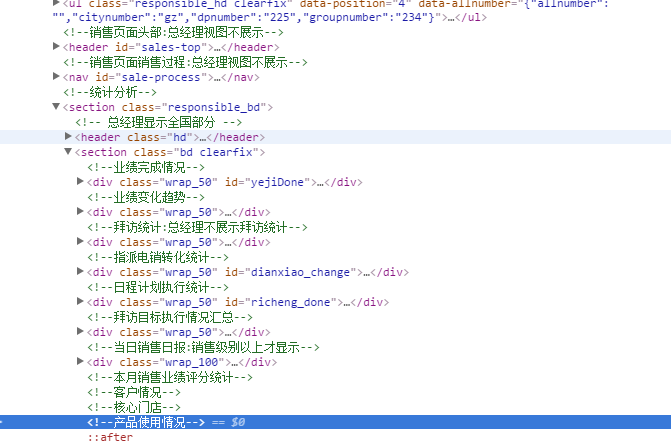
销控首页-纯前端首屏优化

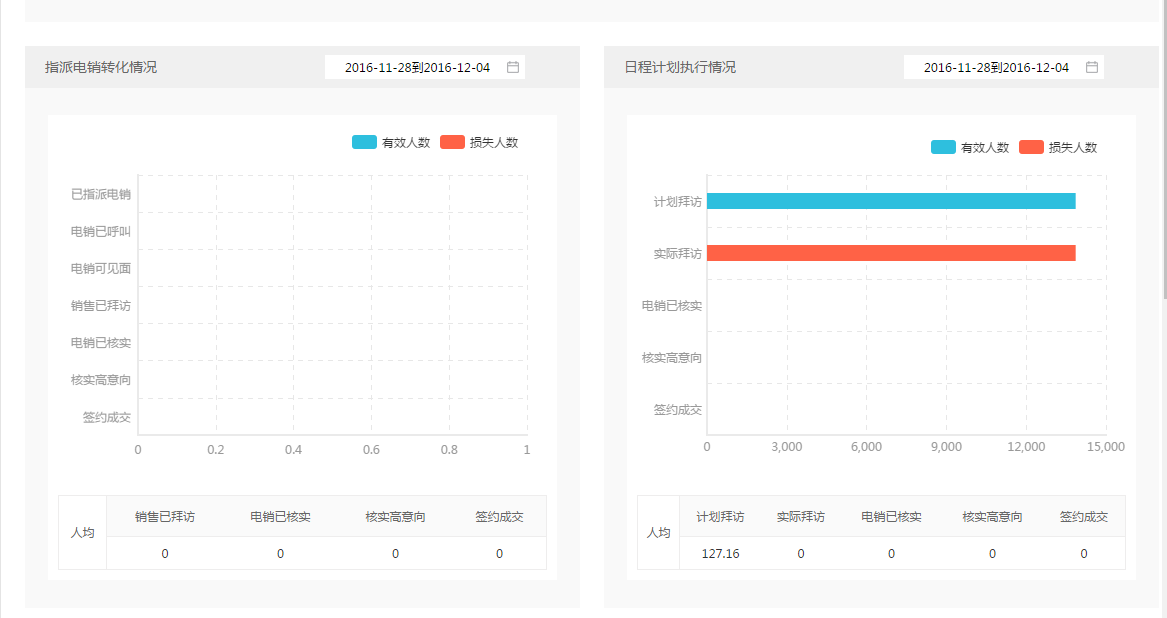
在本次项目的首页部分，因为业务需求，页面包含了极多的数据汇总模块，数据量和类型庞大，角色权限层次多，视图类型多，经过沟通，本次首屏采取视图架构HTML控制由PHP模版输出，架子里面的内容由前端请求接口异步按需渲染承担大部分的页面内容展示，以保证首屏加载的速度。

我们的角色权限分为5级，每级展示的视图模块有区别，这个模块业务的控制由PHP控制，渲染到页面可能就展示出类似于下图的流式布局架构：

前端根据HTML结构，判断模块唯一标识，比如id，来按需请求接口，这个过程中，我们团队划分模块进行开发，把各个模块独立成单个js文件，用webpack进行模块打包，防止冲突。

其次，我把首屏展示的模块的js逻辑和ajax请求置于入口home.js文件的顶部引用，这样可以稍微提前一些发起首屏依赖数据的ajax的请求时间。

第三个，其实多个模块具有可视化需求，引用了体积较大的echarts包依赖，通常遇到这样大体积的第三方包，我都会使用异步加载模块，来防止js同步资源文件的阻塞，原理是基于script标签的*async*属性。

四，组织模块切换条件异步请求接口时，使用缓存代理模式，例如：

这两个模块依赖时间切换，在这种情况，我们很容易想到使用缓存代理模式，保存不同日期条件下的模块数据，达到减少AJAX请求，保存复用的目的。

五，老生常谈的资源请求数减少，使用雪碧图和js合并技术，部分底部模块使用懒加载或者异步。

六，对于请求接口渲染的模块使用等待动画等手段提升用户体验，在这里我们经常使用的是CSS3的伪类：

.content{  
 min-height: 361px;  
 &:empty{  
 background: #f9f9f9 url(../../images/tel\_load.gif) no-repeat center;  
 background-size: 32px 32px;  
 }

就可以达到这样的效果：



七、改变DOM时，能一次性改变尽量不要多次改变，多使用整体结构内的innerHTML改变来重新渲染网页结构内容，而不是循环查找DOM内的标签，逐个更改，页面DOM改变重绘的代价极大，需要高效DOM来优化。

八、其余的就是合适css的复用和结构命名，到达减少样式文件体积的效果，同时保证js代码逻辑的通顺，非阻塞，避免不必要情况下异步的嵌套，同时优化代码结构，达到js代码层面的代码业务复用。

大概能想到的就这些，后续会继续补充。