河北师大软件学院 *@Software College*

前端开发与HTML5
程序设计基础

王岩

自我介绍

- 王岩
- QQ：290320527
- 邮箱：290320527@qq.com

课程教师

- 刘孟祎
  - QQ：2242656886

第一章 了解前端开发

何为前端开发?


那前端是啥?
就是开发你们人类
能看见的那部分。
呃……

“在软件架构和程序设计领域，前端是软件系统中直接和用户交互的部分，而后端控制着软件的输出。将软件分为前端和后端是一种将软件不同功能的部分相互分离的抽象。”

–wikipedia

前端技能应用

- Web应用
  - 举例:
    - 京东
    - 网易云音乐
    - 饿了么
    - processon.com
    - …

前端技能应用

- 移动Web应用
  - 举例：
    - 京东移动Web版
    - 微博移动Web版
    - …

前端技能应用

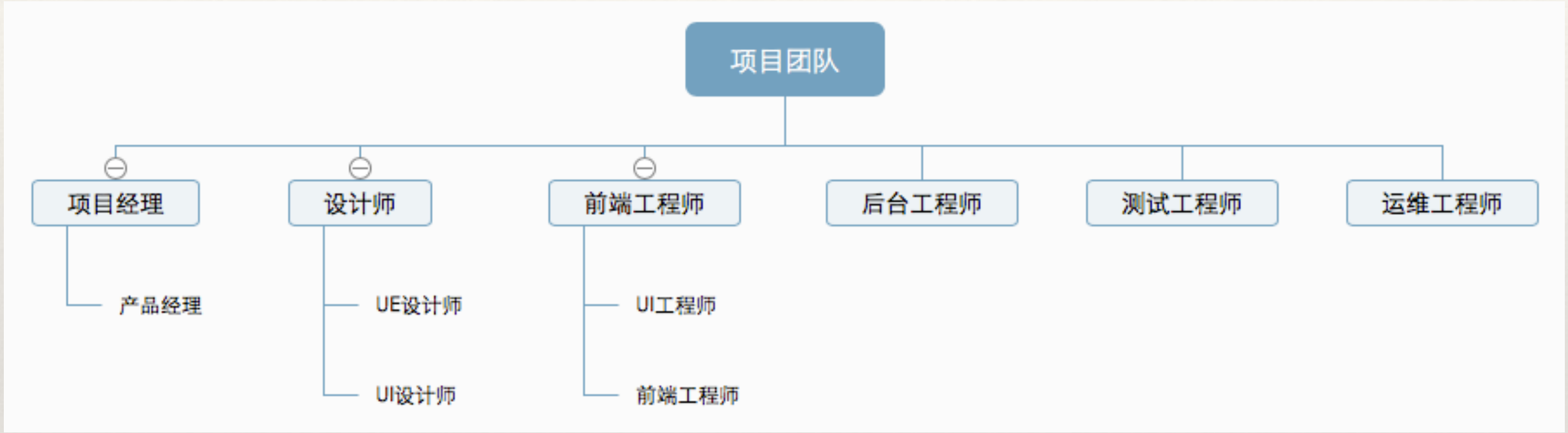
- 游戏应用
  - 网页游戏
  - 移动端网页游戏
  - 举例:
    - 围住神经猫
    - 微信平台，砸金蛋、老虎机...
    - ...

