|
|&gt;alternative|        String     |       已实现格式或者外部规范定义的服务入口信息
↪| No|

**Example Response**
```

HTTP/1.1 200 OK { “endpoint” : { “uris” : [“https://mep-api-gw.mep:8443/VideoService145da803354a611ebb15e”], “addresses” : null, “alternative” : null } }

```
```

10.4 依赖

10.4.1 MEP-auth 三方件使用清单

Artifact	Version
github.com/agiledragon/gomonkey	v2.0.1
github.com/astaxie/beego	v1.12.0
github.com/dgrijalva/jwt-go/v4	v4.0.0-preview1
github.com/go-playground/validator/v10	v10.2.0
github.com/natefinch/lumberjack	v2.0.0
github.com/lib/pq	v1.7.0
github.com/sirupsen/logrus	v1.6.0
github.com/smartystreets/goconvey	v1.6.4
golang.org/x/crypto	v0.0.0-20200302210943-78000ba7a073

10.4.2 MEP-server 三方件使用清单

Artifact	Version
github.com/agiledragon/gomonkey	v2.0.1
github.com/apache/servicecomb-service-center	v0.0.0-20191027084911-c2dc0caef706
github.com/astaxie/beego	v1.12.0
github.com/go-playground/validator/v10	v10.2.0
github.com/satori/go.uuid	v1.2.0
github.com/stretchr/testify	v1.6.1
golang.org/x/crypto	v0.0.0-20200302210943-78000ba7a073
golang.org/x/net	v0.0.0-20200301022130-244492dfa37a

10.4.3 dns-server 三方件使用清单

Artifact	Version
github.com/agiledragon/gomonkey	v2.0.1
github.com/labstack/echo/v4	v4.1.16
github.com/miekg/dns	v1.1.29
github.com/sirupsen/logrus	v1.6.0
github.com/stretchr/testify	v1.6.1
go.etcd.io/bbolt	v1.3.4

10.4.4 MEP-agent 三方件使用清单

Artifact	Version
github.com/agiledragon/gomonkey	v2.0.1
github.com/astaxie/beego	v1.12.0
github.com/go-playground/validator/v10	v10.2.0
github.com/natefinch/lumberjack	v2.0.0
github.com/sirupsen/logrus	v1.6.0
github.com/smartystreets/goconvey	v1.6.4
github.com/stretchr/testify	v1.6.1
gopkg.in/yaml.v2	v2.2.8

10.5 数据库设计

The MEP database uses an open source PostgreSQL database, only api gateway kong and mepauth use database, and the recommended version is 12.3.

- AuthInfoRecord: Store Ak/Sk information.

