前言

在 3.0 工作平台中想要实现一个大且复杂的表单时，犹如置身于泥沼之中，感觉寸步难行。当然，在简单的表单开发时，工作平台的方式简单快速且不易出问题。

所以，这时需要工具帮助我们走出泥沼。vue 是一个不错的选择。

如果你也想这样干，下面会介绍 vue 在 3.0 平台中的使用示例和一些注意事项。

引入

在 `webapp/framework/views/main/index.html` 添加 vue 的依赖

```
<script type="text/javascript" src="https://cdn.bootcss.com/vue/2.6.10/vue.min.js"></script>
```

由于 `main/index.html` 为工作平台的根页面，所以这样引入之后相当于全局引入了 vue。

同时我们也可以加入我们想使用的组件库（element、iView、Ant Design 等）。

快速开始

开门见山，在引入了 vue.js 之后，我们准备开发一个“饮料管理”模块，包含三个字段（id, name, price）

1. 编写工作平台模板文件

- drinks_management.xml

```
<?xml version="1.0" encoding="GBK"?>
<windows>
    <window id="mainView" type="main" title="饮料管理">
        <!-- 搜索区域 省略 -->
        <!-- 数据表格区域 -->
        <panel id="dataView0">
            <dataGrid css="pt_table" funcNo="9019307" id="grid0" defaultForm="mainView_search0_base0">
                <toolbar id="baseTools2">
                    <button id="upload" name="新增" function="add" css="add" isMutiDml="0" isPop="1" popId="a
ddDrinks"></button>
                </toolbar>
                <!-- 列表 fields 省略-->
            </dataGrid>
        </panel>
    </window>

    <!-- 新增饮料弹窗 -->
    <window id="addDrinks" type="pop" title="新增饮料" isMaximize="0" width="300" height="200" style="backgrou
nd:#fff;padding:0;">
        <!-- 工作平台中的 panel 会生成 div 标签，这里相当于写了一个 id 为 app 的 div 标签-->
        <panel id="app" style="padding: 18;"></panel>
        <!-- 相当于引入了 framework/scripts/drinks/drinks_add.js，并向其 init 函数中注入了 jsonParam 对象 -->
        <script>
            var _defaultPage = {"pageCode": "drinks/drinks_add", "jsonParam":{mode: 0}, "isLoad":false,"loadT
ype":"1"};
        </script>
    </window>

</windows>
```

2. 编写处理业务的 js 脚本

- drinks_add.js

```
define("framework/scripts/drinks/drinks_add", function (require, exports, module) {

    function init(param) {
        // 这里的 param 就是上面模板文件中注入的 jsonParam
        var template = '';
        // 同步加载 vue 模板文件 $.getHtmlContent 这个 api 的 base 路径是 framework/views
        $.getHtmlContent('../vue_template/drinks', function (res) { // 加载 vue 模板
            template = res;
        }, false); // false 同步执行 true 异步执行
        initVue(template);
    }

    function initVue(template) {
        // vue 实例
        var app = new Vue({
            el: '#app', // 渲染区域 ID 对应模板中的 <panel id="app"/>
            template: template, // 使用上面加载的模板文件来渲染页面
            data: {
                drinksForm: {
                    name: '',
                    price: ''
                }
            },
            methods: {
                handleSubmit () { // 提交
                    // 开始提交 service.addDrinks(drinksForm, function(res){ ... })
                    console.log("drinksForm: ", drinksForm);
                    this.handleClose();
                    // 添加完成后刷新饮料列表
                    $.gconfig.global.componentBusMap["mainView_dataView0_grid0"].freshData();
                },
                handleClose () { // 关闭弹窗
                    $("#addDrinks").dialog("close");
                }
            }
        });
    }

    function bindPageEvent() {}

    //页面销毁
    function destroy() {}

    var startup = {
        "init": init,
        "bindPageEvent": bindPageEvent,
        "destroy": destroy
    };

    // 暴露对外的接口
    module.exports = startup;
});
```

3. 编写 vue 模板文件，弹框的交互页面

- framework/vue_template/drinks.html

```
<div>
    <input v-model="drinksForm.name" placeholder="请输入饮料名称"><br/>
    <input v-model="drinksForm.price" placeholder="请输入饮料价格"><br/>
    <button v-on:click="handleSubmit">提交</button>
    <button v-on:click="handleClose">关闭</button>
</div>
```

注意事项

1. 传递参数

场景：当我们需要做修改饮料和查看饮料时，通常我们需要传递 `id` 到弹窗的业务控制 js 中，甚至我们可以将表格中的饮料名称和价格（`name`、 `price`）一起传递到控制 js 中。

```
    <!-- 修改饮料弹窗 -->
    <window id="modifyDrinks" type="pop" title="修改饮料" isMaximize="0" width="300" height="200" style="backg
round:#fff;padding:0;">
        <form id="base" name="form" target="mainView_dataView0_grid0" isLoad="1" css="input_list" width="" he
ight="">
            <!-- editConfigType="5" 不用选中即可传入（适用于操作栏，如操作栏的查看饮料操作） editConfigType="4" 需要选
中传入（适用于工具栏，如修改） -->
            <input id="id" displayName="" displayType="hidden" editConfigType="4" conditionId="id"></input>
        </form>
        <!-- 工作平台中的 panel 会生成 div 标签，这里相当于写了一个 id 为 app 的 div 标签-->
        <panel id="app" style="padding: 18;"></panel>
        <!-- 相当于引入了 framework/scripts/drinks/drinks_add.js，并向其 init 函数中注入了 jsonParam 对象 -->
        <script>
            var _defaultPage = {"pageCode": "drinks/drinks_add", "jsonParam":{mode: 1}, "isLoad":false,"loadT
ype":"1"};
        </script>
    </window>
```

这里我们使用隐藏 input 的方式传递参数，在控制 js 中使用 `$('#id').val()` 取值；这里再引入业务控制 js 时，我们依旧使用 `drinks_add.js` 而修改了 `jsonParam`，也就是说这里我们完全可以用 mode 来标识当前操作是什么（如 0：添加 1：修改 2：查看）这样做可以让我们的业务控制 js 和视图模板都用同一份代码，当然，这仅仅适用于这些操作（添加、修改、查看）相似度很高的情况下。

> 注意：当我们这样做了之后，需要及时重命名 drinks_add.js，因为他已经不再是单一责任了。

2. 关闭弹窗

```
$("#modifyDrinks").dialog("close"); // 关闭修改饮料弹窗
```

3. 操作完成后列表刷新

```
// "mainView_dataView0_grid0" 会根据这个很长的值逐层（mainView->dataView0->grid0）的寻找到对应的 dataGrid
$.gconfig.global.componentBusMap["mainView_dataView0_grid0"].freshData(); // 刷新饮料列表
```

4. vue 中的数组变动视图不渲染

Vue 不能检测以下变动的数组:

- 当你利用索引直接设置一个项时，例如： `this.items[indexOfItem] = newValue`
- 当你修改数组的长度时，例如： `this.items.length = newLength`

这里 vue 提供一种解决方法：

```
// 其作用与 this.items[indexOfItem] = newValue 一致，并且会触发 vue 去更新和 this.items 绑定的视图。
this.$set( this.items, indexOfItem , newValue);
```

∞

Ref

- vue: https://cn.vuejs.org
- element: https://element.eleme.io/
- iView: https://iviewui.com/
- Ant Design: https://ant.design/
- 双向数据绑定: https://www.liaoxuefeng.com/wiki/1022910821149312/1109527162256416

- MVVM: https://juejin.im/post/593021272f301e0058273468