**IUAP平台使用指南**

**单表示例**

用友网络科技股份有限公司

二○一八年五月

**此文档为保密文档，未经用友软件股份有限公司书面同意，不得向任何单位或个人提供、转让本文档中的任何内容，用友软件股份有限公司将保留对泄漏文档内容的起诉权利。**

**IUAP平台使用指南**

|  |  |  |
| --- | --- | --- |
| 文件状态：  [ ] 草稿  [**√**]正在修改  [] 正式发布 | 文件标识： |  |
| 当前版本： |  |
| 作者： |  |
| 完成日期： |  |

版本历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 作者 | 参与者 | 起止日期 | 备注 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

审核记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 姓名 | 职位 | 审阅日期 | 备注 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目 录

[第一章概述 1](#_Toc515624816)

[1.1业务场景 1](#_Toc515624817)

[1.2开发方案 1](#_Toc515624818)

[第二章示例配置 2](#_Toc515624819)

[2.1开发环境准备 2](#_Toc515624820)

[2.2数据库准备 2](#_Toc515624821)

[2.3与IUAP集成 3](#_Toc515624822)

[2.3.1创建角色 3](#_Toc515624823)

[2.3.2创建用户 3](#_Toc515624824)

[2.3.3用户分配角色 4](#_Toc515624825)

[2.3.4注册业务节点 4](#_Toc515624826)

[2.3.5业务节点挂接菜单 5](#_Toc515624827)

[2.3.6业务节点授权给角色 5](#_Toc515624828)

[2.4功能验证 6](#_Toc515624829)

[2.4.1数据展示测试 6](#_Toc515624830)

[2.4.2新增记录测试 6](#_Toc515624831)

[2.4.3修改记录测试 6](#_Toc515624832)

[2.4.4删除记录测试 7](#_Toc515624833)

[2.4.5搜索记录测试 7](#_Toc515624834)

[第三章代码解析 8](#_Toc515624835)

[3.1代码结构 8](#_Toc515624836)

[3.2后台代码解析 8](#_Toc515624837)

[3.2.1 Controller常见注解 9](#_Toc515624838)

[3.2.2查询方法 10](#_Toc515624839)

[3.2.3保存 11](#_Toc515624840)

[3.2.4删除 13](#_Toc515624841)

[3.3前台代码解析 15](#_Toc515624842)

[3.3.1 Dictionary.html解析 15](#_Toc515624843)

[3.3.2 viewModel.js 16](#_Toc515624844)

[3.3.3 Dictionary.js解析 16](#_Toc515624845)

# 概述

## 业务场景

在进行业务功能开发过程中，最典型、最简单的就是针对单表的增、删、改、查操作。本示例将以档案管理为例，演示如何与IUAP进行集成。

## 开发方案

### 示例场景设计

对人员档案进行管理，实现增、删、改、查功能。

### 示例库表设计

设计档案管理业务数据表“example\_dictionary”，表结构如下。

表1档案表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 名称 | 类型 | 主键 | 备注 |
| Id | 主键 | Varchar | 是 | Uuid |
| Code | 编码 | Varchar |  | 非空，唯一索引 |
| Name | 名称 | Varchar |  | 可空 |
| sys | 系统预置 | Varchar |  | 可空 |
| creator | 创建人 | Varchar |  | 可空 |
| create\_time | 创建时间 | Varchar |  | 可空 |
| Remark | 备注 | Varchar |  | 可空 |

