**密级：秘密**

文件编号：D00-000000

腾讯智能制造云平台

概要设计报告

**更改履历**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 版本 | 更改时间 | 更改人 | 更改章节 | 状态 | 更改描述 |
| 1 | V1.0 | 2018-2-27 | 曹飞宇 |  | 新建 |  |
|  |  | 2018-3-7 | 汪伟 |  | 新建 |  |
|  |  |  |  |  |  |  |

状态：新建、增加、修改、删除。

目录

[1 引言 1](#_Toc11139)

[1.1 编制目的 1](#_Toc24786)

[1.2 术语表 1](#_Toc18927)

[1.3 参考资料 1](#_Toc16367)

[2 总体设计 1](#_Toc9368)

[2.1 软件体系结构 1](#_Toc21132)

[2.2 系统技术构架 1](#_Toc25877)

[2.2.1 前端技术架构 1](#_Toc1980)

[2.2.2 后端技术架构 1](#_Toc17076)

[2.3 系统开发环境 2](#_Toc21492)

[3 系统设计 2](#_Toc6385)

[3.1 前端页面设计 2](#_Toc25134)

[3.1.1 简介 2](#_Toc30341)

[3.1.2 公共代码封装 2](#_Toc32587)

[3.2 后端代码设计 2](#_Toc4554)

[3.2.1 公共代码设计 2](#_Toc6469)

[3.2.2 产品管理 3](#_Toc13724)

[3.2.3 设备管理 5](#_Toc14640)

[3.2.4 产品计划管理 7](#_Toc4569)

[3.2.5 生产调度管理 9](#_Toc22051)

[4 代码结构 12](#_Toc29986)

[4.1 主要包的功能说明 12](#_Toc4220)

[4.2 主要功能类说明 12](#_Toc24991)

[5 代码说明 13](#_Toc8212)

[5.1 前端接口类 13](#_Toc15559)

[5.2 查询代码 14](#_Toc21866)

[5.3 添加、修改数据 14](#_Toc6780)

[5.4 删除数据 14](#_Toc19659)

[5.5 上传文件 14](#_Toc26108)

[6 其他说明 15](#_Toc11259)

# 引言

## 编制目的

为开发人员提供明确的需求方案、流程设计、数据库设计、包及类结构时间，以明确开发思想。

## 术语表

|  |  |  |
| --- | --- | --- |
| **词汇名称** | **词汇含义** | **备注** |
|  |  |  |
|  |  |  |
|  |  |  |

## 参考资料

# 总体设计

## 软件体系结构

## 系统技术构架

### 前端技术架构

RequireJs + Jquery + ArtTemplate + BootStrap

### 后端技术架构

Springmvc4.0 + mybatis + mysql

## 系统开发环境

Windows

# 系统设计

## 前端页面设计

### 简介

本系统前端架构采用RequireJs作为模块化工具，Jquery作为整体逻辑显示框架，ArtTemplate作为模板渲染工具，bootStrap 3.0作为Ui框架，另外使用gulp作为项目构建工具在node环境下进行开发、测试以及发布。CSS部分使用Less作为预处理工具。

### 公共代码封装

本系统前端模块公共代码被统一封装至Common.js中，主要包括ajax封装、数据处理方法封装、ArtTemplate封装、文件上传封装、页面整体框架封装、分页模块封装、对话框、确认框、提示框封装。

业务模块加载统一由main.js处理，开发中只需要在对应业务模块目录下新建同名业务模块html、js、less文件即可实现页面渲染。

## 后端代码设计

### 公共代码设计

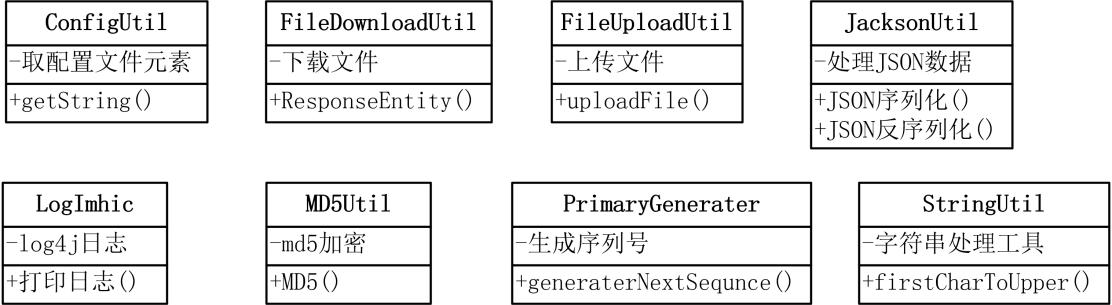
#### 功能说明

工具类所在包com.neuedu.tools提供一些系统的工具类使用

包括json转换、文件上传、生成序列号、获取系统配置等

|  |  |
| --- | --- |
| 类名 | 功能 |
| ConfigUtil | 取配置文件内容 |
| FileDownloadUtil | 文件下载 |
| FileUploadUtil | 文件上传 |
| JacksonUtil | JSON字符处理 |
| LogImhic | LOG4J日志 |
| MD5Util | 字符串MD5加密 |
| PrimaryGenerater | 生产序列号 |
| StringUtil | 字符串工具 |

