# 网站内容导航功能的设计

bootstrap官网和百度贴吧顶部导航的跳转，其实和现在交易系统中的使用的技术是类似的，均是整页面跳转刷新。但是在项目结构的组织和url的跳转规范上，有很多值得借鉴的地方。

这里主要写一些我了解的相关需要注意的技术点：

1. 页面渲染方式

2. 页面url的restful规范

其实每个技术点内，都还有很多细节和优化方案，先不赘述。

## 页面的渲染方式

应用的页面主要是为了展现数据。在用户发出对一个页面的现在在用户请求一个页面的时候一般来说有下面方式渲染页面：

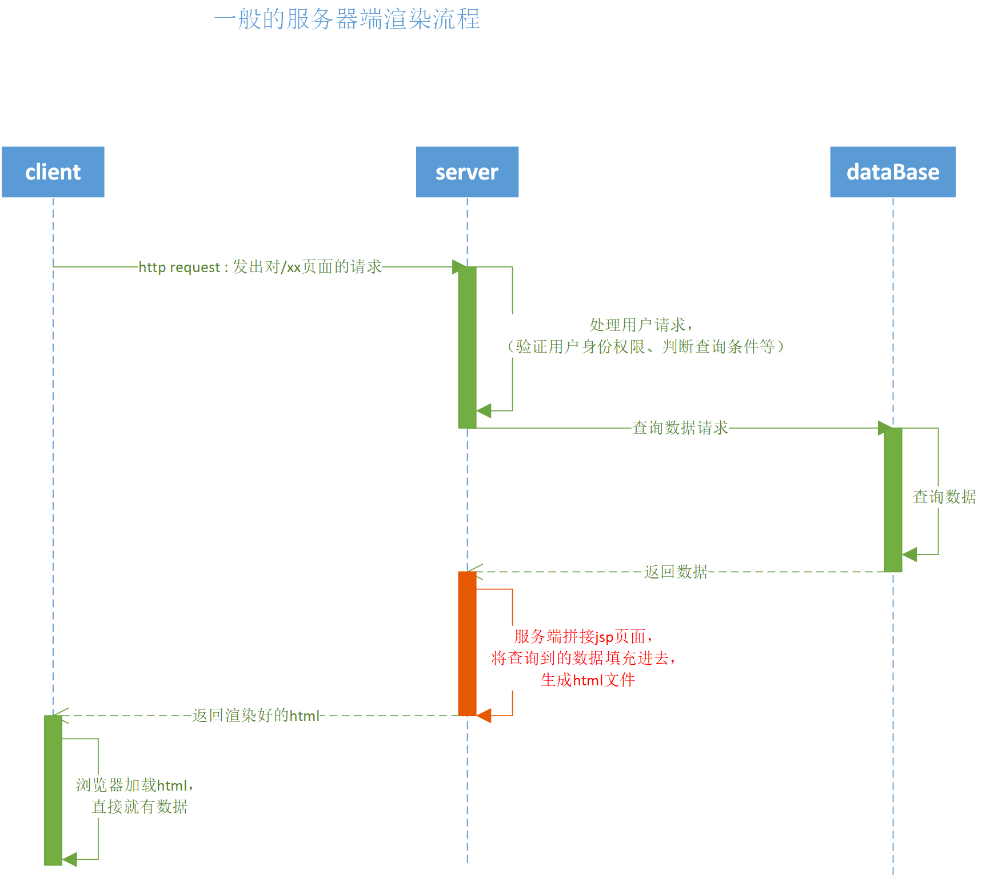
1. 服务端渲染页面（jsp，php，ejs等）