移动App

- HTML5 App
- Hybrid App

相关岗位

项目团队构成


项目团队
项目经理
产品经理
设计师
UE设计师
UI设计师
前端工程师
UI工程师
前端工程师
后台工程师
测试工程师
运维工程师

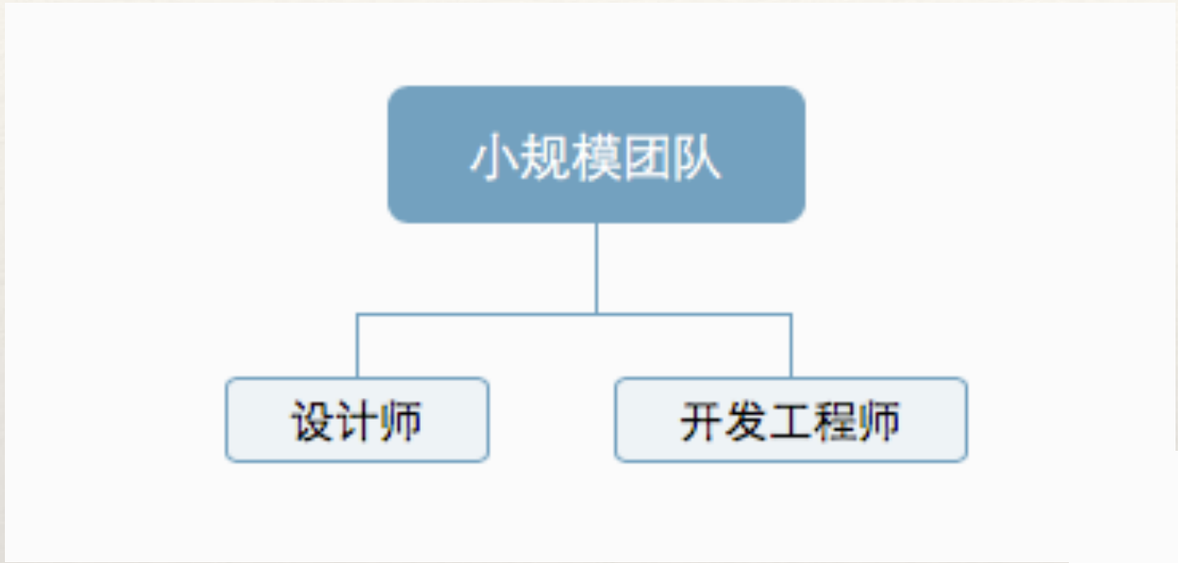
UI工程师

- 主要负责静态页面构建，保证非常精细的设计还原
- 主要技能
  - 切图
  - 页面构建（HTML、CSS）
  - 语义化
  - SEO
  - 页面性能

前端工程师

- 主要负责，页面动态数据、页面交互逻辑与效果。
- 主要技能
  - JavaScript
  - 模块化工具
  - 组件化工具
  - 模式框架
  - 构建工具
  - 自动化测试工具、代码质量工具
  - …

小规模团队



全栈（Full-stack）工程师

- 全栈：为了完成一个项目，所需要的一系列技术集合。

前端领域技术现状

前端大牛们都学过哪些东西?

最近在看bootstrap，发现除了大一的时候看过的html+css,和一些js，JQuery之外，几乎没学什么关于前端的东西。偶尔了解过一些html5。想知道如果作为一个团队的前端负责人还需要学习哪些东西？发现bootstrap与.less有关，除了这个还有哪些是需要学习的？希望得到各方大神的回答。

补充一下，一些基本的计算机方面的东西还是学习过的。就是说除了最基本的像楼下马前辈说的这些之外，还需要哪些?

https://www.zhihu.com/question/22146521

1461

张铮铮

Kim Jee、SThy Zun、王鸿宇 等人赞同

一步一步来。

CSS不能编程？用Less 、Sass 、Stylus 、甚至直接用 Absurd ，框架除了Bootstrap 还有很多。JS写多了很麻烦？jQuery 。移动开发？Zepto.js 。结构不好？找框架，Backbone.js 是MVC，AngularJS 和Ember.js 是MVVM，Twitter还弄了个事件驱动框架Flight 。库多了要优化加载？RequireJS 。

代码质量成问题？Jasmine 、QUnit 、Mocha 做单元测试。各种浏览器都要测？用Karma 。测试通过了部署还有问题？持续集成，用Travis CI 。用户行为也要测？用Selenium 。样式测试还有Viff 。觉得JS都够麻烦的？用CoffeeScript 。

想做动画？Canvas或SVG还有CSS3帮忙，干掉Flash。SVG太难画？用Snap.svg 。想开发游戏？用Canvas。自己写FPS太低？用框架，CreateJS. 。2D太幼稚？three.js 帮你用WebGL开发3D，还不够给力？asm.js 让你在浏览器中拥有虚幻3引擎。

这一堆东西都要配置部署，麻烦，用Grunt ，库太多？用Bower 管理，项目开始要创建各种文件文件夹？用Yeoman 。开源项目太多了，GitHub. 上找，不会？学Git 。顺便用Jekyll 托管博客，不是吧还有Ruby 这玩意...SASS也是Ruby写的，等等Sublime Text 是Python 写的，要写插件？也学一下。调试太难？用Chrome开发者工具，一堆API和功能。

光在电脑浏览器上跑不给力？移动开发HTML5，离开网络就渣了？HTML5离线应用。不如原生应用？用PhoneGap 。想调用原生API？开发Firefox OS 应用吧。浏览器应用也得会吧，Chrome Firefox都有自己的文档。接着是不是把后端甩了，自己来，装Node.js ，所以还得学点服务器知识，想用npm管理node包？linux技巧shell神马的也得学。想前后端通吃？再看看http协议。Web精通了？node-webkit 让你可以写桌面程序了，继续学吧。