#### 类图

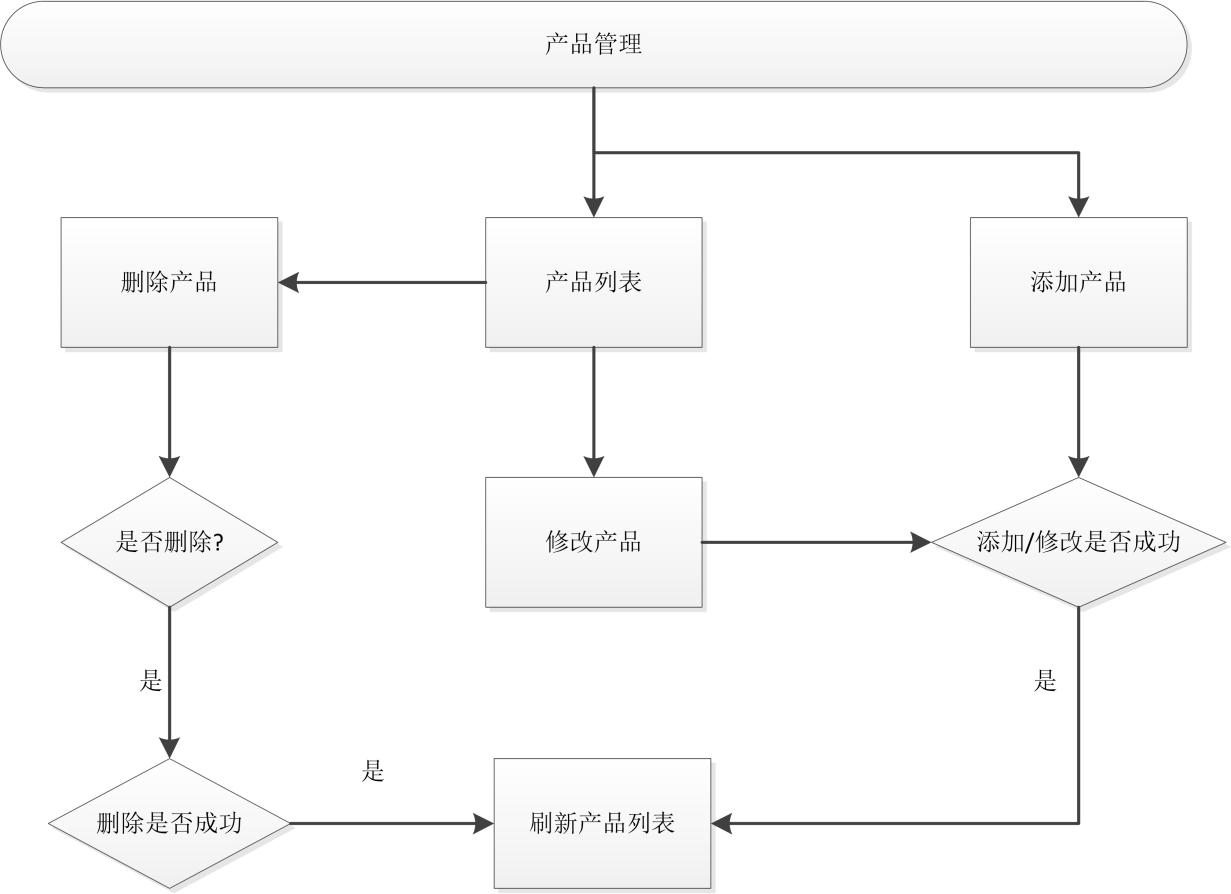


### 产品管理

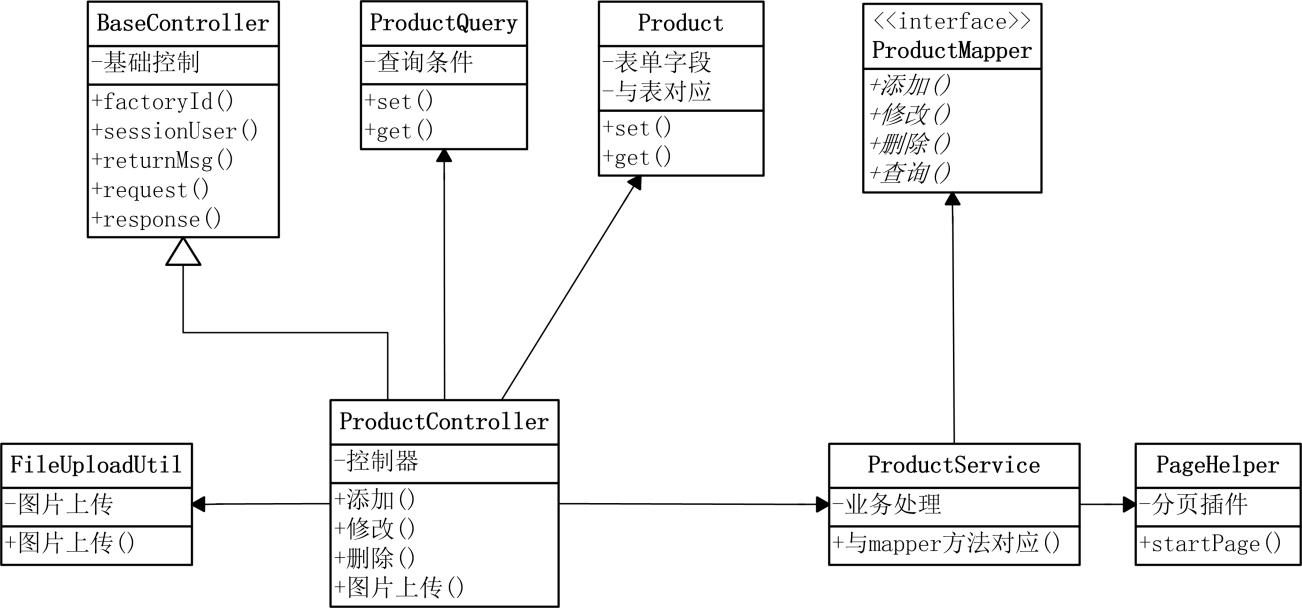
#### 功能说明

实现产品的图片上传、添加、修改、删除，查询列表功能

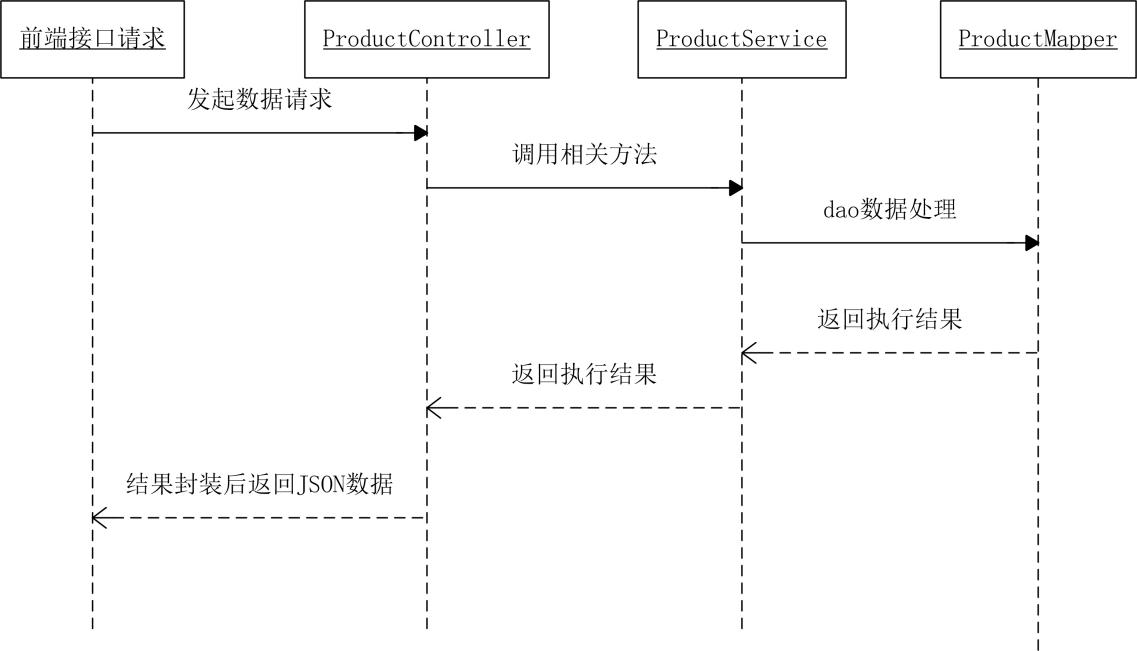
#### 业务流程图



#### 类图



#### 时序图



#### 相关数据表

T\_product表

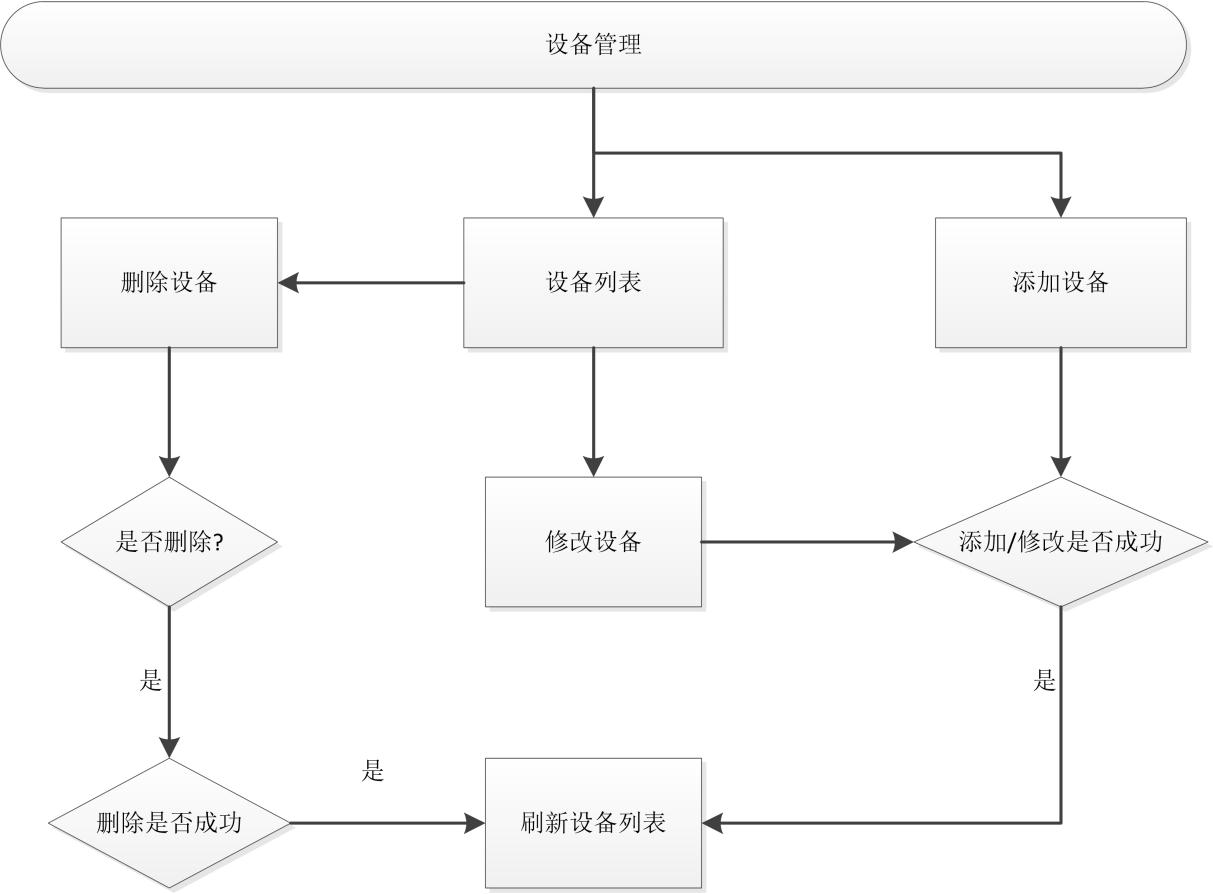
|  |  |  |
| --- | --- | --- |
| Field | Type | Comment |
| id | int(11) NOT NULL | ID标识 |
| flag | int(11) NULL | 有效标识 0：有效 1：无效 |
| create\_time | datetime NULL | 创建时间 |
| create\_userid | int(11) NULL | 创建人ID |
| update\_time | datetime NULL | 修改时间 |
| update\_userid | int(11) NULL | 修改人ID |
| product\_num | varchar(200) NOT NULL | 产品编号 |
| product\_name | varchar(300) NOT NULL | 产品名称 |
| product\_img\_url | varchar(300) NULL | 产品图片所在目录 |
| factory\_id | int(11) NOT NULL | 工厂ID |

