**IUAP平台使用指南**

**打印组件**

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](#_Toc515563125)

[1.1 业务场景 1](#_Toc515563126)

[1.2 开发方案 1](#_Toc515563127)

[第二章 示例配置 2](#_Toc515563128)

[2.1 IUAP打印服务确认 2](#_Toc515563129)

[2.2 打印服务申请 2](#_Toc515563130)

[2.3 数据库配置 4](#_Toc515563131)

[2.4 开发环境配置 5](#_Toc515563132)

[2.5 应用中心打印配置 6](#_Toc515563133)

[2.6 功能验证 9](#_Toc515563134)

[2.7 常见问题 10](#_Toc515563135)

[第三章 代码开发示例 11](#_Toc515563136)

[3.1 后端开发 11](#_Toc515563137)

[3.1.1 编写entity文件 11](#_Toc515563138)

[3.1.2 编写dao文件 12](#_Toc515563139)

[3.1.3 编写service文件 13](#_Toc515563140)

[3.1.4 编写controller文件 13](#_Toc515563141)

[3.1.5 编写xml文件 14](#_Toc515563142)

[3.2 前端开发 14](#_Toc515563143)

[3.2.1 编写viewModel.js 14](#_Toc515563144)

[3.2.2 编写template.html文件 15](#_Toc515563145)

[3.2.3 编写template.css文件 15](#_Toc515563146)

[3.2.4 编写template.js文件 15](#_Toc515563147)

# 概述

## 业务场景

开发者经常遇到打印各种定制化表单的需求，IUAP提供的云打印功能，能让开发者自己定制打印内容，打印格式，只需定制打印格式，编写少量代码即可方便快捷地实现这类需求，典型需求描述如下:

* 打印多种不同表单需求。
* 个性化打印样式需求。

## 开发方案

* 构建一张测试表，用来测试打印表单数据。测试表名称为“example\_print”，表结构如下。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 |
| Remark | 备注 | Varchar |  | 可空 |

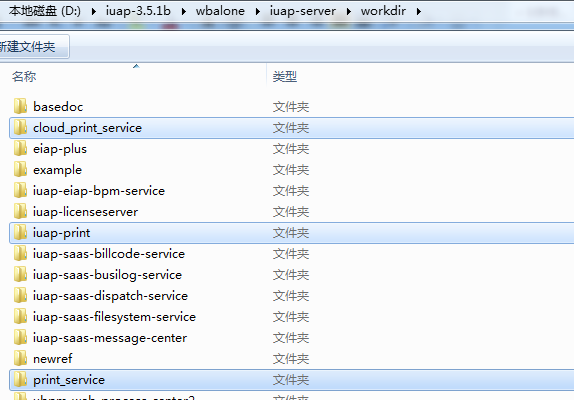
* 在应用中心增加打印业务对象
* 设计打印模板
* 进行资源分配
* 设计示例功能。
* 打印表单内容

# 示例配置

## IUAP打印服务确认

确认部署的IUAP包含了打印相关的服务

* cloud\_print\_service：云打印业务对象维护及打印模板设计的客户端代理服务。
* print\_service：本地打印服务客户端代理。
* iuap-print：本地打印服务

