**一、什么是前后端分离**

1、数据方面：  
所有的展现数据都是后端通过异步接口（ajax/json）提供的,前端只管展示

2、职责方面：  
前端：负责View和Controller层(会处理页面的一些逻辑，减轻后端的压力)。

后端：负责Model层，业务处理/数据汇总输出等。

**二、为什么要前后端分离（优点）**

这种开发模式主要影响工作效率方面（项目的推进）、维护方面、职责混淆

**对于前端开发人员来说：**

在以前传统的网站开发中，前端一般扮演的只是切图的工作，只是简单地将UI设计师提供的原型图实现成静态的HTML页面，而具体的页面交互逻辑，比如与后台的数据交互工作等，可能都是由后台的开发人员来实现的，或者是前端是紧紧的耦合后台。架构决定了前端只能依赖后端。

**对于后端开发来说：**

更有可能后台人员直接兼顾前端的工作，一边实现API接口，一边开发页面，两者互相切换着做，而且根据不同的url动态拼接页面，这也导致后台的开发压力大大增加。

总的来说，这回导致前后端工作分配不均，严重影响开发效率，而且代码难以维护。而前后端分离的话，则可以很好的解决前后端分工不均的问题，将更多的交互逻辑分配给前端来处理，而后端则可以专注于其本职工作，比如提供API接口，进行权限控制以及进行运算工作。而前端开发人员则可以利用nodejs来搭建自己的本地服务器，直接在本地开发，然后通过一些插件来将api请求转发到后台，这样就可以完全模拟线上的场景，并且与后台解耦。前端可以独立完成与用户交互的整一个过程，两者都可以同时开工，不互相依赖，开发效率更快，而且分工比较均衡。

**三、怎么做到前后端分离**

**前端**：负责View和Controller层(会处理页面的一些逻辑，减轻后端的压力)。实现整一个前端页面以及交互逻辑，以及利用ajax与nodejs服务器（中间层)交互

**后端**：负责Model层，业务处理/数据汇总输出等。提供API接口，利用redis来管理session,与数据库交互

具体各端工作的一个流程如下：  
 **前端：**一般来说，要实现前后端分离，前端就需要开启一个本地的服务器来运行自己的前端代码，以此来模拟真实的线上环境，并且，也是为了更好的开发。因为你在实际开发中，你不可能要求每一个前端都去搭建一个java(php)环境，并且在java环境下开发，这对于前端来说，学习成本太高了。但如果本地没有开启服务器的话，不仅无法模拟线上的环境，而且还面临到了跨域的问题，因为你如果写静态的html页面，直接在文件目录下打开的话，你是无法发出ajax请求的(浏览器跨域的限制，除非后端提供的接口允许跨域请求，添加白名单，或者是cors跨站资源共享，[详情见我博客](https://my.oschina.net/u/3407699/blog/1637825)）。因此，你需要在本地运行一个服务器，可是又不想搭建陌生而庞大的java环境，怎么办法呢？nodejs正好解决了这个问题。在我们项目中，我们利用nodejs的express框架来开启一个本地的服务器，然后利用nodejs的一个http-proxy-middleware插件将客户端发往nodejs的请求转发给真正的服务器，让nodejs作为一个中间层。这样，前端就可以无忧无虑的开发了

**后端：** 由于前后端分离后，前端和后台同时开发时，就可能遇到前端已经开发好一个页面了，可是却等待后台API接口的情况。比如说A是负责前端，B是负责后台，A可能用了一周做好了基本的结构，并且需要API接口联调后，才能继续开发，而此时B却还没有实现好所需要的接口，这种情况，怎么办呢？在我们这个项目里，我们是通过了mock来提供一些假数据，我们先规定好了API接口，设计出了一套API文档，然后我们就可以通过API文档，利用mock(http://mockjs.com)来返回一些假数据，这样就可以模拟发送API到接受响应的整一个过程，因此前端也不需要依赖于后端开发了，可以独立开发，等到后台的API全部设计完之后，就可以比较快速的联调。

**四、Nodejs**

**Nodejs做中间层**

1、我觉得引入nodejs主要是为了分层开发，职责划分，nodejs作为前端服务器，由前端开发人员负责，前端开发人员不需要知道java后台是如何实现的，也不需要知道API接口是如何实现的，我们只需要关心我们前端的开发工作，并且管理好nodejs前端服务器，而后台开发人员也不需要考虑如何前端是如何部署的，他只需要做好自己擅长的部分，提供好API接口就可以；

2、nodejs本身有着独特的异步、非阻塞I/O的特点，这也就意味着他特别适合I/O密集型操作，在处理并发量比较大的请求上能力比较强，因此，利用它来充当前端服务器，向客户端提供静态文件以及响应客户端的请求，我觉得这是一个很不错的选择。

最后贴上[个人博客](https://my.oschina.net/u/3407699)以及[美文一篇](https://github.com/lifesinger/blog/issues/184)

**五、现有项目存在的问题**

1、项目的数据量大、页面的交互部分比较吃性能，特别是地图、echarts图标会特别吃电脑性能，严重影响用户体检

2、目前项目绝大部分是有后端人员开发，不仅要处理业务逻辑、数据api数据输出等，还要处理本不擅长前端页面，开发进度缓慢

3、项目维护极其麻烦，严重影响项目推进

面对以上情况，对项目做前后端分离是最好的，整体的开发时间可以很好把握