### 设备管理

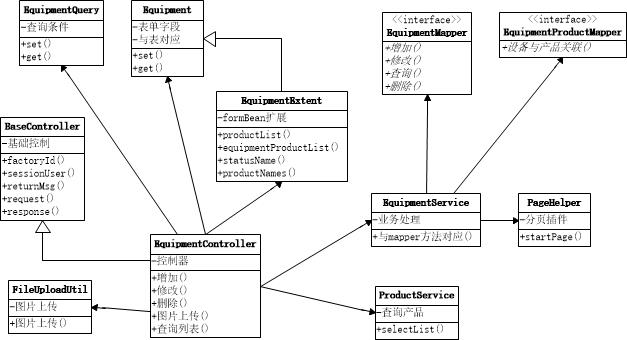
#### 功能说明

实现设备的图片上传、添加、修改、删除，查询列表功能

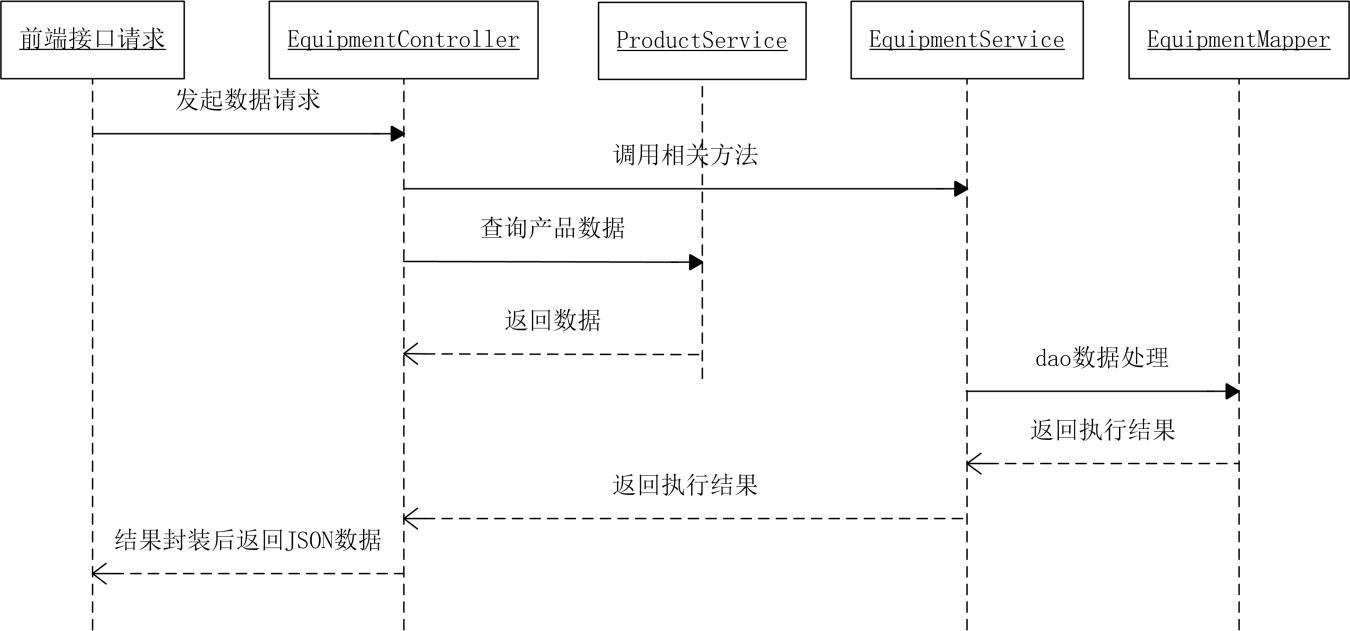
#### 业务流程图



#### 类图



#### 时序图



#### 相关数据表

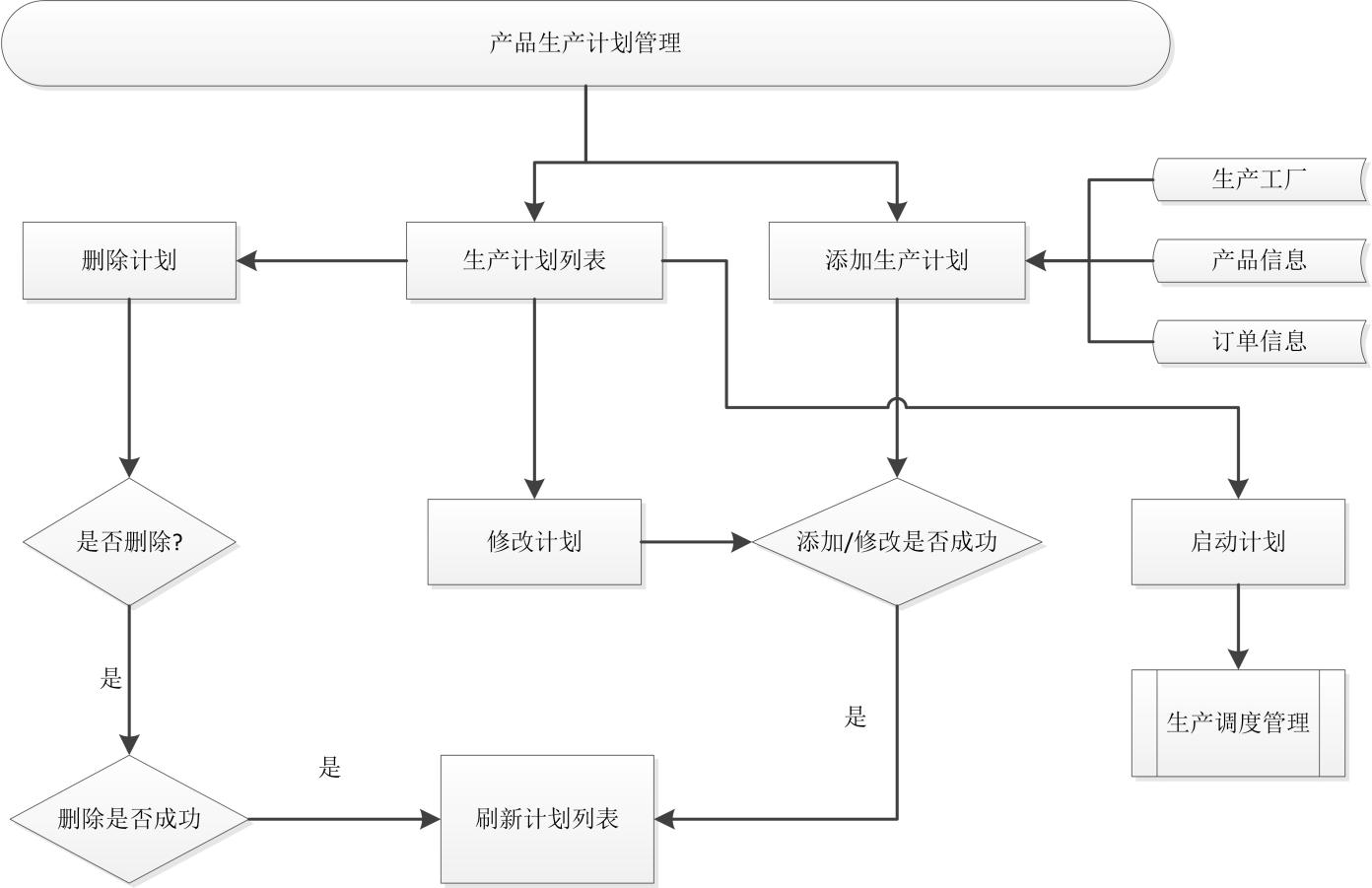
|  |  |  |
| --- | --- | --- |
| Field | Type | Comment |
| id | int(11) NOT NULL | ID标识 |
| flag | int(11) NULL | 有效标识 0：有效 1：失效 |
| create\_time | datetime NULL | 创建时间 |
| create\_userid | int(11) NULL | 创建人ID |
| update\_time | datetime NULL | 修改时间 |
| update\_userid | int(11) NULL | 修改人ID |
| equipment\_seq | varchar(200) NOT NULL | 设备序号 |
| equipment\_name | varchar(300) NULL | 设备名称 |
| equipment\_img\_url | varchar(300) NULL | 设备图片 |
| equipment\_status | int(11) NULL | 设备状态 10：启用 20：停用 30：故障 |
| factory\_id | int(11) NOT NULL | 工厂ID |