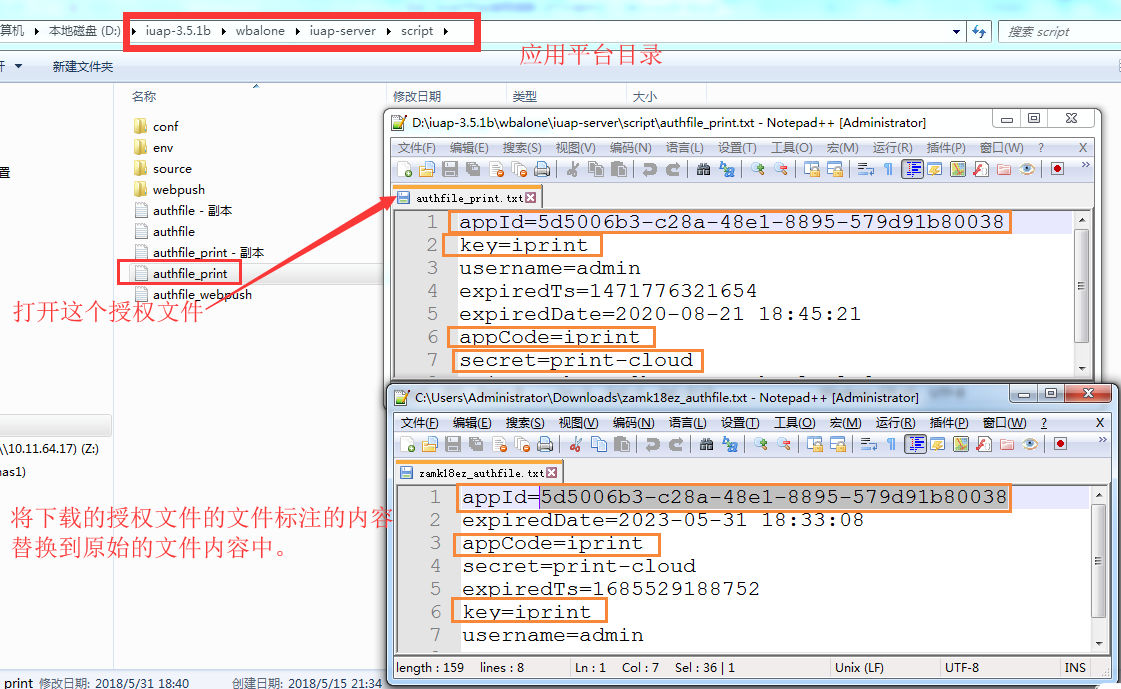


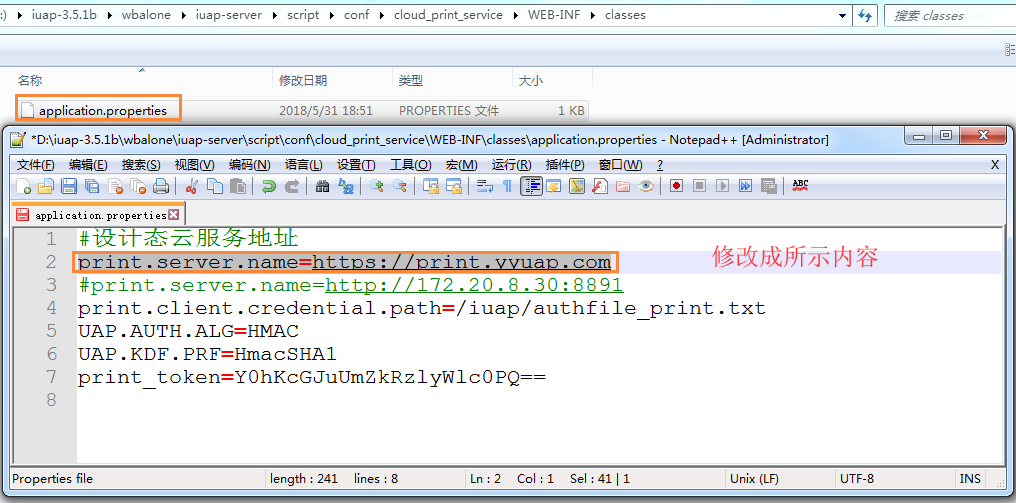
## 打印服务申请

1. 在<https://iuap.yonyoucloud.com/> 上注册账户。
2. 在[https://print.yonyoucloud.com/iuap-print/#/](https://print.yonyoucloud.com/iuap-print/" \l "/)上注册账户。
3. 在云打印网站上登录，进入云工作台，下载授权文件。



