目录

[模块化框架详细设计文档 3](#_Toc36371099)

[框架架构 3](#_Toc36371100)

[Springcloud Config 3](#_Toc36371101)

[Springcloud Gateway 3](#_Toc36371102)

[Springcloud Eureka 4](#_Toc36371103)

[Commonservice 4](#_Toc36371104)

[Org.coud.Core包 4](#_Toc36371105)

[数据库设计 4](#_Toc36371106)

[Mysql： 4](#_Toc36371107)

[Mongodb 4](#_Toc36371108)

[日志存储 4](#_Toc36371109)

[数据接口定义 6](#_Toc36371110)

[系统设计目标 14](#_Toc36371111)

[组件服务 15](#_Toc36371112)

[用户认证 15](#_Toc36371113)

[用户管理 16](#_Toc36371114)

[角色管理 18](#_Toc36371115)

[资源权限管理 23](#_Toc36371116)

[系统接口管理 24](#_Toc36371117)

[数据字典管理 25](#_Toc36371118)

[系统服务接口（按POSTMAN格式） 27](#_Toc36371119)

[环境配置 27](#_Toc36371120)

[测试环境配置 27](#_Toc36371121)

[生产环境 28](#_Toc36371122)

[数据接口 29](#_Toc36371123)

["name": "网关登录", 30](#_Toc36371124)

["name": "用户退出", 33](#_Toc36371125)

["name": "系统资源注册", 44](#_Toc36371126)

["name": "动态SQL查询分页", 47](#_Toc36371127)

["name": "获取环境变量", 49](#_Toc36371128)

["name": "系统资源注册", 51](#_Toc36371129)

["name": "查询接口列表", 53](#_Toc36371130)

["name": "查询接口参数列表", 54](#_Toc36371131)

["name": "查询所有定时任务", 56](#_Toc36371132)

["name": "查询所有运行的定时任务", 58](#_Toc36371133)

["name": "创建或更新定时任务", 60](#_Toc36371134)

["name": "暂停定时任务", 62](#_Toc36371135)

["name": "启动定时任务", 64](#_Toc36371136)

["name": "暂停一个定时任务的job" 66](#_Toc36371137)

["name": "恢复一个定时任务的job", 68](#_Toc36371138)

["name": "立即执行定时任务", 70](#_Toc36371139)

["name": "删除定时任务", 72](#_Toc36371140)

["name": "查询数据字典列表", 74](#_Toc36371141)

["name": "更新或者创建数据字典", 75](#_Toc36371142)

["name": "获取服务列表", 77](#_Toc36371143)

["name": "查看commonservice的EVN", 79](#_Toc36371144)

["name": "根据dic\_code查询items", 80](#_Toc36371145)

["name": "根据diccode查询items", 82](#_Toc36371146)

["name": "查询mongodb的文件", 84](#_Toc36371147)

["name": "mongodb数据接口查询", 86](#_Toc36371148)

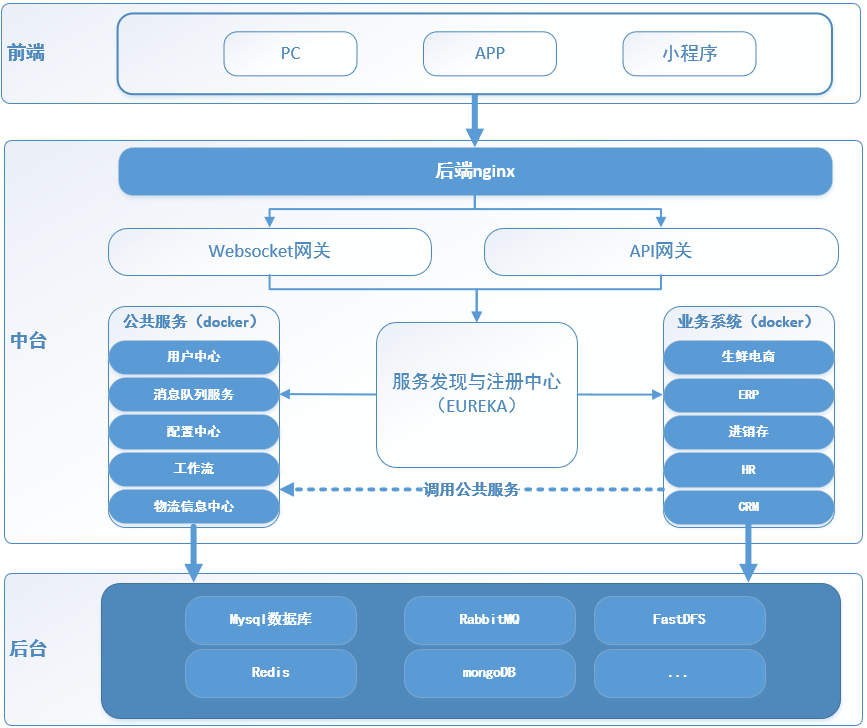
["name": "mongodb数据接口查询", 88](#_Toc36371149)

["name": "动态SQL查询分页", 91](#_Toc36371150)

["name": "动态SQL查询不分页", 93](#_Toc36371151)

# 模块化框架详细设计文档

## 框架架构



### Springcloud Config

配置中心服务，采用springcloud的通用配置中心，配置存储在git空间中，具体的相关技术和文档请参考springcloud的文档。

### Springcloud Gateway

网关服务，采用springcloud gateway 的rect网关，增加spring 的案例认证框架，采用base64认证。是所有服务访问的唯一入口。具体的相关技术和文档请参考springcloud的文档。

### Springcloud Eureka

注册中心服务，服务发现和注册中心，所有服务均会部署后均会注册到服务发现和注册中心，实现服务的扩展和负载均衡。，具体的相关技术和文档请参考springcloud的文档。

