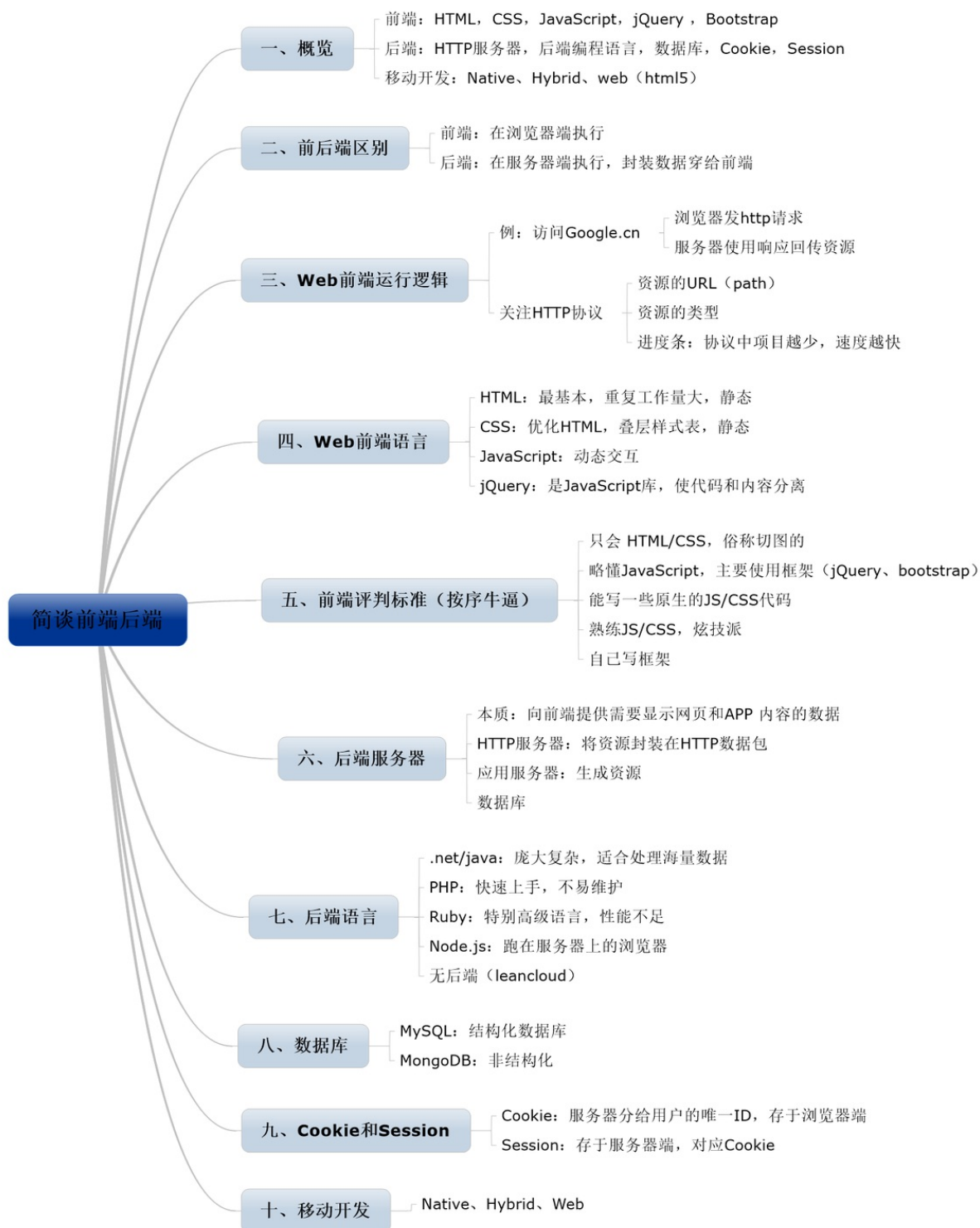


学习：简谈前端后端



作者 Nigel_ 关注

2015.05.28 23:48 字数 2154 阅读 10654 评论 6 喜欢 50



一、概览

前端知识: HTML, CSS, JavaScript, jQuery, Bootstrap

后端知识: HTTP服务器, 后端编程语言, 数据库, Cookie, Session