2. 客户端渲染页面（主要指的现代的单页应用框架vue/react/angular等，前后分离异步加载数据的方式）

**3. 单页应用的服务端渲染（ssr）**

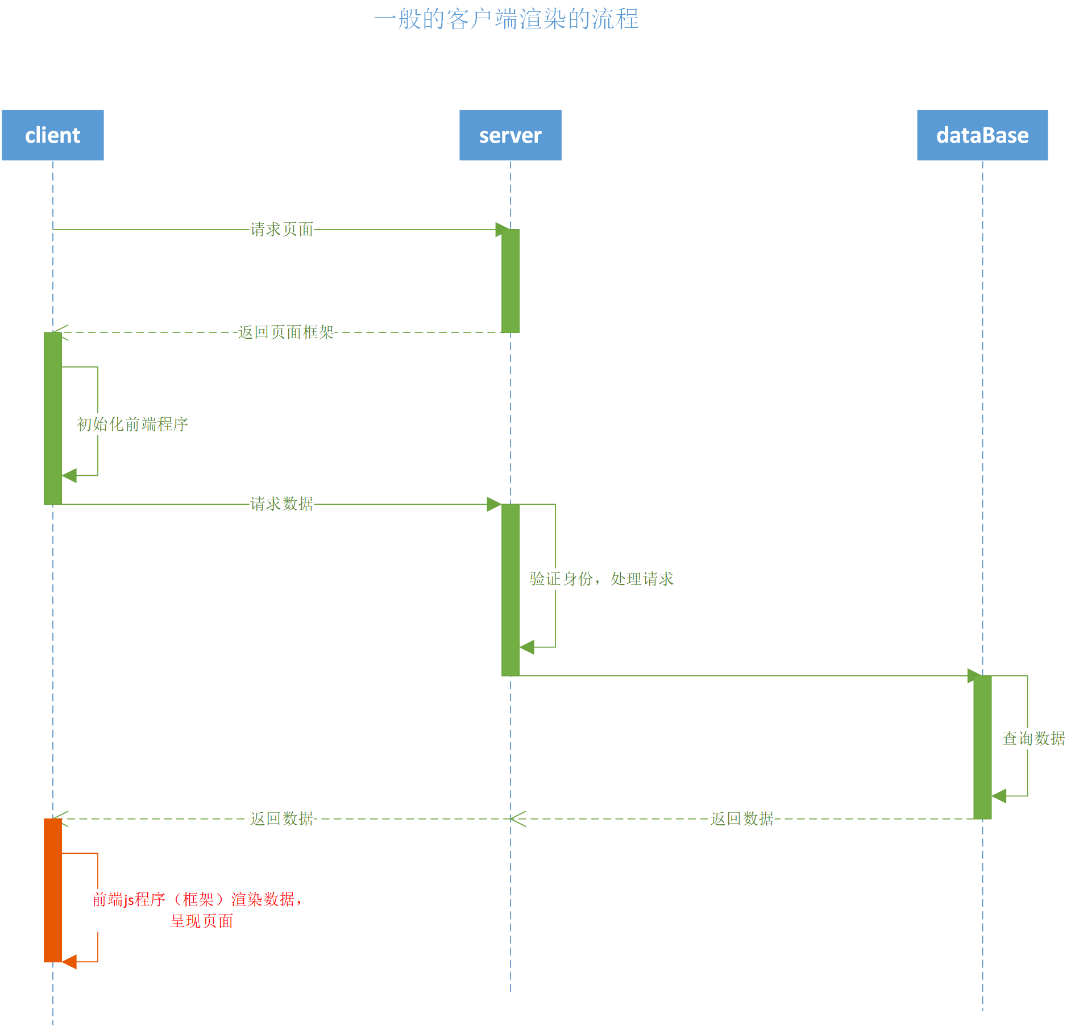
### 1.1 服务端渲染

即以往的jsp和php方式。用户发出请求，服务器处理请求，在服务端生成填充好数据的html页面，返回客户端。



### 1.2客户端渲染

广义的来说，数据使用ajax异步填充应该就属于客户端渲染。现在一般指的就是前端常用的那些单页应用框架。主要特点就是，前端首先加载出一个页面的“框架”，数据是之后通过异步请求，在浏览器端再向页面框架里面填充。