### Commonservice

通用服务，用于管理用户信息，数据字典和用户权限相关的服务。

### Org.coud.Core包

基础core包，实现util类和一些通用的model类的封装。

## 数据库设计

### Mysql：

#### ERD图

请参考ERD图框架设计.pdm（powerdesigner打开），里面有详细的设计，和字段说明。



#### 表结构表格

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 用户相关信息 | t\_frame\_user\_ref |  | | | | |
| 字段中文名 | 字段英文名 | 字段类型 | 注释 | 是否主键 | 是否非空 | 默认值 |
| id | id | bigint |  |  | Y |  |
| login\_id | login\_id | varchar(200) |  | Y | Y |  |
| attribute\_name | attribute\_name | varchar(50) |  |  |  |  |
| attribute\_value | attribute\_value | varchar(200) |  |  |  |  |
| 创建人 | create\_by | varchar(50) |  |  |  |  |
| 创建日期 | create\_date | datetime |  |  |  |  |
| 更新人 | update\_by | varchar(50) |  |  |  |  |
| 更新日期 | update\_date | datetime |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| 用户角色表 | t\_frame\_user\_role |  | | | | |
| 字段中文名 | 字段英文名 | 字段类型 | 注释 | 是否主键 | 是否非空 | 默认值 |
| user\_role\_id | user\_role\_id | bigint |  | Y | Y |  |
| 用户ID | user\_id | bigint | 用户ID |  | Y |  |
| 角色id | role\_id | int | 角色id |  |  |  |
| 有效开始时间 | validator\_start | datetime | 有效开始时间 |  |  |  |
| 有效结束时间 | validator\_end | datetime | 有效结束时间 |  |  |  |
| 创建人 | create\_by | varchar(50) | 创建人 |  |  |  |
| 创建日期 | create\_date | datetime | 创建日期 |  |  |  |
| 更新人 | update\_by | varchar(50) | 更新人 |  |  |  |
| 更新日期 | update\_date | datetime | 更新日期 |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| 组织机构 | t\_frame\_org |  | | | | |
| 字段中文名 | 字段英文名 | 字段类型 | 注释 | 是否主键 | 是否非空 | 默认值 |
| 组织机构编码 | org\_code | VARCHAR(200) |  | Y | Y |  |
| 名称 | org\_name | varchar(200) |  |  |  |  |
| 类型 | org\_type | varchar(20) |  |  |  |  |
| 父结点CODE | parent\_org\_code | varchar(50) |  |  |  |  |
| 状态 | status | varchar(20) |  |  |  |  |
| 有效开始时间 | validator\_start | datetime |  |  |  |  |
| 有效结束时间 | validator\_end | datetime |  |  |  |  |
| 创建人 | create\_by | varchar(50) |  |  |  |  |
| 创建日期 | create\_date | datetime |  |  |  |  |
| 更新人 | update\_by | varchar(50) |  |  |  |  |
| 更新日期 | update\_date | datetime |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| 系统资源 | t\_framework\_resource | 系统资源，用来记录所有的系统的URI服务资源    可以定义参数类型并进行参数类型的检查 | | | | |
| 字段中文名 | 字段英文名 | 字段类型 | 注释 | 是否主键 | 是否非空 | 默认值 |
| resource\_id | resource\_id | bigint |  | Y | Y |  |
| 资源编码 | resource\_code | varchar(200) | 资源编码 |  | Y |  |
| 资源名称 | resource\_name | varchar(1000) | 资源名称 |  | Y |  |
| 资源路径 | resource\_path | varchar(200) | 资源路径 |  | Y |  |
| 访问方式 | method | varchar(200) | 访问方法：get,post |  |  |  |
| 所属微服务 | belong\_microservice | varchar(200) | 所属微服务 |  |  |  |
| 创建人 | create\_by | varchar(50) | 创建人 |  |  |  |
| 创建日期 | create\_date | datetime | 创建日期 |  |  |  |
| 更新人 | update\_by | varchar(50) | 更新人 |  |  |  |
| 更新日期 | update\_date | datetime | 更新日期 |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| 系统菜单 | t\_frame\_menu |  | | | | |
| 字段中文名 | 字段英文名 | 字段类型 | 注释 | 是否主键 | 是否非空 | 默认值 |
| menu\_id | menu\_id | bigint |  | Y | Y |  |
| 菜单编码 | menu\_code | varchar(200) |  |  | Y |  |
| 菜单名称 | menu\_name | varchar(200) |  |  |  |  |
| 菜单URL | menu\_url | varchar(2048) |  |  |  |  |
| 序号 | seq\_no | int |  |  | Y | 1 |
| 状态 | status | int(2) | 1:有效 0:无效 -1:过期 |  | Y | 1 |
| 父级菜单ID | parent\_menu\_id | bigint | 父级菜单ID |  |  |  |
| 创建人 | create\_by | varchar(50) |  |  |  |  |
| 创建日期 | create\_date | datetime |  |  |  |  |
| 更新人 | update\_by | varchar(50) |  |  |  |  |
| 更新日期 | update\_date | datetime |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| 系统角色 | t\_frame\_role | 系统角色 | | | | |
| 字段中文名 | 字段英文名 | 字段类型 | 注释 | 是否主键 | 是否非空 | 默认值 |
| 角色id | role\_id | int | 角色id | Y | Y |  |
| 角色编码 | role\_code | varchar(200) | 角色编码 |  | Y |  |
| 角色名称 | role\_name | varchar(200) | 角色名称 |  |  |  |
| 角色类型 | role\_type | varchar(20) | 角色类型， function(功能角色)、data(数据权限) |  |  |  |
| 创建人 | create\_by | varchar(50) | 创建人 |  |  |  |
| 创建日期 | create\_date | datetime | 创建日期 |  |  |  |
| 更新人 | update\_by | varchar(50) | 更新人 |  |  |  |
| 更新日期 | update\_date | datetime | 更新日期 |  |  |  |
| status | status | int(2) | 1:有效 0:无效 -1:过期 |  | Y | 1 |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| 系统角色菜单 | t\_frame\_role\_menu |  | | | | |
| 字段中文名 | 字段英文名 | 字段类型 | 注释 | 是否主键 | 是否非空 | 默认值 |
| role\_menu\_id | role\_menu\_id | bigint | 系统角色id | Y | Y |  |
| 角色id | role\_id | int | 角色id |  | Y |  |
| menu\_id | menu\_id | bigint |  |  |  |  |
| role\_code | role\_code | varchar(200) |  |  | Y |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| 角色资源列表 | t\_frame\_role\_resource | 角色资源列表 | | | | |
| 字段中文名 | 字段英文名 | 字段类型 | 注释 | 是否主键 | 是否非空 | 默认值 |
| role\_resource\_id | role\_resource\_id | bigint | 角色菜单资源ID | Y | Y |  |
| 角色id | role\_id | int | 角色id |  |  |  |
| resource\_id | resource\_id | bigint |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| SQL脚本数据接口表 | t\_frame\_data\_SQL\_config |  | | | | |
| 字段中文名 | 字段英文名 | 字段类型 | 注释 | 是否主键 | 是否非空 | 默认值 |
| id | id | bigint |  | Y | Y |  |
| 数据编号 | DATA\_CODE | VARCHAR(20) |  |  | Y |  |
| data\_interface\_id | data\_interface\_id | bigint |  |  |  |  |
| SQL内容 | SQL\_CONTENT | text |  |  |  |  |
| 数据源 | data\_source | VARCHAR(20) |  |  |  |  |
| 数据库类型 | database\_type | VARCHAR(20) |  |  |  |  |
| 数据执行DAO | data\_dao\_name | VARCHAR(200) |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| 数据接口定义 | t\_frame\_data\_interface | 数据接口定义 | | | | |
| 字段中文名 | 字段英文名 | 字段类型 | 注释 | 是否主键 | 是否非空 | 默认值 |
| data\_interface\_id | data\_interface\_id | bigint | data\_interface\_id | Y | Y |  |
| 数据编号 | DATA\_CODE | VARCHAR(20) | 数据编号 |  | Y |  |
| 数据名称 | DATA\_NAME | VARCHAR(200) | 数据名称 |  |  |  |
| 数据类型 | DATA\_TYPE | VARCHAR(20) | 数据类型 |  |  |  |
| 数据执行服务名 | DATA\_EXECUTE\_NAME | VARCHAR(200) | 数据执行服务名 |  |  |  |
| 描述 | DESCRIPTION | TEXT | 描述 |  |  |  |
| 创建人 | create\_by | varchar(50) | 创建人 |  |  |  |
| 创建日期 | create\_date | datetime | 创建日期 |  |  |  |
| 更新人 | update\_by | varchar(50) | 更新人 |  |  |  |
| 更新日期 | update\_date | datetime | 更新日期 |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| 数据接口参数定义 | t\_frame\_data\_interface\_params |  | | | | |
| 字段中文名 | 字段英文名 | 字段类型 | 注释 | 是否主键 | 是否非空 | 默认值 |
| id | id | bigint |  | Y | Y |  |
| 数据编号 | DATA\_CODE | VARCHAR(20) |  |  | Y |  |
| 序号 | SEQ | int |  |  | Y |  |
| 参数名称 | param\_name | varchar(20) |  |  | Y |  |
| data\_interface\_id | data\_interface\_id | bigint |  |  |  |  |
| 参数类型 | PARAM\_TYPE | varchar(20) |  |  |  |  |
| 参数描述 | PARAM\_DESC | varchar(200) |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| 存储过程数据接口表 | t\_frame\_data\_SP\_config |  | | | | |
| 字段中文名 | 字段英文名 | 字段类型 | 注释 | 是否主键 | 是否非空 | 默认值 |
| id | id | bigint |  | Y | Y |  |
| 数据编号 | DATA\_CODE | VARCHAR(20) |  | Y | Y |  |
| data\_interface\_id | data\_interface\_id | bigint |  |  |  |  |
| SQL内容 | SQL\_CONTENT | text |  |  |  |  |
| 数据源 | data\_source | VARCHAR(20) |  |  |  |  |
| 数据库类型 | database\_type | VARCHAR(20) |  |  |  |  |
| 数据执行DAO | data\_dao\_name | VARCHAR(200) |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| 角色数据接口权限 | t\_frame\_role\_data\_interface | 角色数据接口权限 | | | | |
| 字段中文名 | 字段英文名 | 字段类型 | 注释 | 是否主键 | 是否非空 | 默认值 |
| id | id | bigint |  | Y | Y |  |
| 角色id | role\_id | int | 角色id |  | Y |  |
| 数据接口的首次创建时名称的MD5值 | data\_interface\_id | varchar(100) | 接口名称的创建时的md5值存放在mongodb中，请参考DataInterFaceVO |  | Y |  |
| 所属微服务 | belong\_microservice | varchar(200) | 所属微服务 |  | Y |  |
| 接口名称 | data\_interface\_name | varchar(200) | 接口名称 |  | Y |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| t\_framework\_resource | t\_framework\_resource2 |  | | | | |
| 字段中文名 | 字段英文名 | 字段类型 | 注释 | 是否主键 | 是否非空 | 默认值 |
| id | id | bigint(20) |  |  | Y |  |
| resource\_code | resource\_code | varchar(200) |  | Y | Y |  |
| resource\_name | resource\_name | varchar(200) |  |  |  |  |
| uri | uri | varchar(2000) |  |  |  |  |
| method | method | varchar(200) |  |  |  |  |
| create\_by | create\_by | varchar(50) |  |  |  |  |
| create\_date | create\_date | datetime |  |  |  |  |
| update\_by | update\_by | varchar(50) |  |  |  |  |
| update\_date | update\_date | datetime |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| REST服务数据接口定义表 | t\_frame\_data\_REST\_config |  | | | | |
| 字段中文名 | 字段英文名 | 字段类型 | 注释 | 是否主键 | 是否非空 | 默认值 |
| id | id | bigint |  | Y | Y |  |
| 数据编号 | DATA\_CODE | VARCHAR(20) |  |  | Y |  |
| data\_interface\_id | data\_interface\_id | bigint |  |  |  |  |
| 数据访问地址 | URI | text |  |  |  |  |
| 访问方式 | method | char(10) |  |  |  |  |
| 是否需要登录 | must\_login | VARCHAR(20) |  |  |  |  |
| 登录服务BEAN | Login\_bean | VARCHAR(200) |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| 系统微服务注册表 | t\_microservice\_register | 系统微服务注册表 | | | | |
| 字段中文名 | 字段英文名 | 字段类型 | 注释 | 是否主键 | 是否非空 | 默认值 |
| microservice\_id | microservice\_id | int |  | Y | Y |  |
| 微服务名称 | microservice\_name | varchar(200) | 微服务名称 |  |  |  |
| 创建人 | create\_by | varchar(50) |  |  |  |  |
| 创建日期 | create\_date | datetime |  |  |  |  |
| 更新人 | update\_by | varchar(50) |  |  |  |  |
| 更新日期 | update\_date | datetime |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| 角色数据权限列表 | t\_frame\_role\_data | 数据权限列表  角色类型为功能权限之外的权限统一为功能权限 | | | | |
| 字段中文名 | 字段英文名 | 字段类型 | 注释 | 是否主键 | 是否非空 | 默认值 |
| data\_auth\_list\_id | data\_auth\_list\_id | int |  | Y | Y |  |
| 角色id | role\_id | int | 角色id |  |  |  |
| 权限维度 | data\_dimension | varchar(200) | 权限维度:： org（组织机构） / micro 微服务 |  | Y |  |
| 权限值 | data\_dimension\_value | varchar(500) | 权限值 |  | Y |  |
| 创建人 | create\_by | varchar(50) |  |  |  |  |
| 创建日期 | create\_date | datetime |  |  |  |  |
| 更新人 | update\_by | varchar(50) |  |  |  |  |
| 更新日期 | update\_date | datetime |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| 系统数据字典主表 | t\_system\_dic\_master | 系统数据字典主表 | | | | |
| 字段中文名 | 字段英文名 | 字段类型 | 注释 | 是否主键 | 是否非空 | 默认值 |
| dic\_master\_id | dic\_master\_id | bigint | id | Y | Y |  |
| 字典编码 | dic\_code | varchar(50) | 字典编码 |  | Y |  |
| 字典名称 | dic\_name | varchar(200) | 字典名称 |  |  |  |
| 备注 | remark | varchar(2000) | 备注 |  |  |  |
| 所属微服务 | belong\_micro\_service | varchar(200) | 所属微服务，默认为General(通用) |  | Y | General |
| 父级id | parent\_master\_id | bigint | 父级id |  |  |  |
| 状态 | status | int(2) | 状态，1/有效 0/无效 -1/过期 |  | Y | 1 |
| 创建人 | create\_by | bigint | 创建人 |  | Y |  |
| 创建日期 | create\_date | timestamp | 创建日期 |  | Y | CURRENT\_TIMESTAMP |
| 更新人 | update\_by | bigint | 更新人 |  | Y |  |
| 更新日期 | update\_date | timestamp | 更新日期 |  | Y | CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| 系统数据字典字典项表 | t\_system\_dic\_item | 系统数据字典字典项表 | | | | |
| 字段中文名 | 字段英文名 | 字段类型 | 注释 | 是否主键 | 是否非空 | 默认值 |
| dic\_item\_id | dic\_item\_id | bigint | id | Y | Y |  |
| dic\_master\_id | dic\_master\_id | bigint | 数据字典主表id |  | Y |  |
| 字典项编码 | dic\_item\_code | varchar(50) | 字典项编码 |  | Y |  |
| 字典项名称 | dic\_item\_name | varchar(200) | 字典项名称 |  |  |  |
| 字典项值 | dic\_item\_value | varchar(200) | 字典项值 |  |  |  |
| 扩展属性1 | ext\_attribut1 | varchar(2000) | 扩展属性1 |  |  |  |
| 扩展属性3 | ext\_attribut3 | varchar(2000) | 扩展属性1 |  |  |  |
| 扩展属性2 | ext\_attribut2 | varchar(2000) | 扩展属性1 |  |  |  |
| 扩展属性4 | ext\_attribut4 | varchar(2000) | 扩展属性1 |  |  |  |
| 扩展属性5 | ext\_attribut5 | varchar(2000) | 扩展属性1 |  |  |  |
| 序号 | seq | int(4) | 序号 |  | Y | 0 |
| 语言 | language | varchar(20) | 语言 |  | Y | zh\_CN |
| 状态 | status | int(2) | 状态，1/有效 0/无效 -1/过期 |  | Y | 1 |
| 创建人 | create\_by | bigint | 创建人 |  | Y |  |
| 创建日期 | create\_date | timestamp | 创建日期 |  | Y | CURRENT\_TIMESTAMP |
| 更新人 | update\_by | bigint | 更新人 |  | Y |  |
| 更新日期 | update\_date | timestamp | 更新日期 |  | Y | CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP |

### Mongodb

#### 日志存储

主要的collection按微服务的名称存储日志，方便日志查询。如下为日志查询脚本。

db.getCollection("COMMON-SERVICE\_logbackLogCollection").find({

createDate: {

$gte: new Date("2020-02-16")

},

logger: /./i, //关键字查询 i表示不区分大小写

message: /./i

, //关键字查询

level: {

$in: ["ERROR"] //表示按正则表达式来查找,{$regex:/INFO/}

}

}, {

// 显示字段配置

\_id: 0,

logger: 1,

message: 1,

level: 1,

createDate: 1

}).sort({

createDate: - 1 //排序

}).skip(0).limit(100);

// 创建索引

//db.getCollection("COMMON-SERVICE\_logbackLogCollection").createIndex({message:-1}, {name:"message\_springgatewaylog"})

//db.getCollection("COMMON-SERVICE\_logbackLogCollection").count()

//

////创建索引

//db.getCollection("HTCLOUD-PAYCENTER\_logbackLogCollection").createIndex({level:-1,createDate:-1}, {name:"idx\_level\_createDate\_HTCLOUD-PAYCENTER\_logbackLogCollection"})