### 示例功能设计

* 记录展示（分页展示）
* 增加记录
* 修改记录
* 删除记录（单行删除、多行删除）
* 查询与筛选

# 示例配置

## 开发环境准备

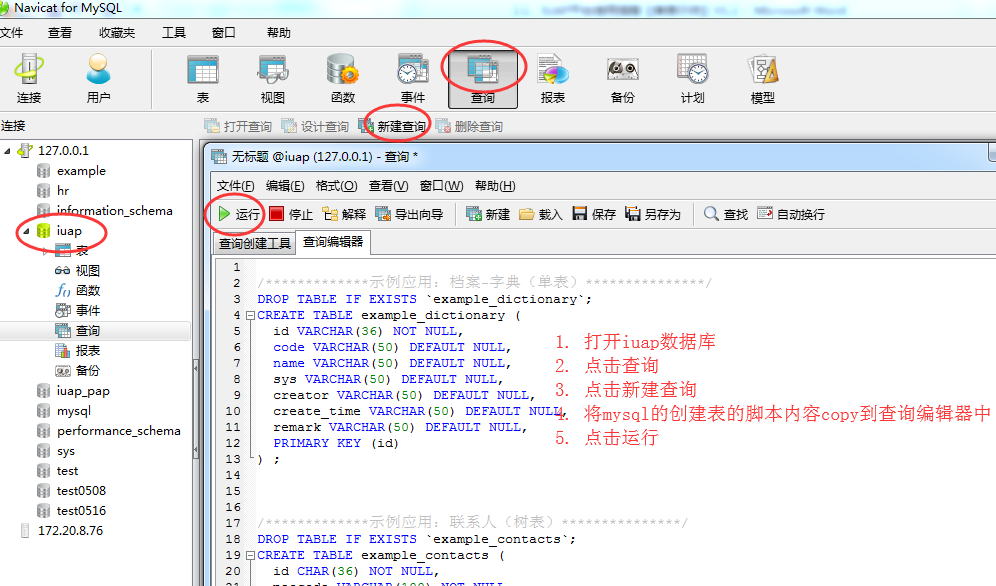
此示例的代码在dictionary文件夹下，运行示例时需要把“dictionary”文件夹下的“java”和“resources”文件夹添加到编译中。

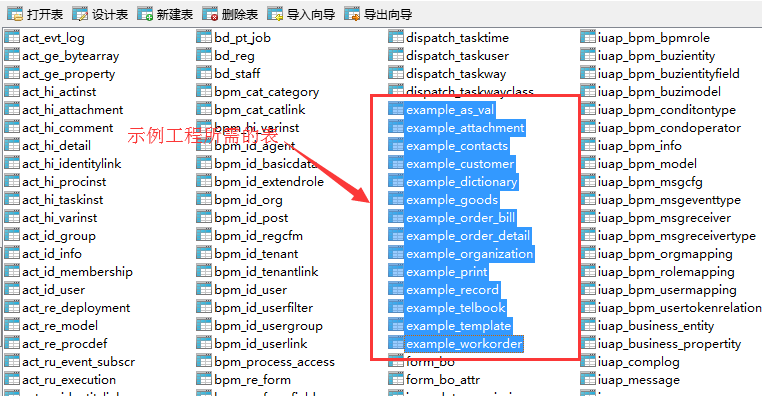


图1后台工程结构

## 数据库准备

示例提供了mysql和oracle两种创建脚本来创建示例工程所需要的表，在sql目录下。所有的表都是以example\_作为前缀的，需要在iuap的数据库中运行脚本。现在以mysql数据库为例，采用navicat工具导入数据库表。