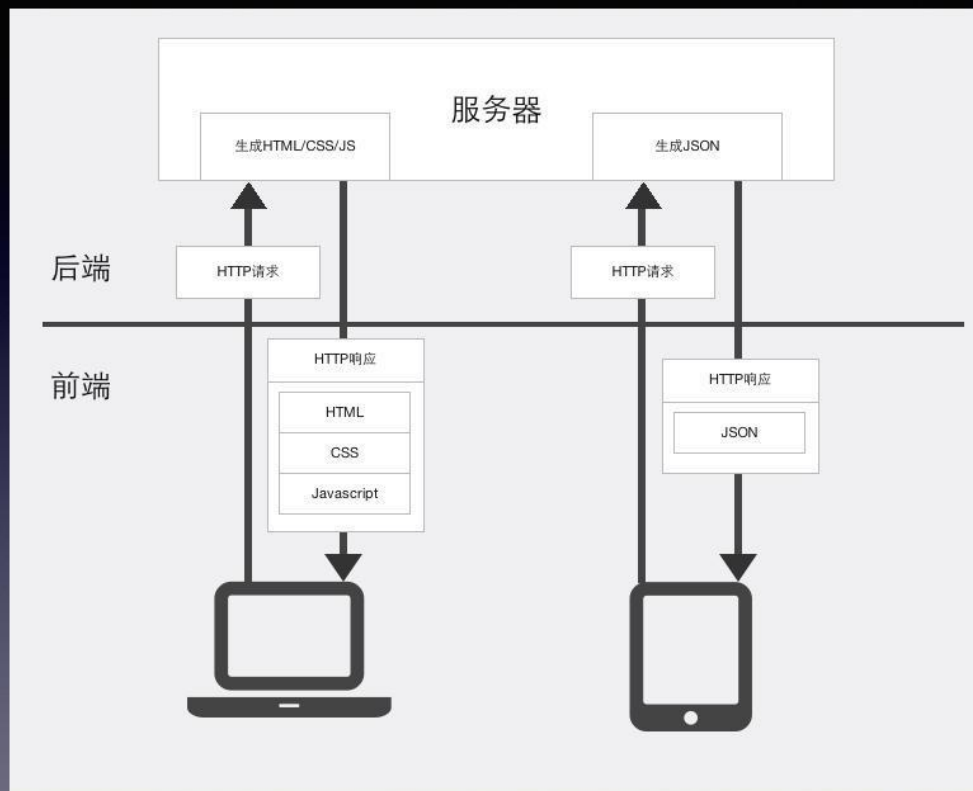
移动开发: 原生, 混合式, HTML5, 不同的移动端技术选择在功能和开发成本上的比较



二、前后端的区别

1、前端: 简单理解就是, 在浏览器端执行, 凡是运行在用户设备上的技术都可以称为前端技术 (比如HTML / CSS / JS, 甚至移动设备的Obj-C / Swift)