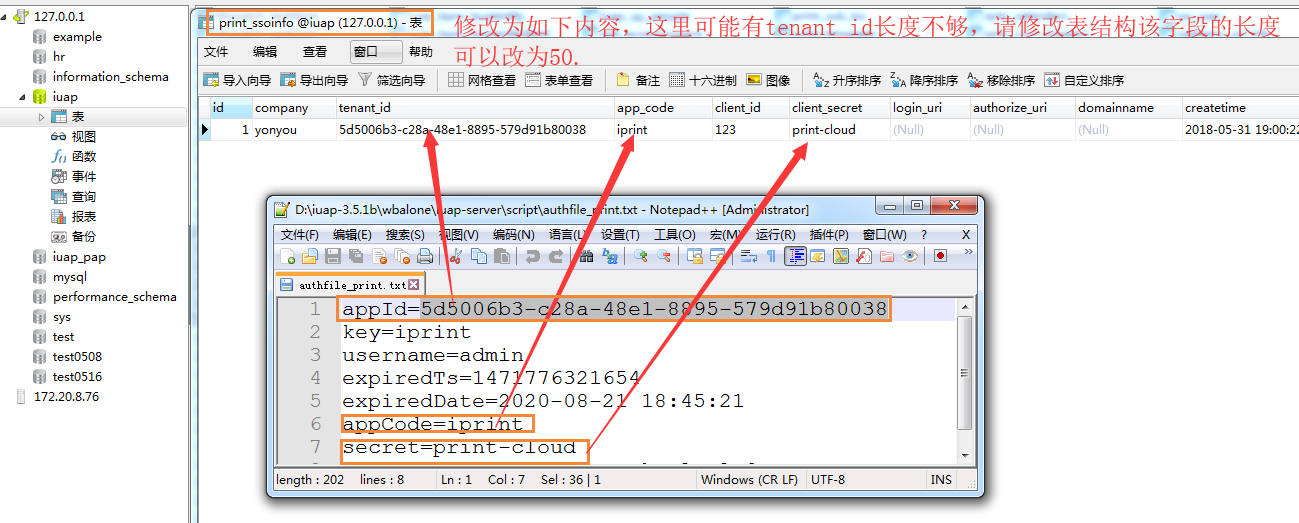
1. 根据下载的授权文件修改相关配置文件。





1. 修改数据库内容

先修改“print\_ssoinfo”表“tenant\_id”字段长度为50。然后修改“tenant\_id”，“app\_code”,“client\_secret”字段。



1. 重启IUAP服务。

## 数据库配置

在IUAP的数据库中创建示例表“example\_print”。

**CREATE TABLE `example\_print` (**

**`id` varchar(36) COLLATE utf8\_unicode\_ci NOT NULL,**

**`code` varchar(50) COLLATE utf8\_unicode\_ci DEFAULT NULL,**

**`name` varchar(50) COLLATE utf8\_unicode\_ci DEFAULT NULL,**

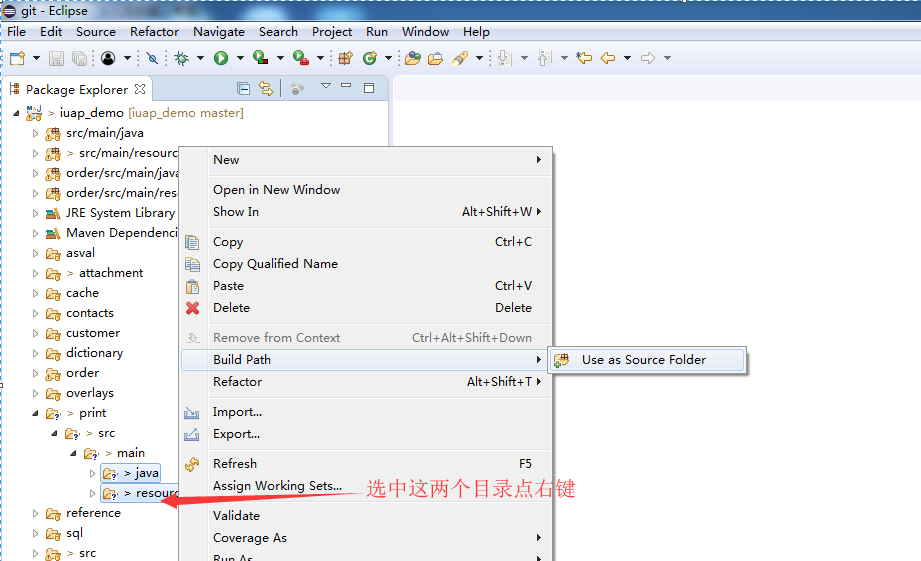
**`remark` varchar(50) COLLATE utf8\_unicode\_ci DEFAULT NULL,**