## 功能测试

### 菜单注册

手工进行菜单注册，为标准方法，不重复介绍。

### 功能验证

注销管理员用户，并使用user1进行登录，默认密码是123456a，首次登录需要修改默认密码，登录成功后在左侧菜单“示例节点”下打开“档案2”。

#### 数据展示测试



图8数据展示

#### 新增记录测试



图9新增测试

#### 修改记录测试

****

图10修改测试

#### 删除记录测试

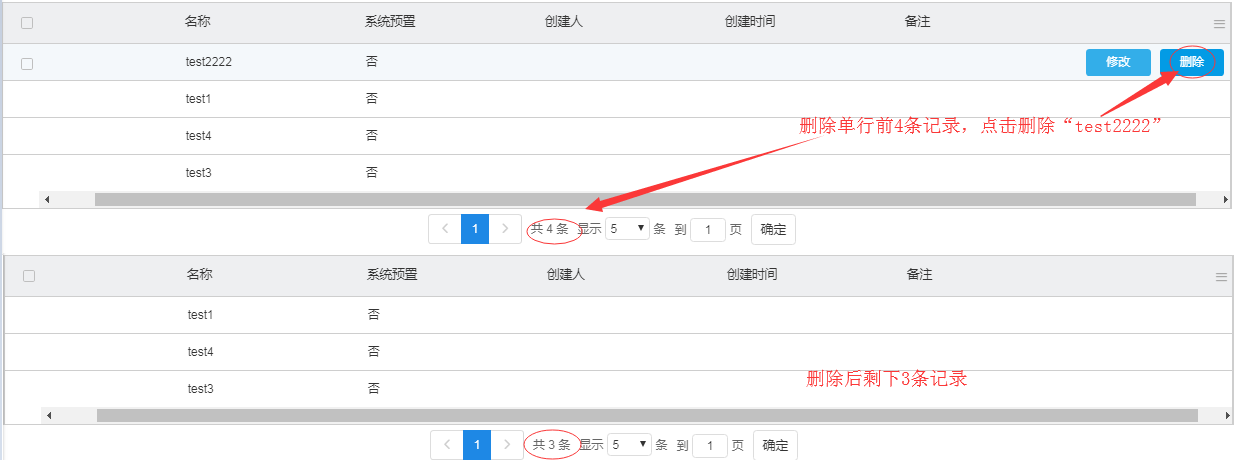


图11删除单行记录



图12批量删除记录

#### 搜索记录测试



图13搜索记录

# 代码解析

## 代码结构

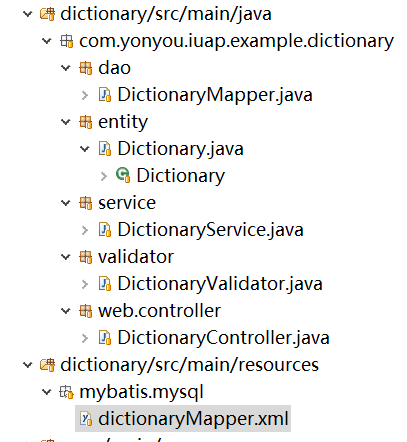


图14代码结构

Dictionary.java: 实体

DictionaryMapper.java 持久层dao

DictionaryService.java 服务层Service

DictionaryController.java web层Controller

dictionaryMapper.xml.xml映射文件

## 后台代码解析

### Controller常见注解

Controller层代码DictionaryController.java类

DictionaryController继承了iuap提供的BaseController基类：

com.yonyou.iuap.example.web.BaseController【提供了各种错误、成功提示信息包装】

* Controller类上必须使用@RestController或者@Controller注解、@RequestMapping注解用户地址映射
* @RestController：继承自@Controller注解，Spring MVC的组件都使用@Controller来标识当前类是一个控制器servlet。使用这个特性，我们可以开发REST服务的时候不需要使用@Controller而专门的@RestController。
* @RequestMapping：是一个用来处理请求地址映射的注解，可用于类或方法上。用于类上，表示类中的所有响应请求的方法都是以该地址作为父路径1、 value， method；value： 指定请求的实际地址，指定的地址可以是URI Template 模式；method：  指定请求的method类型， GET、POST、PUT、DELETE等；

在web.xml中，对所有的路径进行过滤，或者拦截，不拦截的包括 \*.mp3  /  \*.js    /js/  ，等，这些都属于请求静态资源，并不需要访问controller。

针对请求后台的就会进行拦截。然后找到匹配的controller 中的方法进行处理

* @Autowired：用在JavaBean中的注解，通过byType形式，用来给指定的字段或方法注入所需的外部资源。
* @ResponseBody:用于将Controller的方法返回的对象，通过适当的

HttpMessageConverter转换为指定格式后，写入到Response对象的body数据区。【此处可以将我们的实体转为前端的Jason对象，用于前端数据显示，或者从前端的jason对象转为我们的实体】；

获取前台传值到后台的方法，可以用httpservletrequest，更好的方法，是放到一个参数类中。可以直接用参数类来进行钱后台数据交互，会根据你类的字段名，进行自动匹配。