//db.getCollection("COMMON-SERVICE\_logbackLogCollection").createIndex({level:-1,createDate:-1}, {name:"idx\_level\_createDate\_COMMON-SERVICE\_logbackLogCollection"})

//db.getCollection("SPRING-GATEWAY\_logbackLogCollection").createIndex({level:-1,createDate:-1}, {name:"idx\_level\_createDate\_SPRING-GATEWAY\_logbackLogCollection"})

//db.getCollection("XGSIXTEEN\_logbackLogCollection").createIndex({level:-1,createDate:-1}, {name:"idx\_level\_createDate\_XGSIXTEEN\_logbackLogCollection"})。

#### 数据接口定义

相关基础配置数据如下

*/\*  
 Navicat Premium Data Transfer  
  
 Source Server : mongodb-localhost  
 Source Server Type : MongoDB  
 Source Server Version : 40203  
 Source Host : localhost:27017  
 Source Schema : framework  
  
 Target Server Type : MongoDB  
 Target Server Version : 40203  
 File Encoding : 65001  
  
 Date: 18/03/2020 21:06:08  
\*/  
  
  
// ----------------------------  
// Collection structure for dataInterFaceVO  
// ----------------------------*db.getCollection(**"dataInterFaceVO"**).drop();  
db.createCollection(**"dataInterFaceVO"**);  
db.getCollection(**"dataInterFaceVO"**).createIndex({  
 **md5**: NumberInt(**"1"**)  
}, {  
 **name**: **"md5\_1"**,  
 **unique**: **true**});  
db.getCollection(**"dataInterFaceVO"**).createIndex({  
 **interfaceName**: NumberInt(**"1"**)  
}, {  
 **name**: **"interfaceName\_1"**,  
 **unique**: **true**});  
  
*// ----------------------------  
// Documents of dataInterFaceVO  
// ----------------------------*db.getCollection(**"dataInterFaceVO"**).insert([ {  
 **\_id**: ObjectId(**"5e69ec77543ee56a391f1fcb"**),  
 **interfaceName**: **"动态SQL示例所有登录用户"**,  
 **interfaceType**: **"sql"**,  
 **params**: [ ],  
 **urlOrSqlContent**: **"select \* from t\_frame\_user \n<where> \n\t<if test=\"userName!=null\">\n\t\tuser\_name = #{userName,jdbcType=VARCHAR}\n\t</if>\n\t<if test=\"password!=null\">\n\t and\tpassword = #{password,jdbcType=VARCHAR}\n\t</if>\n\t<if test=\"sexList!=null\">\n\t and sex in \n\t\t <foreach item=\"item\" collection=\"sexList\" separator=\",\" open=\"(\" close=\")\" index=\"\">\n\t\t #{item, jdbcType=VARCHAR}\n\t\t </foreach>\n\t</if>\n</where>"**,  
 **microServiceName**: **"COMMON-SERVICE"**,  
 **authMethod**: **"ALLSYSTEMUSER"**,  
 **createdOrUpdateTime**: ISODate(**"2020-03-12T09:07:04.468Z"**),  
 **createdOrUpdateBy**: NumberLong(**"1"**),  
 **createdOrUpdateUserName**: **"admin"**,  
 **status**: NumberInt(**"1"**),  
 **\_class**: **"org.cloud.mongo.DataInterFaceVO"**,  
 **restMethod**: **null**,  
 **md5**: **"50d55445206d7a70022d932c98e2bd7d"**} ]);  
db.getCollection(**"dataInterFaceVO"**).insert([ {  
 **\_id**: ObjectId(**"5e69ec93543ee56a391f1fe2"**),  
 **interfaceName**: **"动态SQL仅查询当前用户"**,  
 **interfaceType**: **"sql"**,  
 **params**: [ ],  
 urlOrSqlContent: **"select \* from t\_frame\_user \n<where> \n\t<if test=\"userName!=null\">\n\t\tuser\_name = #{userName,jdbcType=VARCHAR}\n\t</if>\n\tand id= #{paramsCurrentUserId,jdbcType=BIGINT}\n\t<if test=\"password!=null\">\n\t and\tpassword = #{password,jdbcType=VARCHAR}\n\t</if>\n\t<if test=\"sexList!=null\">\n\t and sex in \n\t\t <foreach item=\"item\" collection=\"sexList\" separator=\",\" open=\"(\" close=\")\" index=\"\">\n\t\t #{item, jdbcType=VARCHAR}\n\t\t </foreach>\n\t</if>\n</where>"**,  
 microServiceName: **"COMMON-SERVICE"**,  
 authMethod: **"ALLSYSTEMUSER"**,  
 createdOrUpdateTime: ISODate(**"2020-03-12T09:05:42.503Z"**),  
 createdOrUpdateBy: NumberLong(**"1"**),  
 createdOrUpdateUserName: **"admin"**,  
 status: NumberInt(**"1"**),  
 **\_class**: **"org.cloud.mongo.DataInterFaceVO"**,  
 **restMethod**: **null**,  
 **md5**: **"23450f7c37ba735d498a0be8575baf2d"**} ]);  
db.getCollection(**"dataInterFaceVO"**).insert([ {  
 **\_id**: ObjectId(**"5e69ecb0543ee56a391f1ff9"**),  
 **interfaceName**: **"动态SQL无需授权示例"**,  
 **interfaceType**: **"sql"**,  
 **params**: [ ],  
 **urlOrSqlContent**: **"select \* from t\_frame\_user \n<where> \n\t<if test=\"userName!=null\">\n\t\tuser\_name = #{userName,jdbcType=VARCHAR}\n\t</if>\n\t<if test=\"password!=null\">\n\t and\tpassword = #{password,jdbcType=VARCHAR}\n\t</if>\n\t<if test=\"sexList!=null\">\n\t and sex in \n\t\t <foreach item=\"item\" collection=\"sexList\" separator=\",\" open=\"(\" close=\")\" index=\"\">\n\t\t #{item, jdbcType=VARCHAR}\n\t\t </foreach>\n\t</if>\n</where>"**,  
 **microServiceName**: **"COMMON-SERVICE"**,  
 **authMethod**: **"NOAUTH"**,  
 **createdOrUpdateTime**: ISODate(**"2020-03-12T09:09:58.275Z"**),  
 **createdOrUpdateBy**: NumberLong(**"1"**),  
 **createdOrUpdateUserName**: **"admin"**,  
 **status**: NumberInt(**"1"**),  
 **\_class**: **"org.cloud.mongo.DataInterFaceVO"**,  
 **restMethod**: **null**,  
 **md5**: **"da1664b5771bce00f4919c0b1942cdcf"**} ]);  
db.getCollection(**"dataInterFaceVO"**).insert([ {  
 **\_id**: ObjectId(**"5e69ecc8543ee56a391f2010"**),  
 **interfaceName**: **"动态SQL需要授权给用户的示例"**,  
 **interfaceType**: **"sql"**,  
 **params**: [ ],  
 **urlOrSqlContent**: **"select \* from t\_frame\_user \n<where> \n\t<if test=\"userName!=null\">\n\t\tuser\_name = #{userName,jdbcType=VARCHAR}\n\t</if>\n\t<if test=\"password!=null\">\n\t and\tpassword = #{password,jdbcType=VARCHAR}\n\t</if>\n\t<if test=\"sexList!=null\">\n\t and sex in \n\t\t <foreach item=\"item\" collection=\"sexList\" separator=\",\" open=\"(\" close=\")\" index=\"\">\n\t\t #{item, jdbcType=VARCHAR}\n\t\t </foreach>\n\t</if>\n</where>"**,  
 **microServiceName**: **"COMMON-SERVICE"**,  
 **authMethod**: **"BYUSERPERMISSION"**,  
 **createdOrUpdateTime**: ISODate(**"2020-03-12T09:07:08.18Z"**),  
 **createdOrUpdateBy**: NumberLong(**"1"**),  
 **createdOrUpdateUserName**: **"admin"**,  
 **status**: NumberInt(**"1"**),  
 **\_class**: **"org.cloud.mongo.DataInterFaceVO"**,  
 **restMethod**: **null**,  
 **md5**: **"6191e5feb221dd9d0acb24ae2895fb80"**} ]);  
db.getCollection(**"dataInterFaceVO"**).insert([ {  
 **\_id**: ObjectId(**"5e6f56df0c730000a10008d9"**),  
 **interfaceName**: **"查询系统角色列表"**,  
 **interfaceType**: **"sql"**,  
 **params**: [ ],  
 **urlOrSqlContent**: **"<!--系统配置，切勿删除-->\nselect \* from t\_frame\_role\n<where>\n<if test=\"roleId!=null\">\n\trole\_id = #{roleId,jdbcType=INTEGER}\n</if>\n<if test=\"roleCode!=null\">\n\tand role\_code like concat('%', #{roleCode,jdbcType=VARCHAR},'%')\n</if>\n<if test=\"createBy!=null\">\n\tand create\_by = #{createBy,jdbcType=VARCHAR}\n</if>\n<if test=\"createDate!=null\">\n\tand create\_Date BETWEEN #{createDate.0,jdbcType=DATE} AND #{createDate.1,jdbcType=DATE}\n</if>\n<if test=\"updateBy!=null\">\n\tand update\_by = #{updateBy,jdbcType=VARCHAR}\n</if>\n<if test=\"updateDate!=null\">\n\tand update\_Date BETWEEN #{updateDate.0,jdbcType=DATE} AND #{updateDate.1,jdbcType=DATE}\n</if>\n<if test=\"status!=null\">\n\tand status= #{status,jdbcType=INTEGER}\n</if>\n<if test=\"roleName!=null\">\n\tand role\_name like concat('%', #{roleName,jdbcType=VARCHAR},'%')\n</if>\n<if test=\"keyWord!=null\">\n\tor role\_name like concat('%', #{keyWord,jdbcType=VARCHAR},'%') or role\_code like concat('%', #{keyWord,jdbcType=VARCHAR},'%')\n</if>\n</where>"**,  
 **microServiceName**: **"General"**,  
 **authMethod**: **"ALLSYSTEMUSER"**,  
 createdOrUpdateTime: ISODate(**"2020-03-18T05:12:56.096Z"**),  
 createdOrUpdateBy: NumberLong(**"1"**),  
 createdOrUpdateUserName: **"admin"**,  
 md5: **"fca0c2236325f81017b12e4da4d32990"**,  
 status: NumberInt(**"1"**),  
 \_class: **"org.cloud.mongo.DataInterFaceVO"**,  
 restMethod: **null**} ]);  
db.getCollection(**"dataInterFaceVO"**).insert([ {  
 \_id: ObjectId(**"5e6f56df0c730000a10008da"**),  
 interfaceName: **"查询需要授权的资源清单列表"**,  
 interfaceType: **"sql"**,  
 params: [ ],  
 urlOrSqlContent: **"SELECT\n\tresource\_id,\n\tbelong\_microservice,\n\tresource\_path,\n\tresource\_name \nFROM\n\tt\_framework\_resource \nWHERE\n\tmethod = 'BYUSERPERMISSION' \n<if test=\"keyWord!=null\">\nand ( belong\_microservice like concat('%',#{keyWord,jdbcType=VARCHAR},\"%\") or resource\_path like concat('%',#{keyWord,jdbcType=VARCHAR},\"%\") or resource\_name like concat('%',#{keyWord,jdbcType=VARCHAR},\"%\") )\n</if>"**,  
 microServiceName: **"General"**,  
 authMethod: **"ALLSYSTEMUSER"**,  
 createdOrUpdateTime: ISODate(**"2020-03-15T12:35:29.41Z"**),  
 createdOrUpdateBy: NumberLong(**"1"**),  
 createdOrUpdateUserName: **"admin"**,  
 md5: **"ff4211d5f54026dc74e8de937a18b790"**,  
 status: NumberInt(**"1"**),  
 \_class: **"org.cloud.mongo.DataInterFaceVO"**,  
 restMethod: **null**} ]);  
db.getCollection(**"dataInterFaceVO"**).insert([ {  
 \_id: ObjectId(**"5e6f56df0c730000a10008db"**),  
 interfaceName: **"查询当前角色操作权限列表"**,  
 interfaceType: **"sql"**,  
 params: [ ],  
 urlOrSqlContent: **"<!-- 请勿删除 -->\nSELECT\n\tfrr.role\_resource\_id,\n\tfrr.role\_id,\n\tfrr.resource\_id,\n\tfr.belong\_microservice,\n\tfr.resource\_path,\n\tfr.resource\_name \nFROM\n\tt\_framework\_resource fr,t\_frame\_role\_resource frr\nWHERE fr.resource\_id = frr.resource\_id \nand fr.method = 'BYUSERPERMISSION' \nand frr.role\_id = #{roleId,jdbcType=INTEGER}\n\t"**,  
 microServiceName: **"General"**,  
 authMethod: **"ALLSYSTEMUSER"**,  
 createdOrUpdateTime: ISODate(**"2020-03-14T12:06:51.044Z"**),  
 createdOrUpdateBy: NumberLong(**"1"**),  
 createdOrUpdateUserName: **"admin"**,  
 md5: **"6a58842f2dab21009867e5d7782985ec"**,  
 status: NumberInt(**"1"**),  
 \_class: **"org.cloud.mongo.DataInterFaceVO"**,  
 restMethod: **null**} ]);  
db.getCollection(**"dataInterFaceVO"**).insert([ {  
 \_id: ObjectId(**"5e6f56df0c730000a10008dc"**),  
 interfaceName: **"查询需要授权的接口清单"**,  
 interfaceType: **"sql"**,  
 params: [ ],  
 urlOrSqlContent: **"SELECT\n\tresource\_id,\n\tbelong\_microservice,\n\tresource\_path,\n\tresource\_name \nFROM\n\tt\_framework\_resource \nWHERE\n\tmethod = 'BYUSERPERMISSION' "**,  
 microServiceName: **"General"**,  
 authMethod: **"ALLSYSTEMUSER"**,  
 createdOrUpdateTime: ISODate(**"2020-03-14T23:19:09.125Z"**),  
 createdOrUpdateBy: NumberLong(**"1"**),  
 createdOrUpdateUserName: **"admin"**,  
 md5: **"99deca6d1f7b31ceeeeaf99e5afa488d"**,  
 status: NumberInt(**"1"**),  
 \_class: **"org.cloud.mongo.DataInterFaceVO"**} ]);  
db.getCollection(**"dataInterFaceVO"**).insert([ {  
 \_id: ObjectId(**"5e6f56df0c730000a10008e2"**),  
 interfaceName: **"查询当前角色的数据接口列表"**,  
 interfaceType: **"sql"**,  
 params: [ ],  
 urlOrSqlContent: **"select \* from t\_frame\_role\_data\_interface where role\_id = #{roleId,jdbcType=INTEGER}"**,  
 microServiceName: **"General"**,  
 authMethod: **"ALLSYSTEMUSER"**,  
 createdOrUpdateTime: ISODate(**"2020-03-15T01:33:44.856Z"**),  
 createdOrUpdateBy: NumberLong(**"1"**),  
 createdOrUpdateUserName: **"admin"**,  
 md5: **"7d458ef7f373239fd213efa213a30091"**,  
 status: NumberInt(**"1"**),  
 \_class: **"org.cloud.mongo.DataInterFaceVO"**} ]);  
db.getCollection(**"dataInterFaceVO"**).insert([ {  
 \_id: ObjectId(**"5e703d0d1e209d1416226564"**),  
 interfaceName: **"查询所有用户的接口"**,  
 interfaceType: **"sql"**,  
 params: [ ],  
 urlOrSqlContent: **"select \* from t\_frame\_user\n<where>\n<if test=\"userName!=null\">\nuser\_name like concat('%',#{userName,jdbcType=VARCHAR},'%')\n</if>\n<if test=\"status!=null and status.size()>0\">\nand status in\n\t\t <foreach item=\"statusItem\" collection=\"status\" separator=\",\" open=\"(\" close=\")\" index=\"\">\n\t\t #{statusItem, jdbcType=INTEGER}\n\t\t </foreach>\n</if>\n</where>"**,  
 microServiceName: **"General"**,  
 authMethod: **"BYUSERPERMISSION"**,  
 createdOrUpdateTime: ISODate(**"2020-03-17T03:28:40.161Z"**),  
 createdOrUpdateBy: NumberLong(**"1"**),  
 createdOrUpdateUserName: **"admin"**,  
 md5: **"3efa00c2c4b06b73758fac9e95e9b6a9"**,  
 status: NumberInt(**"1"**),  
 \_class: **"org.cloud.mongo.DataInterFaceVO"**,  
 restMethod: **null**} ]);  
db.getCollection(**"dataInterFaceVO"**).insert([ {  
 \_id: ObjectId(**"5e71b5ec561cd27d329b5d1f"**),  
 interfaceName: **"按用户ID查询用户的角色列表"**,  
 interfaceType: **"sql"**,  
 params: [ ],  
 urlOrSqlContent: **"<!--查询当前用户的角色列表，请勿删除，系统用-->\nSELECT\n\tUR.role\_id,\n\tR.role\_code,\n\tR.role\_name,\n\tUR.validator\_start,\n\tUR.validator\_end \nFROM\n\tt\_frame\_user\_role ur,\n\tt\_frame\_role r \nWHERE\n\tur.role\_id = r.role\_id \n\tAND ur.user\_id = #{userId,jdbcType=BIGINT}"**,  
 microServiceName: **"General"**,  
 authMethod: **"BYUSERPERMISSION"**,  
 createdOrUpdateTime: ISODate(**"2020-03-18T08:51:48.15Z"**),  
 createdOrUpdateBy: NumberLong(**"1"**),  
 createdOrUpdateUserName: **"admin"**,  
 md5: **"fbf2e7c0ec6c91d9041ccd92509ce309"**,  
 status: NumberInt(**"1"**),  
 \_class: **"org.cloud.mongo.DataInterFaceVO"**,  
 restMethod: **null**} ]);