**PRIMARY KEY (`id`)**

**) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;**

## 开发环境配置

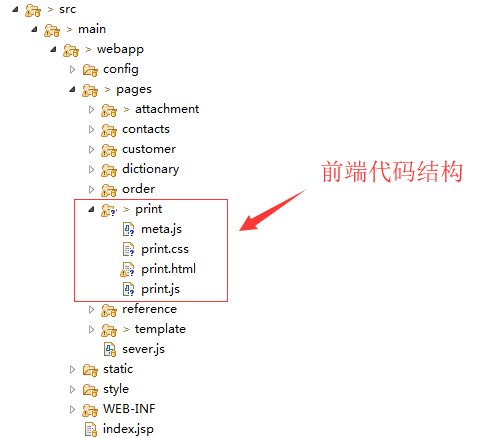
用eclipse载入示例工程，选中print目录下的java和resources目录，然后右键—》Build Path—》Use as Source Folder，将print示例载入编译环境。



后端代码结构



前端代码结构



检查项目无问题后，右击iuap\_demo –》Run As —》run on server。

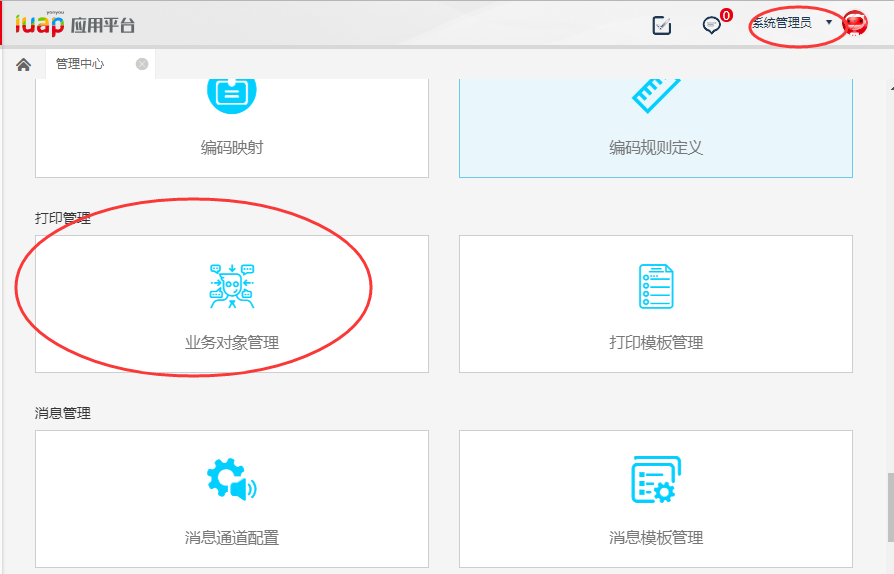
## 菜单注册

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/print/print.js。

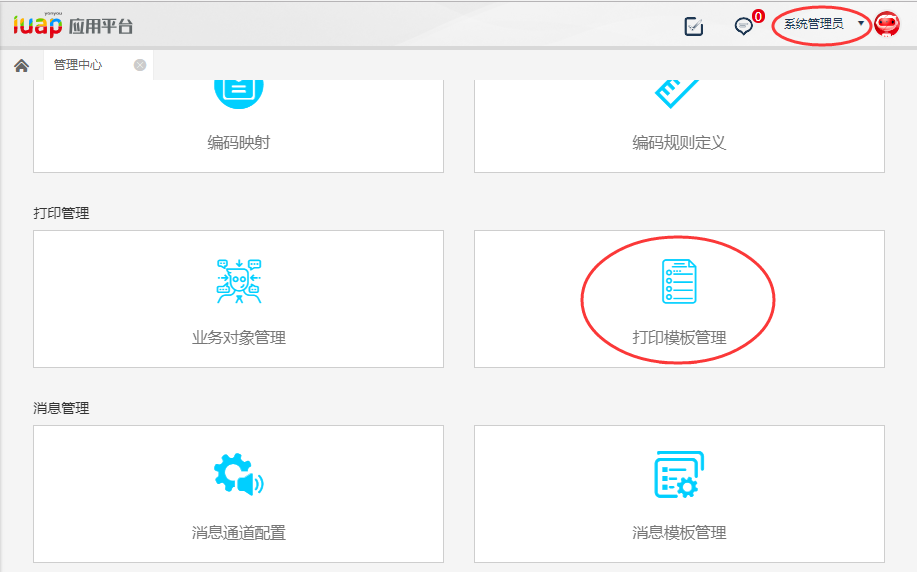
## 应用中心打印配置

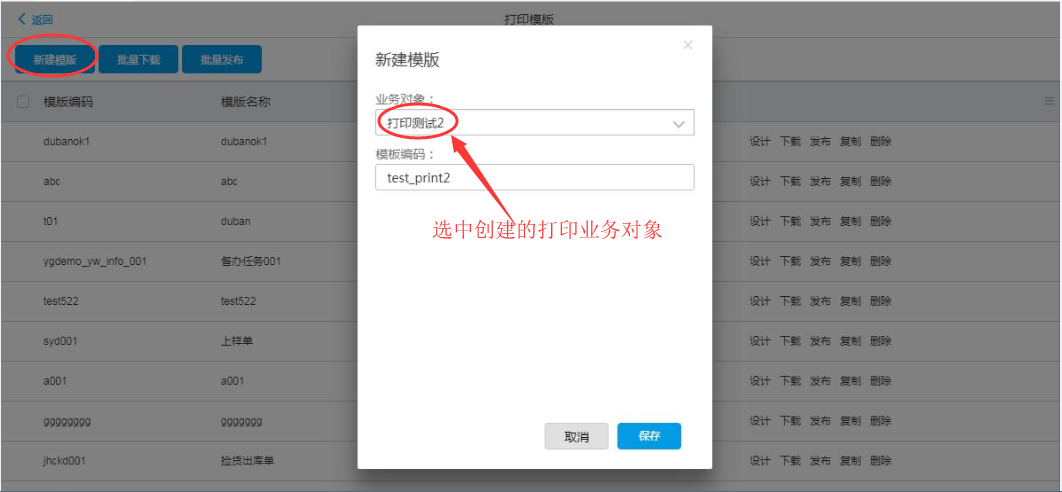
1. 业务对象管理





1. 创建打印模板





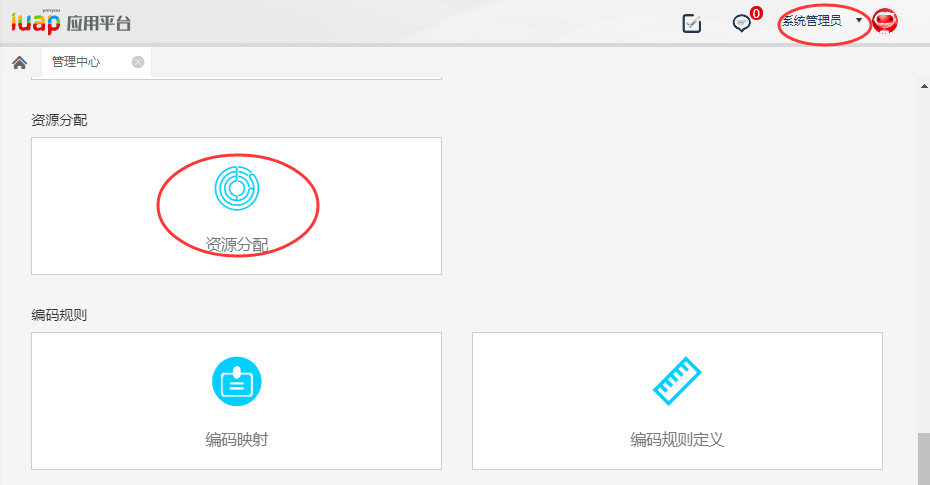






备注（如果在资源分配中找不到这里发布的模板，请点击这里的下载，然后在iuap数据库中执行即可。）

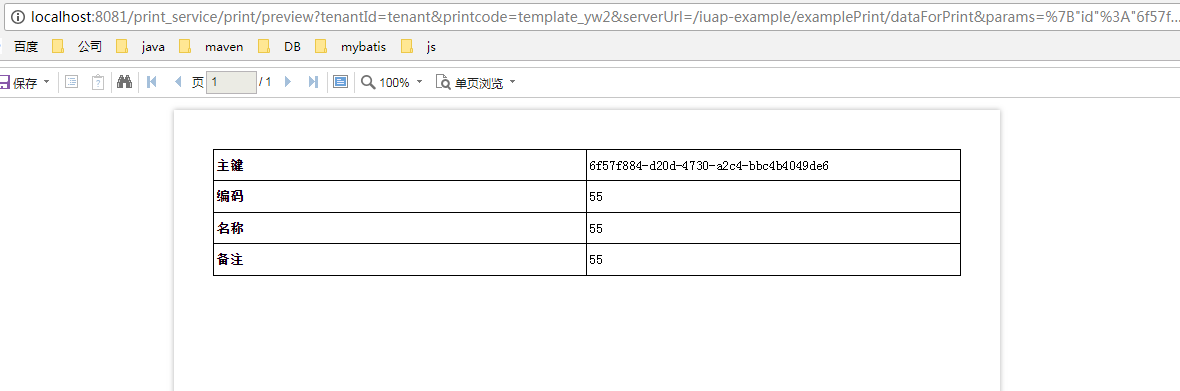
1. 分配资源





## 功能验证





## 常见问题

* 设计打印业务对象的时候，需要注意字段名需要与数据库保持一致。
