**1.数据加载的方式，并行和串行如何实现？**

可以自行选择并行或者串行加载。

|  |
| --- |
| 通过一个对象配置并发的数据获取，比如：  datasource = {  'list': require('./datasource').remote('/model/list'),  'config': require('./datasource').constant('listConfig')  };  通过一个数组配置并发的数据获取，比如：  datasource = [  { 'config': require('./datasource').constant('config') },  { 'list': require('./datasource').remote('/model/list') }  ]; |

**2.etpl模板引擎怎么样？有更好的模板引擎吗?**

Etpl的优点是灵活，复用性高，可使用if、for等分支与循环语句，支持变量定义和替换。但是缺点就是不易读懂，使用HTML注释的形式，容易混淆。且没有错误处理，调试成本较高。

doT.js等，个人觉得，抛却性能上的差异等等，没有什么是最好的，只是不同应用场景和不同开发人员的习惯不同罢了。

**3.结合ubi讲一下er的执行过程？**

1. main.html中er.start()让controller、router和locator开始接管和监听。
2. 当url改变时，此变化被locator监听到，执行其重定向函数，将变化交给router进行处理。
3. router将请求转发给默认的controller。
4. controller将变化映射给对应的action，而UBI中action的注册是在er.start之前，于common/config.js中进行。Action首先触发enteraction，然后执行ation.enter。
5. action负责调用Model进行数据的加载。
6. Model的dataresource利用Deffered对象保证了数据加载完成。
7. Action调用view进行页面渲染。
8. View使用template进行渲染，同时可以进行局部元素的渲染。
9. Action中做初始化并绑定事件。
10. 当用户触发事件时执行相应动作。

**4.table为什么用enterChildAction，有哪些好处？**

代码复用，将table抽象成一个组件，可增加代码复用性。

模块化，增加可读性，同时可以将table数据分离在不同的model，防止因表格数据加载过慢导致页面长时间保持在空白状态。

**5.er的事件顺序？**

enter -> modelloaded -> beforerender -> rendered -> entercomplete -> beforeleave -> leave -> afterleave

**6.this.refreshtable的参数及其作用，这里的this指的是，回调函数里的this指的是什么？**

接受5个参数：obj, i, option, rendered, fail, timeout

|  |  |
| --- | --- |
| {JqueryObject} obj  {number} i  {Object} option  {Function} rendered  {Function} fail  {number} timeout | 时间所在的jquery对象  第几张表格刷新（tablearray顺序与html中table元素顺序一致）或者 刷新的表格名字  刷新表格时所传参数  成功时回调函数  失败时回调函数  从生成ajax的请求起的到超时的毫秒数 |

回调函数里的this指的是

**7.drawcallback是在什么情况下回调的？**

Action中initBechavier()中初始化dataTable对象时回调。此时数据加载、页面渲染都已经完成。 但是在entercomplete时间之前。

**8.前端配置了上维度和左维度，后端没有传，前端怎么显示，如果要做到自动隐藏，如何设置？**

前端配置的数据后端没传，就会显示短横杠'-'。

前端配置了上维度，但是后端没传，如果配置了autohide，则会自动删除此维度的显示。

如果配置了左维度，而后端没传，则会显示短横杠'-'。

自动隐藏，在新建表格对象的时候传入autohide=true，这样没有数据的列会自动隐藏。

**9.表格分页是如何设置的？**

表格对象初始化的时候需要用传入datatable：true，paging：true

**10.如何在绘制趋势图之前，修改config配置好了的属性？**

在新建对象的时候传入参数：setOptionBeforeDraw，它是一个函数，用于在绘制趋势图之前，修改config配置好了的属性。

**11.如何写注释，哪些情况下需要写注释？**

首先要符合代码规范。其次在一下几点需要写注释

1. 关键的函数以及函数名意义不明确的地方。
2. 逻辑不明显的代码段，比如多层循环。
3. 值得注意，有可能被其他函数使用的变量。
4. 重要的复用性高的函数。
5. 调用远处的函数，注明来源。
6. 等等……

**12.代码规范要注意哪些问题，你觉得哪些比较重要？**

首先是注释，在需要的时候写详细的解释。其次是结构清晰，逻辑相似的代码段最好能放在一起，易于阅读。但是大多数只要统一美观易于阅读即可。

**13.ubi现在前端js，css，less，图片的缓存方案是什么，怎么优化？**

缓存优化：

1. 使用代理服务器进行缓存。
2. 设置过期时间，比如每天缓存一次，或者每周缓存一次。

**14.ubi平台页面展示，还有哪些需要改进的方向？**

1. 前端还有许多值得美化和优化的地方。在不影响性能的前提下可以做一些动画效果，比如表格的滑动等等。
2. 展示的图表可以做一些交互的改进，比如拖曳关键词进行组表比较等等。
3. 现在的页面风格给人一种沉重的感觉，可以做的更加轻盈一点。

**14.ubi前端代码，有哪些改进的方向？**

1. 老的代码结构比较混乱，现在正在或者即将开发的代码可以更加整齐，结构清晰一点。
2. 模块化和代码复用方面任然有许多优化的空间。