**IUAP平台使用指南**

**消息组件**

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 1](#_Toc515723746)

[1.1 业务场景 1](#_Toc515723747)

[1.2 开发方案 1](#_Toc515723748)

[第二章 环境准备 2](#_Toc515723749)

[2.1 开发环境配置 2](#_Toc515723750)

[2.1.1 设置编译路径 2](#_Toc515723751)

[2.1.2 设置编码规则地址 2](#_Toc515723752)

[2.2 数据库脚本配置 3](#_Toc515723753)

[2.2.1 创建表 3](#_Toc515723754)

[2.2.2 预置数据 3](#_Toc515723755)

[2.2.3 创建消息模板 5](#_Toc515723756)

[2.3 功能测试 6](#_Toc515723757)

[2.3.1 菜单注册 6](#_Toc515723758)

[2.3.2 功能测试 6](#_Toc515723759)

[第三章 代码开发示例 8](#_Toc515723760)

[3.1.1 后台代码 8](#_Toc515723761)

# 概述

## 业务场景

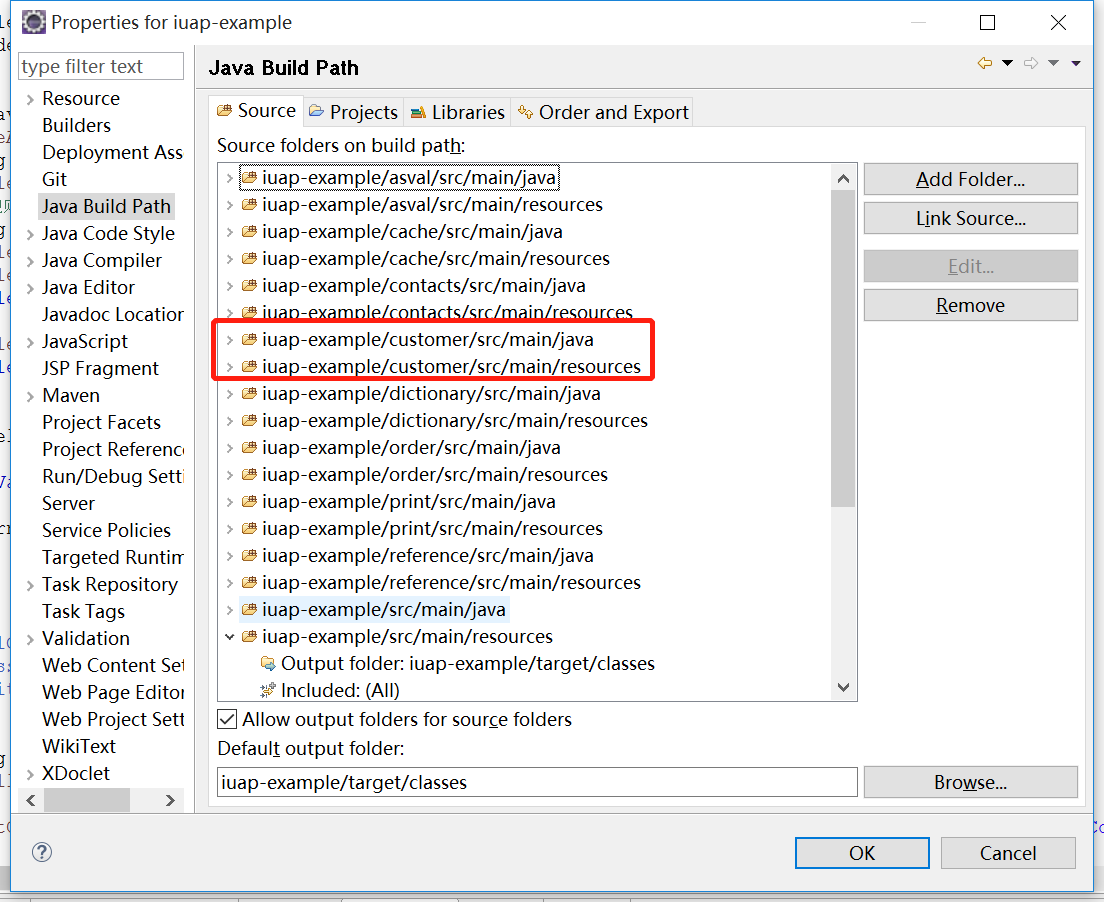
## 开发方案

# 环境准备

## 开发环境配置

### 设置编译路径

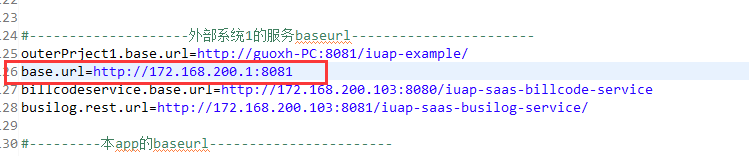