## 系统设计目标

构建一个弹性的，可扩展的，业务系统开发的基础平台，方便快速的开发业务系统的业务逻辑。

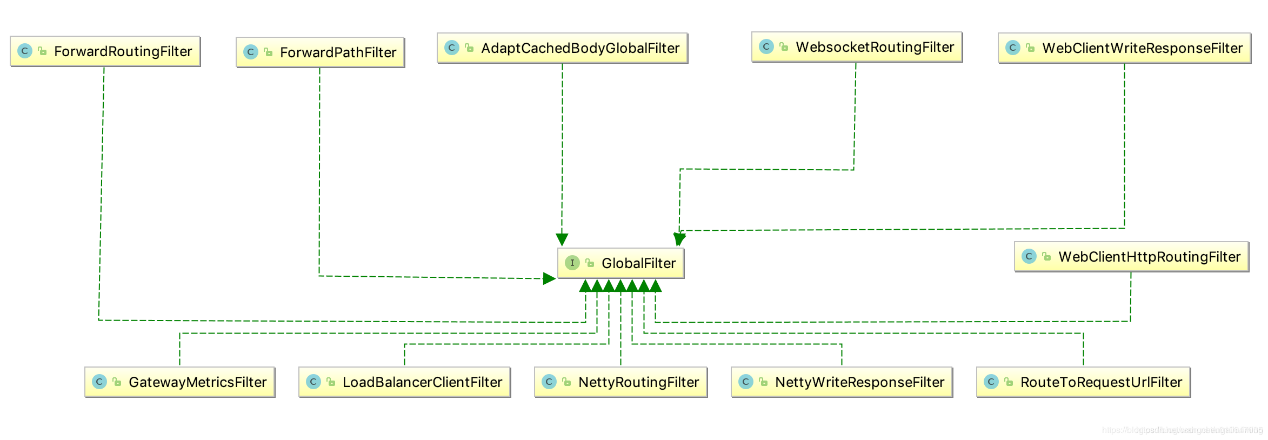
构建一个多租户的平台系统，方便业务系统的扩展。

构建一个以消息为中心的分布式的通信机制，实现系统相互之间的数据交互，实现数据的实效性。

构建一个友好的系统后台管理平台，实现用户权限的标准配置化。并实现一些基础的功能。

## 组件服务

### 用户认证



用户认证模块编写在网关服务中，用于处理所有用户请求的认证，注：只做认证，不做权限的管控，认证信息会通过redis共享给各系统，关键服务类如下，此认证框架感谢<https://blog.csdn.net/wangchengaihuiming/article/details/89193378>和<https://blog.csdn.net/MongolianWolf/article/details/94329980> 提供的帮助

|  |  |
| --- | --- |
| 类名 | 功能说明 |
| com.longyou.gateway.security.SecurityConfig | 安全配置类，配置一些相关的认证服务，如登录成功后的处理，失败后的处理，启动basic认证等等。 |
| com.longyou.gateway.security.SecurityUserDetailsService | 获取用户的接口实现，用于通过用户名获取用户，具体的逻辑请查看代码 |
| com.longyou.gateway.security.handler.AuthenticationFaillHandler | 校验失败的处理类 |
| com.longyou.gateway.security.handler.AuthenticationSuccessHandler | 用户校验后的处理类，用于生成登录token等操作。 |
| com.longyou.gateway.security.handler.CustomHttpBasicServerAuthenticationEntryPoint | Httpbasic鉴权失败后的处理类。返回base64鉴权失败后的信息 |
| com.longyou.gateway.security.handler.CustomServerLogoutSuccessHandler | 退出登录后的处理 |
| com.longyou.gateway.util.MD5PasswordEncoder | 密码校验和加密处理，及密码的加盐处理。 |

### 用户管理

用户管理组件主要用于各个应用的用户的处理，用户管理内置在commonservice微服务中。含微信用户登录后的处理等，如下为相关的类说明

|  |  |
| --- | --- |
| 类名 | 功能说明 |
| com.longyou.comm.mapper.UserInfoMapper | 获取登录用户的相关信息的MAPPER |
| com.longyou.comm.mapper.TFrameUserDao | 实现用户的增加删除和修改的功能mapper |
| com.longyou.comm.admin.service.impl.UserAdminService | 用户管理相关功能的服务类，含新增改的操作，删除方式采用软删除的方式，不删除实际的用户信息，增加用户角色等等。 |
| com.longyou.comm.service.impl.UserInfoService | 处理用户登录信息的查询的服务类 |
| com.longyou.comm.service.impl.WeixinLoginUserGetService | 处理微信用户预登录后的处理，主要用于获取用户的openid |
| org.cloud.userinfo.LoginUserGetInterface | 第三方登录的实现类，根据**config**:  **micro-app**:  **app-list[0]**:  **type**: weixin 的配置实现不同的登录方式，目前仅支持微信 |
| 角色授权时相关查询操作 | 全部配置在数据接口中，列表如下，也可以通过后台的接口管理功能查询。 |
| // 4  {  "\_id": ObjectId("5e71b5ec561cd27d329b5d1f"),  "interfaceName": "按用户ID查询用户的角色列表",  "interfaceType": "sql",  "params": [ ],  "urlOrSqlContent": "<!--查询当前用户的角色列表，请勿删除，系统用-->\nSELECT\n\tUR.role\_id,\n\tR.role\_code,\n\tR.role\_name,\n\tUR.validator\_start,\n\tUR.validator\_end \nFROM\n\tt\_frame\_user\_role ur,\n\tt\_frame\_role r \nWHERE\n\tur.role\_id = r.role\_id \n\tAND ur.user\_id = #{userId,jdbcType=BIGINT}",  "microServiceName": "General",  "authMethod": "BYUSERPERMISSION",  "createdOrUpdateTime": ISODate("2020-03-18T08:51:48.15Z"),  "createdOrUpdateBy": NumberLong("1"),  "createdOrUpdateUserName": "admin",  "md5": "fbf2e7c0ec6c91d9041ccd92509ce309",  "status": NumberInt("1"),  "\_class": "org.cloud.mongo.DataInterFaceVO",  "restMethod": null  }  // 5  {  "\_id": ObjectId("5e703d0d1e209d1416226564"),  "interfaceName": "查询所有用户的接口",  "interfaceType": "sql",  "params": [ ],  "urlOrSqlContent": "select \* from t\_frame\_user\n<where>\n<if test=\"userName!=null\">\nuser\_name like concat('%',#{userName,jdbcType=VARCHAR},'%')\n</if>\n<if test=\"status!=null and status.size()>0\">\nand status in\n\t\t <foreach item=\"statusItem\" collection=\"status\" separator=\",\" open=\"(\" close=\")\" index=\"\">\n\t\t #{statusItem, jdbcType=INTEGER}\n\t\t </foreach>\n</if>\n</where>",  "microServiceName": "General",  "authMethod": "BYUSERPERMISSION",  "createdOrUpdateTime": ISODate("2020-03-17T03:28:40.161Z"),  "createdOrUpdateBy": NumberLong("1"),  "createdOrUpdateUserName": "admin",  "md5": "3efa00c2c4b06b73758fac9e95e9b6a9",  "status": NumberInt("1"),  "\_class": "org.cloud.mongo.DataInterFaceVO",  "restMethod": null  } | |

