**IUAP平台使用指南**

**编码规则组件**

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 2](#_Toc515714519)

[1.1 业务场景 2](#_Toc515714520)

[1.2 开发方案 2](#_Toc515714521)

[第二章 环境准备 3](#_Toc515714522)

[2.1 开发环境配置 3](#_Toc515714523)

[2.1.1 设置编译路径 3](#_Toc515714524)

[2.1.2 设置编码规则地址 3](#_Toc515714525)

[2.2 数据库脚本配置 4](#_Toc515714526)

[2.2.1 创建业务表 4](#_Toc515714527)

[2.2.2 注册编码对象 4](#_Toc515714528)

[2.2.3 注册编码规则定义 5](#_Toc515714529)

[2.3 功能测试 7](#_Toc515714530)

[2.3.1 菜单注册 7](#_Toc515714531)

[2.3.2 功能测试 7](#_Toc515714532)

[第三章 代码开发示例 8](#_Toc515714533)

[3.1 获取单据号 8](#_Toc515714534)

[3.2 单据号退号 9](#_Toc515714535)

[第四章 其它参考 11](#_Toc515714536)

[4.1 编码规则RestAPI 11](#_Toc515714537)

# 概述

## 业务场景

编码规则使用了应用组件iuap-billcode

组件功能：根据编码规则和一些必要信息，生成业务对象编码。组件需要的编码规则可以存储在文件中，甚至是手工构造，或者存储在数据库中。本组件提供了存储在数据库中一种实现，提供了基本的增加，删除，修改编码规则的服务方法，用户可以在此调用此服务，开发出自己的编码规则设计工具，也可以直接设计符合自己要求的编码规则数据库结构和服务，编码规则VO符合组件的模型即可（实现组件的规则接口）。

另外，组件提供了数据库行锁和zookeeper分布式锁的支持，以应对大并发的情况下可能出现的编码重复问题。

## 开发方案

编码规则的获取相对来讲是比较独立的业务，所以从业务上可以把它单独分包。这样如果在获取量非常大的情况下，可以给编码规则单独部署增加服务达到性能优化的途径，同时使用zookeeper分布式锁, 以应对大并发的情况下可能出现的编码重复问题。

