

关于列表自定义列配置的开发

1. 需求背景

系统标准的客户模块中的商机管理列表默认展示的列仅有四列，可查看数据过少。可通过无侵入开发，通过代理 `api` 接口的方式以及表单配置，对列表进行列扩展。

2. 自定义列配置

通过建模新增模块如下

数据库表名 : uf_custom_column

<input type="checkbox"/>	字段显示名	数据库字段名称	字段类型
<input type="checkbox"/>	<div><div>+</div><div>接口URL</div></div>	api_url	单行文本框 类型 文本 文本长度 300
<input type="checkbox"/>	<div><div>+</div><div>字段名</div></div>	field_name	单行文本框 类型 文本 文本长度 300
<input type="checkbox"/>	<div><div>+</div><div>显示列名</div></div>	show_name	单行文本框 类型 文本 文本长度 100
<input type="checkbox"/>	<div><div>+</div><div>是否启用</div></div>	isuse	Check框

- 接口URL：需要代理的接口URL
- 字段名：扩展字段的数据库字段名
- 是否启用：启用则新增，否则过滤

3. 实现方式

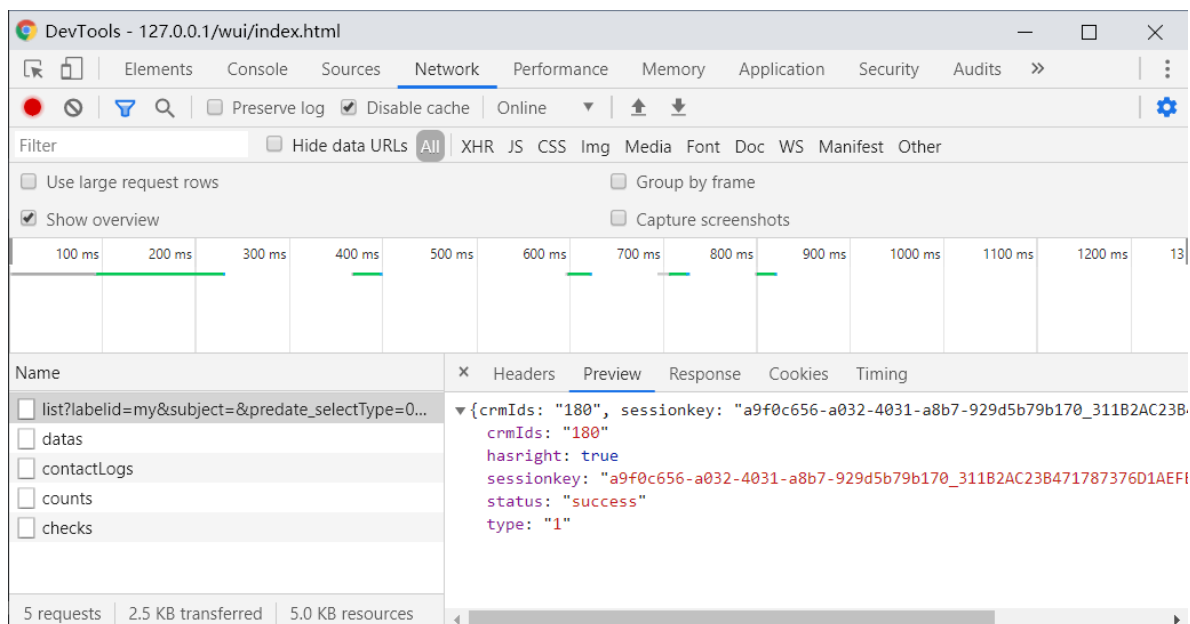
`e9` 列表数据的步骤如下：

- 根据相关接口生成 `dataKey`，并配置表格的相关配置参数。将其写入到缓存中
- 将生成的 `datakey` 传给通用 `/api/ec/dev/table/datas` 接口，从而获取数据。

通过采用云商店目前还在内测的一种无侵入开发模式，使用 `weaverboot-e9` 框架，对**商机管理**列表进行接口代理。通过修改缓存中的相关配置参数，从而实现修改表格数据的效果。

3.1 抓取生成dataKey的接口

如下图中的接口的响应参数中，`sessionkey` 即 `datakey`



3.2 根据响应参数创建ReceiveResultMsg

该类需要继承

`com.weaverboot.weaResultMsg.impl.tableResult.inte.AbstractWeaTableComponentReceiveResultMsg`

实现 `parseAndGetWeaTable` 方法，并且 在该类中，类的成员变量需要与响应参数想对应。

如下面代码中，需要实现 `crmIds`，`hasright`，`sessionId`，`status` 等参数。（必须实现！）

```

1  @Setter
2  @Getter
3  public static class ReceiveResultMsg extends
   AbstractWeaTableComponentReceiveResultMsg {
4
5      private String crmIds;
6
7      private String status;
8
9      private boolean hasRight;
10
11     private String sessionId;
12
13     @Override
14     public <T extends AbstractWeaTable> T parseAndGetWeaTable(Class<T>
   clazz) throws Exception {
15         String sessionId = this.getDatas();
16         T t = weaTableTools.checkTableStringConfig(sessionId, clazz);
17         this.setWeaTable(t);
18         return t;
19     }
20 }

```

3.3 实现接口代理

```

1  @WeaIoCReplaceComponent
2  public class CrmSellChanceTableReplace {
3

```

```
4      @weaReplaceAfter(value = "/api/crm/sellchance/list", order = 1)
5      public String tableReplace(weaAfterReplaceParam param) throws Exception
6      {
7          // 获取接口响应参数
8          ReceiveResultMsg resultMsg =
9          JSONObject.parseObject(param.getData(), ReceiveResultMsg.class);
10         // 根据 sessionId 获取表格对象
11         String sessionId = resultMsg.getSessionKey();
12         DefaultweaTable table =
13         weaTableTools.checkTableStringConfig(sessionId, DefaultweaTable.class);
14         // 开始一顿修改参数操作
15         // ...
16         // 将修改后的参数推送到缓存中
17         weaTableTools.setTableStringVal(sessionId, weaTableTools
18         .toTableString(table));
19         return param.getData();
20     }
21 }
```