### 角色管理

角色管理用于处理角色的相关处理，增加删除和修改功能。

|  |  |
| --- | --- |
| 类名 | 功能说明 |
| com.longyou.comm.admin.service.impl.RoleAdminService | 增加和保存用户的数据权限，后台接口页面为 |
| 角色查询操作 | 全部配置在数据接口中，列表如下，也可以通过后台的接口管理功能查询。 |
| // 1  {  "\_id": ObjectId("5e71b5ec561cd27d329b5d1f"),  "interfaceName": "按用户ID查询用户的角色列表",  "interfaceType": "sql",  "params": [ ],  "urlOrSqlContent": "<!--查询当前用户的角色列表，请勿删除，系统用-->\nSELECT\n\tUR.role\_id,\n\tR.role\_code,\n\tR.role\_name,\n\tUR.validator\_start,\n\tUR.validator\_end \nFROM\n\tt\_frame\_user\_role ur,\n\tt\_frame\_role r \nWHERE\n\tur.role\_id = r.role\_id \n\tAND ur.user\_id = #{userId,jdbcType=BIGINT}",  "microServiceName": "General",  "authMethod": "BYUSERPERMISSION",  "createdOrUpdateTime": ISODate("2020-03-18T08:51:48.15Z"),  "createdOrUpdateBy": NumberLong("1"),  "createdOrUpdateUserName": "admin",  "md5": "fbf2e7c0ec6c91d9041ccd92509ce309",  "status": NumberInt("1"),  "\_class": "org.cloud.mongo.DataInterFaceVO",  "restMethod": null  }  // 2  {  "\_id": ObjectId("5e6f56df0c730000a10008db"),  "interfaceName": "查询当前角色操作权限列表",  "interfaceType": "sql",  "params": [ ],  "urlOrSqlContent": "<!-- 请勿删除 -->\nSELECT\n\tfrr.role\_resource\_id,\n\tfrr.role\_id,\n\tfrr.resource\_id,\n\tfr.belong\_microservice,\n\tfr.resource\_path,\n\tfr.resource\_name \nFROM\n\tt\_framework\_resource fr,t\_frame\_role\_resource frr\nWHERE fr.resource\_id = frr.resource\_id \nand fr.method = 'BYUSERPERMISSION' \nand frr.role\_id = #{roleId,jdbcType=INTEGER}\n\t",  "microServiceName": "General",  "authMethod": "ALLSYSTEMUSER",  "createdOrUpdateTime": ISODate("2020-03-14T12:06:51.044Z"),  "createdOrUpdateBy": NumberLong("1"),  "createdOrUpdateUserName": "admin",  "md5": "6a58842f2dab21009867e5d7782985ec",  "status": NumberInt("1"),  "\_class": "org.cloud.mongo.DataInterFaceVO",  "restMethod": null  }  // 3  {  "\_id": ObjectId("5e6f56df0c730000a10008e2"),  "interfaceName": "查询当前角色的数据接口列表",  "interfaceType": "sql",  "params": [ ],  "urlOrSqlContent": "select \* from t\_frame\_role\_data\_interface where role\_id = #{roleId,jdbcType=INTEGER}",  "microServiceName": "General",  "authMethod": "ALLSYSTEMUSER",  "createdOrUpdateTime": ISODate("2020-03-15T01:33:44.856Z"),  "createdOrUpdateBy": NumberLong("1"),  "createdOrUpdateUserName": "admin",  "md5": "7d458ef7f373239fd213efa213a30091",  "status": NumberInt("1"),  "\_class": "org.cloud.mongo.DataInterFaceVO"  }  // 4  {  "\_id": ObjectId("5e6f56df0c730000a10008d9"),  "interfaceName": "查询系统角色列表",  "interfaceType": "sql",  "params": [ ],  "urlOrSqlContent": "<!--系统配置，切勿删除-->\nselect \* from t\_frame\_role\n<where>\n<if test=\"roleId!=null\">\n\trole\_id = #{roleId,jdbcType=INTEGER}\n</if>\n<if test=\"roleCode!=null\">\n\tand role\_code like concat('%', #{roleCode,jdbcType=VARCHAR},'%')\n</if>\n<if test=\"createBy!=null\">\n\tand create\_by = #{createBy,jdbcType=VARCHAR}\n</if>\n<if test=\"createDate!=null\">\n\tand create\_Date BETWEEN #{createDate.0,jdbcType=DATE} AND #{createDate.1,jdbcType=DATE}\n</if>\n<if test=\"updateBy!=null\">\n\tand update\_by = #{updateBy,jdbcType=VARCHAR}\n</if>\n<if test=\"updateDate!=null\">\n\tand update\_Date BETWEEN #{updateDate.0,jdbcType=DATE} AND #{updateDate.1,jdbcType=DATE}\n</if>\n<if test=\"status!=null\">\n\tand status= #{status,jdbcType=INTEGER}\n</if>\n<if test=\"roleName!=null\">\n\tand role\_name like concat('%', #{roleName,jdbcType=VARCHAR},'%')\n</if>\n<if test=\"keyWord!=null\">\n\tor role\_name like concat('%', #{keyWord,jdbcType=VARCHAR},'%') or role\_code like concat('%', #{keyWord,jdbcType=VARCHAR},'%')\n</if>\n</where>",  "microServiceName": "General",  "authMethod": "ALLSYSTEMUSER",  "createdOrUpdateTime": ISODate("2020-03-18T05:12:56.096Z"),  "createdOrUpdateBy": NumberLong("1"),  "createdOrUpdateUserName": "admin",  "md5": "fca0c2236325f81017b12e4da4d32990",  "status": NumberInt("1"),  "\_class": "org.cloud.mongo.DataInterFaceVO",  "restMethod": null  } | |

### 资源权限管理

资源权限管理用于处理各个微服务的资源的权限，权限点通过注解后自动上传到服务器上并保存，可通过后台的/#/admin/microservice的微服务管理功能同步权限点到服务器。相关类如下。,相关操作文档请参考《权限配置开发文档.docx》

|  |  |
| --- | --- |
| 类名 | 功能说明 |
| org.cloud.controller.SystemResourceRegisterController | 权限注册服务类，用于同步权限点到服务器。 |
| org.cloud.aop.SystemResourceAspect | 权限点校验切面服务类，用于校验权限点是否通过的功能，根据权限注解。 |
| org.cloud.annotation.SystemResource | 权限注解类，注解参数如下。  String menuName() **default ""**; *// 菜单名称，可选，如果不为空，那么会自动创建菜单* String menuCode() **default ""**; *// 菜单编码，可选，如果不为空，那么会自动创建菜单* String parentMenuCode() **default ""**; *// 父级菜单名称，默认为无上级* String parentMenuName() **default ""**; *// 父级菜单名称，默认为无上级* String path() **default ""**; *//资源路径，用于分类资源,例如 商品管理/SPU管理* **int** index() **default** 0; *// 序号* String description() **default ""**; *// 资源名称* String value() **default ""**; *// 资源code* CoreConstant.AuthMethod authMethod() **default** CoreConstant.AuthMethod.***ALLSYSTEMUSER***; *//验证方式,默认为所有的登录用户,ByUserAuth* |

### 系统接口管理

用于管理数据接口定义，相关信息存储在mongodb中，可在/admin/datainterface功能中配置,使用说明，请参考《自定义查询开发文档.docx》

主要服务类如下

|  |  |
| --- | --- |
| 类名 | 功能说明 |
| com.longyou.comm.admin.service.impl.DataInterFaceService | 接口配置服务类，用于处理服务的增加删除功能 |
| com.longyou.comm.admin.controller.DataInterfaceController | 用于查询相关的数据接口的功能，主要用于前台的展示 |
| org.cloud.mybatis.dynamic.DynamicSqlUtil | 动态SQL查询的服务类，用于处理动态SQL的处理，包含分页 |
| org.cloud.controller.DataInterfaceExecController | 动态SQL数据接口执行服务接口，包含权限的验证。 |
| org.cloud.utils.AuthCheckUtils | 数据接口权限验证，用于验证是否有数据接口执行的权限，如果没有则抛出错误。逻辑如下  **if** (dataInterFaceVO == **null**) {  **throw new** BusinessException(**"不存在的动态接口定义，请检查"**, md5); } **if** (!CoreConstant.AuthMethod.***NOAUTH***.equals(dataInterFaceVO.getAuthMethod())) {  **if** (RequestContextManager.*single*().getRequestContext().getUser() == **null**) {  **throw new** BusinessException(**"请先登录！"**, md5, HttpStatus.***UNAUTHORIZED***.value());  } **else if** (CoreConstant.AuthMethod.***BYUSERPERMISSION***.equals(dataInterFaceVO.getAuthMethod())) {  LoginUserDetails loginUserDetails = RequestContextManager.*single*().getRequestContext().getUser();  Set<String> userDataiterfaces = **redisUtil**.hashGet(CoreConstant.***USER\_LOGIN\_SUCCESS\_CACHE\_KEY*** + loginUserDetails.getId(), CoreConstant.UserCacheKey.***DATA\_INTERFACE***.value());  **if** (userDataiterfaces == **null** || userDataiterfaces.isEmpty() || !userDataiterfaces.contains(md5)) {  **logger**.error(loginUserDetails.getUsername() + **",正在非法的请求接口："** + md5);  **throw** HttpClientErrorException.*create*(HttpStatus.***UNAUTHORIZED***, **"没有此接口权限，请联系管理员授权！"**, **null**, **null**, Charset.*forName*(**"utf8"**));  }  } } |

