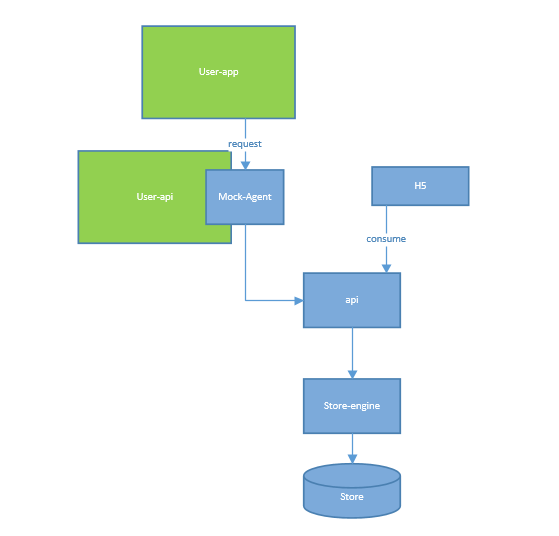
Mock开源平台一期详细设计

1.mock 平台作用

为了降低服务的提供方与消费方的联调成本，使得服务消费方(包括前端H5、Android以及ios)可以提前进行模拟联调测试，从而提升团队间的效率，特此开发一个可配置化的mock平台。

2.架构设计

总体架构图：



Mock品台主要涉及四个组件 mock-agent 、H5(web UI)、api、store-engine：

* mock-agent作为模拟服务，模拟用户应用的处理并返回指定数据给请求方。里mock-agent可以以aop形式接入。
* store-engine 存储引擎抽象层，为上层api提供存储相关的接口，底层实现可以是mysql，也可以是mongodb等。
* api一方面负责为Mock-agent查询返回模拟数据，另一方面为H5(web UI)提供mock数据的增删查改行为。
* H5(web UI) 提供简单的web界面，为用户提供mock数据的编辑及查询。

Ps:api 与 store-engine 是项目agent-service 的子模块

3.存储表格设计（mysql）

* 1. （user\_app）：

|  |  |  |
| --- | --- | --- |
| 字段： | 类型： | comment: |
|  |  |  |
| group | varchar | 项目组 |
| app\_name | varchar | 应用名称 |
|  |  |  |
| create\_time | datetime | 更新时间 |
| update\_time | datetime | 更新时间 |

* 1. （app\_mock）：

|  |  |  |
| --- | --- | --- |
| 字段： | 类型： | comment: |
| Id | bigint | 自增长id |
| group | varchar | 项目组 |
| app\_name | varchar | 应用名称 |
| req\_uri | varchar | 请求绝对地址 |
| result | varchar | 响应json结果 |
| valid\_flag | varchar | true:有效；false:无效 |
| version | bigint | 版本号 |
| create\_time | datetime | 更新时间 |
| update\_time | datetime | 更新时间 |
| method | varchar | 方法 |
| Protocol | bigint | 版本号 |

* 1. （user）：

|  |  |  |
| --- | --- | --- |
| 字段： | 类型： | comment: |
| user\_name | varchar | 登录名 |
| password | varchar | 登录密码 |
| create\_time | datetime | 更新时间 |
| update\_time | datetime | 更新时间 |

1. api接口设计

4.1、mock结果查询接口

|  |  |
| --- | --- |
| **请求URL** | /api/mock/result/query |
| **请求方式** | GET |

“group”: “SETTLEMENT”&“appName”: “account-service”&“reqUri”: “CB”

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **选项** | **说明** |
| group | String | 必填 | 项目组 |
| appName | String | 必填 | 应用名称 |
| reqUri | String | 必填 | 请求地址 |

返回结果

|  |  |
| --- | --- |
| **字段** | **说明** |
| result | Json字符串 |

4.2、mock项目管理查询接口

|  |  |
| --- | --- |
| **请求URL** | /mock/manager/project/query |
| **请求方式** | GET |

“group”:“SETTLEMENT”&“appName”:“account-service”&“reqUri”:“CB”&"pageSize":1&"currentPage":1&"startTime":"2015-01-01 01:01:01"&"endTime":"2015-01-01 01:01:01"

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **选项** | **说明** |
| group | String | 选填 | 项目组 |
| appName | String | 选填 | 应用名称 |
| reqUri | String | 选填 | 请求地址 |
| startTime | String | 选填 | 创建开始时间 |
| endTime | String | 选填 | 创建开始时间 |
| pageSize | int | 必填 | 请求页大小 |
| currentPage | int | 选填 | 当前页 |

返回结果

{

"resultMsg": "操作成功",

"result": [{

"developer": "123",

"group": "123",

"appName": "123"

}],

"resultCode": "1"

}

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 说明 | | |
| resultCode | 1：成功 else 失败 | | |
| resultMsg | 返回信息 | | |
| 结果（result） | Json数组 | | |
|  | id | long | 接口唯一标识 |
|  | group | String | 项目组 |
|  | appName | String | 应用名称 |
|  | developer | String | 开发者 |

4.3、mock项目添加接口

|  |  |
| --- | --- |
| **请求URL** | /mock/manager/ project /create |
| **请求方式** | POST |

{

"developer": "123",

"group": "123",

"appName": "123"

}

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **选项** | **说明** |
| group | String | 必填 | 项目组 |
| appName | String | 必填 | 应用名称 |
| reqUri | String | 必填 | 请求地址 |

返回结果

{

"resultMsg": "操作成功",

"result": "操作成功",

"resultCode": "1"

