# 自定义模板通过接口获取数据的设计说明

1. 接口API模板定义

模板打印增加通过调用接口来获取数据，在模板设计页面中定义接口地址，每个模板可以配置多个接口地址。接口定义表单如下：



**表单字段说明:**

**请求地址：**接口请求地址必须是完整的URL。

**请求方式：**默认请求方式POST（建议统一使用POST）。

**业务入参名称：**默认值ids, 调用改接口时，将会使用当前模板主元数据记录id集合，使用当前参数名称进行传入。

**自定义Token:** 格式为JSON字符串，JSON格式如下所示：

{

“tokenName”: “X-Access-Token”,

“tokenValue”: “值”

}

1. 业务接口定义

**接口参数：**需要包含一个List<String> ids参数，用来获取模板打印时主元数据记录ID参数集合。其中参数名“ids“ 也可以自定义，修改名称需要在接口配置表单中同步调整，示例参数对象IdsDTO。

**接口请求方式：**建议为POST。

**接口返回数据格式：**返回结果为Rsp封装对象，其中data格式为ElementConfigDataDTO对象数组，具体格式如下：

Rsp: {

data: [{

elementConfigId: “元数据配置ID”,

elementDataList: [{},{}…] // 元数据对象的数据集合

}, {

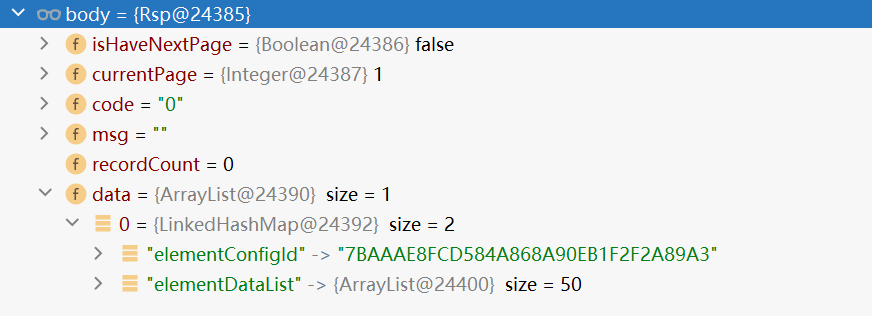
elementConfigId: “元数据配置ID”,

elementDataList: [{},{}…] // 元数据对象的数据集合

}

]

}



1. 在模板中使用接口数据

在模板中配置接口后，在调用模板进行打印时，后台会调用配置的接口获取元数据记录。如果返回的数据存在非模板主元数据记录时，也需要配置元数据之间的关系，才能获取到数据。

接口返回的均为元数据记录，不存在或没有找到符合要求的元数据需要先进行创建。

单元格设置取值和之前完全一样。

1. JAVA接口参考示例

@ApiOperation("测试从接口获取数据")

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

public Rsp getDataByApiTest(HttpServletRequest request, @RequestBody IdsDTO data) {

*// Token*

String token = request.getHeader(HttpUtil.*AUTHENTICAtION\_TOKEN*);

System.*out*.println(token);

*// 获取请求参数*

System.*out*.println(JSONUtil.*toJsonStr*(data));

*// 系统参数*

System.*out*.println(SystemContextHandler.*getName*());

System.*out*.println(SystemContextHandler.*getUserName*());

System.*out*.println(SystemContextHandler.*getTsEnterXid*());

*// 元对象信息查询*

String viewNo = "v\_tt\_testitem";

BaseElementConfig baseElementConfig = new BaseElementConfig();

baseElementConfig.setViewNo(viewNo);

baseElementConfig.setIsDelete("1");

baseElementConfig = baseElementConfigMapper.selectOne(baseElementConfig);

*// 数据查询*

List<TtTestitem> allList = ttTestitemMapper.selectAll();

Rsp rsp;

try {

List<ElementConfigDataDTO> list = new ArrayList<>();

ElementConfigDataDTO item = new ElementConfigDataDTO();

item.setElementConfigId(baseElementConfig.getId());

// 数据封装

List<Map<String, Object>> listDatas = new ArrayList<>();

for (TtTestitem itemObj : allList) {

Map<String, Object> obj = object2map(itemObj);

listDatas.add(obj);

}

item.setElementDataList(listDatas);

list.add(item);

// 数据返回

rsp = ResponseUtil.*buildOKRsp*(list);

} catch (Exception ex) {

ex.printStackTrace();

rsp = ResponseUtil.*buildFailRsp*(null, ex.getMessage());

}

return rsp;

}