编码规则使用customer作为编译路径



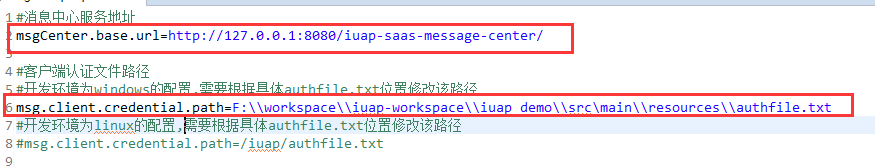
1. 设置编译路径

### 设置编码规则地址

1. 修改开发环境中src/resource/application.properties文件中的base.url为服务器的IP加端口号 或者 nginx监听的地址和端口号



1. 修改开发项目中src/resources/目录下的msg-sdk.properties文件，修改标红的内容：



如果应用中心配置在本机，则msgCenter.base.url 配置为本机的IP加对应的端口号即可；如果应用中心配置在远程服务器上，则将msgCenter.base.url配置为远程服务器的IP加对应的端口号；

Msg.client.credential.path要修改为本机存放该文件的位置；上图展示的是为本机为windows系统的配置方式；如果需要在linux系统中配置，则参照第8行的配置方式配置。

1. 修改authrbac-sdk.properties文件

地址为wbalone的地址：

authrbac.server.http.url=http://192.168.184.128:8080/wbalone

## 数据库脚本配置

### 创建表

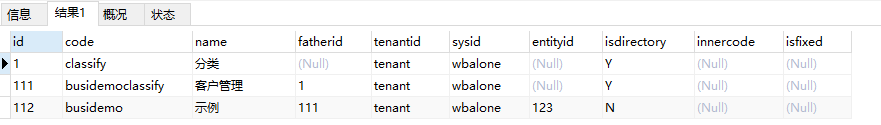
创建数据库表example\_customer

### 预置数据

a. 在iuap\_msgtemplate\_manage表中添加数据

MySQL/Oracle通用:

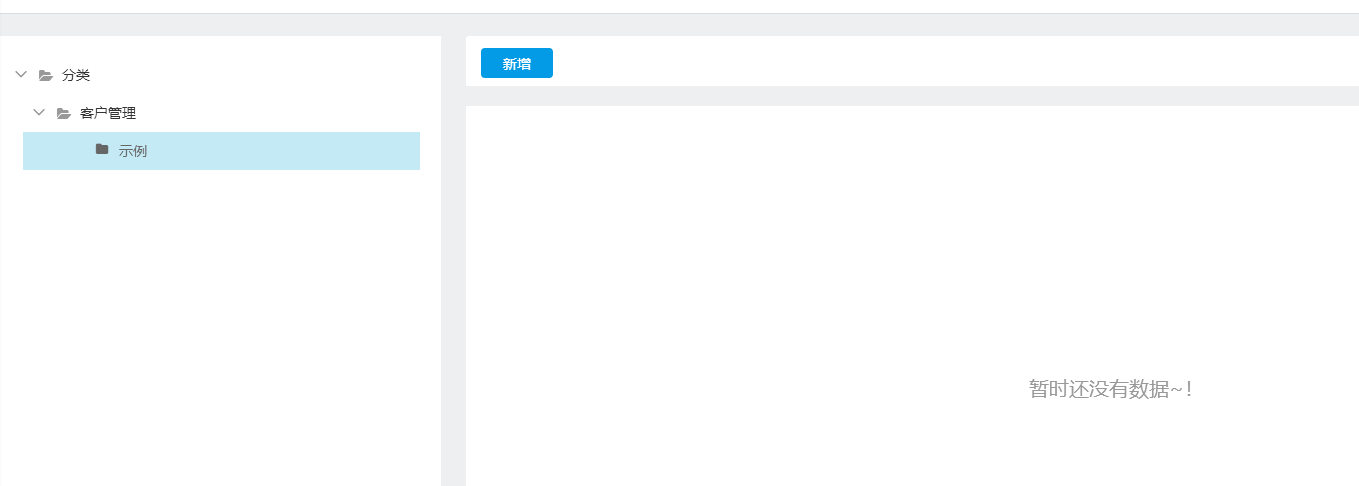
|  |
| --- |
| INSERT INTO iuap\_msgtemplate\_manage (id, code, name, fatherid, tenantid, sysid, entityid, isdirectory, innercode, isfixed)  VALUES ('1', 'classify', '分类', NULL, 'tenant', 'wbalone', NULL, 'Y', NULL, NULL);  INSERT INTO iuap\_msgtemplate\_manage (id, code, name, fatherid, tenantid, sysid, entityid, isdirectory, innercode, isfixed)  VALUES ('111', 'busidemoclassify', '客户管理', '1', 'tenant', 'wbalone', NULL, 'Y', NULL, NULL);  INSERT INTO iuap\_msgtemplate\_manage (id, code, name, fatherid, tenantid, sysid, entityid, isdirectory, innercode, isfixed)  VALUES ('112', 'busidemo', '示例', '111', 'tenant', 'wbalone', '123', 'N', NULL, NULL); |



注意:

1.isdirectory字段表示是否是目录，Y表示是，N表示否；

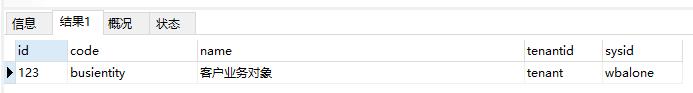
此表中数据为消息模板管理界面左侧树



b. 在iuap\_business\_entity表格中手动增加客户业务对象消息实体

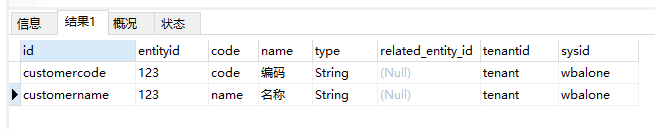
MySQL/Oracle通用：

|  |
| --- |
| INSERT INTO iuap\_business\_entity (id, code, name, tenantid, sysid) VALUES ('123', 'busientity', '客户业务对象', 'tenant', 'wbalone'); |



c. 在iuap\_business\_propertity表中增加客户关系表业务对象实体属性字段名，用于创建消息模板

|  |
| --- |
| INSERT INTO iuap\_business\_propertity (id, entityid, code, name, type, related\_entity\_id, tenantid, sysid) VALUES ('customername', '123', 'name', '名称', 'String', NULL, 'tenant', 'wbalone');  INSERT INTO iuap\_business\_propertity (id, entityid, code, name, type, related\_entity\_id, tenantid, sysid) VALUES ('customercode', '123', 'code', '编码', 'String', NULL, 'tenant', 'wbalone'); |