想学模块化开发？看看CommonJS 和AMD 规范。理解JS有偏差？看看ECMA-262 ，等等不知道什么时候第6版就要出了。浏览器各不相同，弄不清该怎么兼容？看看W3C 标准，HTML写出来人看的懂，机器读不懂？要SEO，要支持残障人士？看HTML语义化，全会了但IE就是不支持？叫不出名字的浏览器尼玛连JS都不知道是啥？渐进增强。想一次把各种设备全搞定？响应式设计。

然后上面这些不过是一些讨巧的小技术。公司做什么业务的？了解一下行业信息。面向大众的产品？交互设计。美工不给力？UI设计。外包和咨询？设计模式、重构方法、算法、数据结构。知道软件工程吗？了解一下敏捷开发 ，或许还可以试试TDD、ATDD、BDD。

看了这么多东西，第一反应是不是求中文文档？学英语去吧。

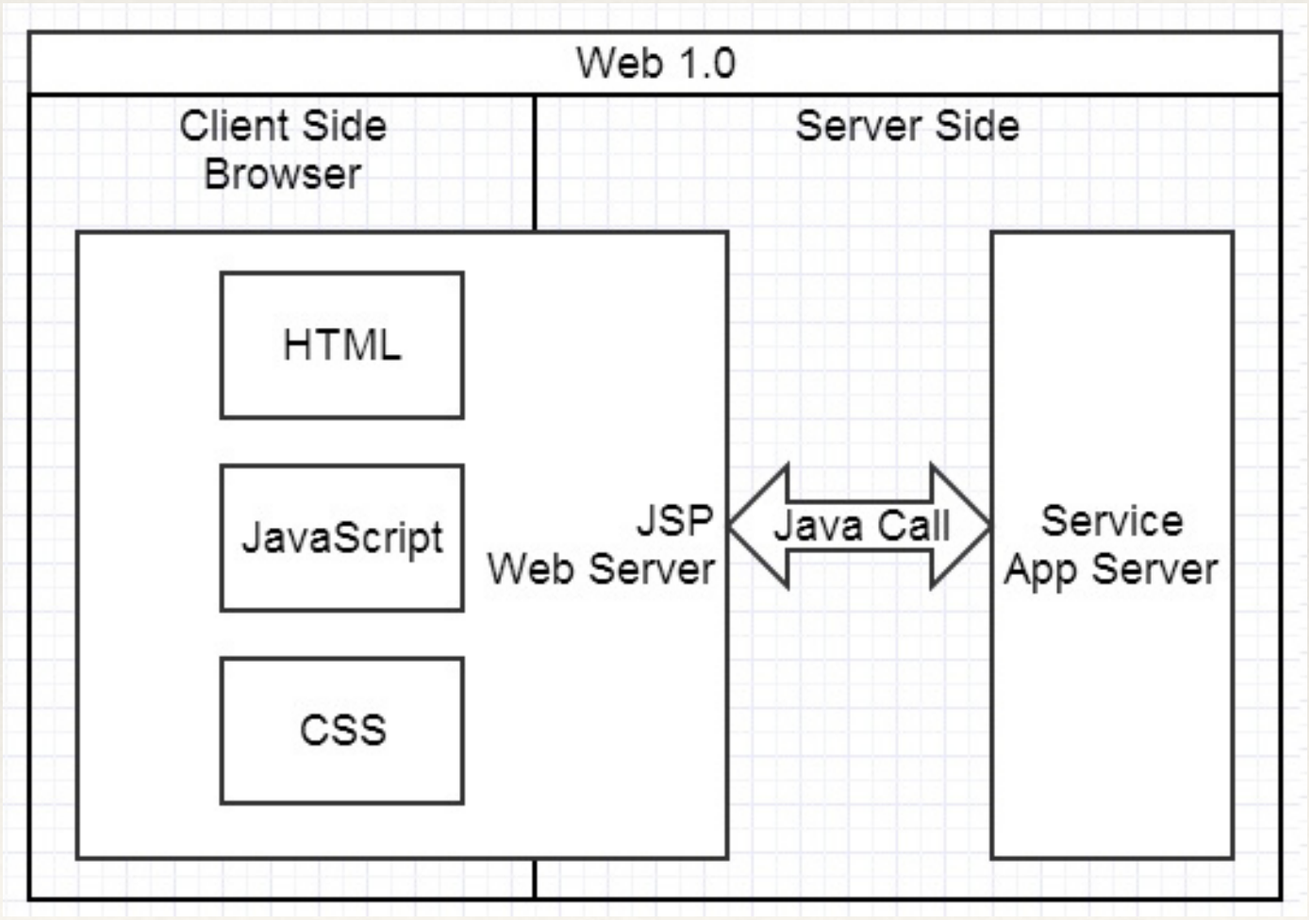
Web Page -> Web App

- 编程技术及生态发展的三个阶段（@xufei）
  - 最初的时候人们忙着补全各种API，代表着他们拥有的东西还很匮乏，需要在语言跟基础设施上继续完善
  - 然后就开始各种模式，标志他们做的东西逐渐变大变复杂，需要更好的组织了
  - 然后就是各类分层MVC，MVP，MVVM之类，可视化开发，自动化测试，团队协同系统等等，说明重视生产效率了，也就是所谓工程化