* 打印模板设计需要在内网环境中进行，暂时没有开通外网。

# 代码开发示例

## 后端开发

### 编写entity文件

编写实体对象，需要继承BaseEntity对象，并重写getMetaDefinedName方法和getNamespace方法。

package com.yonyou.iuap.example.print.entity;

import com.yonyou.iuap.persistence.vo.BaseEntity;

publicclassExamplePrintextendsBaseEntity {

privateStringid;

privateStringcode;

privateStringname;

privateStringremark;

publicExamplePrint() {

}

publicStringgetId() {

returnthis.id;

}

publicvoidsetId(Stringid) {

this.id = id;

}

publicStringgetCode() {

returnthis.code;

}

publicvoidsetCode(Stringcode) {

this.code = code;

}

publicStringgetName() {

returnthis.name;

}

publicvoidsetName(Stringname) {

this.name = name;

}

publicStringgetRemark() {

returnthis.remark;

}

publicvoidsetRemark(Stringremark) {

this.remark = remark;

}

@Override

publicStringgetMetaDefinedName() {

return"example\_print";

}

@Override

publicStringgetNamespace() {

return"iuap\_example";

}

}

### 编写dao文件

在dao文件中新增getByIds方法。

@MyBatisRepository

publicinterfaceExamplePrintMapper {

//这里还有其他方法

//

//

PageResult<ExamplePrint>getByIds(List<String>ids);

}

### 编写service文件

新增queryByPK方法。

publicExamplePrintqueryByPK(Stringid) {

        List<String>ids = newArrayList<String>();

        ids.add(id);

        List<ExamplePrint>list = mapper.getByIds(ids);

        if (list != null&&list.size() >0) {

            ExamplePrintentity = list.get(0);

            return entity;

        } else {

            returnnull;