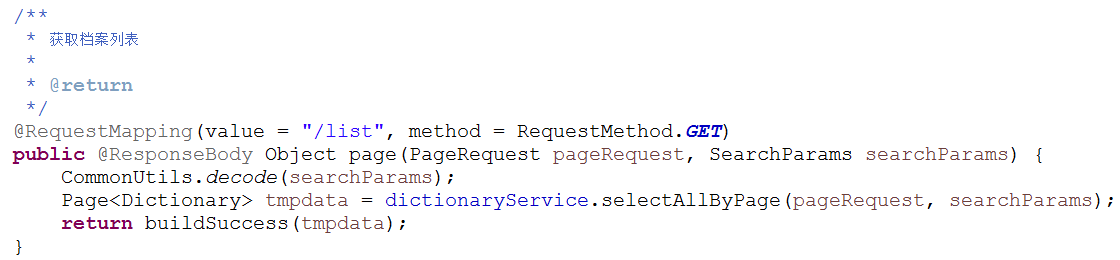
### 查询方法

打开档案页面，默认查询页面数据

**1) DictionaryController**

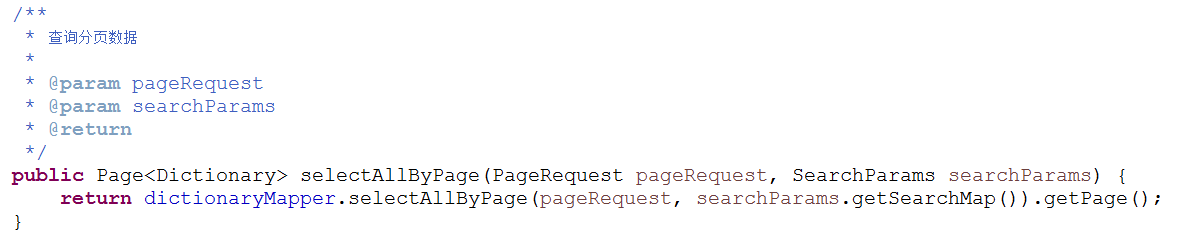
根据前台传入的参数，构造后台查询参数

PageRequest：pageNumber页数，pagesize页面数据大小，Sort排序



**2）DictionaryService**

调用Dao方法查询

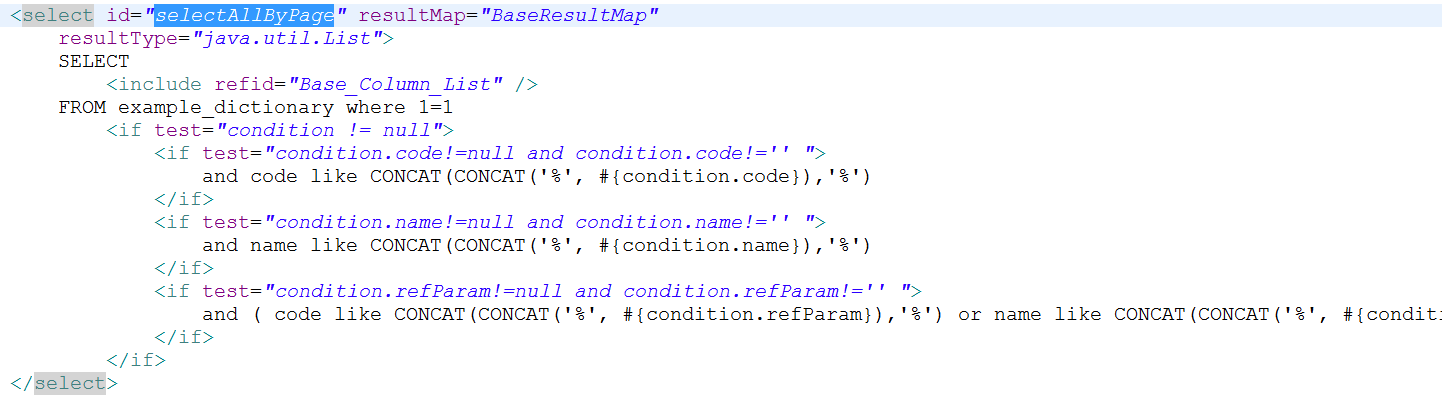


**3）**DictionaryMapper

接口方法定义



**4）**映射文件xml



iuap-mybatis组件会加载插件处理分页参数，并封装结果集返回

在applicationContext-persistence.xml配置文件中如下所示：

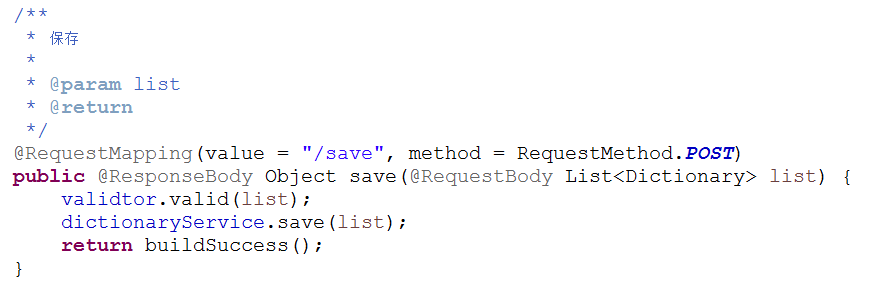