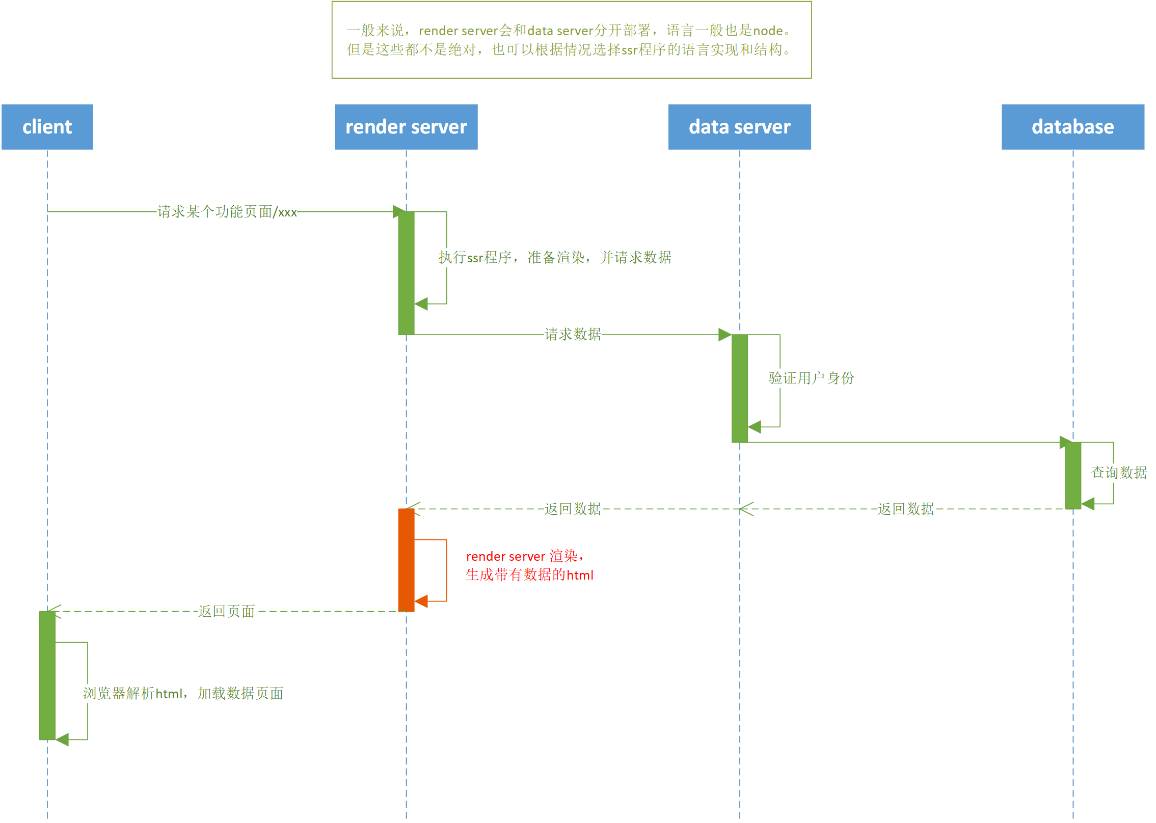


### 1.3 现在的前端框架的ssr

其实ssr就是server side render，也就是服务端渲染。1.2中的客户端渲染最大问题是对搜索引擎不友好。因为搜索引擎抓到的页面没有数据。

Ssr就是通过一个中间层服务器（一般来说，由前端开发人员维护），该服务器用来处理用户的请求，并将1.2中客户端请求数据和渲染数据的步骤交由该中间层实现。用户直接收到渲染好的页面。

这种情况下，用户的交互都是直接和node server进行。和数据紧密关联的后端服务都是通过node server转发，客户端完全不会感知。



### 1.4 三种方式各自适合的场景

其实，无论怎么渲染，最终还是一个在页面渲染数据的过程，不同就是，渲染数据的时机不同（图中红色的部分）。而且，后两种方式，其实都是前后端分离的开发方式出来后，才衍生出的技术。

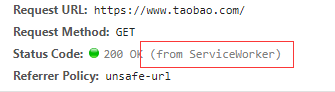
前后端分离，根本上来说，一是为了大型项目中的可维护性，传统项目中的，样式，js和后端逻辑代码耦合的过于严重，难以维护；二是随着前端的发展，前端也有了自己的模块开发方式。