        }

}

### 编写controller文件

新增获取数据的RequestMapping “/dataForPrint”。

@RequestMapping(value = "/dataForPrint", method = RequestMethod.POST)

@ResponseBody

publicObjectgetDataForPrint(HttpServletRequest request) {

Stringparams = request.getParameter("params");

JSONObjectjsonObj = JSON.parseObject(params);

Stringid = (String) jsonObj.get("id");

ExamplePrintvo = service.queryByPK(id);

JSONArraymainDataJson = newJSONArray();// 主实体数据

List<String>lsAttr = vo.getAllAttributeNames();

JSONObjectmainData = newJSONObject();

for (String attr : lsAttr) {

if (BaseEntityUtils.lsAttrExclude.contains(attr)) {

continue;

}

mainData.put(attr, vo.getAttribute(attr));

}

mainDataJson.add(mainData);// 主表只有一行

JSONObjectboAttr = newJSONObject();

boAttr.put("example\_print", mainDataJson);

System.out.println(boAttr.toString());

returnboAttr.toString();

}

### 编写xml文件

新增getByIds的select。

<selectid="getByIds"resultMap="BaseResultMap"parameterType="java.util.List">

    SELECT

    <includerefid="Base\_Column\_List"/>

    from example\_print where

    id in

    <foreachcollection="list"item="item"index="index"

        separator=","open="("close=")">

        #{item,jdbcType=VARCHAR}

    </foreach>

</select>

## 前端开发

### 编写viewModel.js

viewModel.js中定义数据模型，可根据实际业务需求修改。

var meta = {

meta: {

id: { type: "string" },

code: {

type: "string",

required: true,

nullMsg: "不能为空!"

},

name: {

type: "string",

required: true,

nullMsg: "不能为空!"

},

remark: {

type: "string"

}

}

};

var viewModel = {

draw: 1,

//页数(第几页)

pageSize: 5,

gridData: new u.DataTable(meta),

formData: new u.DataTable(meta),

};

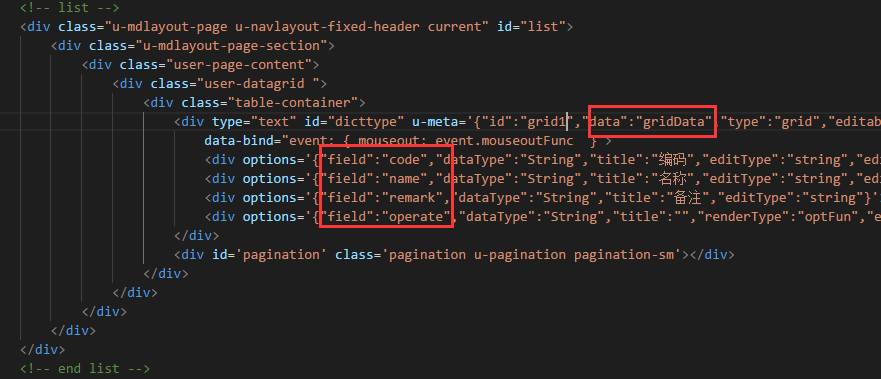
### 编写template.html文件

新增一个打印按钮，绑定打印方法printPage

<buttonclass="u-button u-button-primary raised gray-btn u-button-pa margin-right-20"id="user-action-print"data-bind="click: event.printPage">打印

</button>

添加表格对象，绑定表格字段到meta中定义的filed字段。



### 编写template.css文件

文件中定义的样式可以根据实际情况修改。

### 编写template.js文件

* 添加对文件的引用，

define(['text!./print.html',

    "css!../../style/common.css",

    'css!./print.css',

    '../../config/sys\_const.js',

    "../../utils/utils.js",

    "../../utils/ajax.js",

    "../../utils/tips.js",

    "./viewModel.js"]

* 修改调用服务的地址

  ctx = cookie.appCtx + "/examplePrint";

* 打印数据

在printPage方法中，将url填写为自己定义的打印业务对象和打印模板，在取数据的函数printPageByTemplateCode中修改请求地址为：'/examplePrint/dataForPrint'。