不仅欣欣向荣，而且野蛮生长。

一、简单明快的早期时代

- 不区分前后端，页面由JSP、PHP等工程师在后台生成。

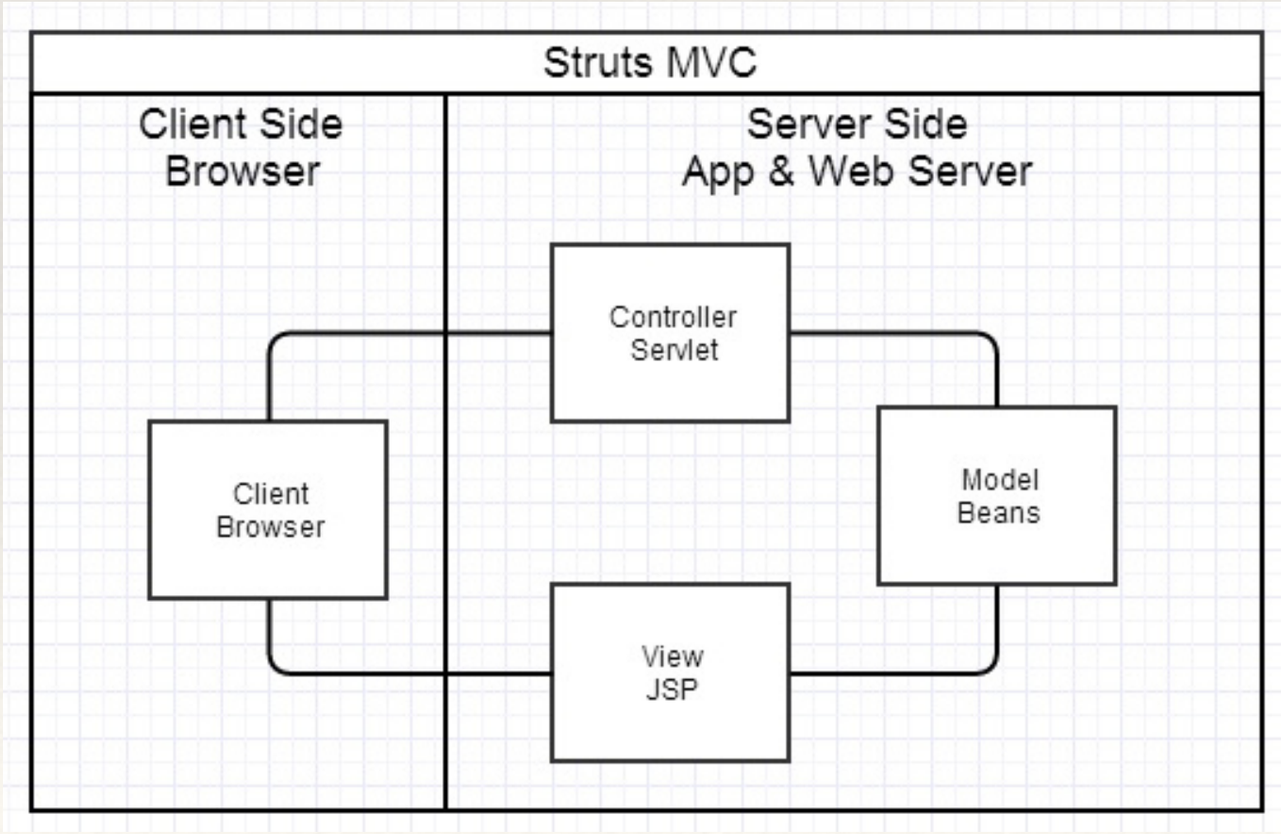


一、简单明快的早期时代

- 典型问题
  - JSP 等代码的可维护性越来越差
  - 服务越来越多，调用关系变复杂

二、后端为主的 MVC 时代

- 以后端为出发点，Web Server中架构升级，比如：Struts、Spring MVC
- 前后端开发人员分离

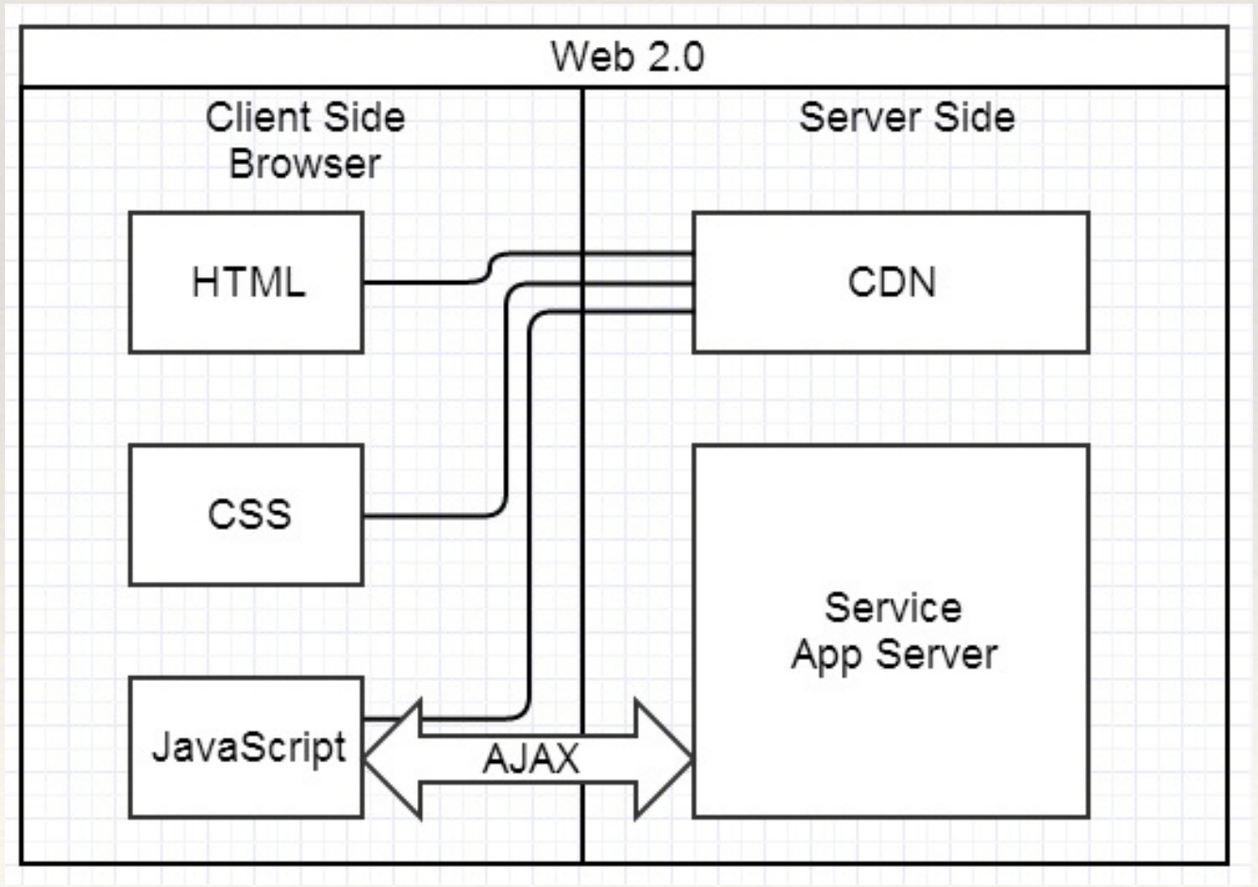


二、后端为主的 MVC 时代

- 典型问题
  - 前端开发重度依赖开发环境
  - 前后端职责依旧纠缠不清

三、Ajax带来的SPA时代

- SPA （Single Page Application 单页面应用）
- 用户体验的提升
- 前后端的分工非常清晰，前后端的关键协作点是 Ajax 接口。

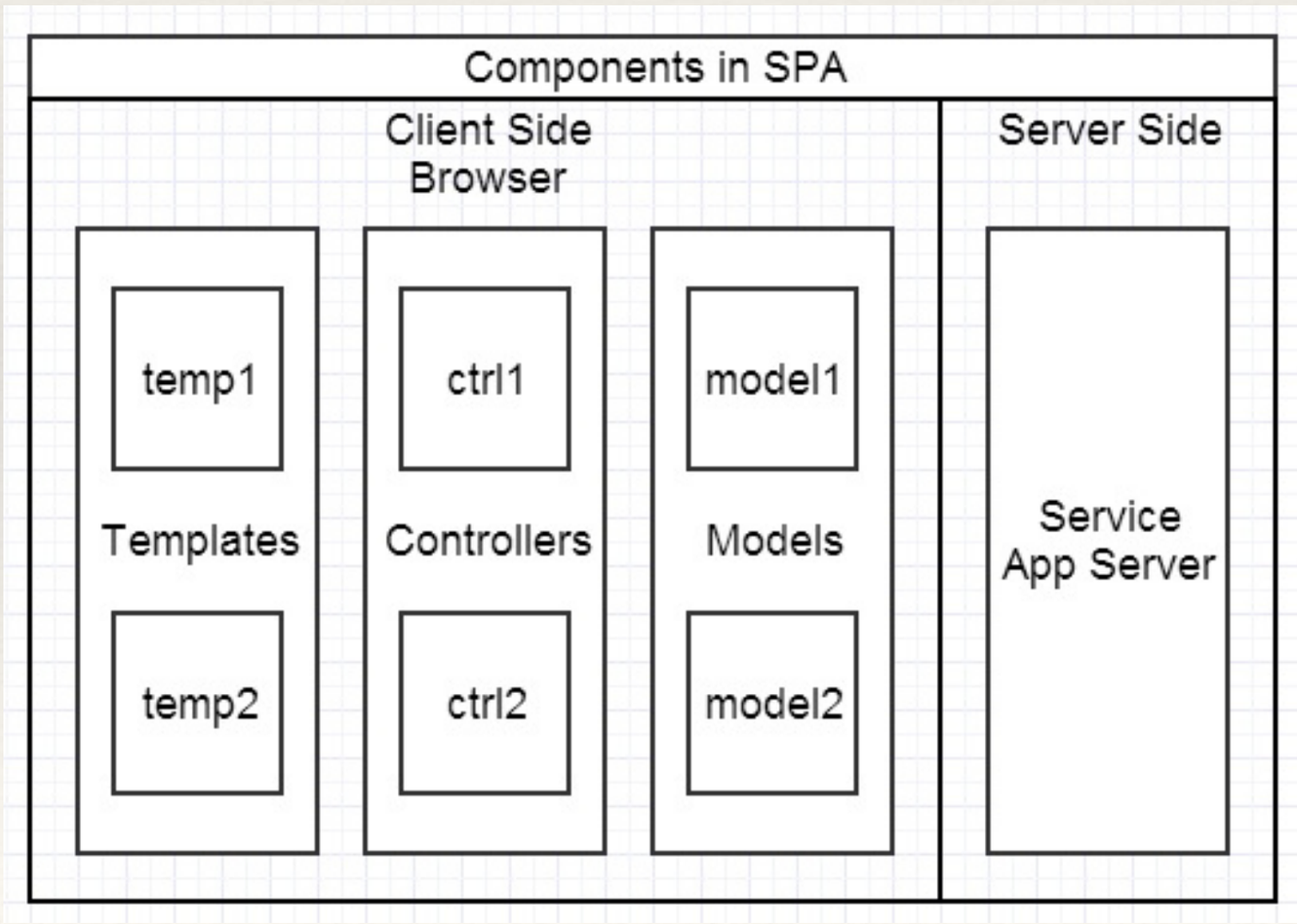


三、Ajax带来的SPA时代

- 典型问题
- 前端开发复杂度提升
- 前后端接口的约定

四、前端的 MV* 时代