应用场景的简单总结见下表：

|  |  |
| --- | --- |
| 渲染方式 | 适用场景和特点 |
| 纯粹服务端渲染 | 1. 项目前端比较简单，业务变更不频繁，前端不需要频繁修改维护的项目。 2. 纯粹信息展示类的网站。 3. 需要seo |
| 客户端渲染 | 1. 业务模块变更和新增比较频繁，比如后台管理系统这类网站。 2. 前端逻辑比较重，需要频繁修改前端的项目。 3. 不考虑seo |
| Ssr | Ssr现在主要指使用前端框架实现的服务端渲染。一些其他互联网公司可能也有自己实现的服务端渲染框架   1. 需要seo 2. 需要快速展示数据，省掉页面加载后，异步加载数据的延迟。 |

现在项目中，其实一个大型网站的页面都是**多种方式的结合**。纯粹服务端渲染和ssr都会增加服务器的压力，因而淘宝，京东这些商城网站首页都是直接在服务端渲染部分比较重要的数据，比如最热门商品，而其他分类商品数据都是客户端动态渲染的。

另外，为了优化速度，一般静态化技术和缓存技术都有应用，现代还出了一个service worker（PWA）技术来提高体验。淘宝就已经使用了service worker。



## 分析百度和bootstrap网站的导航的实现

一般来说，根据是否刷新页面，导航分为两类：**跳页刷新**和**局部刷新**。

局部刷新主要是现代的框架实现的页面内容区域刷新，同时更改了当前页面的url。

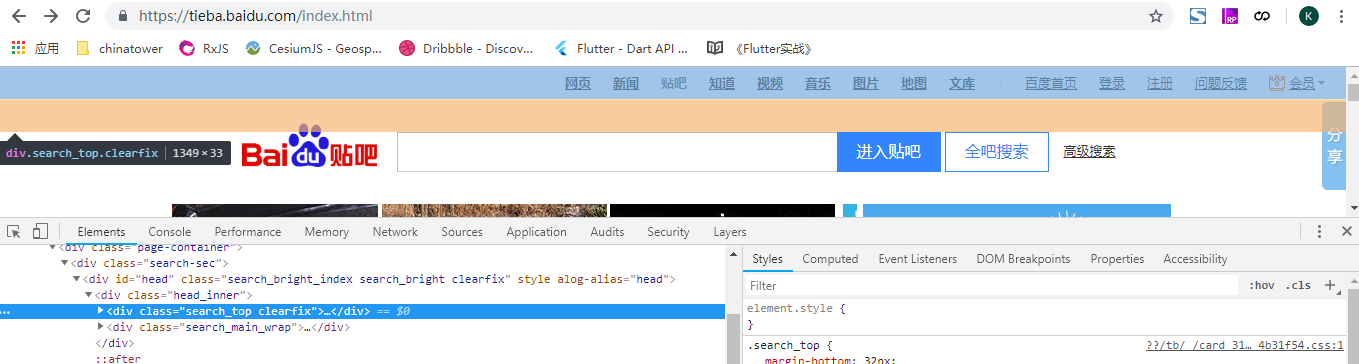
具体使用那种方式，其实是需要根据需求，产品，设计体验人员来研究决定的。技术上需要考虑的是，实现时候，url和代码结构的规范性。

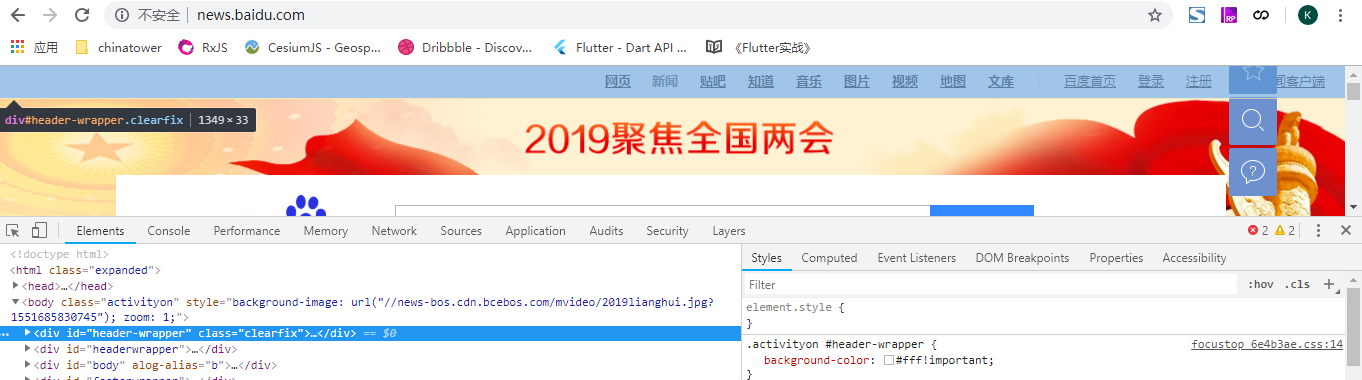
第一节写了一些渲染方式，是因为导航功能实现上，页面内容的加载是离不开渲染过程的。