}

|  |  |
| --- | --- |
| 字段 | 说明 |
| resultCode | 1：成功 else 失败 |
| resultMsg | 返回信息 |
| 结果（result） | Json数组 |

4.5、mock项目修改接口

|  |  |
| --- | --- |
| **请求URL** | /mock/manager/ project /update |
| **请求方式** | POST |

{

"id": "123",

"developer": "123",

"group": "123",

"appName": "123"

}

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **选项** | **说明** |
| group | String | 选填 | 项目组 |
| appName | String | 选填 | 应用名称 |
| reqUri | String | 选填 | 请求地址 |
| id | String | 必填 |  |

返回结果

{

"resultMsg": "操作成功",

"result": "操作成功",

"resultCode": "1"

}

|  |  |
| --- | --- |
| 字段 | 说明 |
| resultCode | 1：成功 else 失败 |
| resultMsg | 返回信息 |
| 结果（result） |  |

4.6、mock项目删除接口

|  |  |
| --- | --- |
| **请求URL** | /mock/manager/ project /delete |
| **请求方式** | POST |

{

"id": "123"

}

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **选项** | **说明** |
| id | String | 必填 |  |

返回结果

{

"resultMsg": "操作成功",

"result": "操作成功",

"resultCode": "1"

}

|  |  |
| --- | --- |
| 字段 | 说明 |
| resultCode | 1：成功 else 失败 |
| resultMsg | 返回信息 |
| 结果（result） | Json数组 |

4.7、mock接口查询接口

|  |  |
| --- | --- |
| **请求URL** | /mock/manager/api/query |
| **请求方式** | GET |

“group”:“SETTLEMENT”&“appName”:“account-service”&"pageSize":1&"currentPage":1&"startTime":"2015-01-01 01:01:01"&"endTime":"2015-01-01 01:01:01"

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **选项** | **说明** |
| group | String | 选填 | 项目组 |
| appName | String | 选填 | 应用名称 |
| startTime | String | 选填 | 创建开始时间 |
| endTime | String | 选填 | 创建开始时间 |
| pageSize | int | 必填 | 请求页大小 |
| currentPage | int | 选填 | 当前页 |

返回结果

{

"resultMsg": "操作成功",

"result": [{

"id": "123",

"reqUri": "123",

"group": "123",

"appName": "123",

"result": "123",

"validFlag": "true"

}],

"resultCode": "1"

}

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 说明 | | |
| resultCode | 1：成功 else 失败 | | |
| resultMsg | 返回信息 | | |
| 结果（result） | Json数组 | | |
|  | id | long | 接口唯一标识 |
|  | group | String | 项目组 |
|  | appName | String | 应用名称 |
|  | reqUri | String | 请求地址 |
|  | result | String | Mock返回结果 |
|  | validFlag | String | 接口开关 |

4.8、mock接口添加

|  |  |
| --- | --- |
| **请求URL** | /mock/manager/ api /create |
| **请求方式** | POST |

{

"group": "123",

"appName": "123",

"reqUri": "123",

"result": "123"

}

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **选项** | **说明** |
| group | String | 必填 | 项目组 |
| appName | String | 必填 | 应用名称 |
| reqUri | String | 必填 | 请求地址 |
| result | String | 必填 | Mock返回结果 |

返回结果

{

"resultMsg": "操作成功",

"result": "操作成功",

"resultCode": "1"

}

|  |  |
| --- | --- |
| 字段 | 说明 |
| resultCode | 1：成功 else 失败 |
| resultMsg | 返回信息 |
| 结果（result） | Json数组 |

4.9、mock项目修改接口

|  |  |
| --- | --- |
| **请求URL** | /mock/manager/ api /update |
| **请求方式** | POST |

{

"id": "123",

"reqUri": "123",

"group": "123",

"appName": "123",

"result": "123"

}

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **选项** | **说明** |
| group | String | 选填 | 项目组 |
| appName | String | 选填 | 应用名称 |
| reqUri | String | 选填 | 请求地址 |
| id | String | 必填 |  |
| result | String | 选填 | Mock结果 |
| validFlag | String | 选填 | 生效开关 |

返回结果

{

"resultMsg": "操作成功",

"result": "操作成功",

"resultCode": "1"

}

|  |  |
| --- | --- |
| 字段 | 说明 |
| resultCode | 1：成功 else 失败 |
| resultMsg | 返回信息 |
| 结果（result） | Json数组 |

4.10、mock项目删除接口

|  |  |
| --- | --- |
| **请求URL** | /mock/manager/ api /delete |
| **请求方式** | POST |

{

"id": "123"

}

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **选项** | **说明** |
| id | String | 必填 |  |

返回结果

{

"resultMsg": "操作成功",

"result": "操作成功",

"resultCode": "1"

}

|  |  |
| --- | --- |
| 字段 | 说明 |
| resultCode | 1：成功 else 失败 |
| resultMsg | 返回信息 |
| 结果（result） | Json数组 |

5.使用大致流程

5.1 初始化mysql存储实例，启动api项目（spring boot）。

5.2前端开发者首先通过H5配置所需要的mock接口及返回数据。

5.3 后端应用可以以filter(这里针对spring mvc形式)形式接入mock-agent。

5.3 前端应用通过http形式访问app应用，mock-agent 通过api获取mock数据，并且返回前端。

6.后续规划

6.1、持续假如rpc框架的mock-agent，例如，业界的dubbo、gRpc、thrrift等。

6.2、加入用户鉴权管理模块。