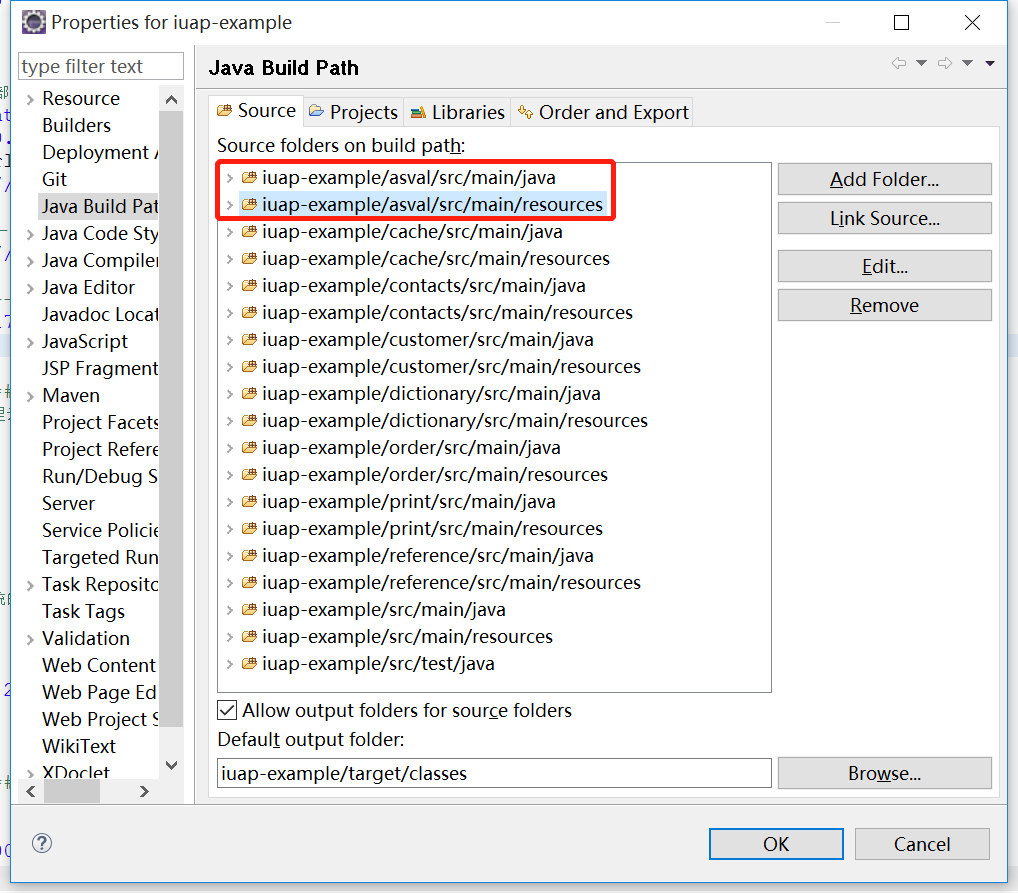
新版本中使用RestAPI方式调用并获取编码规则。

# 环境准备

## 开发环境配置

### 设置编译路径

编码规则使用asVal作为编译路径



1. 设置编译路径

### 设置编码规则地址

配置项目中src/resources/目录下的application.properties文件，配置编码规则提供的服务地址。

如果使用了本机的编码规则服务，则如下配置

billcodeservice.base.url=http://127.0.0.1:8080/iuap-saas-billcode-service

如果应用中心部署在远程服务器中,则将IP地址(127.0.0.1)和端口号(8080)分别修改为服务器上的地址,如下配置:

billcodeservice.base.url=http://127.0.0.1:8080/iuap-saas-billcode-service

## 数据库脚本配置

### 创建业务表

创建数据库表example\_as\_val

### 注册编码对象

编码实体手工注册到数据库表pub\_bcr\_obj中；

Pub\_bcr\_obj中的数据为应用中心-管理中心-编码规则定义左侧的树节点。

如果pub\_bcr\_obj表中无数据，则执行如下语句：

MySQL:

|  |
| --- |
| INSERT INTO `pub\_bcr\_obj`  (`pk\_billobj`, `code`, `name`, `pk\_parent`, `iscatalog`, `assigntype`, `createdate`, `sysid`, `tenantid`)  VALUES  ('000000100000000ROOT1', 'root', 'root', NULL, 'Y', NULL, NOW(), 'wbalone', 'tenant');  INSERT INTO `pub\_bcr\_obj`  (`pk\_billobj`, `code`, `name`, `pk\_parent`, `iscatalog`, `assigntype`, `createdate`, `sysid`, `tenantid`)  VALUES  ('ASVAL01', 'asval', '编码示例', '000000100000000ROOT1', 'N', NULL, NOW(), 'wbalone', 'tenant'); |

Oracle：

|  |
| --- |
| INSERT INTO pub\_bcr\_obj  (pk\_billobj, code, name, pk\_parent, iscatalog, assigntype, createdate, sysid, tenantid)  VALUES  ('000000100000000ROOT1', 'root', 'root', NULL, 'Y', NULL, SYSDATE, 'wbalone', 'tenant');  INSERT INTO pub\_bcr\_obj  (pk\_billobj, code, name, pk\_parent, iscatalog, assigntype, createdate, sysid, tenantid)  VALUES  ('ASVAL01', 'asval', '编码示例', '000000100000000ROOT1', 'N', NULL, SYSDATE, 'wbalone', 'tenant'); |



注意:

1.iscatalog要做赋值，否则页面加载不出数据来

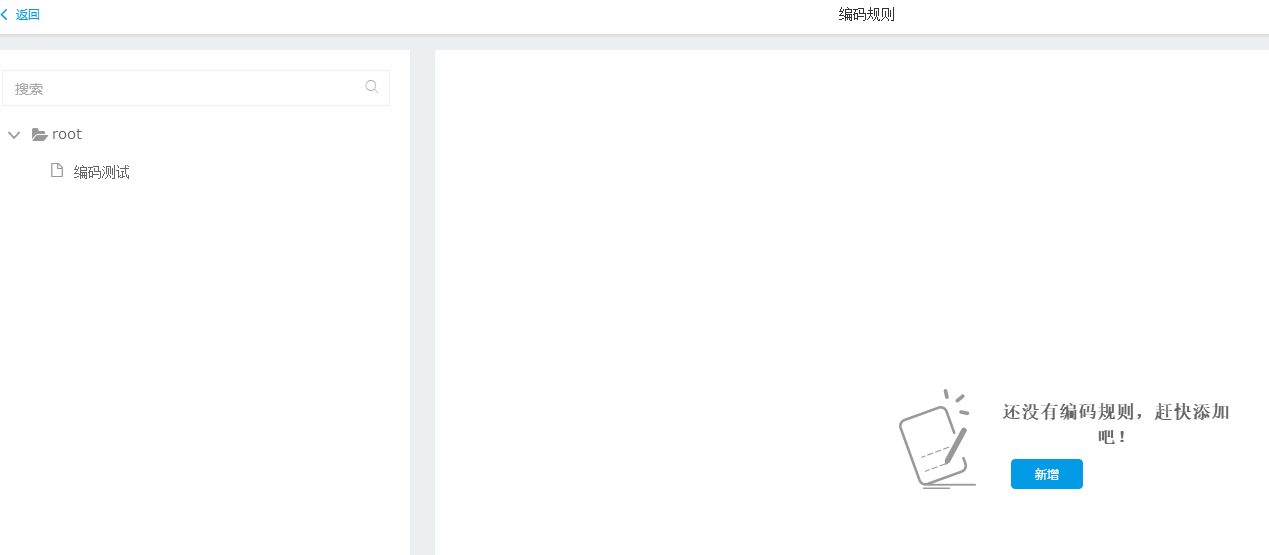
2.sysid默认为wbalone,tenant\_id默认为tenant

### 注册编码规则定义

在应用平台里已经提供了编码规则定义的节点，在管理中心-编码规则的节点就可以正常打开



编码规则定义节点可以注册编码规则。在数据库里注册编码对象后，在编码规则定义节点左侧就会出现对象的树型结构



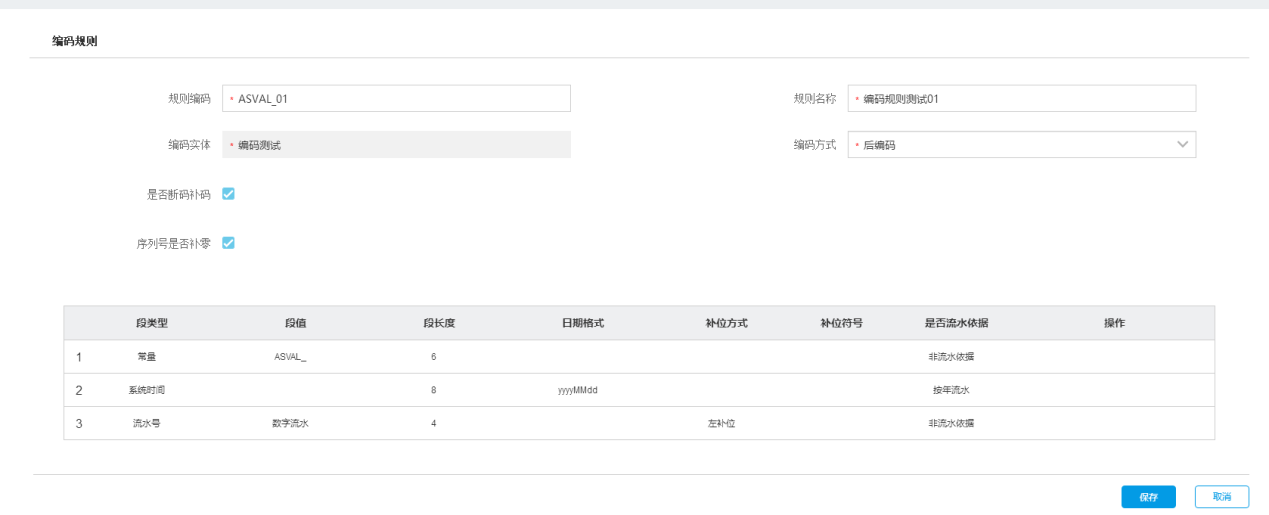
选择刚才注册的示例节点，点击新增，录入数据后保存

注意这里的规则编码在后面代码部分会用到。

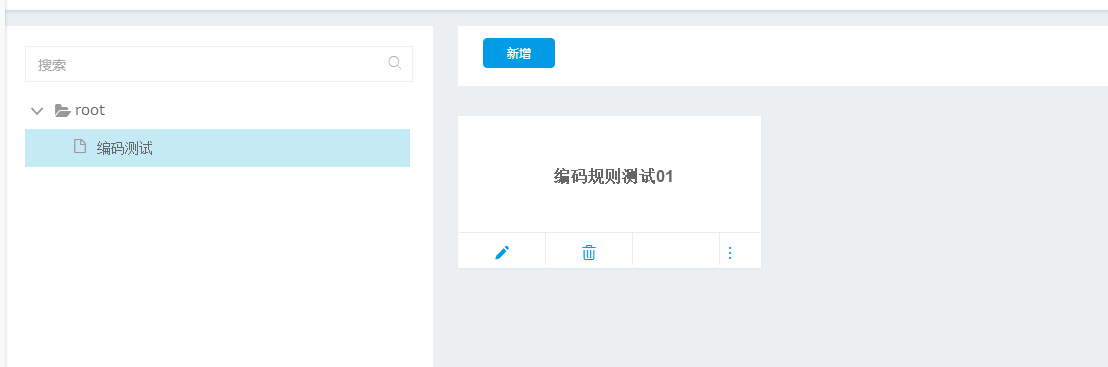
我们按照“CUST“+日期YYMMDD+数字流水号完成。

编码规则细节整理：

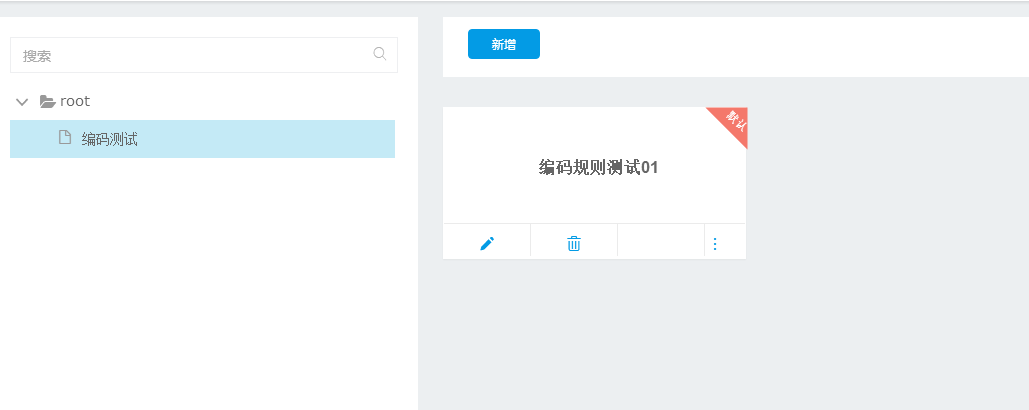
|  |
| --- |
| 规则编码:必填,唯一值建议字母或者数字组合;  规则名称:必填,建议中文描述  编码实体:必填,建议中文描述  编码方式:必填,建议选择后编码  是否断码补码:如果需要保持序列号连续，不中断则需要勾选该项  序列号是否补零: 建议勾选;  序列号是段类型有常量,字段值,系统时间,流水号,随机号  字段值这块具体的设计目前不清楚如何使用; |



保存成功后如下图所示



需要将编码规则设置为默认，操作如下图



此时编码规则的注册就完成了

注意：保存完毕后,设置为默认值;否则编码规则不生效;可以配置多个编码规则,但是同时只有一个编码规则设置为默认,即同时只有一个编码规则生效。

## 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。

功能节点设置js地址为：/iuap-example/pages/asVal/asval.js。

### 功能测试

登录应用环境，配置功能管理、菜单和分配功能权限后，打开界面，点击新增，输入记录后，点击保存操作，如下图所示：





# 代码开发示例

## 获取单据号

调用编码规则的RestAPI，调用服务获取到单据号后，保存即可。

|  |
| --- |
