**Vue实现方式**

**核心：**拖拽 基于sortable.js的draggable.vue

参考：<https://github.com/SortableJS/Vue.Draggable>

<https://github.com/RubaXa/Sortable>

推送：papparse(解析CSV文件) <https://github.com/mholt/PapaParse>

**其他：**loadsh, moment, vue-awesome-swipe, element-ui

------------------------------------------- -独立负责------------------------------------------------------

Conf配置逻辑熟悉:

Banner.vue => conf -block.vue=> conf-fram.vue => banner-conf.vue

出现弹窗 默认渲染数据

展示属性的列表控制 showcase-conf 重点typeOption6个 md模块的typeOption3个

触发List数据 => 触发vuex 中的mutationtype => conf配置库引入

新建sticky模块流程：

Styicky.vue sticky-conf.vue select-template.vue // sticky 模块

Conf-frame.vue(显示sticky-conf)

mutation-types.js conf-block.js

**最终渲染的数据的问题：**

Select-template中的信息要返回给弹窗，弹窗中的数据即是

两个关键的按键： config弹窗的确认键，select框的确认键

可完善的地方：

Banner模块 品牌，商品转义的完善； option 多用valuel等内置属性

Goods模块TypeOptions的完善；

两种不同功能实现需要用到同一模块的时候，要尽量运用分模块思维；比如：

Goods运用到基本相同methods的组件，可以分模块来写 ==》methods分模块

Swiper-sm-conf 中的选择商品按钮的动态变化；

看别人代码的有效方法：

Html结构 中的每一个事件都去看看。Html结构当成事件的大纲

Methods事件 ，事件才是核心逻辑；

自我启示：

命名方式： 名称+操作的组合式命名

看参考文档不如敲参考文档

--------------------------------------------------------------------------------- end --------------------------------

**JQ实现方式**

**核心技术：**

**拖拽：**基于jquery-ui(视图与节点同步变化)的 拖拽操作 sortable与draggable函数

参考网址：

中文文档：<http://www.jqueryui.org.cn/api>

官方文档：https://jqueryui.com/draggable/#sortable

**解析**：jsonToTemplate && templateToJson

参考JS：json-init.js && json-dzh.js

沟通学习：

产品： 开发过程中，多次跟产品确认需求，保持经常联系与反馈；

测试 在上线前主动跟测试人员多沟通，多反馈，不然临近测试会很多bug;

项目着手：

先理清项目的模块思路，然后查文档，整理好思路再写。

笔记整理：

每晚将当天的重点记录下来，最后将重点汇总成一个手册；

代码学习：

变量的变通： 一个变量如果不行，就尝试用其他变量；或者利用代码的顺序给出变量；

域名不要写死，要能根据环境变化而变化，域名是一个变量；

价格的计算，尽量交给后端处理，否则后端数据一旦改变，前端逻辑就要改变；

业务学习：

就大功能先做出来，再慢慢修复小细节。不在细节上面拖累整体进度；

所有涉及地址域名的，不要写死，给一个变量，随环境改变而改变；

找问题：

触发条件 + 检测问题是否会再次出现的地方 + =》 解决错误操作；

资料搜索：

树干检索法：比如：如果要找日历插件，去JQ之家找。

树叶检索法： 吴倩将问题分解成一个一个原理的知识点去搜索；

代码交接：

代码交接的时候，扔出方要在代码里面的逻辑归类并写好注释，一遍接收方的快速上手；