# 概念区别

**库，框架，架构，平台**

库是工具箱。比如宜家买个便宜的架子，自己装，螺丝刀拧的手痛，就买了个电动螺丝刀，生产力立马跃迁好几倍。螺丝刀，电动螺丝刀，就是不同的库，虽然用法不同，API不同，但想换就换了，改动不大。

**框架是一套通用的解决方案**。比如刚才买的架子，可以放书，可以放花，可以放DVD。也可以个性化，有的地方装上门，装抽屉，但是，它还是个架子，不能在上面睡觉。对应软件框架来说，通常针对某类典型问题，预先规定好在什么地方做什么事情，不好僭越。更换框架，也是麻烦，就像架子一样，之前装的门，抽屉都要丢掉，架子里东西也要迁移，重新安置。

**架构是高度抽象的需求，是系统中的不变量**。比如架子装好了，要有个房子放。买房子，房型图就是架构，这屋书房，那屋睡觉，二人世界，70平搞定！装修，北欧风，中国风，都可以；打通墙，换马桶，也行，但承重墙不能动，70平不会多，这就是架构。开心的住了一年，怀孕了，三胞胎，噗噗噗，生了，婆婆岳母都来了，乌压压的一屋人，晚上客厅里睡倒一片，书房改胶囊旅店也塞不下了，怎么办？没办法！架构在那里，需求的发展超出了架构师的想象，只能认怂，有钱就买新房，换架构，没钱就受着，谁让你当初低估生殖力呢？

平台是所有可能做的事的集合。比如带着三胞胎搬进了新小区。想溜娃，去小区的滑梯；想运动，去小区的健身角；想游泳，物业说，不好意思，咱是中档小区，大哥你出门，打车，奥体。在家里，小区就是平台，你能做的受限于小区的设施。同样，网页上，js操作 dom，风生水起，但想要随意动硬盘里的文件，就困难了，因为它跑在浏览器里，浏览器是平台，有所限制。如果js跑另一个平台上，操作文件也不是大问题。微信之于小程序，浏览器之于前端，JVM之于Java，皆是平台，既提供了服务，也限制了可能。

具体到实际操作，平台不需太费脑筋，没多少可选；架构比较考验人，设计的不好以后就被虐；框架看团队偏好，一般选熟悉的；库自己选顺手的即可。总之，边做边学，不用追求完美，有太大压力，谁知道以后需求怎么变呢。

# 前端Js框架

　　有些日子没有正襟危坐写博客了，互联网飞速发展的时代，技术更新迭代的速度也在加快。看着Java、Js、Swift在各领域心花路放，也是煞是羡慕。寻了寻.net的消息，也是振奋人心，.net core 1，mono，xamarin等等，但大多都还在狂吼的阶段。其实一直以来对技术的理解是技术服务于业务和产品，产品又在不同程度的推进着技术的演进。

　　Web、无线、物联网、VR、PC从不同方向推进着技术的融合与微创新。程序员在不同业务场景下的角色互换。而随着node.js的出现语言的角色也在发生着转变，Js扮演了越来越重要的角色。也就有了茶余饭后也把了解到的知识整理一下。看过“你的知识需要管理”后，强烈的意识到杂乱且范范的知识储备远不如整理后条理清晰的知识带来好处多。所以，是的，我们需要时不时的回来对掌握的知识梳理归类，以备不时之需。

# 前端框架库

**1.Zepto.js**

* **地址：**http://www.css88.com/doc/zeptojs/
* **描述：**Zepto是一个轻量级的针对现代高级浏览器的JavaScript库， 它与jquery有着类似的api。 如果你会用jquery，那么你也会用zepto。

**2.SUI Mobile**

* **地址：**http://m.sui.taobao.org
* **描述：**SUI Mobile 是一套基于 [Framework7](http://framework7.taobao.org/) 开发的UI库。它非常轻量、精美，只需要引入我们的CDN文件就可以使用，并且能兼容到 iOS 6.0+ 和 Android 4.0+，非常适合开发跨平台Web App。
* **用途：**你也看到了，他是用于无线端的Web App的开发。

**3.Node.Js**

* **地址：**http://www.runoob.com/nodejs/nodejs-tutorial.html (中文网)
* **描述：**Node.js是一个Javascript运行环境(runtime)。实际上它是对Google V8引擎进行了封装。V8引 擎执行Javascript的速度非常快，性能非常好。Node.js对一些特殊用例进行了优化，提供了替代的API，使得V8在非浏览器环境下运行得更好。

　　Node.js是一个基于Chrome JavaScript运行时建立的平台， 用于方便地搭建响应速度快、易于扩展的网络应用。Node.js 使用[事件驱动](http://baike.baidu.com/view/536048.htm" \t "_blank)， 非阻塞[I/O](http://baike.baidu.com/subview/300881/11169495.htm) 模型而得以轻量和高效，非常适合在分布式设备上运行数据密集型的实时应用。