```
create table auth_info_record (
    app_ins_id  varchar(255) not null,
    ak  varchar(255) not null,
    sk  varchar(255) not null,
    nonce  varchar(255) not null,
    primary key (app_ins_id)
);
```

10.6 How to Contribute 如何贡献

10.6.1 如何贡献

详细信息，请参考代码仓库

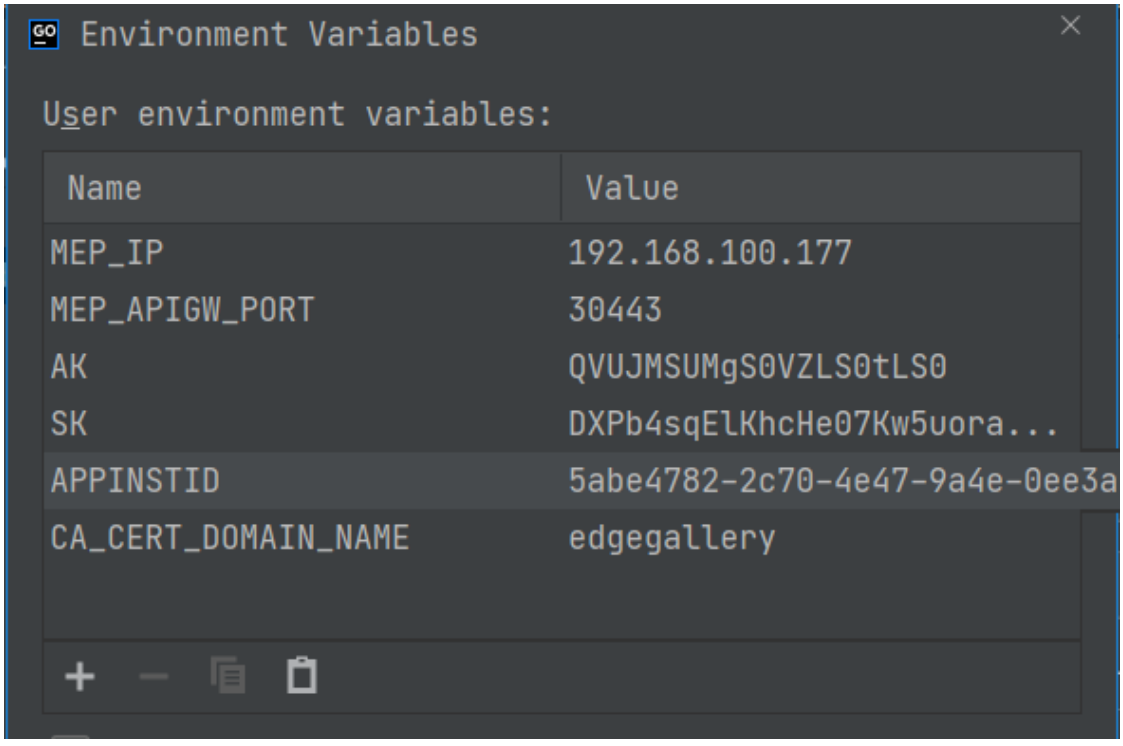
10.6.2 Contribution to mep-agent

This section provides details on how to participate/contribute to the mep-agent project.

Build local development environment

This section will guide developers to build mep-agent module locally and help developers to quickly understand and develop code

- Clone mep-agent code
- Open folder by goland
- Configure the runtime environment variables MEP_IP=192.168.100.177; MEP_APIGW_PORT=30443; AK=QVUJMSUMgS0VZLS0tLS0; SK=DXPb4sqElKhcHe07Kw5uorayETwId1JOjjOIRomRs5wyszoCR5R7AtVa28KT3lSc; APPINSTID=5abe4782-2c70-4e47-9a4e-0ee3a1a0fd1f; CA_CERT_DOMAIN_NAME=edgegallery



- Run main.go to start mep-agent

- After start, can call the api by postman

CHAPTER 11

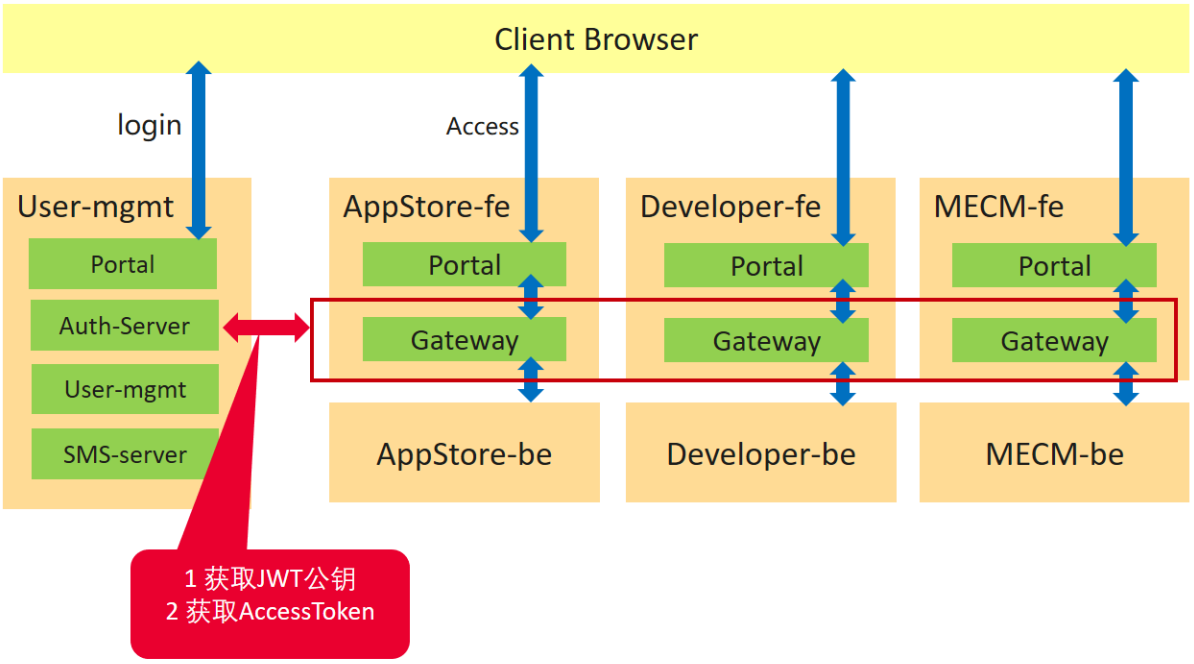
User Management

11.1 User Overview

11.1.1 简介

User Management 模块是为 EdgeGallery 项目提供用户管理的基本能力，包括用户注册/登录/权限认证等功能。为 AppStore/开发者平台/MECM/ATP/Lab 提供统一的用户鉴权和认证服务。

11.1.2 部署关系



User-mgmt 模块主要包含四个模块：

- Portal：提供登录/登出的界面操作
- Auth-server：提供 JWT 服务，用于 API 接口访问的 token 生成
- User-mgmt：提供用户帐号管理，包括获取/修改用户信息、密码找回等功能
- SMS：提供短信服务
- Mail: 提供邮箱服务

11.1.3 User-mgmt 和其他模块之间的关系

User-mgmt 是 EdgeGallery 的用户管理模块：

- 用户首先通过 User Management 登录到 EdgeGallery 平台
- User Management 会给登录成功的用户签发 AccessToken，用于后台 API 接口的访问，token 默认超时时间 1 小时，token 携带以下数据：
  - userId: 登录用户 ID
  - enableSms：是否开启短信服务
  - enableMail：是否开启邮箱服务
  - ssoSessionId：全局 SessionId，用户单点登录

  - userName：用户名

有关单点登录的详细实现过程，请参考章节《特性设计-单点登录》

11.2 特性设计

- 用户注册
- 单点登录
- 用户中心
- 密码找回
- 访客登录
- 用户列表管理
- 邮箱验证
- 手机验证

Feature Info

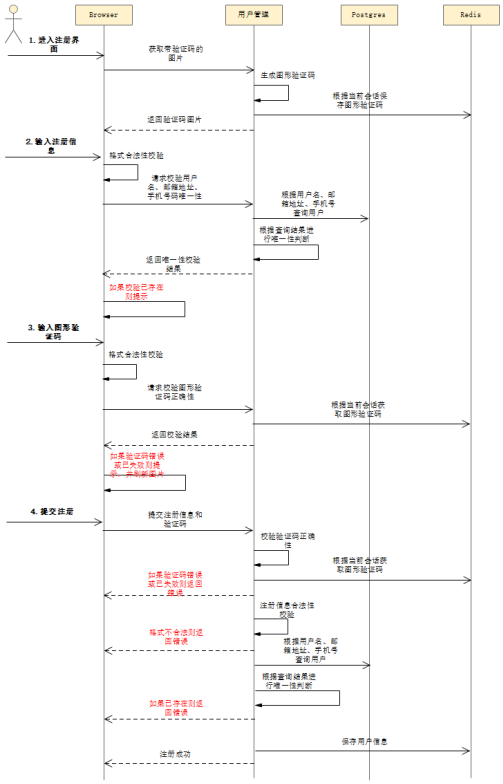
特性名称	特性描述	设计约束
用户注册	提供用户注册功能。注册时除提供必须的用户名外，还可以选择提供邮箱地址、手机号码。建议二者至少填写一项真实信息，便于忘记密码后找回密码时进行验证。	注册时提供的用户名、邮箱地址、手机号码必须是唯一的，如果已存在，则无法成功注册。目前只支持中国大陆手机号注册。为了方便用户注册，当前不对邮箱地址和手机号码进行验证。
单点登录	可以通过用户名、手机号码或邮箱地址登录各应用平台Portal。EdgeGallery 采取单点登录方式（Single Sign On），即多个应用平台共用一个登录认证平台，一次登录就可以访问互相信任的其他应用系统。	目前单点登录系统适用于开发者平台（Developer Portal）、应用仓库（Application Store）、管理系统（MEC Manager）、应用测试认证平台（ATP Portal）、社区实验室管理平台（Lab Portal）。
用户中心	用户登录到各应用平台后，可以通过“我的帐号”进入用户中心。在用户中心可以查看/修改个人信息、修改密码。	目前不支持修改用户名，只能修改手机号码和邮箱地址。