| 保存方法（在原有保存方法上新增一个编码规则的取值和赋值操作）： |
|  |
| 获取编码规则方法，通过RestAPI的方式调用获取编码，红色标注的内容是需要根据业务传值手动修改的： |
|  |

## 单据号退号

在删除方法里调用退号的RestAPI

参数还需要传入当前业务单据的编号，如果勾选了是否退码补码，会保证删除的编号还会重新出现。

|  |
| --- |
| 删除方法，在原有的删除方法上新增一个退号操作： |
|  |
| 退号方法，红色标注的内容是需要根据业务传值手动修改的： |
|  |
| 注意：还需要将customerCode从前台传到后台，所以在js代码customer.js中还需要加如下标红的一段，否则在删除的时候后台会报错： |
|  |

注意事项：

1. RESTAPI中调用的billObjCode为pub\_bcr\_obj表中的code值，而不是具体编码规则定义中的规则编码；

# 其它参考

## 编码规则RestAPI

官网API：组件提供的接口

* 获取编码规则

服务API“/billcoderest/getBillCode”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数字段 | 必选 | 类型 | 长度 | 说明 |
| billObjCode | True | String |  | 编码规则对象的编码 |
| pkAssign | False | String |  | 分配对象 |
| billVo | False | Object |  | 实体对象 |

* 退号服务

服务URL：” /billcoderest /returnBillCode”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数字段 | 必选 | 类型 | 长度 | 说明 |
| billObjCode | True | String |  | 编码规则对象的编码 |
| pkAssign | False | String |  | 分配对象 |
| billVo | False | Object |  | 实体对象 |
| billCode | True | String |  | 生成的编码 |