### 数据字典管理

数据字典管理，用于管理系统相关的数据字典，区分各个微服务和语言。主要服务类如下。

|  |  |
| --- | --- |
| 类名 | 功能说明 |
| com.longyou.comm.admin.controller.DicController | @RequestMapping(value = **"/saveOrUpdate"**, method = RequestMethod.***POST***) @SystemResource(value = **"保存或者更新"**,description = **"保存或者更新数据字典"**,authMethod = CoreConstant.AuthMethod.***BYUSERPERMISSION***)  分页查询  @RequestMapping(value = **"/page/list/{page}/{pageSize}"**, method = RequestMethod.***POST***)  按masterId查询数据字典项列表  @RequestMapping(value = **"/selectDicByDicMasterId/{dicMasterId}"**, method = RequestMethod.***GET***) |
| com.longyou.comm.admin.service.impl.DicServiceImpl | **public int** SaveOrUpdate(TSystemDicMaster systemDicMaster) **throws** Exception;  **public** TSystemDicMaster getDicMasterById( Long dicMasterId) **throws** Exception;  **public** TSystemDicItem getDicItemById(Long dicMasterId) **throws** Exception;  **public** List<TSystemDicItem> getDicItemsByMasterId(Long dicMasterId) **throws** Exception;  **public** List<TSystemDicItem> getDicItemsByDicCode(Map<String,Object> params) **throws** Exception;  **public** Page<TSystemDicMaster> listPage(QueryParamVO queryParams) **throws** Exception;  **public void** refreshCache() **throws** Exception; |

## 系统服务接口（按POSTMAN格式）

### 环境配置

#### 测试环境配置

{

"id": "2991fc61-8d7c-4340-9cc4-efc2a4dc5555",

"name": "测试环境",

"values": [

{

"key": "homeUrl",

"value": "http://172.16.4.22:20003",

"enabled": true

},

{

"key": "group",

"value": "",

"enabled": true

}

],

"\_postman\_variable\_scope": "environment",

"\_postman\_exported\_at": "2020-03-29T02:26:08.987Z",

"\_postman\_exported\_using": "Postman/7.21.1"

}

#### 生产环境

{

"id": "3a9dca1f-22d6-45f6-af57-f7e063a159eb",

"name": "production",

"values": [

{

"key": "homeUrl",

"value": "https://api.ly-huangtu.cn:444",

"type": "text",

"enabled": true

},

{

"key": "group",

"value": "",

"type": "text",

"enabled": true

}

],

"\_postman\_variable\_scope": "environment",

"\_postman\_exported\_at": "2020-03-29T02:26:18.248Z",

"\_postman\_exported\_using": "Postman/7.21.1"

}

### 数据接口

{

"info": {

"\_postman\_id": "019caa1f-7d92-4754-bed9-447163171c8c",

"name": "htcloud",

"schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json"

},

"item": [

{

"name": "springgatewary",

"item": [

{

#### "name": "网关登录",

"request": {

"method": "POST",

"header": [

{

"key": "Content-Type",

"value": "application/x-www-form-urlencoded"

}

],

"body": {

"mode": "urlencoded",

"urlencoded": [

{

"key": "username",

"value": "admin",

"type": "text"

},

{

"key": "password",

"value": "123456",

"type": "text"

}

]

},

"url": {

"raw": "{{homeUrl}}/auth/login",

"host": [

"{{homeUrl}}"

],

"path": [

"auth",

"login"

]

}

},

"response": []

},

{

"name": "获取登录用户信息",

"request": {

"method": "GET",

"header": [

{

"key": "Authorization",

"value": "basic YWRtaW46OGZkYmU1ZjQ2MGIyYmRmM2I5ZWY5YzAwNmY0YjIzODAlYTFiMmMwazNkNHk4JWQzYWM1YzdmNmNlYWIwZTYyNWZkYjUzYmI5NzcwOTI5",

"disabled": true

},

{

"key": "userType",

"value": "weixin-xgsixteen",

"disabled": true

}

],

"url": {

"raw": "{{homeUrl}}/{{group}}SPRING-GATEWAY/user/info/authentication",

"host": [

"{{homeUrl}}"

],

"path": [

"{{group}}SPRING-GATEWAY",

"user",

"info",

"authentication"

],

"query": [

{

"key": "",

"value": ""

}

]

}

},

"response": []

},

{

#### "name": "用户退出",

"request": {

"method": "POST",

"header": [

{

"key": "Content-Type",

"value": "application/json"

}

],

"body": {

"mode": "raw",

"raw": ""

},

"url": {

"raw": "{{homeUrl}}/auth/logout",

"host": [

"{{homeUrl}}"

],

"path": [

"auth",

"logout"

]

}

},

"response": []

},

{

"name": "查看路由详情",

"protocolProfileBehavior": {

"disableBodyPruning": true

},

"request": {

"method": "GET",

"header": [

{

"key": "Content-Type",

"name": "Content-Type",

"value": "application/json",

"type": "text"

},

{

"key": "Authorization",

"value": "basic YWRtaW46ODAyNThiMmM2NTMwODJhYTAyZDc3NjEwZWMwMGY4ZjUlYTFiMmMwazNkNHk4JWU5MDBjNWJmMjI1MzU4ZDA5NDY0NDU2YTFhODM1ZDk2",

"disabled": true

}

],

"body": {

"mode": "raw",

"raw": "{\n \"jobClass\": \"org.cloud.scheduler.test.TestJob\",\n \"jobName\": \"TestJob\",\n \"jobGroupName\": \"org.cloud.scheduler.test\",\n \"jobTime\": \"3600aaafdf\",\n \"jobTimes\": -1,\n \"jobDataStr\": \"{\\\"description\\\":\\\"System auto create!\\\"}\",\n \"jobData\": {\n \"description\": \"System auto create!\"\n }\n}",

"options": {

"raw": {

"language": "json"

}

}

},

"url": {

"raw": "{{homeUrl}}/monitor/gateway/routes",

"host": [

"{{homeUrl}}"

],

"path": [

"monitor",

"gateway",

"routes"

]

}

},

"response": []

}

],

"protocolProfileBehavior": {}

},

{

"name": "COMMON-SERVICE",

"item": [

{

"name": "http://{{homeUrl}}:8085/{{group}}COMMON-SERVICE/monitor/env",

"request": {

"method": "GET",

"header": [

{

"key": "Authorization",

"value": "basic YWRtaW46OTYwMmM0MjBkMzA3MzQ4Mjg0YjU1ZmZiODZlMzJhNDI6OjE1NzYxMzc3OTc2NzQ="

}

],

"url": {

"raw": "http://{{homeUrl}}:8085/{{group}}COMMON-SERVICE/monitor/env",

"protocol": "http",

"host": [

"{{homeUrl}}"

],

"port": "8085",

"path": [

"{{group}}COMMON-SERVICE",

"monitor",

"env"

]

}

},

"response": []

},

{

"name": "http://{{homeUrl}}:8085/{{group}}COMMON-SERVICE/userinfo/getUserByName?userName=admin",

"request": {

"method": "POST",

"header": [

{

"key": "Authorization",

"value": "basic YWRtaW46OTYwMmM0MjBkMzA3MzQ4Mjg0YjU1ZmZiODZlMzJhNDI6OjE1NzYxMzc3OTc2NzQ=",

"disabled": true

},

{

"key": "Content-Type",

"name": "Content-Type",

"value": "application/json",

"type": "text"

}

],

"body": {

"mode": "raw",

"raw": "{\n\t\"userName\":\"admin\"\n}",

"options": {

"raw": {

"language": "json"

}

}

},

"url": {

"raw": "{{homeUrl}}/{{group}}COMMON-SERVICE/userinfo/getUserByName",

"host": [

"{{homeUrl}}"

],

"path": [

"{{group}}COMMON-SERVICE",

"userinfo",

"getUserByName"

]

}

},

"response": []

},

{

"name": "网关登录(微信)",

"request": {

"method": "POST",

"header": [

{

"key": "Content-Type",

"value": "application/x-www-form-urlencoded"

},

{

"key": "microAppindex",

"value": "0"

},

{

"key": "userType",

"value": "",

"disabled": true

}

],

"body": {

"mode": "urlencoded",

"urlencoded": [

{

"key": "username",

"value": "admin",

"type": "text",

"disabled": true

},

{

"key": "password",

"value": "023nfzb70t5sVB1SuWc70SHub70nfzb8",

"type": "text"

}

]

},

"url": {

"raw": "{{homeUrl}}/auth/login",

"host": [

"{{homeUrl}}"

],

"path": [

"auth",

"login"

]

}

},

"response": []

},

{

"name": "权限同步",

"request": {

"method": "POST",

"header": [

{

"key": "Content-Type",

"name": "Content-Type",

"value": "application/json",

"type": "text"

},

{

"key": "userType",

"value": "",

"disabled": true

}

],

"body": {

"mode": "raw",

"raw": "{\n\n}",

"options": {

"raw": {

"language": "json"

}

}

},

"url": {

"raw": "{{homeUrl}}/{{group}}COMMON-SERVICE/system/resource/register/all",

"host": [

"{{homeUrl}}"

],

"path": [

"{{group}}COMMON-SERVICE",

"system",

"resource",

"register",

"all"

]

}

},

"response": []

}

],

"protocolProfileBehavior": {}

},

{

"name": "guanmaiapi",

"item": [

{

#### "name": "系统资源注册",

"request": {

"method": "GET",

"header": [],

"url": {

"raw": "{{homeUrl}}/{{group}}XGSIXTEEN/system/resource/register/all",

"host": [

"{{homeUrl}}"

],

"path": [

"{{group}}XGSIXTEEN",

"system",

"resource",

"register",

"all"

]

}

},

"response": []

},

{

"name": "增加物流单号",

"request": {

"method": "POST",

"header": [

{

"key": "Content-Type",

"name": "Content-Type",

"value": "application/json",

"type": "text"

},

{

"key": "Authorization",

"value": "basic YWRtaW46ZWUyY2JhZmQzMzVjMDBkYWMzNDI3ZGRkMGRkYTM5MGUlYTFiMmMwazNkNHk4JWU0ZWJkYzE1NGYzZDk2ODhiMDgzOTU5ZTkwYzNiYjBl",

"type": "text"

}

],

"body": {

"mode": "raw",

"raw": "{\"orderId\":262,\"logisticsNumber\":\"12321312123\"}",

"options": {

"raw": {

"language": "json"

}

}

},

"url": {

"raw": "{{homeUrl}}/{{group}}XGSIXTEEN/order/insertOrderLogisticsNumber",

"host": [

"{{homeUrl}}"

],

"path": [

"{{group}}XGSIXTEEN",

"order",

"insertOrderLogisticsNumber"

]

}

},

"response": []

},

{

#### "name": "动态SQL查询分页",

"request": {

"method": "POST",

"header": [

{

"key": "Content-Type",

"value": "application/json"

},

{

"key": "Authorization",

"value": "basic YWRtaW46ODM1MzI0OWY0ZjM1OGNiMGFiMjNjOGZjNGQ2NWRjMmMlYTFiMmMwazNkNHk4JTk1ZWZhNWM2YWNhMjc5NTlhYWIxOTExYmNjOTYwNjY1",

"disabled": true

}

],

"body": {

"mode": "raw",

"raw": "{\n \"params\": {\n \t\n },\n \"sorts\": \"transaction\_id desc\"\n}"

},

"url": {

"raw": "{{homeUrl}}/{{group}}XGSIXTEEN/common/dynamic/sql/paged/5e69f269543ee56a391f218d/1/5",

"host": [

"{{homeUrl}}"

],

"path": [

"{{group}}XGSIXTEEN",

"common",

"dynamic",

"sql",

"paged",

"5e69f269543ee56a391f218d",

"1",

"5"

]

}

},

"response": []

}

],

"protocolProfileBehavior": {}

},

{

#### "name": "获取环境变量",

"request": {

"method": "GET",

"header": [

{

"key": "Authorization",

"value": "basic YWRtaW46YTRmZjhlOWEwMWFiZDgzMjA2YmIxMDMxNTgwYTVlYzNhZG5mNTZkczM0ZmoyMzJsbno4MWg3YTkxZDVmNDNhMzQzbjN6YjBnYWFkbWluW2JiN2E4ZjBmXSA=",

"disabled": true

}

],

"url": {

"raw": "{{homeUrl}}/LONGYOUXGSIXTEEN/monitor/env",

"host": [

"{{homeUrl}}"

],

"path": [

"LONGYOUXGSIXTEEN",

"monitor",

"env"