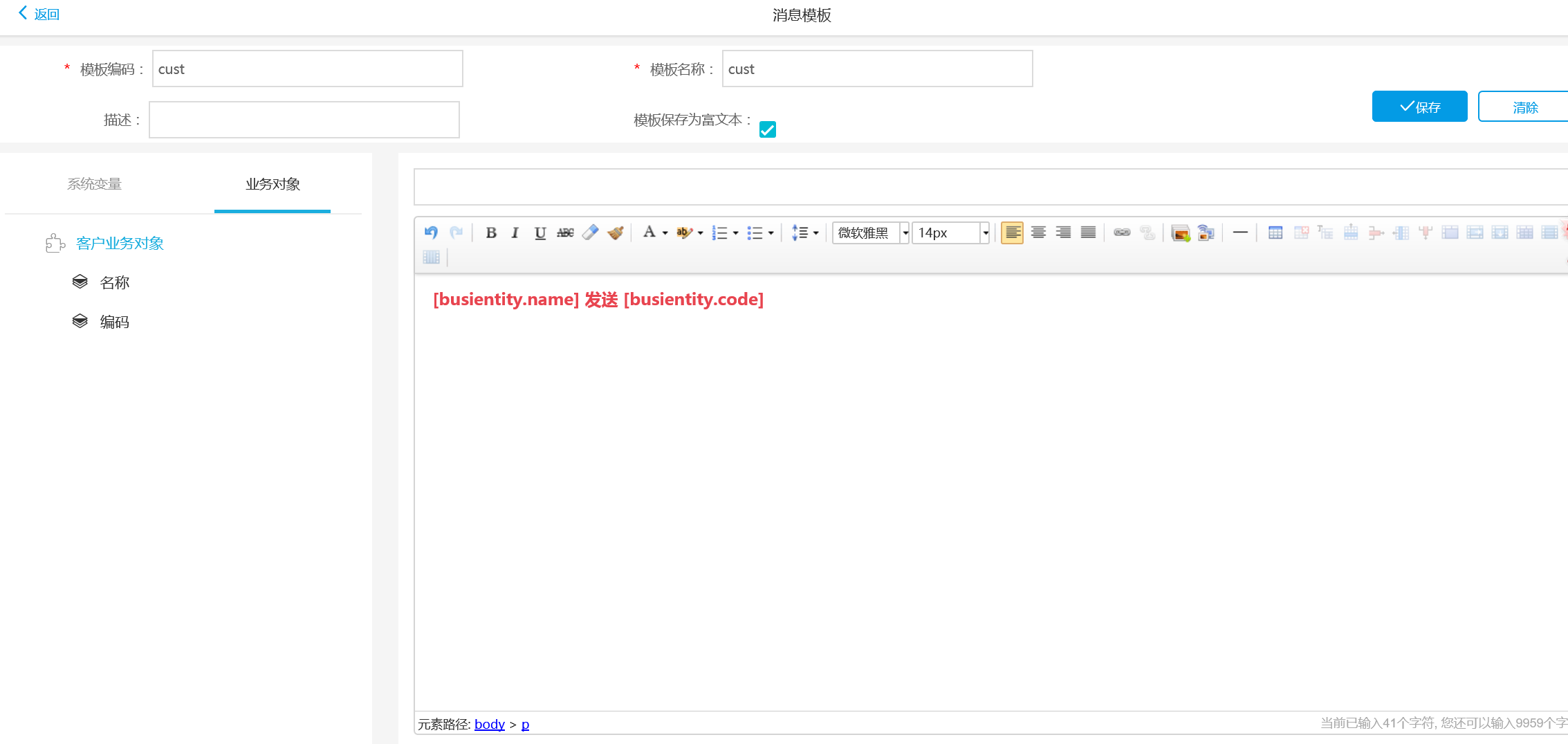


在管理中心-消息模板管理中，选中“客户管理”下的“示例”节点，点击新增按钮，如下图所示：



### 创建消息模板

通过点击系统变量和业务对象下的节点可以组织成一个消息模板，如下图所示



点击保存，注意记住该模板的模板编码，稍后在后台配置时需要用到。

## 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。

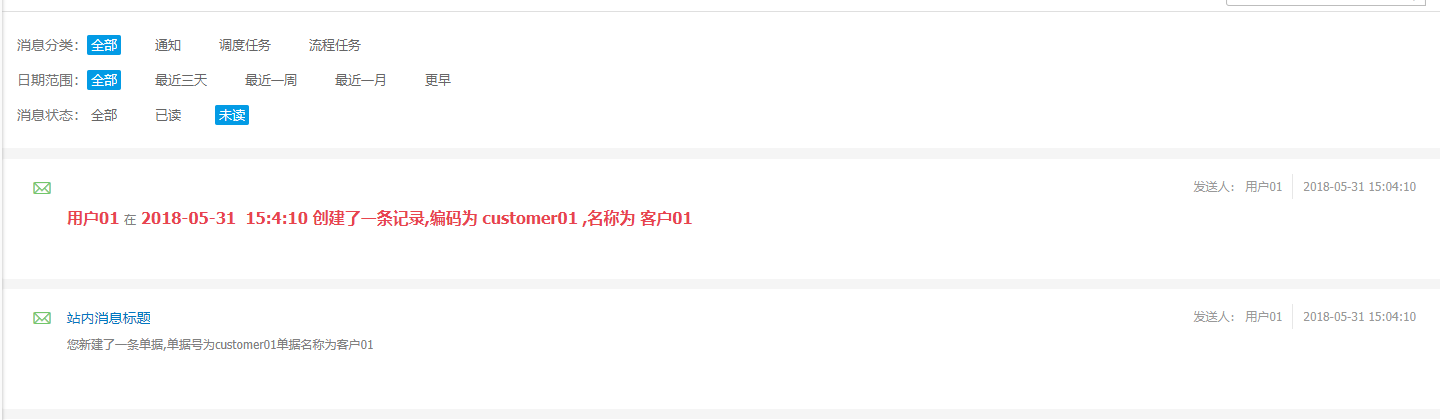
功能节点设置js地址为：/iuap-example/pages/customer/customer.js。

### 功能测试

新建一条记录，点击保存；然后点击右上角的消息中心，会看到如下记录：



打开消息中心会看到消息中心会有一条消息



# 代码开发示例

### 后台代码

在新增数据后发送消息

|  |
| --- |
|  |
| 消息发送代码，红色标注内容是需要根据业务情况进行调整的代码 |
|  |
| sendMessage中调用的convertRefName方法: |
| private Map<String, String> convertRefName(String url, Map<String, String> mapParameter) {  Map<String, String> mapRefName = new ConcurrentHashMap<String, String>();  try {  if (mapParameter != null && mapParameter.size() > 0) {  int index = 0;  for (Map.Entry<String, String> entry : mapParameter.entrySet()) {  String joinSymbol = null;  if (index == 0) {  joinSymbol = "?";  } else {  joinSymbol = "&";  }  url = String.format("%s%s%s=%s", url, joinSymbol, entry.getKey(), entry.getValue());  index++;  }  url = PropertyUtil.getPropertyByKey("base.url") + url;  JSONObject getbillcodeinfo = RestUtils.getInstance().doPost(url, null, JSONObject.class);  Map<String, Object> mapJson = (Map<String, Object>) JSON.parse(getbillcodeinfo.toString());  JSONArray array = (JSONArray) mapJson.get("data");  for (Object obj : array) {  Map<String, String> map = (Map<String, String>) obj;  mapRefName.put(map.get("id"), map.get("name"));  }  return mapRefName;  }  } catch (Exception ex) {  logger.error(ex.getMessage(), ex);  mapRefName.clear();  }  return mapRefName;  } |

注意事项：

在WebappMessageConst常量中定义了一些常用常量

通道类型：

邮件email(CHANNEL\_EMAIL)、

短信note(CHANNEL\_SMS)、

upush-messagepush(CHANNEL\_UPUSH)、

系统消息sys(CHANNEL\_SYS)

消息类型：

通知notice（MESSAGETYPE\_NOTICE）、

任务task(MESSAGETYPE\_TASK)、

文本消息text(MESSAGECONTENTTYPE\_TEXT)、

模板消息template(MESCONTENTTYPE\_TEMPLATE)