- 为了降低前端复杂度，出现大量框架，进行前端应用切分。
- 例如：Backbone、Knockout JS、Angular JS



四、前端的 MV* 时代

- 优势：
  - 前后端职责清晰
  - 前端开发复杂度可控
- 不足：
  - 代码不能复用
  - 全异步，对SEO不利
  - SPA不能满足所有需求，依旧存在大量多页面应用

前端的痛处

JavaScript的问题（一）

- 命名冲突
- 依赖关系

```
<script src="${path}/theme/plugins/zui/lib/jquery/jquery.js"></script>
<script type="text/javascript" src="${path}/theme/plugins/zui/js/zui.min.js"></script>
<script type="text/javascript" src="${path}/theme/plugins/zui/lib/datetimepicker/datetimepicker.min.js"></script>
<script type="text/javascript" src="${path}/theme/js/app.js"></script>
<script type="text/javascript" src="${path}/theme/plugins/DataTables-1.10.8/js/jquery.dataTables.min.js"></script>
<script type="text/javascript" src="${path}/js/LodopFuncs.js"></script>
<script type="text/javascript" src="${path}/js/print.js"></script>
<object id="LODOP_OB" classid="clsid:2105C259-1E0C-4534-8141-A753534CB4CA" width=0 height=0>
    <embed id="LODOP_EM" type="application/x-print-lodop" width=0 height=0></embed>
</object>
<script type="text/javascript" src="${path}/theme/js/chargeMain.js"></script>
```