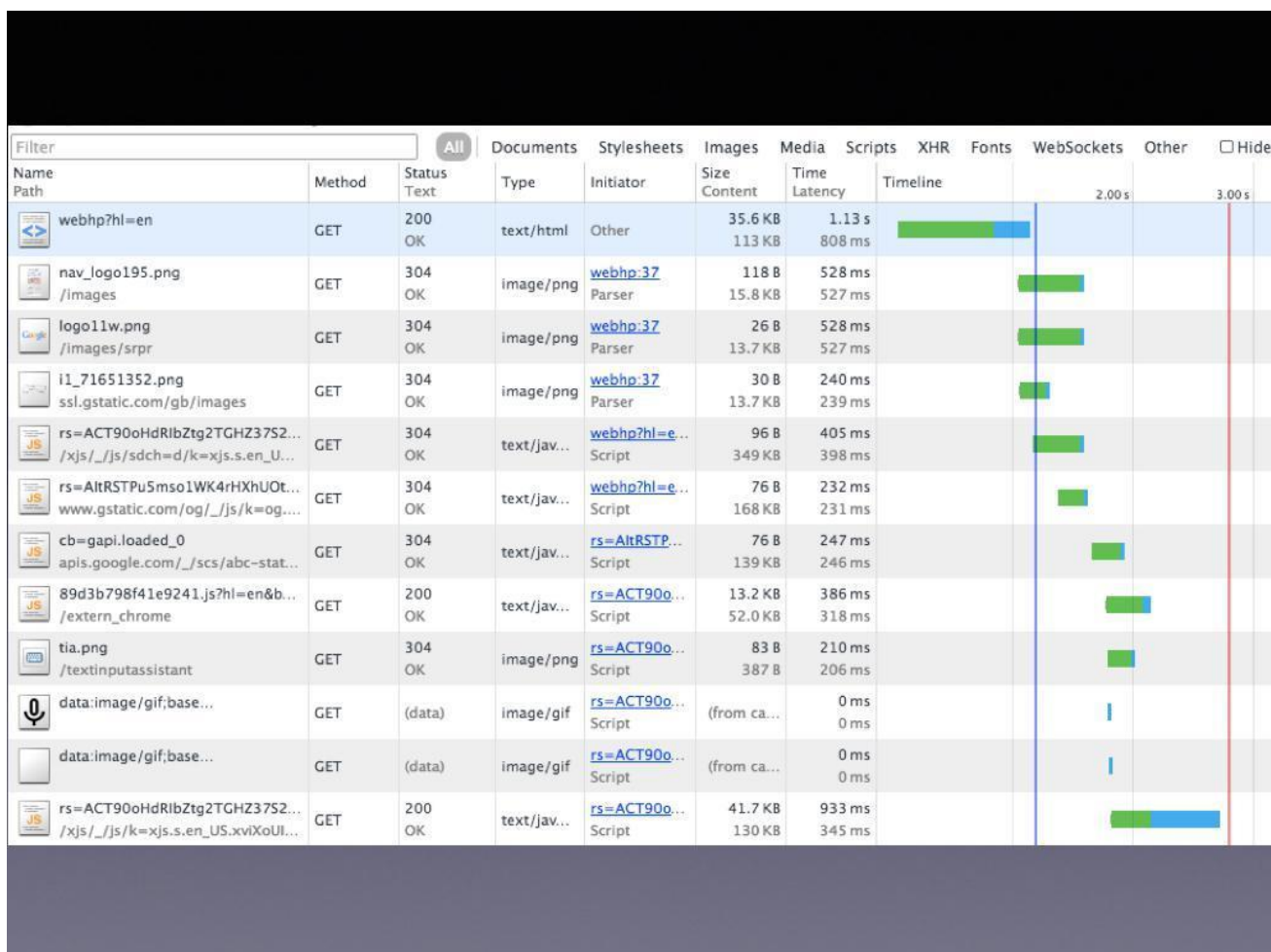
2、后端: 在服务器端执行, 负责将上述代码封装在http的数据包中然后通过网络传送到前端; 另外一个功能是保存和提供用户数据, 比如移动端常见的JSON就是目前最流行的在后端和前端之间传输的一个文件格式。



三、Web前端的运行逻辑

例：访问Google.cn

- 1、浏览器向Google的服务器发送一个http请求
- 2、服务器使用一个http响应，把显示这个网页所需要的资源传回给了浏览器



可以通过Chrome浏览器的开发者工具来进一步观察HTTP协议的运行情况；

上图为Google的HTTP协议运行情况，关键部分为：

第一列，即资源的URL（path）；第四列是这个资源的类型；

在第一个请求和后续的请求之间有一根蓝线，即进度条。而HTTP协议中运行的项目越少，浏览器加载的速度越快。图中Google就处理得很好，只有10个左右的请求。

四、Web前端语言

1、HTML和带样式的HTML

HTML就是一组标签和文本的组合，是一个最基本的网页。它已经包含了网页常见的元素，实际上在Web早期的很长一段时期内，网页都是这个样子。后来随着使用网络的人群越来越广泛，在HTML3.0中引入了对网页样式的定义，某种程度上可以说，也是从这个时候开始产生了网页设计师的角色。（现在已经是html5）

2、CSS

带样式的HTML也拥有一个缺点，它需要为每个标题和文字都设定样式，工作量非常庞大。

CSS就是在这样的情况下诞生了。CSS，又称叠层样式表，简言之是一种用来表现HTML文件样式的样式设计语言。CSS能够对网页中的对象的位置排版进行像素级的精确控制，实现基础的静态的交互设计；而CSS目前的最新版本CSS3能够真正做到网页表现与内容分离。

3、JavaScript

差不多在CSS诞生的同一时间，大家开始觉得这样静态的网页似乎略显无聊，能不能给网页加入一些可以动起来的元素？比如点击一个按钮之后变个颜色。当时网景公司的工程师Brendan Eich就给他们自家的浏览器引入了这种实现动态效果的脚本语言，这就是Javascript（简称JS）的诞生。所以通俗来说，Javascript就是用来给HTML网页增加动态功能，实现更炫酷的交互。

