**杭州培训总结**

**培训时间**：

2018年5⽉4⽇～2018年5⽉25⽇

**培训⽬标**：

1. 了解果品交易平台技术栈，前、后端框架结构；
2. 基于gitflow的开发管理流程；
3. 熟悉后端基础架构，技术栈；
4. 熟悉页面及组件开发流程，前端技术栈，装修系统的使用以及在开发和运维中充当的角色；
5. 达到修改界面风格能力，组件的修改和制作。

**总结⼈：**

信 昕

**总结时间**：

2018年5⽉24⽇

**培训结果：**

截⾄24⽇，在端点科技的学习结束现已基本具备如下能力：

1. 独立页面开发，hbs模板页；

装修组件划分，根据页面活动区域划分；完整组件包括样式，模板，交互，初始化，装修配置5部分。可依据页面位置区域划分，原则是较少的并且竟可能单一的接口访问性质，可复用性，不可替代性，灵活性，任何组件的变更和新增需要有详细的效果和功能说明，例如，写明可配置参数；

通过组件和inject标签实现非装修页面手动拼装：只有新加的独立的1级页面可以这样操作；2级页由于路由限制和数据来源等问题，不建议通过这种方式修改，因为会涉及路由配置变更，数据加载等问题。这种手动引入的弊端是大量内容无法通过装修系统更改，现有的大量页面都是装修页，由于装修页没有实际存在的页面代码路径（所有通过装修系统生成的页面只有数据库里的组件位置关系数据而没有可见的直接能修改的页面代码），此方式只适合相对独立且大多数内容固定的页面使用比如位于app/views内的登录和注册页面，友好错误提示页等。

因此得出设计原则：如需修改已有相同功能的组件的样式就在原组件上直接修改，或开发新组件的hbs模板结构覆盖原组件的hbs模板结构，如果没有相同功能的组件需要新做组件，最后通过装修系统把新增或更改的组件装修到页面上，达到一次更改全部生效的目的；

1. 装修组件开发能力，装修组件存代码目录components和components\_vendor和components\_eevee的所有子目录均有；主要通过装修系统的组件名称，在各路径下resources的front\_config.yaml中查找这些组件的位置查找顺序components-》components\_vendor-》app/files；

config.coffee或config.se6配置文件，每一段参数设置代码是一个定义好的参数配置类的对象（引入的类定义在文件最上方），参数含义name：参数key值，label：参数标识装修系统可见，description：参数描述，type：参数类型，useData：true标识handlebars可直接引用，false不可引用，reRender：true标识改变此参数失去焦点后装修系统可立即看到效果，可配置参数对装修系统开放模仿现有配置，通过装修系统给组件配置少量关键性css样式、图片及文字等数据。

extend.coffee,extend.es6代表对原有view.coffee和view.es6的继承和扩展，首先要extends父类然后补充内容，我们修改时可以这样继承原有父类，也可以直接复制父类内容到components对应位置再做修改；

resources/yaml文件的首次数据装载backend\_config用于入参设定和后端访问接口配置，front\_config用于指定backend\_config和出参封装最终交给hbs模板；其中service默认\_DATA\_结果集，services可自定义结果集满足执行多接口访问，满足页面多原数据展示；

组件的二次ajax数据查询、表单提交、数据展示目前可模仿现有内容编写，复杂实现要依赖对handlebars，coffee.js和es6等大前端技术栈的掌握程度和长期经验积累。