注意在方法名要以selectAllByPage结尾



### 保存

**1) DictionaryController**

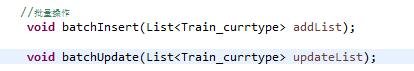
传入的参数为list结果集



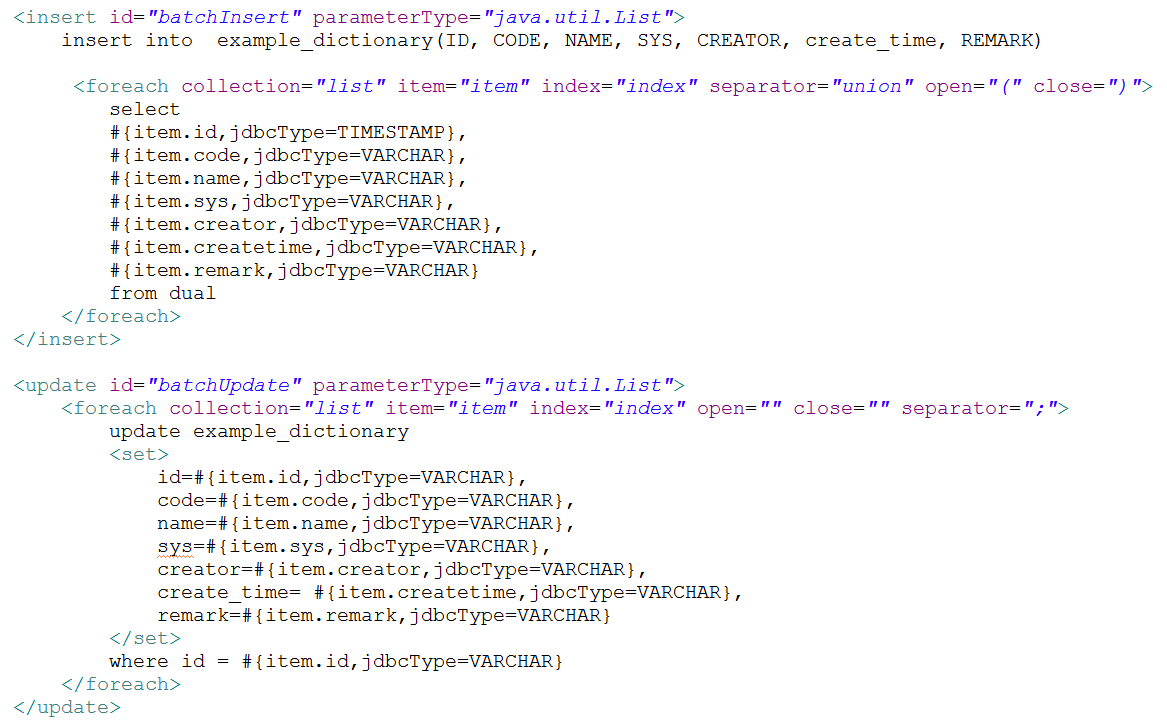
2）DictionaryService



3)DictionaryMapper



4）映射文件xml



说明：<foreach>标签实现sql条件的循环，可完成类似批量的sql

<foreach>标签的用法：

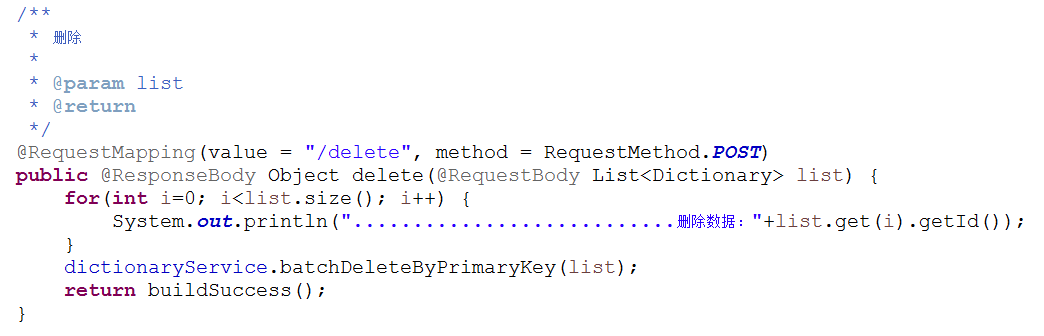
六个参数：

* collection：要循环的集合
* index：循环索引
* item：集合中的一个元素（item和collection，按foreach循环理解）
* open：以什么开始
* close：以什么结束