密码找回	如果用户注册时提供了真实有效的手机号码或邮箱地址（或者在用户中心修改个人信息时提供），当忘记密码时可以通过手机号码或邮箱地址重置密码以完成密码找回。通过手机号码找回密码时需要进行手机验证，通过邮箱地址找回密码时需要进行邮箱验证，验证通过后才可以重置密码。	只有当 EdgeGallery 平台开启了短信服务才可以通过手机号码找回密码；只有当 EdgeGallery 平台开启了邮箱服务才可以通过邮箱地址找回密码
访客登录	访客是系统内置的一个 GUEST 账号，访问首页时自动登录。在访客登录时，用户可以查看系统的大部分功能，但是不能在系统中进行创建/删除/修改登录操作。	访客的默认用户名和密码是 guest/guest，部署时如需修改密码请通过登录数据库修改，如果想停用 GUEST 账号，请从数据库中直接删除。
用户列表管理	用户管理模块默认内置 admin 用户，该用户作为系统的超级管理员。为该用户提供一个统一的用户列表管理功能，可以查询用户、启用/停用用户、设置用户角色权限。	当停用用户后，用户将无法登录；无法设置已停用用户的角色权限。
邮箱验证	系统提供向已注册邮箱地址发送验证码的服务，用来对邮箱进行验证。	只有当 EdgeGallery 平台开启了邮箱服务才可以向邮箱发送验证码
手机验证	系统提供向已注册手机号码发送验证码的服务，用来对手机进行验证。	只有当 EdgeGallery 平台开启了短信服务才可以向手机发送验证码

11.2.1 用户注册

用户注册时，除提供必须的用户名外，还可以选择提供手机号码、邮箱地址。建议手机号码和邮箱地址二者至少填写一项真实信息，便于忘记密码后找回密码时进行验证。

注册时系统会进行以下校验：

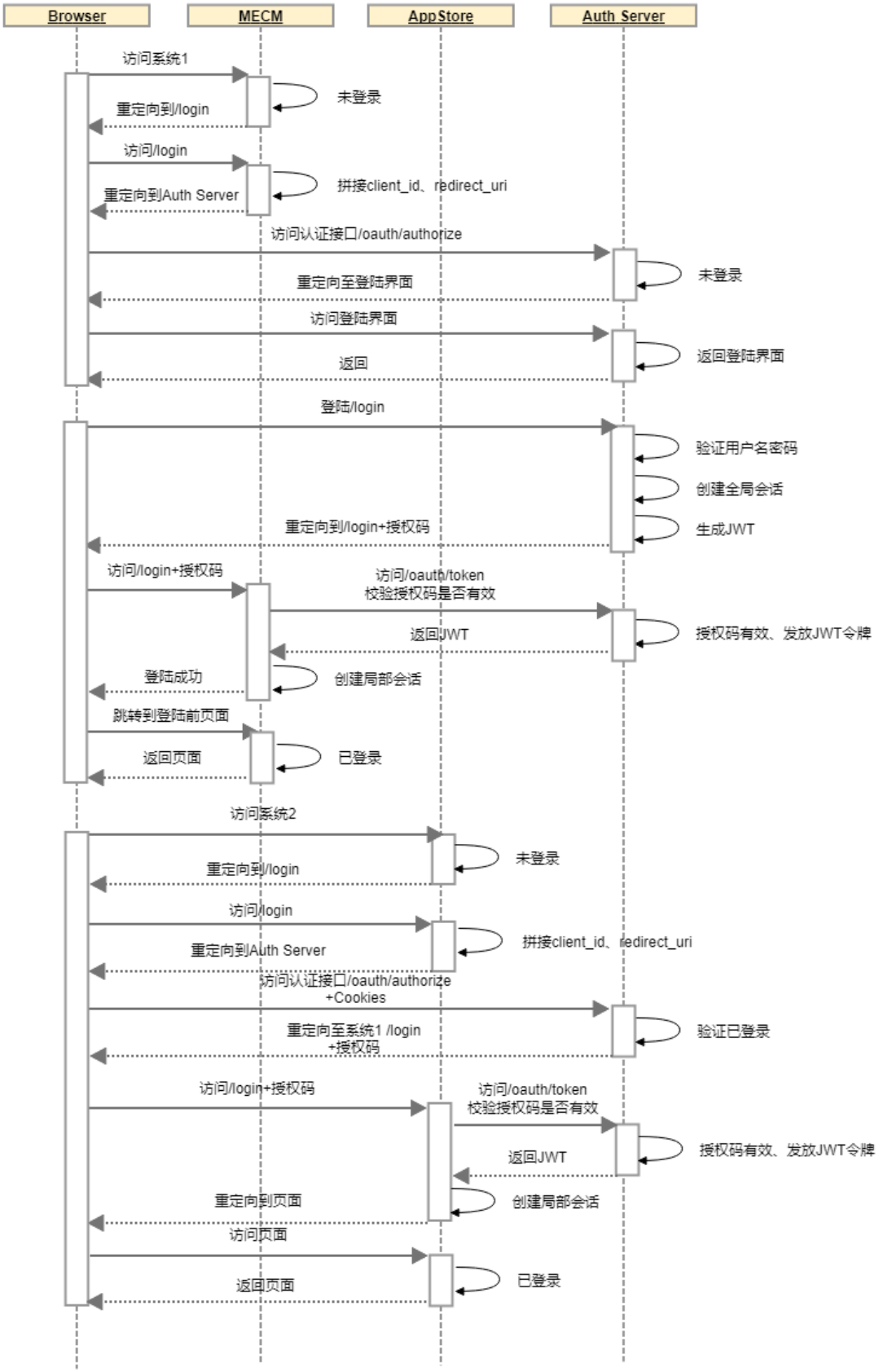
1. 用户名必须大于 6 位，同时会校验用户名是否已经被注册。
2. 如果提供了邮箱地址，校验其是否符合格式要求的邮箱地址，同时会校验邮箱地址是否已经被注册。
3. 如果提供了手机号码，校验其是否符合格式要求的中国大陆手机号码，同时会校验手机号码是否已经被注册。
4. 图形数字验证码

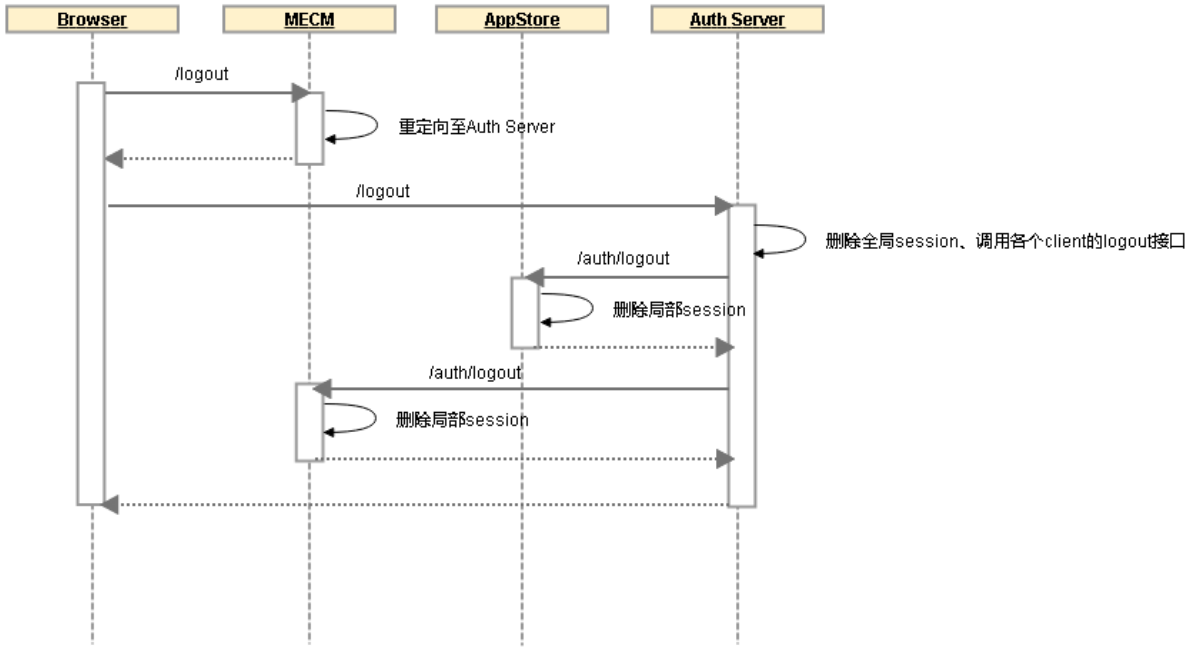


11.2.2 单点登录

单点登录系统是基于 ServiceComb、Spring Security、Oauth2、JWT、Vue 等技术实现，采用 OAuth2 协议的授权码模式（authorization code）。MECM、AppStore、Developer、Atp、Lab 应用平台作为客户端，与认证服务器 user-mgmt-be 进行交互，实现单点登陆与单点注销功能。使用 Cookie+Session 机制实现用户认证鉴权和超时退出功能，使用 JWT 存储用户基本信息，减轻认证服务器压力。

这里以 MECM 和 AppStore 两个应用平台作为客户端为例来说明单点登录和单点注销的处理时序。


Browser
MECM
AppStore
Auth Server
访问系统1
未登录
重定向到/login
访问/login
拼接client_id、redirect_uri
重定向到Auth Server
访问认证接口/oauth/authorize
未登录
重定向至登陆界面
访问登陆界面
返回登陆界面
返回
登陆/login
验证用户名密码
创建全局会话
生成JWT
重定向到/login+授权码
访问/login+授权码
访问/oauth/token
校验授权码是否有效
授权码有效、发放JWT令牌
返回JWT
登陆成功
创建局部会话
跳转到登陆前页面
返回页面
已登录
访问系统2
未登录
重定向到/login
访问/login
拼接client_id、redirect_uri
重定向到Auth Server
访问认证接口/oauth/authorize
+Cookies
验证已登录
重定向至系统1 /login
+授权码
访问/login+授权码
访问/oauth/token
校验授权码是否有效
授权码有效、发放JWT令牌
返回JWT
重定向到页面
创建局部会话
访问页面
已登录
返回页面



11.2.3 用户中心

用户登录到各应用平台后，可以通过“我的帐号”进入用户中心。

在用户中心，用户可以查看/修改个人信息、修改密码。

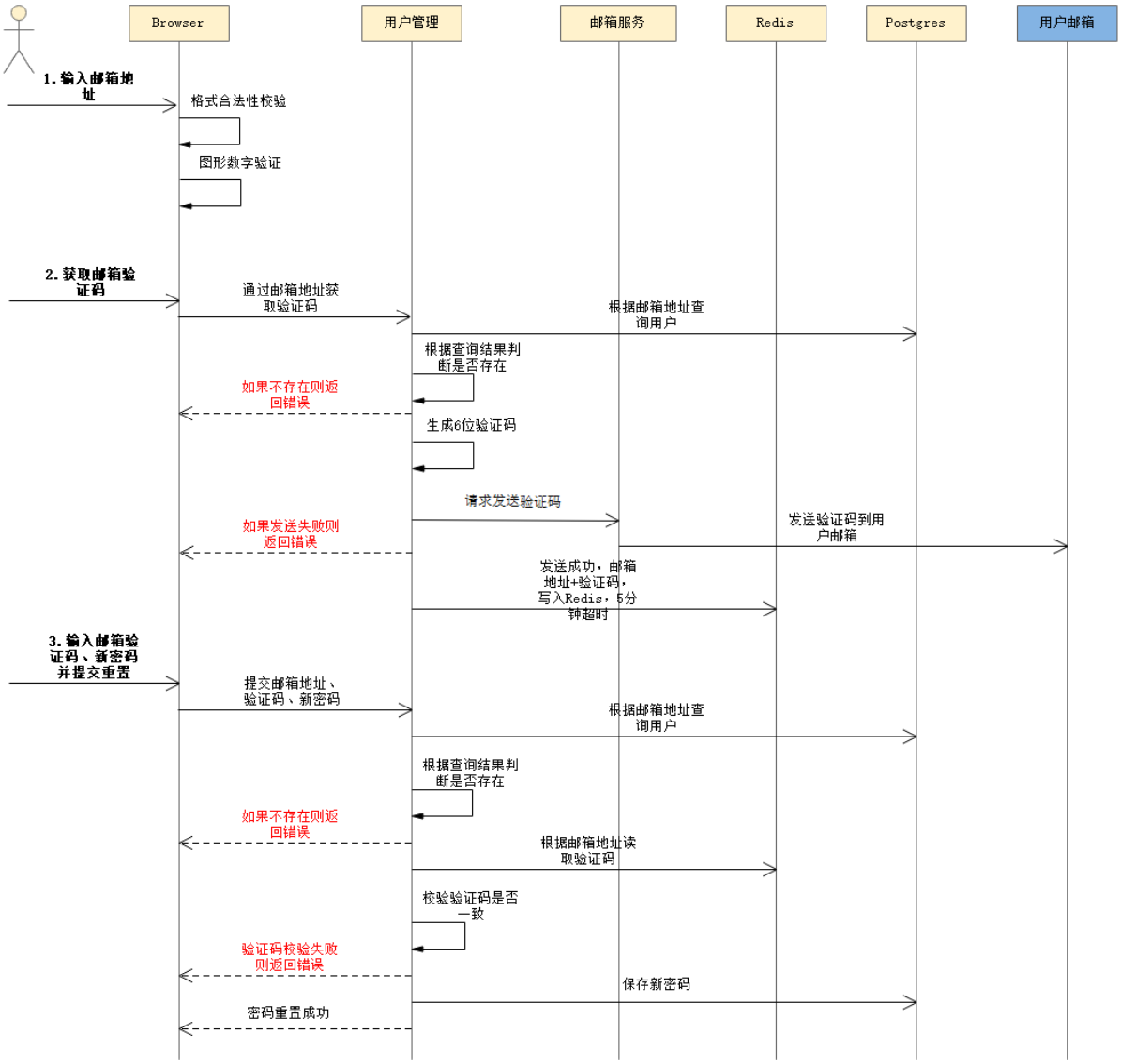
目前不支持修改用户名，只能修改手机号码和邮箱地址。

11.2.4 密码找回

通过邮箱地址找回密码

用户忘记密码后通过邮箱地址找回密码的主要交互处理：

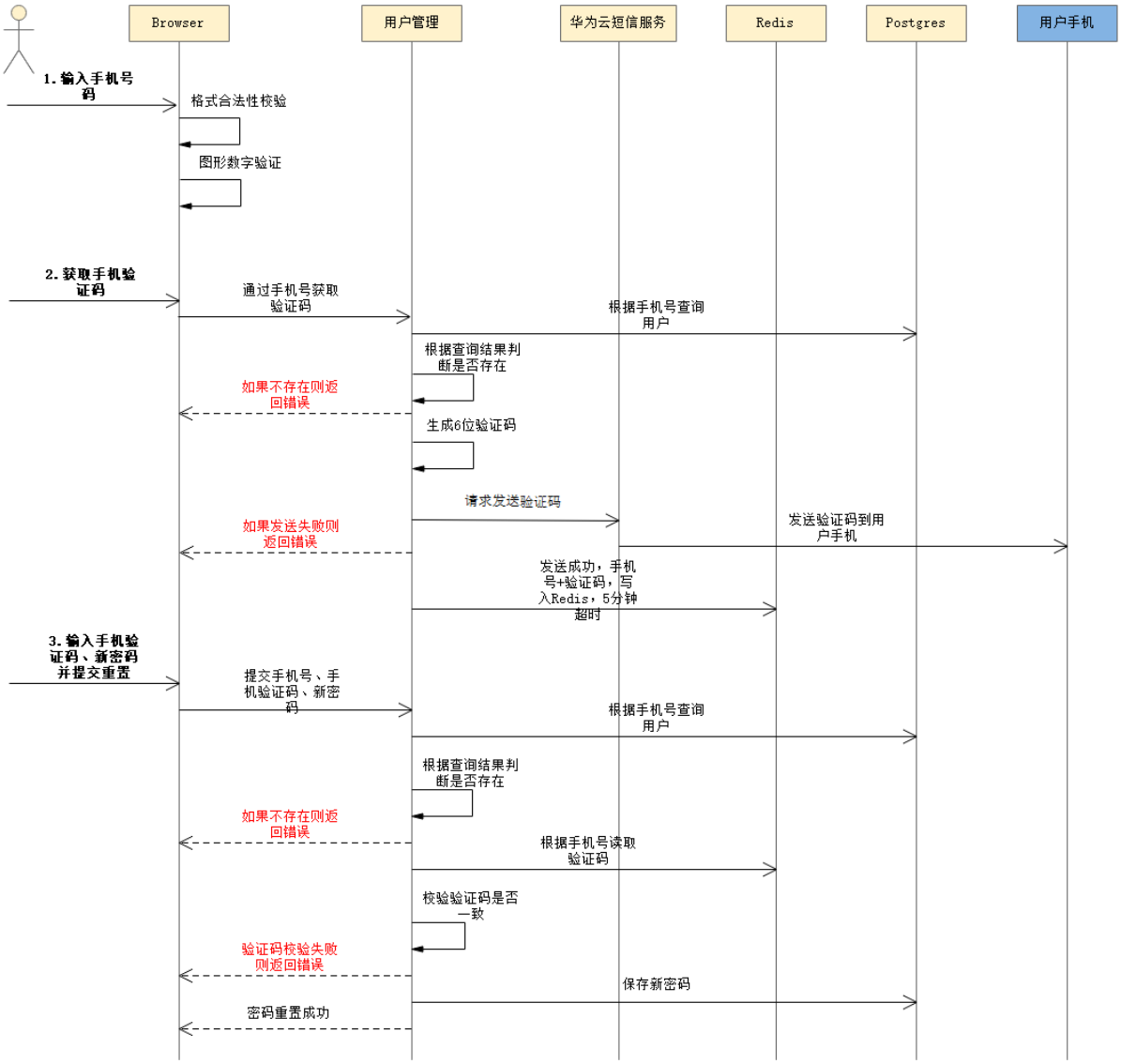
1. 用户输入邮箱地址，完成图形数字验证，然后请求获取验证码。系统判断邮箱地址是否已注册，如果已注册，则发送验证码到邮箱。如果没有注册，提示用户邮箱地址不存在。
2. 用户输入收到的验证码和新密码，修改成功后，返回登录界面。



通过手机号码找回密码

用户忘记密码后通过手机号码找回密码的主要交互处理：

1. 用户输入手机号码，完成图形数字验证，然后请求获取验证码。系统判断手机号码是否已注册，如果已注册，则通过手机短信发送验证码。如果没有注册，提示用户手机号码不存在。
2. 用户输入收到的验证码和新密码，修改成功后，返回登录界面。



11.2.5 访客登录

访客是系统内置的一个 GUEST 账号，访问首页时自动登录。

在访客登录时，用户可以查看系统的大部分功能，但是不能在系统中进行创建/删除/修改登录操作。

访客的默认用户名和密码是 guest/guest，部署时如需修改密码请通过登录数据库修改，如果想停用 GUEST 账号，请从数据库中直接删除。

11.2.6 用户列表管理

用户管理模块默认内置 admin 用户，该用户作为系统的超级管理员。

系统为该用户提供一个统一的用户列表管理功能，可以查询用户、启用/停用用户、设置用户角色权限。