### 2.1分析百度和bootstrap网站

首先，两个网站都是使用的跳转页面的方式刷新的，导航条部份也是使用的服务端渲染。

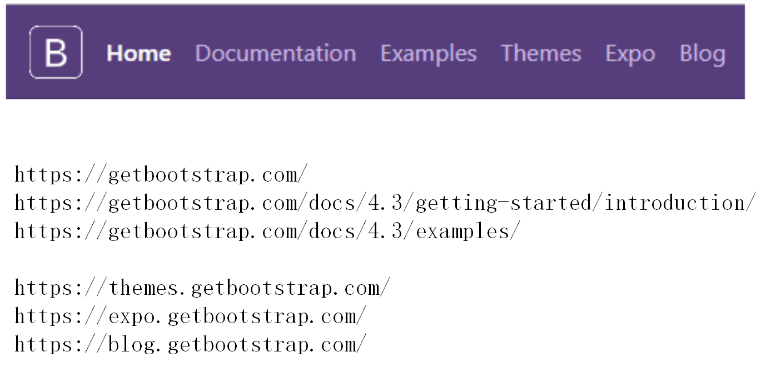
百度贴吧上面导航的每个链接，其实是面对的百度各个项目模块。在每个项目模块中，他们都是单独实现的顶部导航条，每个链接对应的都是**当前域名外的项目**，所以，百度这个，实际和我们交易项目并不相符。





如上面两张图所示，百度新闻和贴吧的顶部导航其实只是每个域名下的项目单独实现的导航条，并没有任何关联。本身就是两个项目中的东西，实现结构也不相同。

bootstrap网站也是服务端渲染。他的前三个链接对应的是同域名下**同一项目内**的不同模块。后三个链接则和百度那个一样，是纯粹的页面跳转，打开的是其他域名下的网站项目。



我们交易项目中，其实本应该就是做成同域名下的不同模块跳转，也就是bootstrap网站前三个导航那样。

这里需要注意，其实按照现代的导航设计要求来说，url接口地址应该按照restful的风格设计，这也是最好管理的方式。根据需求设计，将每种功能模块抽象成资源，每个资源对应层级式的url，数据的操作使用对应的http方法。

比如上面的同一站内的前3个url。根路径”/”对应Home；”/docs”对应文档部分；而文档根据版本又可以单独对应路径”/docs/4.3”或者”/docs/4.2”；并且，还可以继续向下细分功能模块。

目前我们项目中的页面地址和接口url以及请求方法，并没有一个统一的规范，项目的结构并不好立刻理清。

### 实现上的代码组织

其实，上面写了很多，仅考虑这个导航的话，实现上并不难，目前我们项目中也不需要引入新技术。

但在目前交易项目的基础上，改成这样的难点在于：目前项目的结构太乱，而且没有一个统一设计规范，并且代码的重复程度太高了。

仅仅头部组件，一个项目中就有10个左右。而且目前交易项目，虽是一个项目，但是却需要在3个端口部署3次，导航时候也不好理清应该跳转到哪个端口。如果跳错可能就会影响到功能。

理想情况的服务端渲染的，导航功能设计应该是下面这样：

1. 一个项目内，只有一个头部导航组件。
2. 用户请求来的时候，根据url就可以判断，用户请求的是哪个功能模块页面。从而可以在单独的一个头部组件内，十分简单的就能判断应该“激活哪个选项“。