　　简单的说 Node.js 就是运行在服务端的 JavaScript。

　　Node.js 是一个基于Chrome JavaScript 运行时建立的一个平台。

　　Node.js是一个事件驱动I/O服务端JavaScript环境，基于Google的V8引擎，V8引擎执行Javascript的速度非常快，性能非常好。

* **用途：**

　　1. RESTful API

　　这是NodeJS最理想的应用场景，可以处理数万条连接，本身没有太多的逻辑，只需要请求API，组织数据进行返回即可。它本质上只是从某个数据库中查找一些值并将它们组成一个响应。由于响应是少量文本，入站请求也是少量的文本，因此流量不高，一台机器甚至也可以处理最繁忙的公司的API需求。

　　2. 统一Web应用的UI层

　　目前MVC的架构，在某种意义上来说，Web开发有两个UI层，一个是在浏览器里面我们最终看到的，另一个在server端，负责生成和拼接页面。（MVVC？）

不讨论这种架构是好是坏，但是有另外一种实践，面向服务的架构，更好的做前后端的依赖分离。如果所有的关键业务逻辑都封装成REST调用，就意味着在上层只需要考虑如何用这些REST接口构建具体的应用。那些后端程序员们根本不操心具体数据是如何从一个页面传递到另一个页面的，他们也不用管用户数据更新是通过Ajax异步获取的还是通过刷新页面。

　　3. 大量Ajax请求的应用

例如个性化应用，每个用户看到的页面都不一样，缓存失效，需要在页面加载的时候发起Ajax请求，NodeJS能响应大量的并发请求。

总而言之，NodeJS适合运用在高并发、I/O密集、少量业务逻辑的场景。

**4.angular.Js**

* **地址：**http://www.runoob.com/angularjs/angularjs-tutorial.html (中文网)
* **描述：**AngularJS 诞生于2009年，由Misko Hevery 等人创建，后为Google所收购。是一款优秀的前端JS框架，已经被用于Google的多款产品当中。AngularJS有着诸多特性，最为核心的是：MVVM、模块化、自动化双向数据绑定、语义化标签、依赖注入等等。
* **用途：**通过描述我们应该就能很好的明白AngularJS的真实用途了，MVVM，模块化，自动化双向数据绑定等等。除了简单的dom操作外，更能体现Js编程的强大。当然应用应该视场合而定。

**5.JQuery Mobile**

* **地址：**http://www.w3school.com.cn/jquerymobile/    (中文网)
* **描述：**Query Mobile是[jQuery](http://baike.baidu.com/view/1020297.htm) 在手机上和平板设备上的版本。jQuery Mobile 不仅会给主流移动平台带来jQuery核心库，而且会发布一个完整统一的jQuery移动UI框架。支持全球主流的移动平台。jQuery Mobile开发团队说：能开发这个项目，我们非常兴奋。移动Web太需要一个跨浏览器的框架，让开发人员开发出真正的移动Web网站。
* **用途：**jQuery Mobile 是创建移动 web 应用程序的框架。

　　　　　　jQuery Mobile 适用于所有流行的智能手机和平板电脑。

　　　　　　jQuery Mobile 使用 HTML5 和 CSS3 通过尽可能少的脚本对页面进行布局。

**6.requirejs**

* **地址：**http://www.requirejs.cn/
* **描述：**RequireJS的目标是鼓励代码的模块化，它使用了不同于传统<script>标签的脚本加载步骤。可以用它来加速、优化代码，但其主要目的还是为了代码的模块化。它鼓励在使用脚本时以module ID替代URL地址。

RequireJS以一个相对于[baseUrl](http://makingmobile.org/docs/tools/requirejs-api-zh/#config-baseUrl)的地址来加载所有的代码。 页面顶层<script>标签含有一个特殊的属性data-main，require.js使用它来启动脚本加载过程，而baseUrl一般设置到与该属性相一致的目录。

* **用途：**模块化动态加载。

**7.Vue.js**

* **地址：**http://cn.vuejs.org/
* **描述：**Vue.js 是用于构建交互式的 Web  界面的库。
* 它提供了 [MVVM](http://www.yyyweb.com/) 数据绑定和一个可组合的组件系统，具有简单、灵活的 API。
* 从技术上讲， Vue.js 集中在 [MVVM](http://www.yyyweb.com/) 模式上的视图模型层，并通过双向数据绑定连接视图和模型。实际的 DOM 操作和输出格式被抽象出来成指令和过滤器。相比其它的 MVVM 框架，Vue.js 更容易上手。