```
(1) When the user is disabled, the user will not be able to log in.
(2) The role permissions of disabled users cannot be set.
```

11.2.7 邮箱验证

对邮箱地址进行验证时需要发送验证码到邮箱，需要开启邮箱服务。

开启邮箱服务需要修改 user-mgmt 下的配置文件 application.properties：

[mail.enabled=true] //邮箱服务开关，默认 false

[spring.mail.host=] //邮箱服务器

[spring.mail.port=] //邮箱服务器发送邮件的端口

[spring.mail.username=] //邮件发件人

[spring.mail.password=] //发件人授权码

11.2.8 手机验证

对手机号码进行验证时需要通过短信发送验证码到手机，需要开启短信服务。

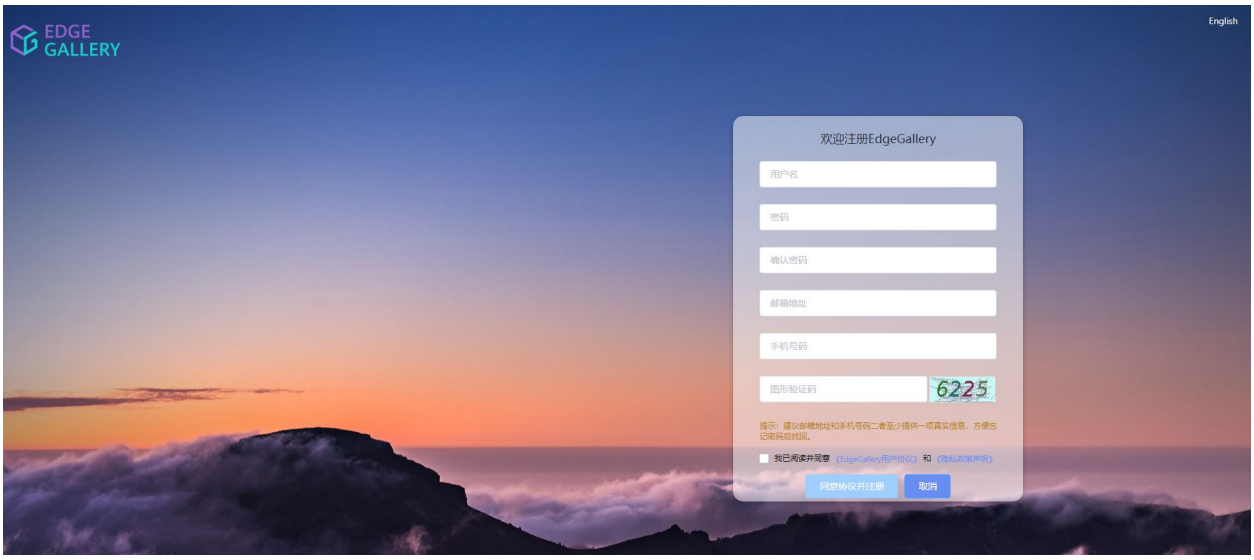
短信业务使用华为云消息 & 短信服务，详细信息请点击这里。