模块化

- 模块化工具
  - CommonJS 规范
    - Node JS
  - AMD 规范
    - Require JS
  - CMD 规范
    - Sea.js
- ECMAScript 6的模块系统

JavaScript的问题（二）

- JavaScript语法缺陷
- this引用
- 原型继承机制

语法改良

- 衍生语言
  - CoffeeScript
  - TypeScript
  - …
- ECMAScript 6
  - Arrow Functions
  - Class

CSS的问题

- 样式庞杂，书写、维护困难

```
6779  .visible-print-inline {
6780    display: none !important;
6781  }
6782  @media print {
6783    .visible-print-inline {
6784      display: inline !important;
6785    }
6786  }
6787  .visible-print-inline-block {
6788    display: none !important;
6789  }
6790  @media print {
6791    .visible-print-inline-block {
6792      display: inline-block !important;
6793    }
6794  }
6795  @media print {
6796    .hidden-print {
6797      display: none !important;
6798    }
6799  }
6800  /*# sourceMappingURL=bootstrap.css.map */
```

以程序的方式书写CSS

- 变量
- 混合
- 嵌套规则
- 函数 & 运算

自动构建工具

- 对于需要反复重复的任务，例如压缩（minification）、编译、单元测试等，使用自动构建工具能够大大减轻重复性工作

组件化

- 面临的问题
  - 繁杂的HTML、CSS
  - 重复的界面代码
- 组件化
  - 拆分
  - 重用
  - 语义化
- 组件式框架
  - React
  - Angular JS 2

工程化

- 前端模式框架
  - MVC、MVP、MVVM
- 流行模式框架
  - Backbone
  - Knockout JS
  - Angular JS
- 样式框架
  - Bootstrap、Zui

前端不止于前端

移动应用的解决方案

- Native App
- HTML5 App
- Hybrid App
- React Native

Native App

- 体验丰富，功能强大
- 开发、维护成本高

HTML5 App

- 跨平台、易于传播
- 运行效率低
- 移动开发框架
  - 界面构建
    - jQuery Mobile、Zepto.js、…
  - 本地资源调用
    - PhoneGap、Weixin JS API、…