]

},

"description": "获取配置文件"

},

"response": []

}

],

"protocolProfileBehavior": {}

}

{

"info": {

"\_postman\_id": "f4ef5fa8-5a3c-47c1-86ed-43a3fdffd460",

"name": "common-service",

"schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json"

},

"item": [

{

#### "name": "系统资源注册",

"request": {

"method": "GET",

"header": [

{

"key": "Authorization",

"value": "basic YWRtaW46OTYzOTkzMjVmYjA2ZmVhNDMxMzZjM2I5MTU2M2RhODUlYTFiMmMwazNkNHk4JWFkbWluWzE2ZmU1MzViXSA=",

"type": "text",

"disabled": true

}

],

"url": {

"raw": "{{homeUrl}}/{{group}}COMMON-SERVICE/system/resource/register/all",

"host": [

"{{homeUrl}}"

],

"path": [

"{{group}}COMMON-SERVICE",

"system",

"resource",

"register",

"all"

]

}

},

"response": []

},

{

#### "name": "查询接口列表",

"request": {

"method": "POST",

"header": [

{

"key": "Content-Type",

"value": "application/json"

},

{

"key": "Authorization",

"value": "basic YWRtaW46NDY0ZWMzZDE0MGU4ZjY2Y2Y3MDA0NTU1NjRjMTI4MWQlYTFiMmMwazNkNHk4JWU3OTA4MjA2YmY4MGNjZDlmMGJjZThlZjA5ZDI2YmJl"

}

],

"body": {

"mode": "raw",

"raw": "{}"

},

"url": {

"raw": "{{homeUrl}}/{{group}}COMMON-SERVICE/common/config/datainterface/page/list/1/15",

"host": [

"{{homeUrl}}"

],

"path": [

"{{group}}COMMON-SERVICE",

"common",

"config",

"datainterface",

"page",

"list",

"1",

"15"

]

}

},

"response": []

},

{

#### "name": "查询接口参数列表",

"request": {

"method": "POST",

"header": [

{

"key": "Content-Type",

"value": "application/json"

},

{

"key": "Authorization",

"value": "basic YWRtaW46NTAyODZkY2FkNGU4ZmNkMWNjN2NiMzA4M2JjMzkyNDQlYTFiMmMwazNkNHk4JTJmNDQwOGU0YzZjMzE4ZDE4ZDEwMmRhZmJiMTExYzli"

}

],

"body": {

"mode": "raw",

"raw": "{}"

},

"url": {

"raw": "{{homeUrl}}/{{group}}COMMON-SERVICE/common/config/datainterface/page/list/1/15",

"host": [

"{{homeUrl}}"

],

"path": [

"{{group}}COMMON-SERVICE",

"common",

"config",

"datainterface",

"page",

"list",

"1",

"15"

]

}

},

"response": []

},

{

#### "name": "查询所有定时任务",

"protocolProfileBehavior": {

"disableBodyPruning": true

},

"request": {

"method": "GET",

"header": [

{

"key": "Content-Type",

"name": "Content-Type",

"value": "application/json",

"type": "text"

}

],

"body": {

"mode": "raw",

"raw": "",

"options": {

"raw": {

"language": "json"

}

}

},

"url": {

"raw": "{{homeUrl}}/{{group}}COMMON-SERVICE/quartz/job/all",

"host": [

"{{homeUrl}}"

],

"path": [

"{{group}}COMMON-SERVICE",

"quartz",

"job",

"all"

]

}

},

"response": []

},

{

#### "name": "查询所有运行的定时任务",

"protocolProfileBehavior": {

"disableBodyPruning": true

},

"request": {

"method": "GET",

"header": [

{

"key": "Content-Type",

"name": "Content-Type",

"value": "application/json",

"type": "text"

}

],

"body": {

"mode": "raw",

"raw": "",

"options": {

"raw": {

"language": "json"

}

}

},

"url": {

"raw": "{{homeUrl}}/{{group}}COMMON-SERVICE/quartz/job/allRunJob",

"host": [

"{{homeUrl}}"

],

"path": [

"{{group}}COMMON-SERVICE",

"quartz",

"job",

"allRunJob"

]

}

},

"response": []

},

{

#### "name": "创建或更新定时任务",

"request": {

"method": "POST",

"header": [

{

"key": "Content-Type",

"name": "Content-Type",

"value": "application/json",

"type": "text"

},

{

"key": "Authorization",

"value": "basic YWRtaW46M2ZiMWM3MjEwZWM2MzJjODQzZGYxNDM4ZThmMDA0ZjQlYTFiMmMwazNkNHk4JWU5NWUxZWM2ZjllNWMxMWIwNzIyYjM4MDdlMzYxNTFm",

"disabled": true

}

],

"body": {

"mode": "raw",

"raw": " {\r\n \"jobClass\": \"org.cloud.scheduler.test.TestJob\",\r\n \"jobName\": \"TestJob\",\r\n \"jobGroupName\": \"org.cloud.scheduler.test\",\r\n \"jobTime\": \"50001123fddsf\",\r\n \"jobData\": {\r\n \"description\": \"System auto create!\"\r\n }\r\n }",

"options": {

"raw": {

"language": "json"

}

}

},

"url": {

"raw": "{{homeUrl}}/{{group}}COMMON-SERVICE/quartz/job/createOrUpdate",

"host": [

"{{homeUrl}}"

],

"path": [

"{{group}}COMMON-SERVICE",

"quartz",

"job",

"createOrUpdate"

]

}

},

"response": []

},

{

#### "name": "暂停定时任务",

"request": {

"method": "POST",

"header": [

{

"key": "Content-Type",

"name": "Content-Type",

"value": "application/json",

"type": "text"

},

{

"key": "userType",

"value": "",

"disabled": true

}

],

"body": {

"mode": "raw",

"raw": "{\n \"jobName\": \"TestJob\",\n \"jobGroupName\": \"org.cloud.scheduler.test\"\n}",

"options": {

"raw": {

"language": "json"

}

}

},

"url": {

"raw": "{{homeUrl}}/{{group}}COMMON-SERVICE/quartz/job/pause",

"host": [

"{{homeUrl}}"

],

"path": [

"{{group}}COMMON-SERVICE",

"quartz",

"job",

"pause"

]

}

},

"response": []

},

{

#### "name": "启动定时任务",

"request": {

"method": "POST",

"header": [

{

"key": "Content-Type",

"name": "Content-Type",

"value": "application/json",

"type": "text"

},

{

"key": "userType",

"value": "",

"disabled": true

}

],

"body": {

"mode": "raw",

"raw": "{\n \"jobName\": \"TestJob\",\n \"jobGroupName\": \"org.cloud.scheduler.test\"\n}",

"options": {

"raw": {

"language": "json"

}

}

},

"url": {

"raw": "{{homeUrl}}/{{group}}COMMON-SERVICE/quartz/job/resume",

"host": [

"{{homeUrl}}"

],

"path": [

"{{group}}COMMON-SERVICE",

"quartz",

"job",

"resume"

]

}

},

"response": []

},

{

"name": "暂停一个定时任务的job",

"request": {

"method": "POST",

"header": [

{

"key": "Content-Type",

"name": "Content-Type",

"value": "application/json",

"type": "text"

},

{

"key": "userType",

"value": "",

"disabled": true

}

],

"body": {

"mode": "raw",

"raw": "{\n \"jobName\": \"TestJob\",\n \"jobGroupName\": \"org.cloud.scheduler.test\"\n}",

"options": {

"raw": {

"language": "json"

}

}

},

"url": {

"raw": "{{homeUrl}}/{{group}}COMMON-SERVICE/quartz/job/pauseJob",

"host": [

"{{homeUrl}}"

],

"path": [

"{{group}}COMMON-SERVICE",

"quartz",

"job",

"pauseJob"

]

}

},

"response": []

},

{

#### "name": "恢复一个定时任务的job",

"request": {

"method": "POST",

"header": [

{

"key": "Content-Type",

"name": "Content-Type",

"value": "application/json",

"type": "text"

},

{

"key": "userType",

"value": "",

"disabled": true

}

],

"body": {

"mode": "raw",

"raw": "{\n \"jobName\": \"TestJob\",\n \"jobGroupName\": \"org.cloud.scheduler.test\"\n}",

"options": {

"raw": {

"language": "json"

}

}

},

"url": {

"raw": "{{homeUrl}}/{{group}}COMMON-SERVICE/quartz/job/resumeJob",

"host": [

"{{homeUrl}}"

],

"path": [

"{{group}}COMMON-SERVICE",

"quartz",

"job",

"resumeJob"

]

}

},

"response": []

},

{

#### "name": "立即执行定时任务",

"request": {

"method": "POST",

"header": [

{

"key": "Content-Type",

"name": "Content-Type",

"value": "application/json",

"type": "text"

},

{

"key": "userType",

"value": "",

"disabled": true

}

],

"body": {

"mode": "raw",

"raw": "{\n \"jobName\": \"TestJob\",\n \"jobGroupName\": \"org.cloud.scheduler.test\",\n \"jobData\": {\n \t\"test1\":\"运行时参数！\"\n }\n}",

"options": {

"raw": {

"language": "json"

}

}

},

"url": {

"raw": "{{homeUrl}}/{{group}}COMMON-SERVICE/quartz/job/runNow",

"host": [

"{{homeUrl}}"

],

"path": [

"{{group}}COMMON-SERVICE",

"quartz",

"job",

"runNow"

]

}

},

"response": []

},

{

#### "name": "删除定时任务",

"request": {

"method": "POST",

"header": [

{

"key": "Content-Type",

"name": "Content-Type",

"value": "application/json",

"type": "text"

},

{

"key": "userType",

"value": "",

"disabled": true

}

],

"body": {

"mode": "raw",

"raw": "{\n \"jobName\": \"TestJob\",\n \"jobGroupName\": \"org.cloud.scheduler.test\"\n}",

"options": {

"raw": {

"language": "json"

}

}

},

"url": {

"raw": "{{homeUrl}}/{{group}}COMMON-SERVICE/quartz/job/delete",

"host": [

"{{homeUrl}}"

],

"path": [

"{{group}}COMMON-SERVICE",

"quartz",

"job",

"delete"

]

}

},

"response": []

},

{

#### "name": "查询数据字典列表",

"request": {

"method": "POST",

"header": [

{

"key": "Content-Type",

"value": "application/json"

},

{

"key": "Authorization",

"value": "basic YWRtaW46ODM1MzI0OWY0ZjM1OGNiMGFiMjNjOGZjNGQ2NWRjMmMlYTFiMmMwazNkNHk4JTk1ZWZhNWM2YWNhMjc5NTlhYWIxOTExYmNjOTYwNjY1",

"disabled": true

}

],

"body": {

"mode": "raw",

"raw": "{\n\n\t\n}"

},

"url": {

"raw": "{{homeUrl}}/{{group}}COMMON-SERVICE/admin/dic/page/list/1/15",

"host": [

"{{homeUrl}}"

],

"path": [

"{{group}}COMMON-SERVICE",

"admin",

"dic",

"page",

"list",

"1",

"15"

]

}

},

"response": []

},

{

#### "name": "更新或者创建数据字典",

"request": {

"method": "POST",

"header": [

{

"key": "Content-Type",

"value": "application/json"

},

{

"key": "Authorization",

"value": "basic YWRtaW46ODM1MzI0OWY0ZjM1OGNiMGFiMjNjOGZjNGQ2NWRjMmMlYTFiMmMwazNkNHk4JTk1ZWZhNWM2YWNhMjc5NTlhYWIxOTExYmNjOTYwNjY1",