```
Description:
If you have not purchased the SMS service, the mobile phone verification function can␣
↪be blocked by configuration. The specific steps are as follows:
Configuration file application.properties under user-mgmt, set sms.enabled=false
```

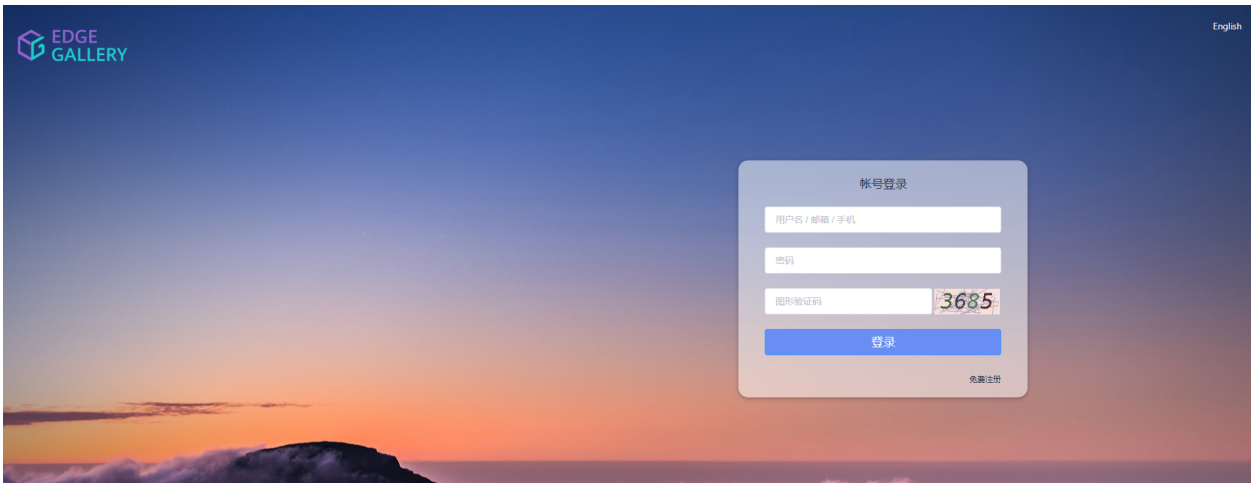
11.2.9 界面设计

- 用户注册界面
- 用户登录界面
- 忘记密码后找回密码界面
- 用户中心界面
- 用户列表管理界面

用户注册界面

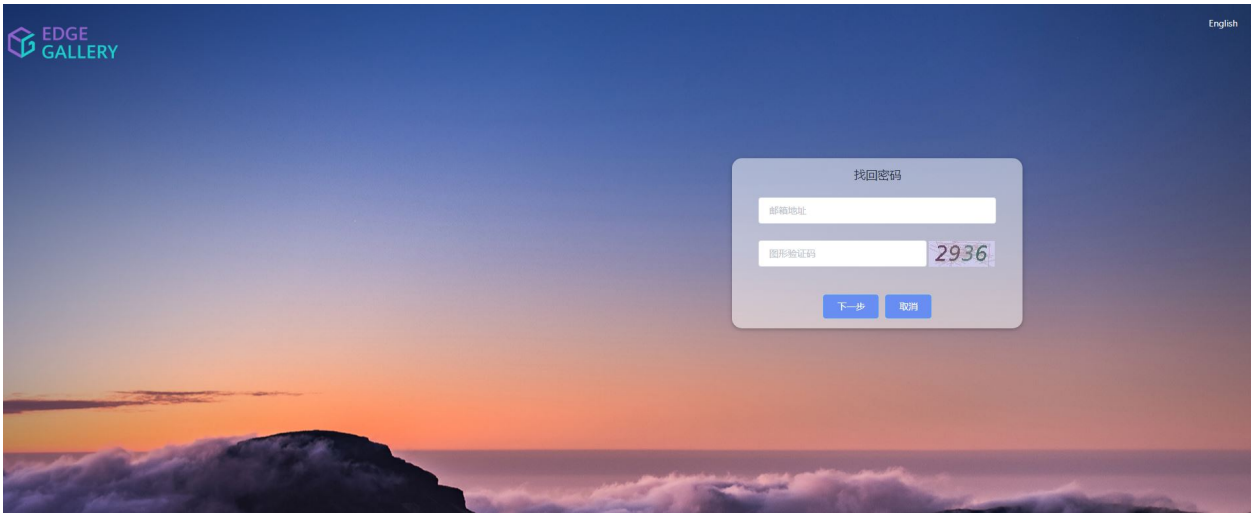


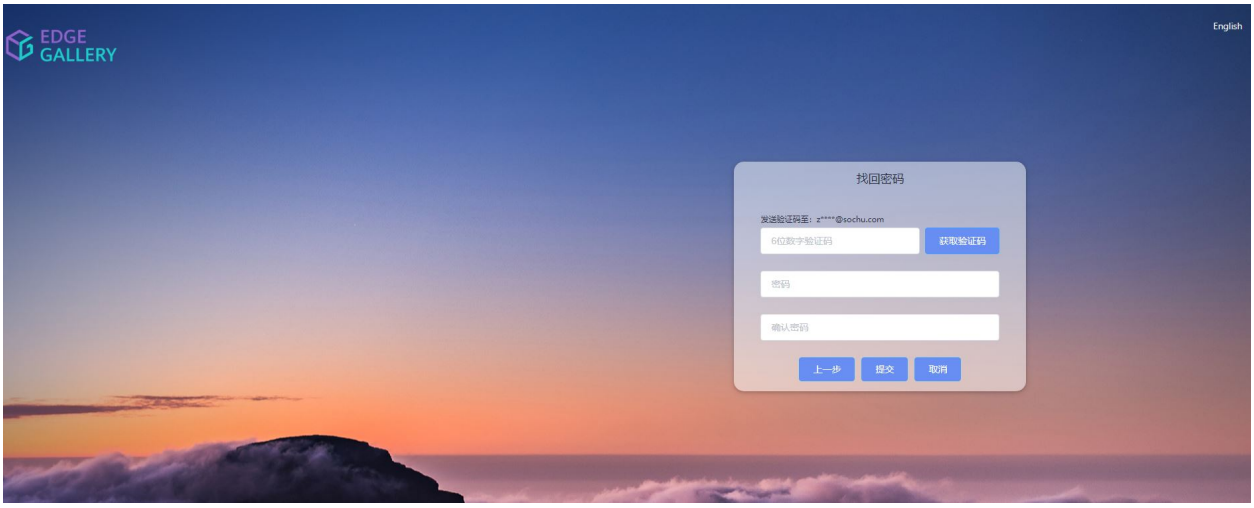
用户登录界面



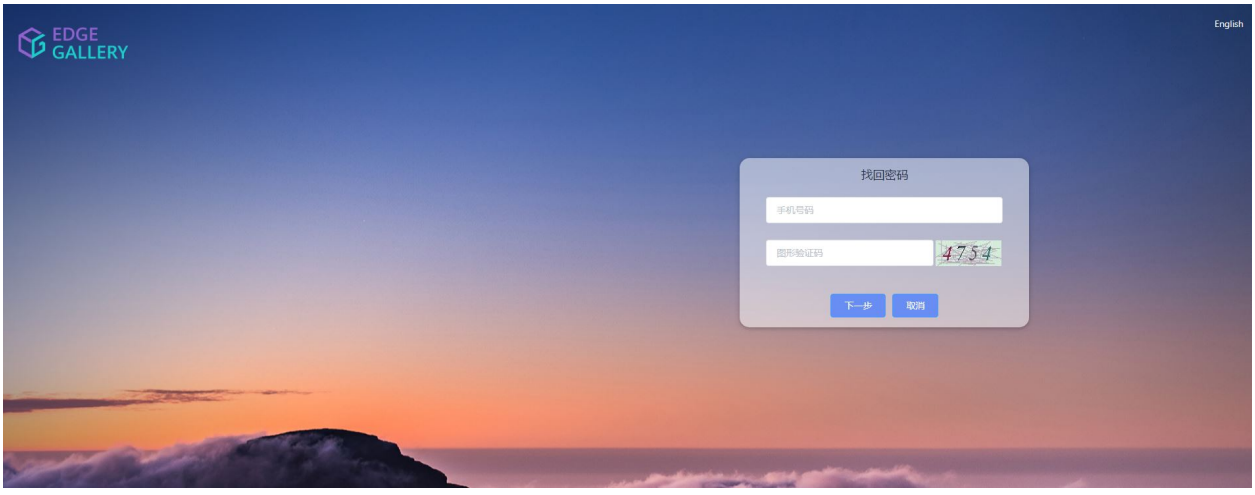
忘记密码后找回密码界面

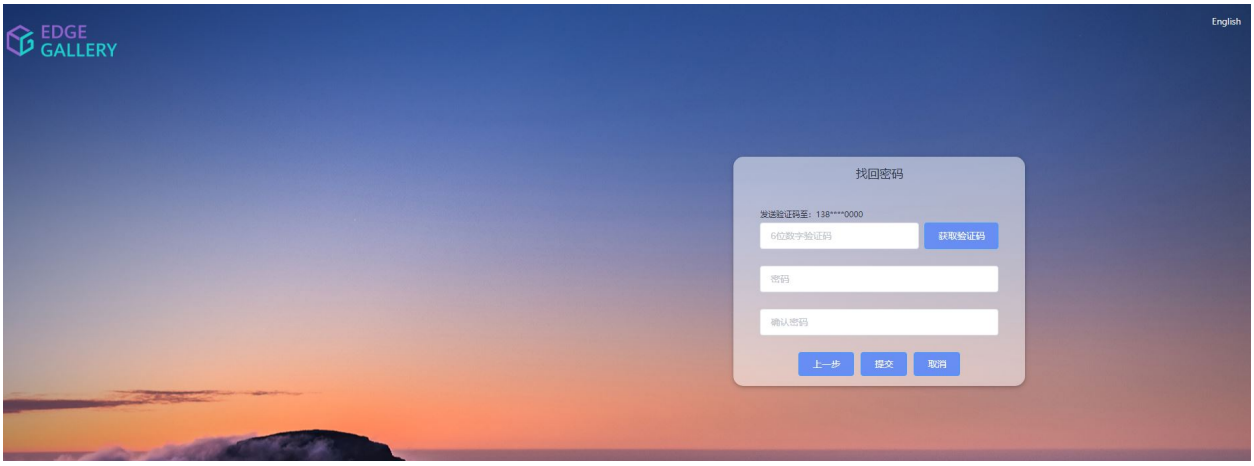
（1）当只开启了邮箱服务时，只能通过邮箱找回



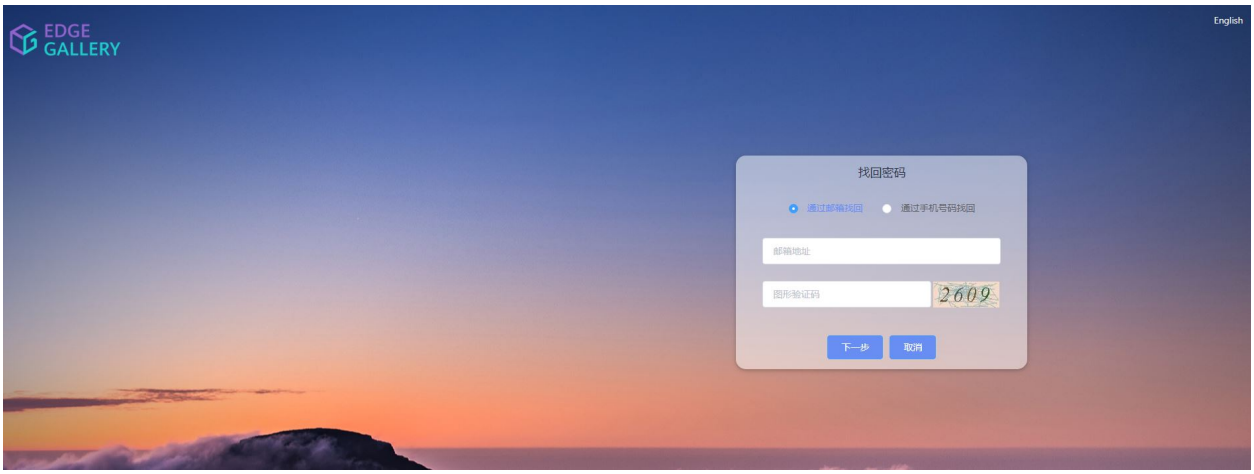


（2）当只开启了短信服务时，只能通过手机找回





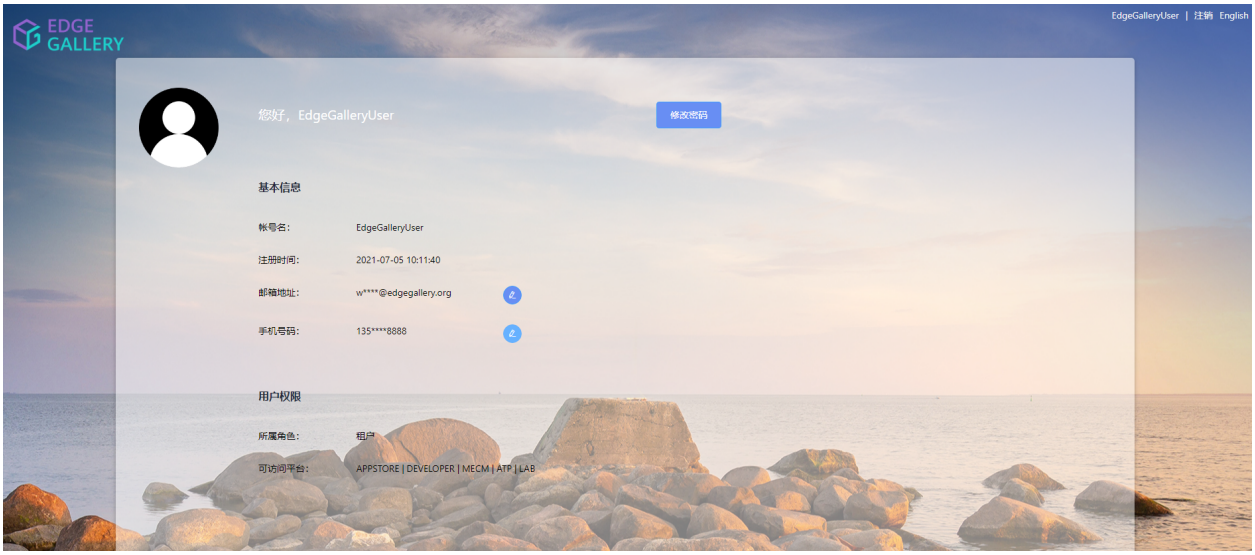
（3）当邮箱服务和短信服务同时开启时，可以二选一