1. 现有组件的修改和简单扩展能力，如果是来自eevee装修库或者vender基本库的组件需要在components中加入需要修改的组件的原有文件，shepherd通过扫描策略即可达到覆盖原有组件的效果，然后在components的新路径下对这个组件进行功能扩展，针对装修系统开放更多的可配置参数，然后重新用新组件装修达到扩展效果；
2. 前端目录结构大致了解，组件来源分为自定义库components，端点公用基本库components\_vender，装修系统基本库components\_eevee，shepherd打包时根据正则匹配文件方式，首先扫描components路径下，再扫描components\_vendor路径下，如果前者拥有同路径的hbs、scss、coffee或es6文件，components\_vendor中的同路径同后缀将不会加载，修改现有scss文件可以达到其中一小段内容单独提出来修改；
3. 理解前端架构router路由配置的原理和作用：简单的说是对特定url进行拦截，完成入参封装等业务操作，当后端接口与前端请求不能完全匹配时，可通过对url做路由定义来匹配后端接口，最终重定向到其他路径或实现页面层跳转，相当于前端的controller层；
4. 理解Shepherd扫描路径基本配置，打包路径修改，css、js、coffee、es6打包结构配置，新增独立打包；windows执行shepherd.cmd打包会首先shepherd.cmd clean清空public，components\_vender及components\_eevee目录，之后重新从端点服务器下载代码和打包；shepherd配置了base缺省路径，components、components\_vender、app/files的扫描顺序；
5. 理解Herd配置，装修数据库配置，redis负责登录状态，web-ui为了加载装修数据，注意url要和eevee数据库配置一致，本地运行配置host回调，本地配置默认pampasfile.js只开发过程中使用；

designer装修模板配置，这里编辑例如“有头模板”，“无头模板”等，对应app/view/eevee内hbs模板文件，这里配置是为装修系统要装修一个新的空白页所配置的布局方式，已知designPart中scope=site代表此区域被admin编辑保存后，其他角色不能更改。通过装修系统新建的页面不是项目路径下真实存在的页面，而是数据库中详细的位置关系数据。可以通过url访问，如果访问的url同时也是views内真实存在的hbs会优先加载这个hbs页面而不通过数据库渲染出装修页面。

1. 前端通用配置：

app/scripts安装js、jquery和第三方插件的目录，项目所有依赖的第三方插件，均要在shepherd.js内bundles下配置下载路径，每次打包时shepherd会自动拉取这些插件并根据打包配置加载到页面，配置方式参照现有bundles配置，查询可通过bootCDN官网查询路径；

app/style框架样式文件以及pokeball主题配置文件，简单的颜色改动可通过\_theme.scss配置整体更换；

用户身份权限配置方式，位于app/files/auth-config.js中配置了各种身份对应的访问路径权限，可加入路径绕过登录验证；

app/images存放有系统按钮、装饰等图片；

server/handlebars下存放有handlebars的自定义扩展标签，例如时间格式、加减法、比较等；

server/routers存放前端请求地址的路由配置；

1. 可操作Gitflow创建开发分支，合并分支，具备端点人员同等的代码管理模式；
2. 后端项目依赖关系已梳理清晰，user-用户等，item-商品等，彼此独立优先启动，trade-业务逻辑中间层，启动依赖user和item；web-ui-前台交互及接口与admin-ui-后台交互及接口，彼此独立均依赖trade。后端子工程启动依赖zookeeper-服务治理负载均衡、redis-缓存、rocketmq-消息中间件或其他mq、mysql-数据库、elasticsearch-搜索工具基础设施；
3. 后端自定义数据增、删、改、查接口实现，以及前端交互，数据展示的基本面完整流程能力，实际开发均需要基于完整业务流程及其设计实现；
4. 了解pass平台发布、使用方式，学习项目发布流程，feature分支合并，通过git substree命令发布并启动。

**总结：**

我此次参加培训的⽬标已完成，但任然需要保留对端点相关人员中短期持续追问和深究系统细节的通道，从而降低二次开发的风险，提高工作效率。前端界面基本能够持续性修改，需要我司美工及专业前端人员配合样式、图片美化等内容修改和编码。

同时也需要项目相关人员理解装修系统使用方式，具备与本平台开发模式相匹配的需求设计思维，新组件设计的能力、拿到完整页面后针对页面组件的拆分能力，合理确定哪些内容可以通过装修系统和现有组件实现，哪些内容通过技术人员修正现有组件样式实现，哪些内容需要技术人员开发新组实现，以便能够对日后新需求的工作进度和范围做合理的评估和提高工作效率。

对于现有运营系统、前台电商系统的业务，需要在端点提供完善⽂档的前提下，随着时间的积累，处理问题经验的积累，逐步完成对电商系统业务、运营系统业务的理解。