"disabled": true

}

],

"body": {

"mode": "raw",

"raw": "[\n {\n \"dicMasterId\": 13,\n \"dicCode\": \"systemBaseConfig7\",\n \"dicName\": \"系统基础配置\",\n \"remark\": \"系统基础配置\",\n \"belongMicroService\": \"General\",\n \"parentMasterId\": null,\n \"status\": 1,\n \"createBy\": 1,\n \"createDate\": \"2020-02-22T08:32:24.000+0000\",\n \"updateBy\": 1,\n \"updateDate\": \"2020-02-22T08:32:24.000+0000\",\n \"items\": [\n {\n \"dicItemId\": 12,\n \"dicItemCode\": \"dicItemCode2\",\n \"dicItemName\": \"dicItemName1\",\n \"dicItemValue\": \"dicItemValue1\",\n \"extAttribut1\": \"extAttribut1\",\n \"extAttribut2\": \"extAttribut2\",\n \"status\": 1,\n \"createBy\": 1,\n \"updateBy\": 1\n }\n ]\n }\n]"

},

"url": {

"raw": "{{homeUrl}}/COMMON-SERVICE/admin/dic/saveOrUpdate",

"host": [

"{{homeUrl}}"

],

"path": [

"COMMON-SERVICE",

"admin",

"dic",

"saveOrUpdate"

]

}

},

"response": []

},

{

#### "name": "获取服务列表",

"protocolProfileBehavior": {

"disableBodyPruning": true

},

"request": {

"method": "GET",

"header": [

{

"key": "Content-Type",

"value": "application/json"

},

{

"key": "Authorization",

"value": "basic YWRtaW46ODM1MzI0OWY0ZjM1OGNiMGFiMjNjOGZjNGQ2NWRjMmMlYTFiMmMwazNkNHk4JTk1ZWZhNWM2YWNhMjc5NTlhYWIxOTExYmNjOTYwNjY1",

"disabled": true

}

],

"body": {

"mode": "raw",

"raw": "{\n\t\"dicCode\":\"systemBaseConfig\",\n\t\"dicItemCode\":\"company\"\n}"

},

"url": {

"raw": "{{homeUrl}}/{{group}}COMMON-SERVICE/discovery/client/getServices",

"host": [

"{{homeUrl}}"

],

"path": [

"{{group}}COMMON-SERVICE",

"discovery",

"client",

"getServices"

]

}

},

"response": []

},

{

#### "name": "查看commonservice的EVN",

"request": {

"method": "GET",

"header": [],

"url": {

"raw": "{{homeUrl}}/COMMON-SERVICE/monitor/env",

"host": [

"{{homeUrl}}"

],

"path": [

"COMMON-SERVICE",

"monitor",

"env"

]

}

},

"response": []

},

{

#### "name": "根据dic\_code查询items",

"protocolProfileBehavior": {

"disableBodyPruning": true

},

"request": {

"method": "GET",

"header": [

{

"key": "Content-Type",

"value": "application/json"

},

{

"key": "Authorization",

"value": "basic YWRtaW46ODM1MzI0OWY0ZjM1OGNiMGFiMjNjOGZjNGQ2NWRjMmMlYTFiMmMwazNkNHk4JTk1ZWZhNWM2YWNhMjc5NTlhYWIxOTExYmNjOTYwNjY1",

"disabled": true

}

],

"body": {

"mode": "raw",

"raw": "{\n\n\t\n}"

},

"url": {

"raw": "{{homeUrl}}/{{group}}COMMON-SERVICE/admin/dic/selectDicItemsByDicCode?dicCode=commStatus",

"host": [

"{{homeUrl}}"

],

"path": [

"{{group}}COMMON-SERVICE",

"admin",

"dic",

"selectDicItemsByDicCode"

],

"query": [

{

"key": "dicCode",

"value": "commStatus"

}

]

}

},

"response": []

},

{

#### "name": "根据diccode查询items",

"request": {

"method": "POST",

"header": [

{

"key": "Content-Type",

"value": "application/json"

},

{

"key": "Authorization",

"value": "basic YWRtaW46ODM1MzI0OWY0ZjM1OGNiMGFiMjNjOGZjNGQ2NWRjMmMlYTFiMmMwazNkNHk4JTk1ZWZhNWM2YWNhMjc5NTlhYWIxOTExYmNjOTYwNjY1",

"disabled": true

}

],

"body": {

"mode": "raw",

"raw": "{\n\n\t\n}"

},

"url": {

"raw": "{{homeUrl}}/{{group}}COMMON-SERVICE/admin/dic/page/list/1/15",

"host": [

"{{homeUrl}}"

],

"path": [

"{{group}}COMMON-SERVICE",

"admin",

"dic",

"page",

"list",

"1",

"15"

]

}

},

"response": []

},

{

#### "name": "查询mongodb的文件",

"request": {

"method": "POST",

"header": [

{

"key": "Content-Type",

"value": "application/json"

},

{

"key": "Authorization",

"value": "basic YWRtaW46ODM1MzI0OWY0ZjM1OGNiMGFiMjNjOGZjNGQ2NWRjMmMlYTFiMmMwazNkNHk4JTk1ZWZhNWM2YWNhMjc5NTlhYWIxOTExYmNjOTYwNjY1",

"disabled": true

}

],

"body": {

"mode": "raw",

"raw": "{\n\n\t\"filename\":\"plugin\"\n}"

},

"url": {

"raw": "{{homeUrl}}/COMMON-SERVICE/admin/mongo/gridfs/list/1/23",

"host": [

"{{homeUrl}}"

],

"path": [

"COMMON-SERVICE",

"admin",

"mongo",

"gridfs",

"list",

"1",

"23"

]

}

},

"response": []

},

{

#### "name": "mongodb数据接口查询",

"request": {

"method": "POST",

"header": [

{

"key": "Authorization",

"value": "basic YWRtaW46OTYzOTkzMjVmYjA2ZmVhNDMxMzZjM2I5MTU2M2RhODUlYTFiMmMwazNkNHk4JWFkbWluWzE2ZmU1MzViXSA=",

"type": "text",

"disabled": true

},

{

"key": "Content-Type",

"name": "Content-Type",

"value": "application/json",

"type": "text"

}

],

"body": {

"mode": "raw",

"raw": "{\n \"params\": [\n {\n \"name\": \"interfaceName\",\n \"relationalOperator\": \"AND\",\n \"operator\": \"REGEX\",\n \"value\": \"数据接口\"\n },\n {\n \"name\": \"params.0.fieldName\",\n \"relationalOperator\": \"AND\",\n \"operator\": \"EXISTS\",\n \"value\": true\n },\n {\n \"name\": \"params.2.fieldType\",\n \"relationalOperator\": \"AND\",\n \"operator\": \"REGEX\",\n \"value\": \"Number\"\n }\n ],\n \"orders\": [\n {\n \"direction\": \"DESC\",\n \"property\": \"interfaceName\"\n }\n ]\n}",

"options": {

"raw": {

"language": "json"

}

}

},

"url": {

"raw": "{{homeUrl}}/{{group}}COMMON-SERVICE/admin/mongo/config/dataInterface/list/2/10",

"host": [

"{{homeUrl}}"

],

"path": [

"{{group}}COMMON-SERVICE",

"admin",

"mongo",

"config",

"dataInterface",

"list",

"2",

"10"

]

},

"description": "{\r\n\"params\"[\r\n\t{\"name\":\"interfaceName\",\"relationalOperator\":\"AND\",\"operator\":\"REGEX\",value:\"数据接口\"},\r\n\t{\"name\":\"params.0.fieldName\",\"relationalOperator\":\"AND\",\"operator\":\"EXISTS\",value:true},\r\n\t{\"name\":\"params.2.fieldType\",\"relationalOperator\":\"AND\",\"operator\":\"REGEX\",value:\"Number\"}\r\n],\r\n\"sort\":[\r\n\t{\"direction\":\"DESC\",\"property\":\"interfaceName\"}\r\n]\r\n\r\n}"

},

"response": [

{

#### "name": "mongodb数据接口查询",

"originalRequest": {

"method": "POST",

"header": [

{

"key": "Authorization",

"value": "basic YWRtaW46OTYzOTkzMjVmYjA2ZmVhNDMxMzZjM2I5MTU2M2RhODUlYTFiMmMwazNkNHk4JWFkbWluWzE2ZmU1MzViXSA=",

"type": "text",

"disabled": true

},

{

"key": "Content-Type",

"name": "Content-Type",

"value": "application/json",

"type": "text"

}

],

"body": {

"mode": "raw",

"raw": "{\n \"params\": [\n {\n \"name\": \"interfaceName\",\n \"relationalOperator\": \"AND\",\n \"operator\": \"REGEX\",\n \"value\": \"数据接口\"\n },\n {\n \"name\": \"params.0.fieldName\",\n \"relationalOperator\": \"AND\",\n \"operator\": \"EXISTS\",\n \"value\": true\n },\n {\n \"name\": \"params.2.fieldType\",\n \"relationalOperator\": \"AND\",\n \"operator\": \"REGEX\",\n \"value\": \"Number\"\n }\n ],\n \"sort\": [\n {\n \"direction\": \"DESC\",\n \"property\": \"interfaceName\"\n }\n ]\n}",

"options": {

"raw": {

"language": "json"

}

}

},

"url": {

"raw": "{{homeUrl}}/{{group}}COMMON-SERVICE/admin/mongo/config/dataInterface/list/1/10",

"host": [

"{{homeUrl}}"

],

"path": [

"{{group}}COMMON-SERVICE",

"admin",

"mongo",

"config",

"dataInterface",

"list",

"1",

"10"

]

}

},

"\_postman\_previewlanguage": null,

"header": [],

"cookie": [],

"body": null

}

]

},

{

#### "name": "动态SQL查询分页",

"request": {

"method": "POST",

"header": [

{

"key": "Content-Type",

"value": "application/json"

},

{

"key": "Authorization",

"value": "basic YWRtaW46ODM1MzI0OWY0ZjM1OGNiMGFiMjNjOGZjNGQ2NWRjMmMlYTFiMmMwazNkNHk4JTk1ZWZhNWM2YWNhMjc5NTlhYWIxOTExYmNjOTYwNjY1",

"disabled": true

}

],

"body": {

"mode": "raw",

"raw": "{\n \"params\": {\n \"userName\": null,\n \"sexList\": [\n \"F\",\n \"M\"\n ]\n },\n \"sorts\": \"user\_Name desc\"\n}"

},

"url": {

"raw": "{{homeUrl}}/{{group}}COMMON-SERVICE/common/dynamic/sql/paged/fca0c2236325f81017b12e4da4d32990/1/5",

"host": [

"{{homeUrl}}"

],

"path": [

"{{group}}COMMON-SERVICE",

"common",

"dynamic",

"sql",

"paged",

"fca0c2236325f81017b12e4da4d32990",

"1",

"5"

]

}

},

"response": []

},

{

#### "name": "动态SQL查询不分页",

"request": {

"method": "POST",

"header": [

{

"key": "Content-Type",

"value": "application/json"

},

{

"key": "Authorization",

"value": "basic YWRtaW46ODM1MzI0OWY0ZjM1OGNiMGFiMjNjOGZjNGQ2NWRjMmMlYTFiMmMwazNkNHk4JTk1ZWZhNWM2YWNhMjc5NTlhYWIxOTExYmNjOTYwNjY1",

"disabled": true

}

],

"body": {

"mode": "raw",

"raw": "{\n \"params\": {\n \"userName\": \"teacher\_王五\",\n \"sexList\": [\n \"F\",\n \"M\"\n ]\n },\n \"sorts\": \"user\_Name desc\"\n}"

},

"url": {

"raw": "{{homeUrl}}/{{group}}COMMON-SERVICE/common/dynamic/sql/list/5e69ecb0543ee56a391f1ff9",

"host": [

"{{homeUrl}}"

],

"path": [

"{{group}}COMMON-SERVICE",

"common",

"dynamic",

"sql",

"list",

"5e69ecb0543ee56a391f1ff9"

]

}

},

"response": []

}

],

"protocolProfileBehavior": {}

}