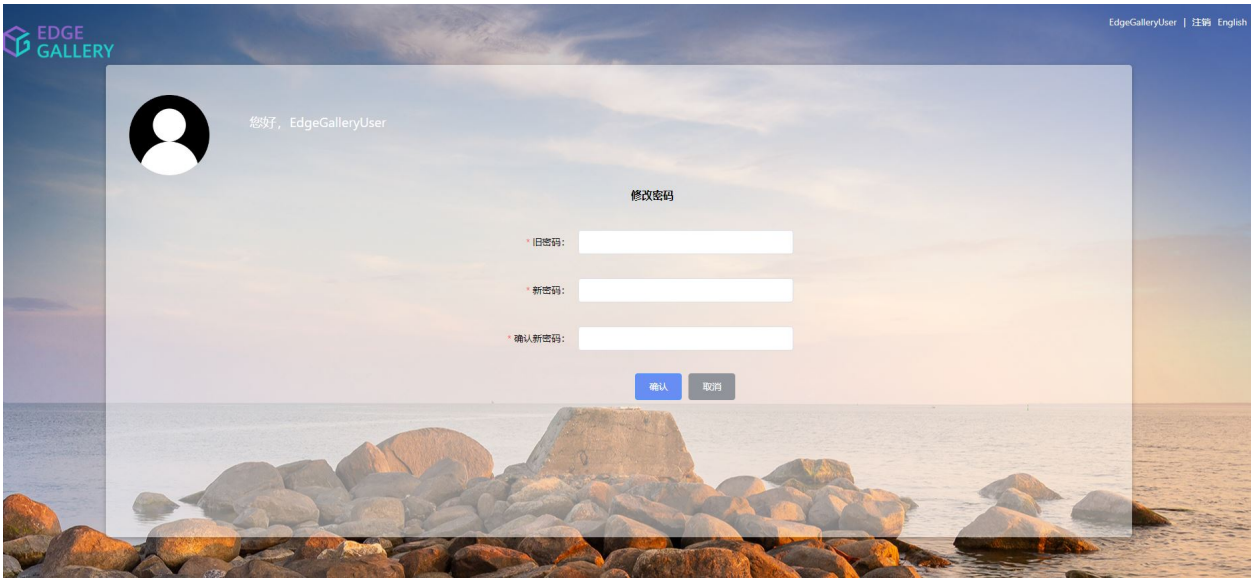
点击“下一步”后的界面，与前面的界面一样。

用户中心界面

（1）主界面

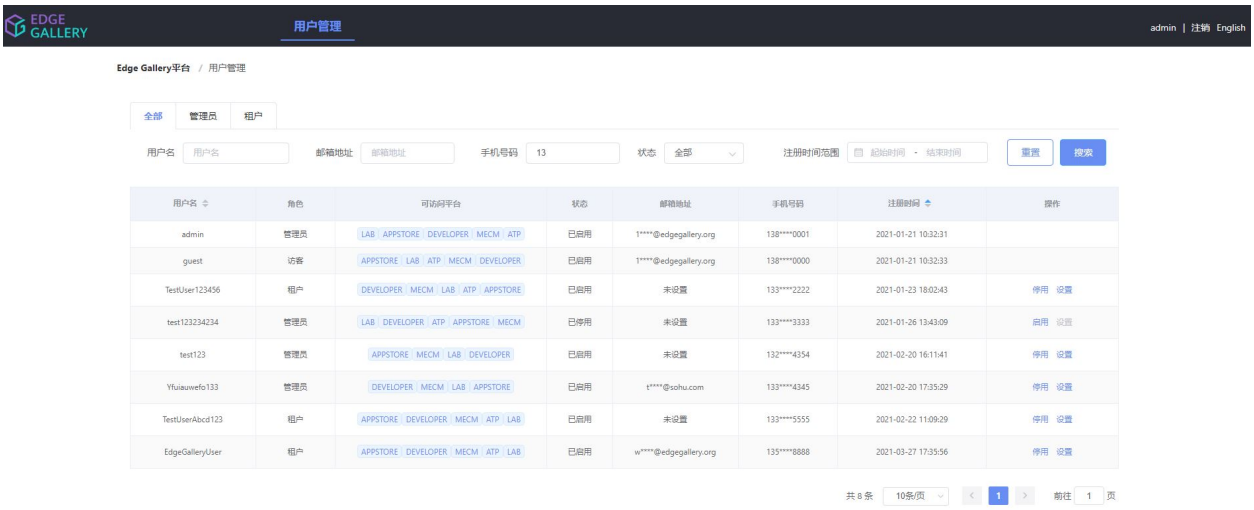


（2）修改密码

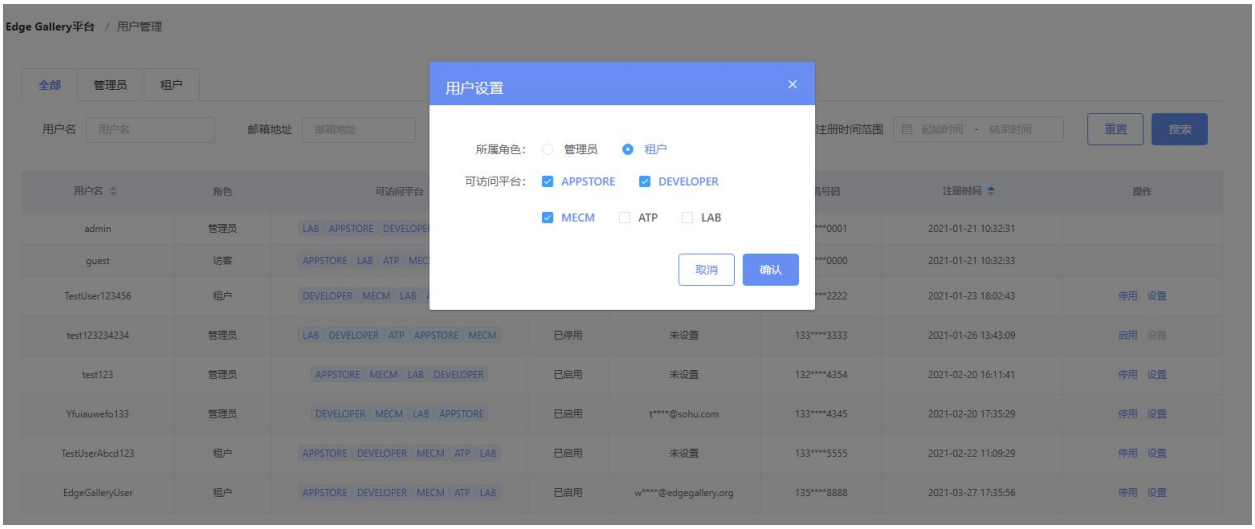


用户列表管理界面

（1）主界面



（2）设置用户角色



11.3 User Interfaces

- *User Interfaces*
- *User-mgmt-Server*
  - *1. AUTH*
    - *1.1* 登录
    - *1.2* 获取当前登录用户信息
    - *1.3* 登出
  - *2. USER-MGMT*

    * *2.1* 注册用户
    * *2.2* 修改密码（个人修改密码 & 找回密码）
    * *2.3* 校验重复用户
    * *2.4* 查询用户列表
    * *2.5* 删除用户
    * *2.6* 修改用户个人信息
    * *2.7* 停用用户
    * *2.8* 启用用户
    * *2.9* 设置用户角色
  - *3. IDENTITY*
    * *3.1* 获取短信验证码
    * *3.2* 获取邮件验证码
    * *3.3* 获取图形验证码
    * *3.4* 预校验图形验证码正确性

11.3.1 1. AUTH

1.1 登录

登录系统

登录前需要先获取图形验证码，通过 verifyCode 参数传递。