Hybrid App

- 结合Native App与HTML5 App
- 原生应用程序中某些模块应用HTML5页面

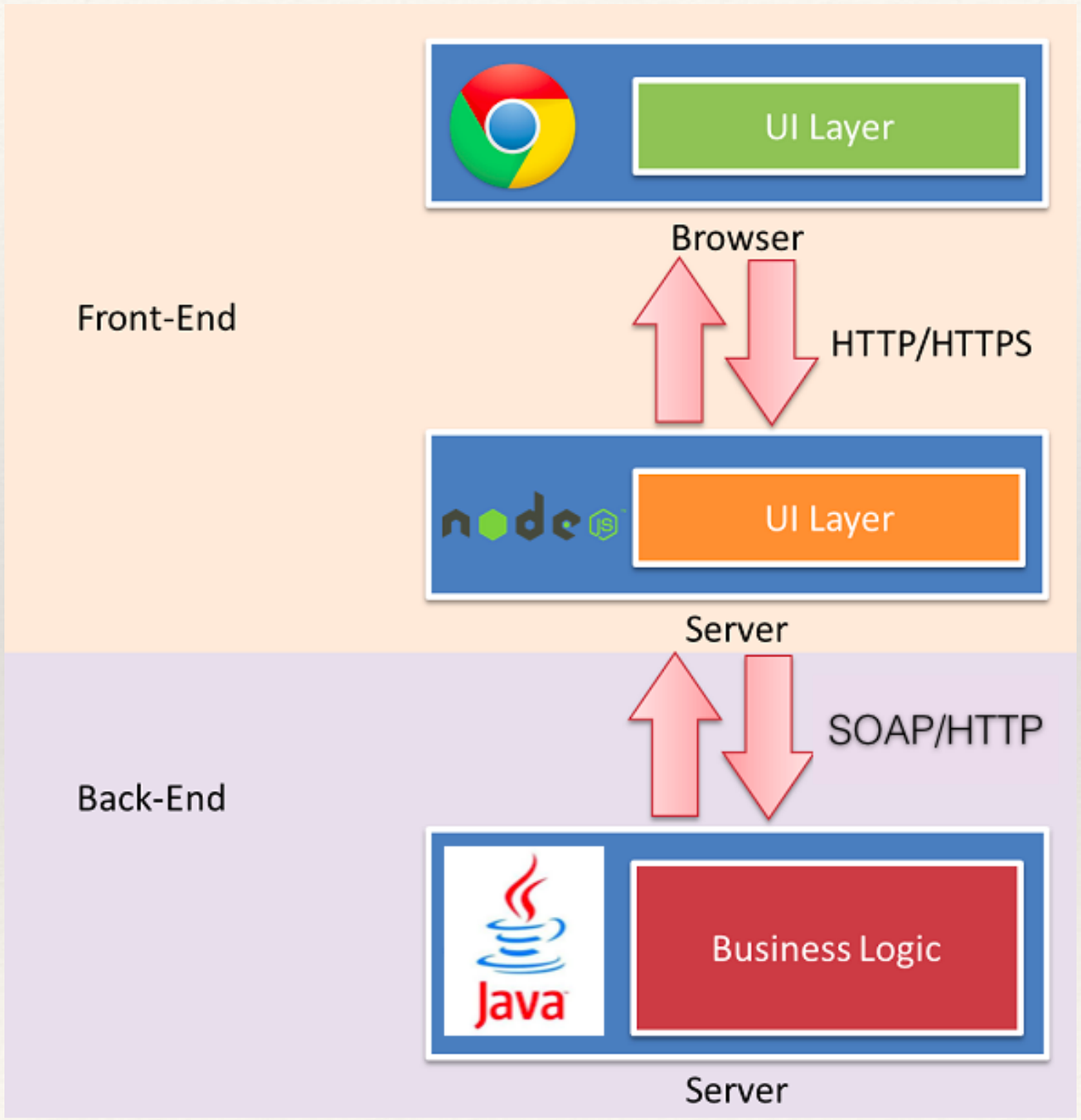
React Native

- 构建目标：开发者只需学习一种语言就能轻易为任何平台高效地编写代码。
- 结合 Web 应用和 Native 应用的优势，可以使用 JavaScript 来开发 iOS 和 Android 原生应用。
- 在 JavaScript 中用 React 抽象操作系统原生的 UI 组件，代替 DOM 元素来渲染等。提供出色的应用体验。

后端

- Node JS
- 为构建高性能Web服务器诞生，JS程序的运行平台。
- 通过npm包管理工具，构建出一整套生态环境
- Browserify组件使程序可以在浏览器中运行

淘宝的全栈架构


Front-End
UI Layer
Browser
HTTP/HTTPS
node
UI Layer
Server
Back-End
SOAP/HTTP
Java
Business Logic
Server

JavaScript一统三端

- Web端
- 后端
  - Node JS 平台
- 移动端
  - React Native

课程介绍

课程目标

- 前端工程师的基础课程
  - 理解Web应用程序架构、Web数据通讯
  - 熟练进行页面构建
  - 掌握JavaScript语言高级特性
  - 掌握HTML5标准与应用
  - 熟悉移动Web开发

课程考核

- 期末考试：50%
- 课程设计：20%
- 作业 && 实验：20%
- 平时表现：10%

课程内容


课程内容
一、服务器端基础（PHP）
二、前端基础回顾
三、JavaScript程序高级
四、HTML5程序设计
五、移动Web开发基础
六、Web优化

如何拿到阿里巴巴，百度，腾讯的前端实习生offer?

从初学前端到面试通过，整个的过程是怎样的？ 小弟目标大三去实习，希望各位有经验的大神详细说说，不甚感激！ ——————————————————我是分割线———————————————— 怎么不知不觉楼就歪到了211,985上去了。。。题主我并没有表达BAT实习生offer只属于211,985这种观点啊，实际上小弟是211的，不过我觉得互联网企业还是主要看能力，学历什么的真心不重要，认识的非211,985的朋友里面也有很多拿到很好的offer的 这个问题吧，其实是小弟想了解一下那些拿到offer的童鞋之前都付出了那些努力，水平大概达到什么程度才能通过考核。我现在大二，因为已经准备大三去面阿里或网易杭州了，所以特别希望有相关经验的大犇学长们分享一下自己的经历~很感谢各位~

4 条评论 分享

https://www.zhihu.com/question/29448457

推荐资源

- 阮一峰博客：http://www.ruanyifeng.com
- 百度Web前端研发部博客：http://fex.baidu.com
- 腾讯全端 AlloyTeam 团队 Blog：http://www.alloyteam.com
- 《Web全栈工程师的自我修养》—余果

谢谢！