* separator：循环内容之间以什么分隔

### 删除

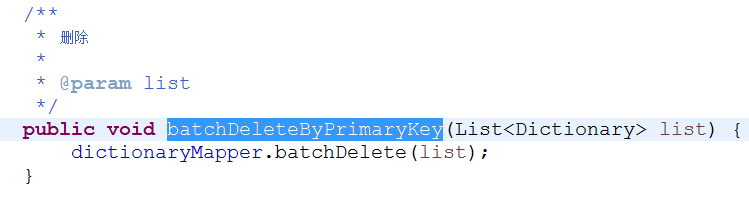
**1) DictionaryController.del**

传入的参数为被删除的档案实体数组，同时调用后台删除



**2）DictionaryService**

调用mapper的删除方法

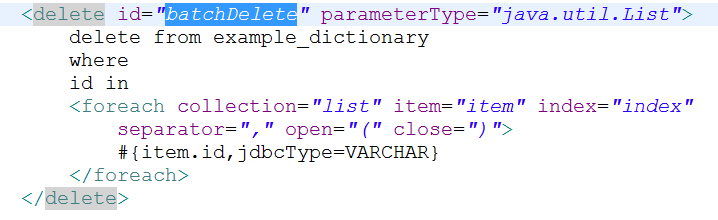


**3）DictionaryMapper**

批量删除方法



**4）映射文件xml**



## 前台代码解析

### Dictionary.html解析

Html页面的具体内容和布局方式可以根据实际业务场景设计来确定，本示例设计html内容包含4个主要部分。

1. 标题区域：描述业务信息，本示例为档案管理。
2. 搜索区域：按编码和名称搜索。
3. 列表区域：显示业务记录对象。
4. 表单区域：新增或者编辑记录时候显示，默认隐藏。