```
Resource URI: /login
Method: POST
```

Name	Definition	Type	Required	Describe
verifyCode	图形验证码	query	是	
username	用户名/邮箱/手机号	body	是	
password	密码	body	是	

Example response:

```
response 200 OK
```

1.2 获取当前登录用户信息

用户登录成功后，通过此接口获取当前登录用户的信息。

```
Resource URI: auth/login-info
Method: GET
```

Example response:

```
response 200 OK
{
  "username": "TestUser1",
  "mailAddress": "test@edgegallery.org",
  "telephone": "13812345678",
  "createTime":"2021-02-24 04:39:23",
  "allowed": true,
  "userId": "37423702-051a-46b4-bf2b-f190759cc0b8",
  "permissions": [
    {
      "platform": "APPSTORE",
      "role": "GUEST"
    }
  ]
}
```

1.3 登出

登出系统

```
Resource URI: /auth/logout
Method: GET
```

Example response:

```
200 OK
Succeed
```

11.3.2 2 USER-MGMT

2.1 注册用户

注册用户接口，注册成功返回 200，并返回已经注册成功的用户信息；接口数据不全返回 400。

注册前需要先获取图形验证码，通过 verifyCode 参数传递。

```
Resource URI: /v1/users
Method: POST
```

Name	Definition	Type	Required	Describe
verifyCode	图形验证码	query	是	
username	用户名	body	是	必须是字母或者字母和数字的组合，必须以字母开头，长度在 6~30 个字符之间
password	密码	body	是	必须满足复杂度要求，必须是数字/字母/特数字符的组合，长度在 6~18 个字符之间
mailAddress	邮箱地址	body	否	如果填写，必须符合邮箱地址的格式要求
telephone	手机号码	body	否	如果填写，必须为 11 位有效数字，且以 1 开头

Example request:

```
{
  "username": "TestUser1",
  "mailAddress": "test@edgegallery.org",
  "telephone": "13812345678",
  "password": "123.qwe"
}
```

Example response:

```
response 201 OK
{
  "username": "TestUser1",
  "mailAddress": "test@edgegallery.org",
  "telephone": "13812345678",
  "userId": "37423702-051a-46b4-bf2b-f190759cc0b8",
  "permissions": [
    {
```

(下页继续)

(续上页)

```
      "platform": "APPSTORE",
      "role": "GUEST"
    }
  ]
}

response 400 Bad Request
{
  "code": 0,
  "message": "string"
}

response 403 FORBIDDEN
{
  "code": 0,
  "message": "string"
}

response 500 INTERNAL ERROR
{
  "code": 0,
  "message": "string"
}
```

2.2 修改密码（个人修改密码 & 找回密码）

1. 个人修改密码，包括三种场景：

（1）用户登录后在个人中心通过原密码验证方式修改个人密码

（2）admin 首次登录后强制修改个人默认密码

（3）非 guest 用户密码超过 90 天后过期，登录后强制修改密码

1. 如果用户注册时提供了真实的邮箱地址或手机号（也可以在个人中心填写这两项信息），可以通过邮箱或手机验证码的方式重置密码。需要开启邮件服务或短信服务才能使用。

```
Resource URI: /v1/users/password
Method: PUT
```

Name	Definition	Type	Required	Describe
type	修改密码的类型	body	是	1–原密码验证修改; 2–密码找回
newPassword	修改后的新密码	body	是	必须满足复杂度要求，必须是数字/字母/特数字符的组合，长度在 6~18 个字符之间
oldPassword	原密码	body	否	当 type=1 时必填
telephone	手机号码	body	否	当 type=2 时，且需要通过手机号找回时，填写已经注册过的手机号码。与 mailAddress 有效性互斥
mailAddress	邮箱地址	body	否	当 type=2 时，且需要通过邮箱找回时，填写已经注册过的邮箱地址。与 telephone 有效性互斥
verificationCode	验证码	body	否	6 位有效数字。当 type=2 时必填

Example request:

```
{
  "type": 2,
  "newPassword": "123.qwe",
  "oldPassword": "",
  "telephone": "",
  "mailAddress": "test@edgegallery.org",
  "verificationCode": "123456"
}
```

Example response:

```
response 200 OK

response 400 Bad Request
{
  "code": 0,
  "message": "string"
}

response 403 FORBIDDEN
{
  "code": 0,
  "message": "string"
}

response 500 INTERNAL ERROR
```

(下页继续)

(续上页)

```
{
  "code": 0,
  "message": "string"
}
```

2.3 校验重复用户

判断用户名或手机号是否已经注册过，不允许重复注册。

```
Resource URI: /v1/users/action/uniqueness
Method: POST
```

Name	Definition	Type	Required	Describe
username	用户名	body	是	必须是字母或者字母和数字的组合，必须以字母开头，长度在 6~30 个字符之间
mailAddress	邮箱地址	body	否	如果填写，必须符合邮箱地址的格式要求
telephone	手机号码	body	否	如果填写，必须为 11 位有效数字，且以 1 开头

Example request:

```
{
  "username": "TestUser1",
  "mailAddress": "test@edgegallery.org",
  "telephone": "13812345678"
}
```

Example response:

```
response 200 OK
{
    "username": true or false,
    "mailAddress": true or false,
    "telephone": true or false
}
```

2.4 查询用户列表

用户管理员查询用户列表

```
Resource URI: /v1/users/list
Method: POST
Role: APPSTORE_ADMIN or DEVELOPER_ADMIN or MECM_ADMIN or LAB_ADMIN or ATP_ADMIN
User: admin
```

Name	Definition	Type	Required	Describe
username	用户名	body	否	用于模糊查询条件的用户名关键字
mailAddress	邮箱地址	body	否	用于模糊查询条件的邮箱地址关键字
telephone	电话号码	body	否	用于模糊查询条件的电话号码关键字
role	角色	body	是	ALL–全部;ADMIN–管理员;TENANT–租户;GUEST–访客
status	状态	body	是	-1–全部;0–停用;1–启用
createTimeBegin	创建时间 (即注册时间) 范围–起始	body	否	起始时间，格式：2021-1-21
createTimeEnd	创建时间 (即注册时间) 范围–终止	body	否	终止时间，格式：2021-2-1
offset	查询起始位置	body	是	分页控制参数，表示查询起始位置
limit	查询记录数	body	是	分页控制参数，表示每页查询记录数
sortBy	排序字段	body	否	用于排序的字段名。USERNAME–按用户名排序;CREATETIME–按创建 (注册) 时间排序
sortOrder	排序顺序	body	否	排序顺序。ASC–升序;DESC–降序

Example request:

```
{
  "username": "",
  "mailAddress": "",
  "telephone": "",
  "role": "",
  "status": -1,
  "createTimeBegin": "2021-1-21",
  "createTimeEnd": "2021-2-1",
  "queryCtrl": {
    "offset": 0,
    "limit": 10,
```

(下页继续)

(续上页)

```
    "sortBy": "USERNAME",
    "sortOrder": "DESC"
  }
}
```

Example response:

```
response 200 OK
{
  "totalCount": 20,
  "userList": [
    {
      "username": "TestUser1",
      "mailAddress": "test@edgegallery.org",
      "telephone": "13812345678",
      "userId": "37423702-051a-46b4-bf2b-f190759cc0b8",
      "permissions": [
        {
          "platform": "APPSTORE",
          "role": "GUEST"
        }
      ],
      "allowed": true,
      "createTime": "2021-1-21 10:24:45"
    }
  ]
}

response 400 Bad Request
{
  "code": 0,
  "message": "string"
}

response 403 FORBIDDEN
{
  "code": 0,
  "message": "string"
}

response 500 INTERNAL SERVER ERROR
{
  "code": 0,
  "message": "string"
```

(下页继续)

(续上页)

```
}
```

2.5 删除用户

用户管理员可以根据用户 ID 删除任何用户

```
Resource URI: /v1/users/{userId}
Method: DELETE
Role: APPSTORE_ADMIN or DEVELOPER_ADMIN or MECM_ADMIN or LAB_ADMIN or ATP_ADMIN
User: admin
```

Name	Definition	Type	Required	Describe
userId	用户 ID	path	是	uuid

Example request:

```

```

Example response:

```
response 200 OK

response 400 Bad Request
{
  "code": 0,
  "message": "string"
}

response 403 FORBIDDEN
{
  "code": 0,
  "message": "string"
}
```

2.6 修改用户个人信息

用户可以修改个人信息，返回修改后的信息。

```
Resource URI: /v1/users/{userId}
Method: PUT
```

Name	Definition	Type	Required	Describe
userId	用户 ID	path	是	uuid
username	用户名	body	是	必须是字母或者字母和数字的组合，必须以字母开头，长度在 6~30 个字符之间
mailAddress	邮箱地址	body	否	如果填写，必须符合邮箱地址的格式要求
telephone	手机号码	body	否	如果填写，必须为 11 位有效数字，且以 1 开头

Example request:

```
{
  "username": "TestUser1",
  "mailAddress": "test@edgegallery.org",
  "telephone": "13812345678"
}
```

Example response:

```
response 200 OK
{
  "username": "TestUser1",
  "mailAddress": "test@edgegallery.org",
  "telephone": "13812345678",
  "userId": "37423702-051a-46b4-bf2b-f190759cc0b8",
  "permissions": [
    {
      "platform": "APPSTORE",
      "role": "GUEST"
    }
  ]
}

response 400 Bad Request
{
```

(下页继续)

(续上页)

```
  "code": 0,
  "message": "string"
}

response 403 FORBIDDEN
{
  "code": 0,
  "message": "string"
}
```

2.7 停用用户

用户管理员可以根据用户 ID 停用用户，其中用户 ID 为非内置用户的用户 ID。用户停用后将无法登录。

```
Resource URI: /v1/users/status/{userId}/disallow
Method: PUT
Role: APPSTORE_ADMIN or DEVELOPER_ADMIN or MECM_ADMIN or LAB_ADMIN or ATP_ADMIN
User: admin
```

Name	Definition	type	Required	Describe
userId	用户 ID	path	是	uuid

Example request:

```

```

Example response:

```
response 200 OK

response 400 Bad Request
{
  "code": 0,
  "message": "string"
}

response 403 FORBIDDEN
{
  "code": 0,
  "message": "string"
}
```

2.8 启用用户

用户管理员可以根据用户 ID 启用用户，其中用户 ID 为非内置用户的用户 ID。

```
Resource URI: /v1/users/status/{userId}/disallow
Method: PUT
Role: APPSTORE_ADMIN or DEVELOPER_ADMIN or MECM_ADMIN or LAB_ADMIN or ATP_ADMIN
User: admin
```

Name	Definition	Type	Required	Describe
userId	用户 ID	path	是	uuid

Example request:

```

```

Example response:

```
response 200 OK

response 400 Bad Request
{
  "code": 0,
  "message": "string"
}

response 403 FORBIDDEN
{
  "code": 0,
  "message": "string"
}
```

2.9 设置用户角色

用户管理员可以根据用户 ID 设置其所属角色和可访问平台。