**8.backbone.js**

* **地址：**http://www.css88.com/doc/backbone/
* **描述：**[Backbone](http://baike.baidu.com/view/342697.htm) 为复杂Javascript应用程序提供模型(models)、集合(collections)、视图(views)的结构。其中模型用于绑定键值数据和自定义事件；集合附有可枚举函数的丰富API； 视图可以声明事件处理函数，并通过RESTful JSON接口连接到应用程序。

**9.React**

* **地址：**http://reactjs.cn/react/docs/why-react.html
* **描述：**React 是一个 Facebook 和 Instagram 用来创建用户界面的 JavaScript 库。很多人认为 React 是 [**MVC**](http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller) 中的 **V**（视图）。我们创造 React 是为了解决一个问题：**构建随着时间数据不断变化的大规模应用程序**。

**10.Ionic**

* **地址：**http://www.ionic.wang/js\_doc-index.html
* **描述：**Ionic既是一个CSS框架也是一个Javascript UI库。许多组件需要Javascript才能产生神奇的效果，尽管通常组件不需要编码，通过框架扩展可以很容易地使用，比如我们的AngularIonic扩展。

Ionic遵循视图控制模式，通俗的理解和 Cocoa 触摸框架相似。在视图控制模式中，我们将界面的不同部分分为子视图或包含其他视图的子视图控制器。然后视图控制器“驱动”内部视图来提供交互和UI功能。一个很好的例子就是标签栏（Tab Bar）视图控制器处理点击标签栏在一系列可视化面板间切换。

浏览我们的API文档来了解视图控制器和Ionic中可用的Javascript实用工具。

Ionic 是目前最有潜力的一款 HTML5 手机应用开发框架。通过 SASS 构建应用程序，它 提供了很多 UI 组件来帮助开发者开发强大的应用。 它使用 JavaScript MVVM 框架和 AngularJS 来增强应用。提供数据的双向绑定，使用它成为 Web 和移动开发者的共同选择。

# 前端UI框架

**1.Pure**

* **地址：**http://purecss.org/layouts/
* **描述：**Pure精心设计，只为可以在任何Web项目中使用。为了例证这一点，我们制作了如下几个模板。这些模板都是响应式的，并且没有使用任何JavaScript。
* **用途：**真的是很精美的一个样式框架，便于我们快事构建一些个人产品，当然也可以服务于工作中的一些项目。

**2.bootstrap**

* **地址：**http://www.bootcss.com/
* **描述：**简洁、直观、强悍的前端开发框架，让web开发更迅速、简单。

**3.EasyUI**

* **地址：**http://www.jeasyui.net/  (中文网)
* **描述：**easyui是一种基于jQuery的用户界面插件集合。

　　　　　　easyui为创建现代化，互动，JavaScript应用程序，提供必要的功能。

　　　　　　使用easyui你不需要写很多代码，你只需要通过编写一些简单HTML标记，就可以定义用户界面。

　　　　　　easyui是个完美支持HTML5网页的完整框架。

　　　　　　easyui节省您网页开发的时间和规模。

　　　　　　easyui很简单但功能强大的。

**4.Ant Design**

* **地址：**http://ant.design/
* **描述：**一个 UI 设计语言，一套提炼和应用于企业级后台产品的交互语言和视觉体系

# 三、可视化组件

**1.Echarts**

* **地址：**http://echarts.baidu.com/
* **描述：**ECharts，一个纯 Javascript 的图表库，可以流畅的运行在 PC 和移动设备上，兼容当前绝大部分浏览器（IE8/9/10/11，Chrome，Firefox，Safari等），底层依赖轻量级的 Canvas 类库[ZRender](https://github.com/ecomfe/zrender)，提供直观，生动，可交互，可高度个性化定制的数据可视化图表。

**2.tableau(收费)**

* **地址：**http://www.yuandingit.com/special/tableau/index.html
* **描述：**Tableau 是桌面系统中最简单的商业智能工具软件，Tableau 没有强迫用户编写自定义代码，新的控制台也可完全自定义配置。在控制台上，不仅能够监测信息，而且还提供完整的分析能力。Tableau控制台灵活，具有高度的动态性。

# 四、前端构建工具

1.gulp

地址：http://www.gulpjs.com.cn/

描述：易于使用

通过代码优于配置的策略，Gulp 让简单的任务简单，复杂的任务可管理。

构建快速：利用 Node.js 流的威力，你可以快速构建项目并减少频繁的 IO 操作。插件高质：Gulp 严格的插件指南确保插件如你期望的那样简洁高质得工作。

易于学习：通过最少的 API，掌握 Gulp 毫不费力，构建工作尽在掌握：如同一系列流管道。

# 五、博客搭建

1.技术组合

　　HEXO+Github,搭建属于自己的博客。

**站点：**http://www.jianshu.com/p/465830080ea9

**HEXO介绍：**Hexo是一个开源的静态博客生成器,用node.js开发,作者是台湾大学生tommy351

**准备：**git  + node.js + markdown编辑 + gitcafe + 域名