### 产品计划管理

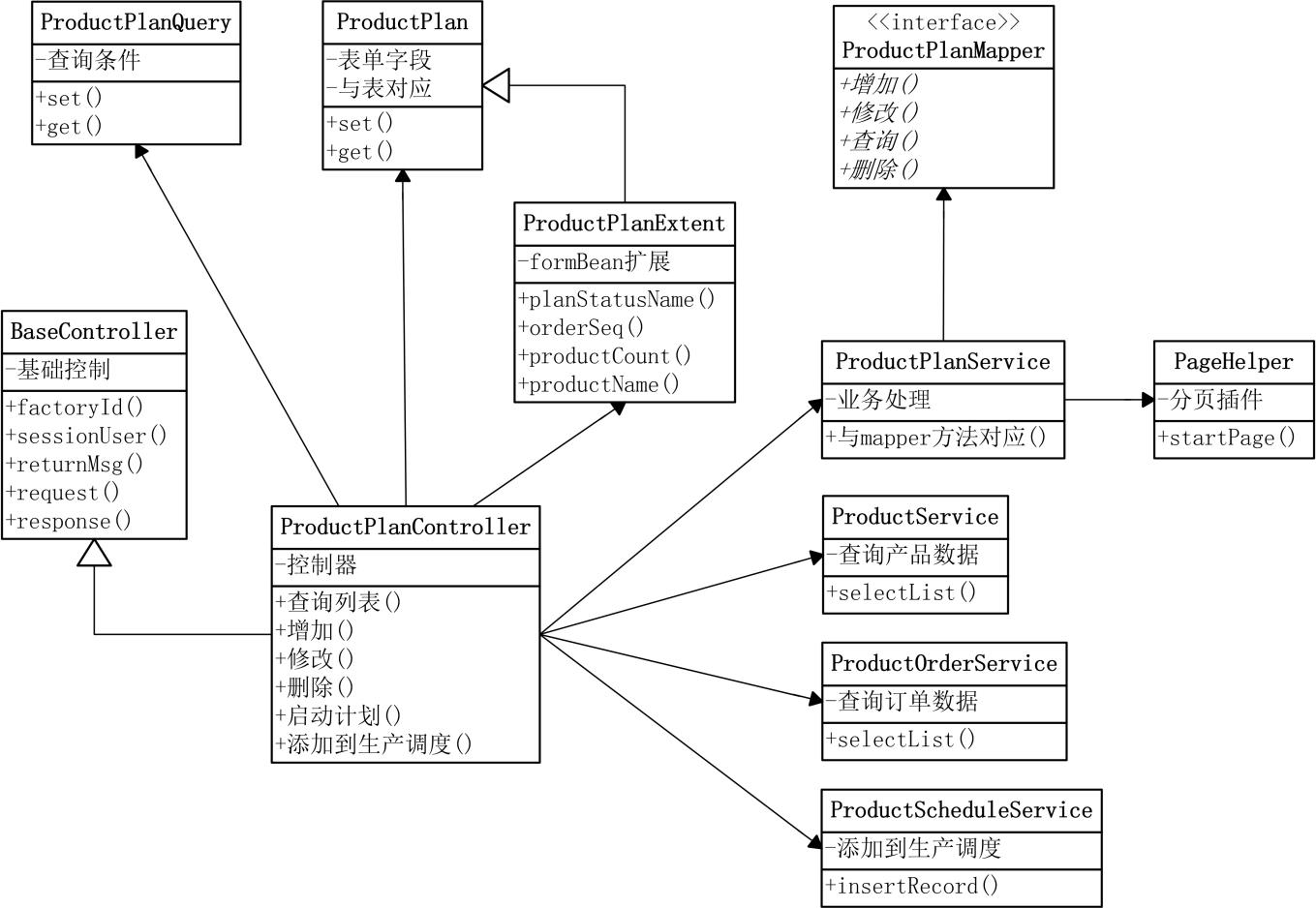
#### 功能说明

实现产品计划的添加、修改、删除，查询列表功能

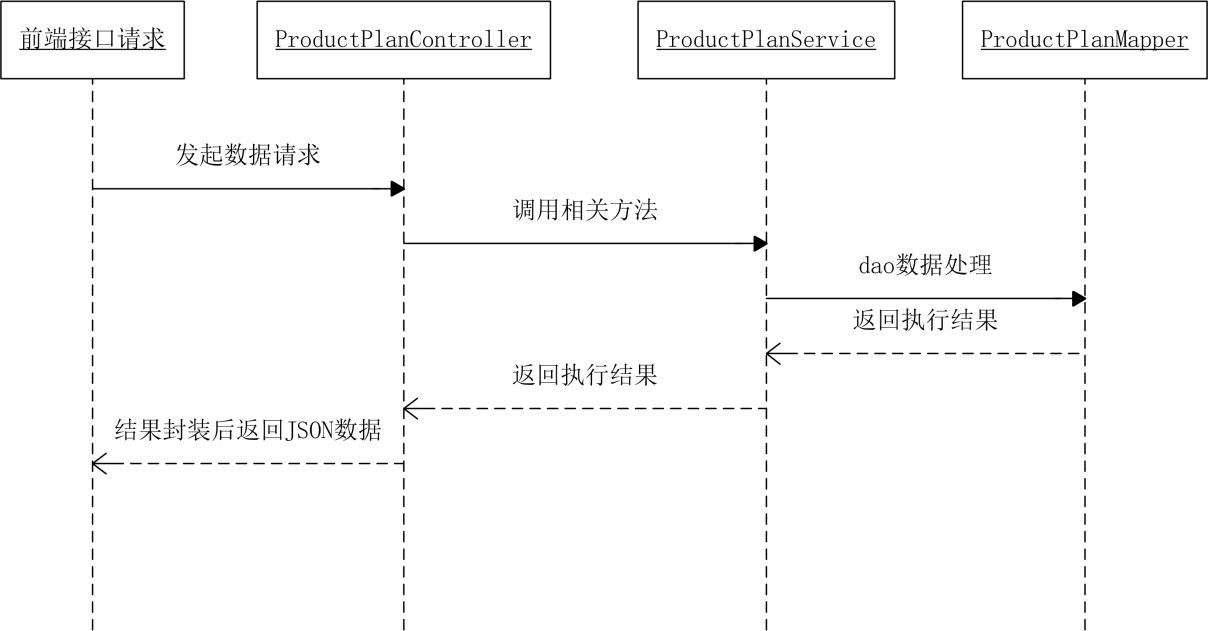
#### 业务流程图



#### 类图



#### 时序图



#### 相关数据表

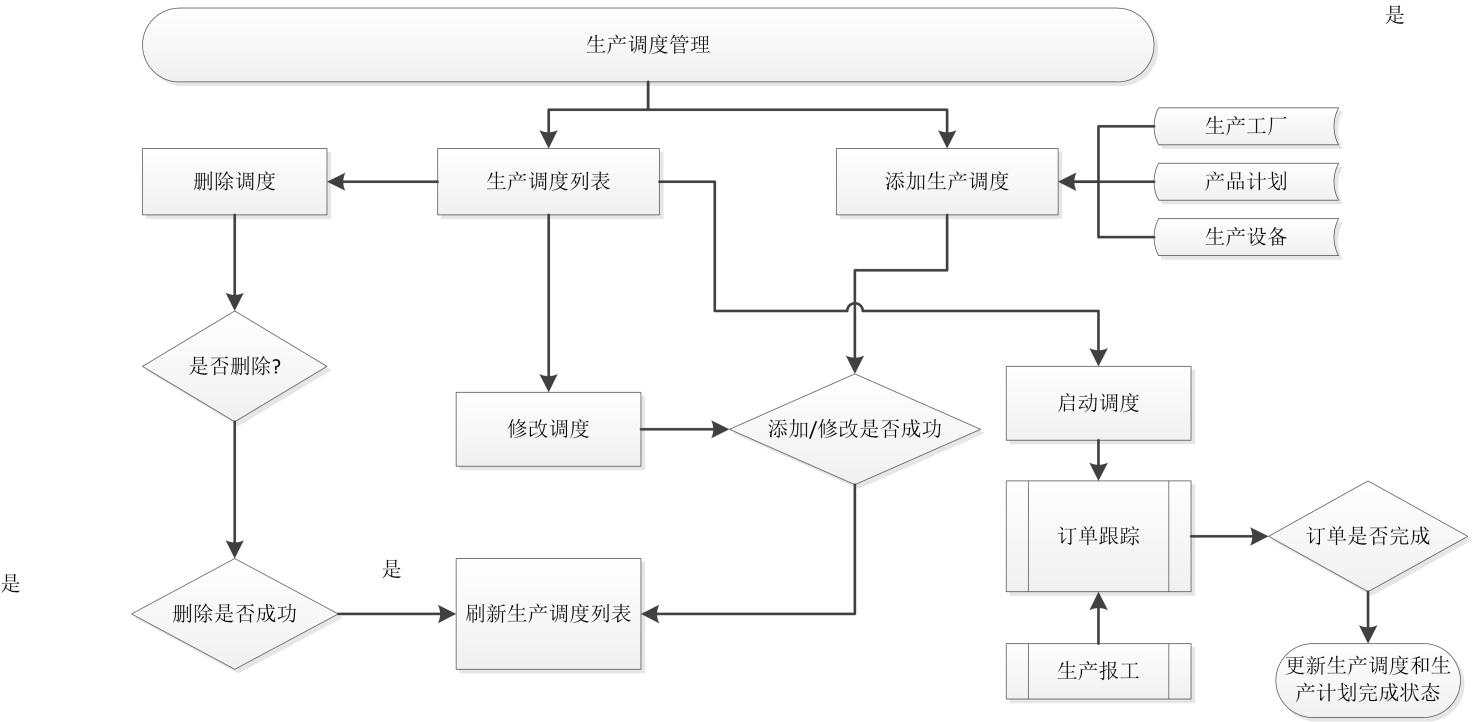
|  |  |  |
| --- | --- | --- |
| Field | Type | Comment |
| id | int(11) NOT NULL | ID标识 |
| flag | int(11) NULL | 有效标识 0：有效 1：无效 |
| create\_time | datetime NULL | 创建时间 |
| create\_userid | int(11) NULL | 创建人ID |
| update\_time | datetime NULL | 修改时间 |
| update\_userid | int(11) NULL | 修改人ID |
| plan\_seq | varchar(200) NULL | 计划编号 |
| order\_id | int(11) NOT NULL | 订单ID |
| product\_id | int(11) NOT NULL | 产品ID |
| plan\_count | int(11) NULL | 计划数量 |
| delivery\_date | date NULL | 交货日期 |
| plan\_start\_date | date NULL | 计划开始日期 |
| plan\_end\_date | date NULL | 计划结束日期 |
| plan\_status | int(11) NULL | 计划状态 10：未启动 20：已启动 30：已完成 |
| factory\_id | int(11) NOT NULL | 工厂ID |