图15搜索区域

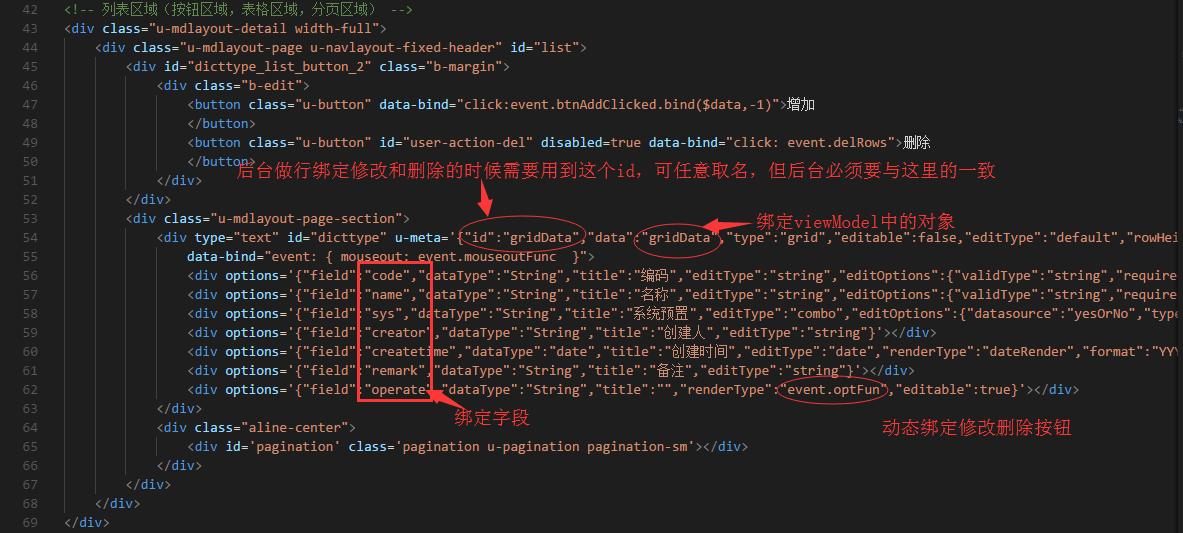


图16表格区域

### viewModel.js

向导生成数据模型，字段默认类型为String



图17数据模型

### Dictionary.js解析

使用requirejs的方式引用，对应的html、js、css。其中Dictionary.html,Dictionary.css,viewModel.js是业务对应的资源，其他为公共资源，如果公用的样式无法满足页面调整需求，可以在对应的业务css文件中（如：Dictionary.css）做修改。

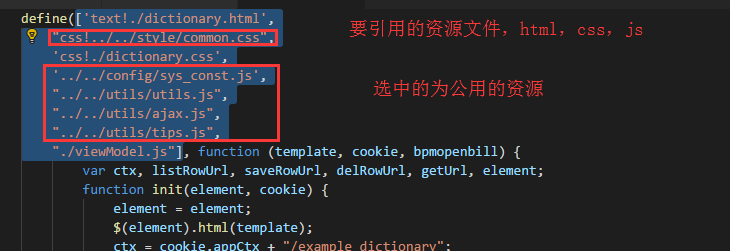


图18引用

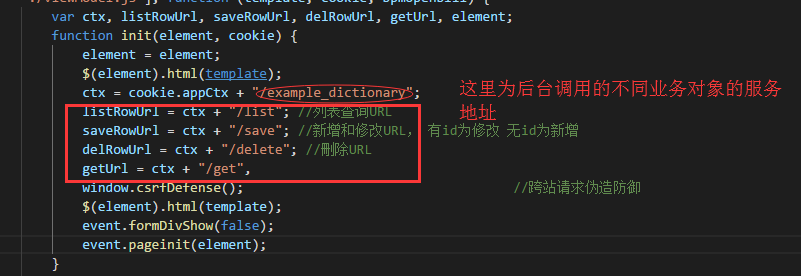


图19请求地址

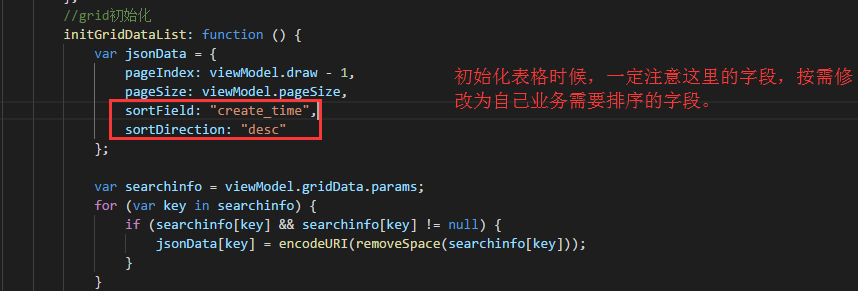


图20排序字段