```
Resource URI: /v1/users/settings/{userId}
Method: PUT
Role: APPSTORE_ADMIN or DEVELOPER_ADMIN or MECM_ADMIN or LAB_ADMIN or ATP_ADMIN
User: admin
```

Name	Definition	Type	Required	Describe
userId	用户 ID	path	是	uuid
platform	可访问平台	body	是	用户可访问的平台
role	角色	body	是	用户所属角色

Example request:

```
{
  "permissions": [
    {
      "platform": "APPSTORE",
      "role": "GUEST"
    }
  ]
}
```

Example response:

```
response 200 OK

response 400 Bad Request
{
  "code": 0,
  "message": "string"
}

response 403 FORBIDDEN
{
  "code": 0,
  "message": "string"
}
```

11.3.3 3 IDENTITY

3.1 获取短信验证码

发送验证码到指定的手机号上，发送成功返回 200，发送失败返回 417。

发送短信验证码前需要先获取图形验证码，通过 verifyCode 参数传递。

```
Resource URI: /v1/identity/sms
Method: POST
```

Name	Definition	Type	Required	Describe
verifyCode	图形验证码	query	是	
telephone	电话号码	body	是	11 位有效数字，必须以 1 开头

Example request:

```
{
  "telephone": "15191881159"
}
```

Example response:

```
response 200 OK

response 417 Expectation Failed
{
  "code": 0,
  "message": "string"
}
```

3.2 获取邮件验证码

发送验证码到指定的邮箱，发送成功返回 200，发送失败返回 417。

发送邮件验证码前需要先获取图形验证码，通过 verifyCode 参数传递。

```
Resource URI: /v1/identity/mail
Method: POST
```

Name	Definition	Type	Required	Describe
verifyCode	图形验证码	query	是	
mailAddress	邮箱地址	body	是	用以接收验证码的邮箱地址

Example request:

```
{
  "mailAddress": "test@edgegallery.org"
}
```

Example response:

```
response 200 OK

response 400 Bad Request
{
  "code": 0,
  "message": "string"
}

response 417 Expectation Failed
{
  "code": 0,
  "message": "string"
}
```

3.3 获取图形验证码

获取带验证码的图片数据流

```
Resource URI: /v1/identity/verifycode-image
Method: GET
```

- 该接口不带任何请求参数，返回二进制的图片数据流
- 验证码 3 分钟内有效

3.4 预校验图形验证码正确性

预校验用户输入的验证码是否正确

```
Resource URI: /v1/identity/verifycode-image/precheck
Method: GET
```

Name	Definition	Type	Required	Describe
verifyCode	图形验证码	query	是	

Example response:

```
response 200 OK
{
  "checkResult": true
}
```

11.4 数据库设计

11.4.1 tbl_tenant 租户表

记录所有的租户信息

字段	类型	主键	描述
tenantid	string	key	主键
username	string	-	注册时的用户名，需要唯一性。[5-29] 个字符之间
password	string	-	注册时设置的密码，注册时需要满足密码复杂度要求，满足数字/字母/特殊字符至少两种以上的组合，[6-18] 个字符之间。数据库中的密码使用 PBKDF2 加盐值后存储
company	string	-	用户公司名称
telephonenumber	string	-	用户手机号码，用于找回密码时短信验证用户身份，必须满足 11 位有效数字
mailaddress	string	-	用户邮箱地址，用于找回密码时邮箱验证用户身份，必须为有效的邮箱格式
gender	string	-	用户性别
isallowed	boolean	-	帐号是否允许登录
createtime	timestamp	-	创建时间（即注册时间）
modifytime	timestamp	-	修改时间
pweffecttime	timestamp	-	密码生效时间

初始值：

```
              tenantid               |    username    |                                                  password                                          | company | telephonenumber |  mailaddress | gender | isallowed | createtime | modifytime | pweffecttime
--------------------------------------+----------------+----------------------------------------------------------------------------------------------------+---------+-----------------+--------------+--------+-----------+------------+------------+--------------
 de3565b1-a7c2-42b9-b281-3f032af29ff7 | guest          | 59756fda85ebddee6024d5cc0e6edcde3226693257a9c1eb662c56426b8a4f232b3d56c321adbd91 | company | 13800000000     | 13800000000@edgegallery.org | 1      | t         | 初始数据入库时间 (now 函数值) | 初始数据入库时间 (now 函数值) | NULL
```

(续上页)

```
 39937079-99fe-4cd8-881f-04ca8c4fe09d | admin          |␣
↪472645ad1af0101adaa6769cc865fec3b29fedeba6dc912900a59b1364b7a6bb17bb9a0575854547 |␣
↪company | 13800000001     | 13800000001@edgegallery.org | 1       | t | 初始数据入库时间
(now 函数值) | 初始数据入库时间 (now 函数值) | 初始数据入库时间 (now 函数值) 往前推 10 年的时间

The default password for the guest account is: guest
The default password for the admin account is: Admin@321
对于 admin 用户，首次登录后会强制修改默认密码。
对于非 guest 用户，密码超过 90 天后过期，登录时会强制修改密码。
```

11.4.2 tbl_role 用户角色表

配置支持的平台的和用户角色，该表在启动时初始化，不支持动态添加

字段	类型	主键	描述
id	string	key	主键
platform	string	-	支持的平台
role	string	-	支持的角色定义

初始值:

```
 id | platform  |  role
----+-----------+--------
  1 | APPSTORE  | GUEST
  2 | APPSTORE  | TENANT
  3 | APPSTORE  | ADMIN
  4 | DEVELOPER | GUEST
  5 | DEVELOPER | TENANT
  6 | DEVELOPER | ADMIN
  7 | MECM      | GUEST
  8 | MECM      | TENANT
  9 | MECM      | ADMIN
 10 | ATP       | GUEST
 11 | ATP       | TENANT
 12 | ATP       | ADMIN
 13 | LAB       | GUEST
 14 | LAB       | TENANT
 15 | LAB       | ADMIN
```

说明:

- APPSTORE：应用仓库平台

- DEVELOPER：应用开发集成平台
- MECM：MEC 管理平台
- ATP：应用认证测试平台
- LAB：实验室管理平台

11.4.3 tbl_tenant_role 关系表

用于描述租户和角色的关系

字段	类型	主键	描述
tenantid	string		租户 ID
roleid	string		角色 ID

初始值：

```
               tenantid                | roleid
---------------------------------------+--------
 de3565b1-a7c2-42b9-b281-3f032af29ff7 |      1
 de3565b1-a7c2-42b9-b281-3f032af29ff7 |      4
 de3565b1-a7c2-42b9-b281-3f032af29ff7 |      7
 de3565b1-a7c2-42b9-b281-3f032af29ff7 |     10
 de3565b1-a7c2-42b9-b281-3f032af29ff7 |     13
 39937079-99fe-4cd8-881f-04ca8c4fe09d |      3
 39937079-99fe-4cd8-881f-04ca8c4fe09d |      6
 39937079-99fe-4cd8-881f-04ca8c4fe09d |      9
 39937079-99fe-4cd8-881f-04ca8c4fe09d |     12
 39937079-99fe-4cd8-881f-04ca8c4fe09d |     15

Contrast with the initialization data of tbl_tenant,
de3565b1-a7c2-42b9-b281-3f032af29ff7It is a guest user and has the GUEST role of each␣
↪platform by default;
39937079-99fe-4cd8-881f-04ca8c4fe09dIt is an admin user and has the ADMIN role of␣
↪each platform by default;
```

11.5 如何参与贡献

详细信息，请参考代码仓库

CHAPTER 12

Kubernetes NodePort List

Component	Pod Name	Service Name	Node Port	Internal Port
appstore-be	appstore-be-0	appstore-be-svc	ClusterIP	8099
appstore-fe	appstore-fe	appstore-fe-svc	30091	8443
appstore-be-postgres	appstore-be-postgres-0	appstore-be-postgres-svc	ClusterIP	5432
developer-be	developer-be-0	N/A	N/A	N/A
developer-fe	developer-fe	developer-fe-svc	30092	8443
developer-be-postgres	developer-be-postgres-0	developer-be-postgres-svc	ClusterIP	5432
file-system	file-system	file-system-svc	30090	9500
filesystem-postgres	filesystem-postgres	filesystem-postgres	ClusterIP	5432
healthcheck	healthcheck	healthcheck-svc	32759	9527
healthcheck-m	healthcheck-m	healthcheck-m-svc	32757	9529
user-mgmt	user-mgmt	user-mgmt-svc	30067	8067
user-mgmt-postgres	user-mgmt-postgres-0	user-mgmt-postgres-svc	ClusterIP	5432
user-mgmt-redis	user-mgmt-redis-0	user-mgmt-redis-svc	ClusterIP	6379
service-center	service-center	service-center	ClusterIP	30100
mecm-fe	mecm-fe	mecm-fe-svc	30093	8443
mecm-appo	mecm-appo	mecm-appo-svc	30201	8091
mecm-apm	mecm-apm	mecm-apm-svc	30202	8092
mecm-inventory	mecm-inventory	mecm-inventory-svc	30203	8093
mecm-applcm-controller	mecm-applcm-controller	mecm-applcm-controller-svc	30204	8094
mecm-applcm-k8splugin	mecm-applcm-k8splugin	mecm-applcm-k8splugin-svc	30205	8095
mepserver	mepserver-deploy	mepserver-service	ClusterIP	8080
mepauth	mepauth-deploy	mepauth-service	ClusterIP	8088
kong-apigw	apigw-kongv	kong-service	30443	8443

CHAPTER 13

ServiceComb NodePort List

Component	Service Name	REST Port	Spring Server Port (default 8080)
appstore-be	mec-appstore	8099	8099
appstore-fe	appstore-fe	8080	8443
developer-be	mec-developer	9082	9082
developer-fe	developer-fe	8080	8443
user-mgmt	user-mgmt-be	8067	8067
user-mgmt-fe	user-mgmt-fe	8080	8443

```
All services registered to the service center
```