### 生产调度管理

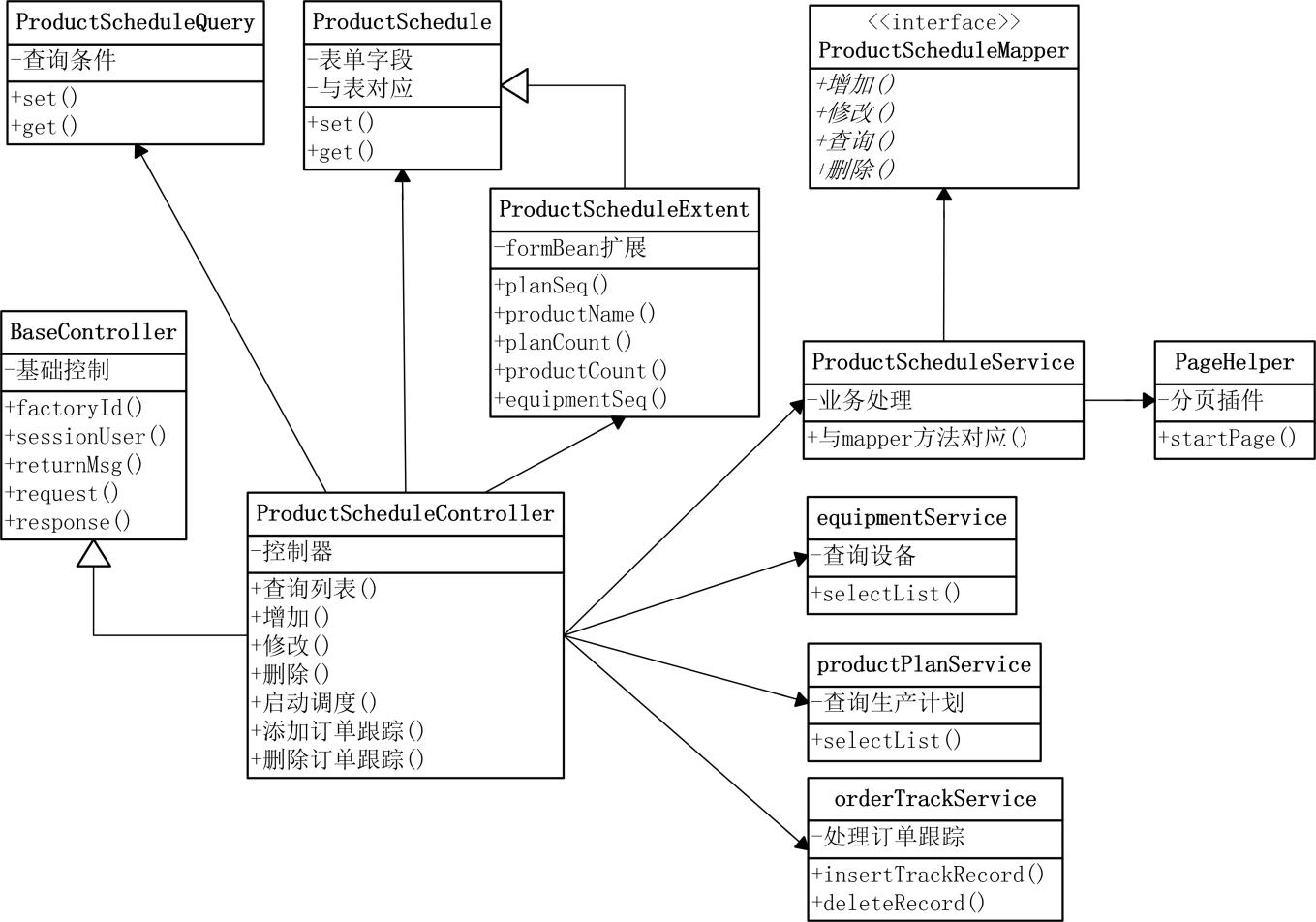
#### 功能说明

实现生产调度的添加、修改、删除，查询列表功能

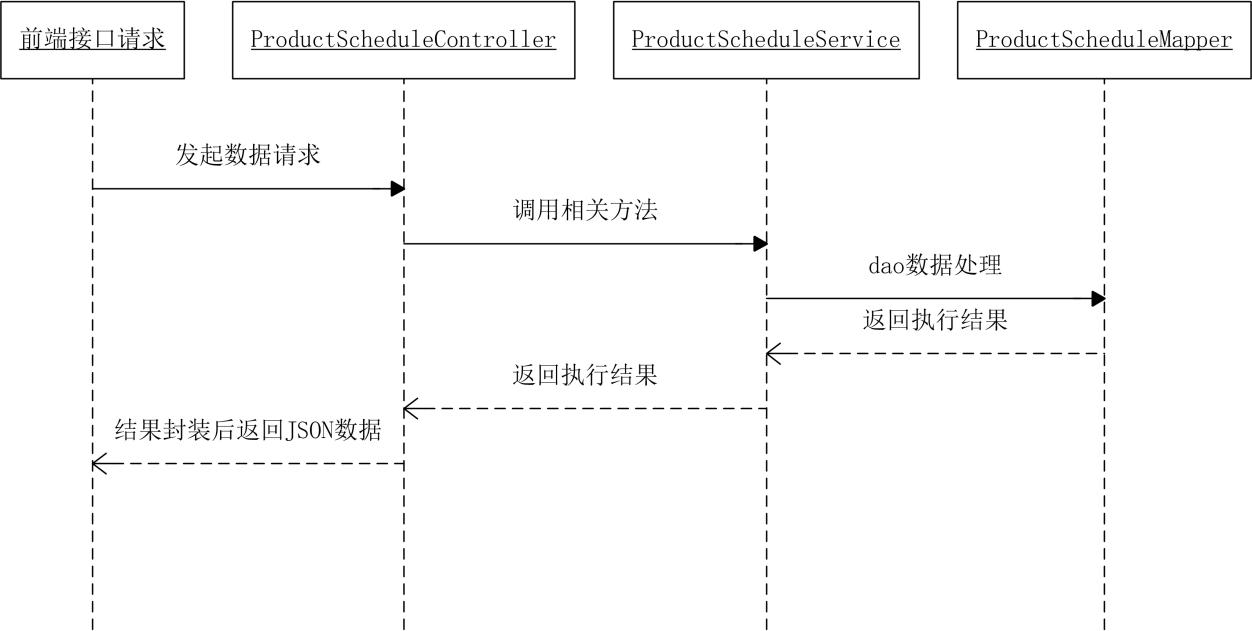
#### 业务流程图



#### 类图



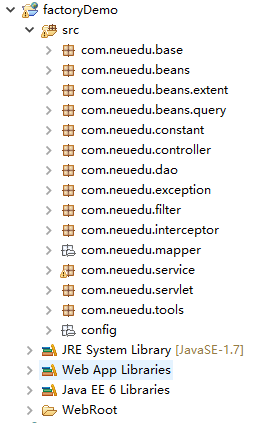
#### 时序图



#### 相关数据表

|  |  |  |
| --- | --- | --- |
| Field | Type | Comment |
| id | int(11) NOT NULL | ID标识 |
| flag | int(11) NULL | 有效标识 0：有效 1：无效 |
| create\_time | datetime NULL | 创建时间 |
| create\_userid | int(11) NULL | 创建人ID |
| update\_time | datetime NULL | 修改时间 |
| update\_userid | int(11) NULL | 修改人ID |
| schedule\_seq | varchar(200) NULL | 工单编号 |
| schedule\_count | int(11) NULL | 工单数量 |
| schedule\_status | int(11) NOT NULL | 工单状态 10：未开始 20：生产中 30：已完成 |
| plan\_id | int(11) NOT NULL | 计划ID |
| product\_id | int(11) NOT NULL | 产品ID |
| equipment\_id | int(11) NOT NULL | 设备ID |
| start\_date | date NULL | 开始日期 |
| end\_date | date NULL | 结束日期 |
| factory\_id | int(11) NOT NULL | 工厂ID |

# 代码结构



## 主要包的功能说明

base包中放置一些基础的功能类

extent包中的类继承beans包中实体bean用于扩展查询使用

query包中的类是查询时使用包括定义一些基本的查询条件和分页信息

constant包的中类定义一些常量类，如状态值的说明

tools包中定义一些工具类

config为配置文件所在目录

## 主要功能类说明

BaseController控制器父类主要设置当前用户与工厂id