4、jQuery

提到Javascript，就得提一下jQuery。jQuery是一个优秀的Javascript库。jQuery使用户能更方便地处理HTML，它能够使用户的HTML页面保持代码和内容分离,通过jQuery，可以不用在HTML里面插入一堆JS来调用命令，只需要定义ID即可。此外，由Twitter设计师Mark Otto和Jacob Thornton合作开发的Bootstrap也是一个受欢迎的前端框架。

五、优秀前端评判标准（按序牛逼）

- 1、只会HTML/CSS的，这种我们俗称切图的，基本上就是淘宝几十块切一张图的；
- 2、懂一些简单的Javascript，主要是使用一些现成的框架，比如jQuery，bootstrap等等；
- 3、知道jQuery和Bootstrap的局限，必要时能写一些原生的JS/CSS代码；
- 4、对JS/CSS非常了解，执着于使用浏览器的各种最新特性来实现各种炫酷效果，这种我们成为炫技派；
- 5、可以自己写出类似jQuery / Bootstrap这样的前端框架供其他人使用。

六、后端服务器

- 1、本质：是向前端提供需要显示网页和APP内容的数据，可能是HTML，也可能是JSON数据，也可以是音视频或者PDF文件。
- 2、一个服务器包含3个部分：
 - a)HTTP服务器：把（需要返回给客户端的）资源文件封装在HTTP数据包里；
这个资源有可能是它后面的应用服务器动态生成的，也有可能是保存在硬盘上的静态文件；
这个资源是所有后端程序都必须有的，也是直接和我们的浏览器通信，返回给我们数据的程序；
这个资源作用就是把它（后面的编程语言生成的各种HTML/CSS/Javascript），打包成一个HTTP请求，然后再封装到一个TCP/IP的数据包里发回给浏览器端；
最常用的两个HTTP服务器叫做Apach和Nginx。
 - b)应用服务器：生成前端需要的HTML/CSS/JS交给浏览器
 - c)数据库

七、后端语言

- 1、.net/java
庞大，复杂。但Java的优点就是适合处理特别大的数据量，如果你的项目会很快实现大爆发，需要处理海量的请求，那么Java是一个不错的选择。
- 2、PHP
可以快速上手，相比其他语言，可以更快的为应用添加各种新功能。当然，可维护性就另当别论了。
- 3、Ruby
非常接近自然语言，基本上即使不懂编程，也能看明白70%或80%。，快的同时，他的最大短板就是性能。Twitter最早就是使用Ruby on Rails开发的，但是随着用户数的逐步增长，Twitter的宕机开始变得非常频繁，后来他们迫不得已将整个系统从Ruby迁移到了一个从Java派生出来的语言Scala。
- 4、Node.js
简单来说，可以把node.js理解为跑在服务器上的javascript，再直白一点，就是一个跑在服务器上的浏览器，因为node.js最早就是从chrome浏览器的Javascript引擎V8中剥离出来的。相比Ruby，Node.js程序可以获得更高的并发性能，这在一些高并发的场景下（比如群聊，多人协作等）会很有优势。
- 5、无后端（leancloud）
无后端编程是最近的一个新趋势，但她并非说是真的没有后端，而只是把后端交给一些第三方的云平台，比如Leancloud，Firebase等。如果你开发一个手机App，这样的好处就是你可以在早期没有后端程序员的情况下快速开工，像Leancloud这样的云平台已经可以胜任大部分的应用场景，如果后期业务逻辑复杂之后再寻找合适的后端工程师迁移也不迟。

八、数据库

- 1、MySQL（结构化数据库）：表结构固定，查询需用SQL（结构化查询语言）
- 2、MongoDB（非结构化）：所以可以操作某一行数据，而不用增加新列

九、Cookie和Session

Cookie理解为是服务器给每个用户分配的唯一ID，这个ID由用户浏览器保存；
Session则是服务器为了维护这个会话在服务器端保存的与cookie对应的用户数据。

十、移动开发

- 1、移动端和浏览器的区别就在于，大部分App，我们打开的一瞬间，就已经看到了它的界面，而不用再去向服务器来拿显示界面的HTML等文件。所以移动端，开发原生应用所运用到的技术（比如Objective C，swift）就相当于前端的HTML，只不过它是直接保存在应用本地的（原生）。
- 2、这样就产生了一个问题：如何来获取应用数据？如果是网页应用，我们可以直接将数据包含在HTML中一并反馈给浏

览器；但是对于移动应用就需要有一个专门的协议来传送应用需要的数据，这就是JSON。

3、移动应用三种类型：原生、HTML5、混合型