BaseQuery查询query的父类主要设置分页数据

PageObject前端页面接口类

ConfigUtil 读取baseconfig.properties文件内容

FileUploadUtil 文件上传

JacksonUtil json转换

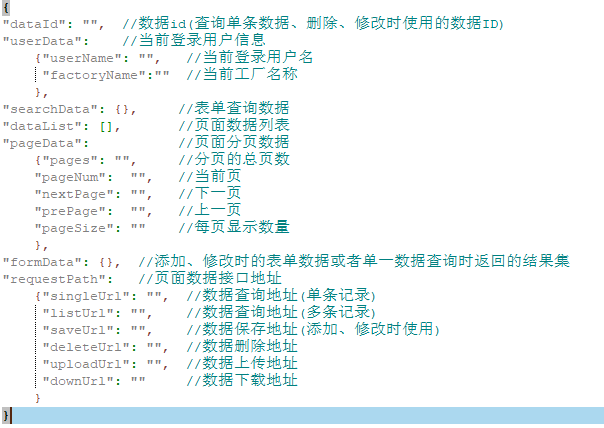
PrimaryGenerater 生成序列号

# 代码说明

## 前端接口类

com.neuedu.base.PageObject

主要定义一些前端接口数据以Map方式实现最终生成一组json数据



## 查询代码

查询时的条件要全部在query对象中定义，query对象会继承BaseQuery用于分页查询

前端要将查询的表单内容封装成一组json字符串提交到后台用query对象接收，

数据不能为空，如果查询条件为空要提交一个空的json对象”{}”

Service以query对象为查询条件进行分页或不分页查询

返回前台如果是多条记录就放到PageObject中的dataList中

如果只是查询是单条件记录可以放到PageObject中的formData中主要是编辑时使用,最后将PageObject对象转成json格式返回前端

## 添加、修改数据

前端将表单数据封装成json提交到后端以后用bean对象接收，调用Service方法以后

判断id是否为空，如果为空就执行添加操作，否则执行修改操作,将结果返回前端

json格式为{“status”:””,”msg”:””}

## 删除数据

前端将要删除的数据id封装成json格式{“id”:””}提交到后端，用bean对象接收调用Service删 除方法，将删除结果返回前端

json格式为{“status”:””,”msg”:””}

## 上传文件

上传文件使用com.neuedu.tools.FileUploadUtil类

前端异步提交以后调用此方法并将文件保存的地址返回到前端

json格式为{“status”:””,”msg”:”文件相对地址”}

# 其他说明

1、目前后端代码只接收post方式提交的数据，并将json对象转为json字符串提交

2、在拦截器com.neuedu.interceptor.HandlerInterceptorAdapter中做了模拟登录的操作并使用了session方式，后期需要更改实现方式，比如用cookie

3、用户登录后并设置工厂id，所有操作基本条件最是以工厂ID为基础