

前端架构及选型之路

前端开发部-冯伟平

2017.10



Why?



前端需要架构思想

移动应用的兴起，越来越多的前端团队开始承担起APP客户端的开发；

NodeJS的兴起，越来越多的前端团队开始承接后端轻逻辑的开发；

前后端分工协作问题凸显，更多的前端团队开始通过一些php承接接入层的开发；

用户对前端页面体验的要求，更小的资源消耗、更快速度、更高性能；



What?



前端架构组成

1. 组件框架
2. 工具平台
3. 代码结构
4. 开发规范
5. 流程边界



组件框架

1. **前端架构**是指在前端开发工作过程中用于提升开发效率、质量、性能和可维护性的保障体系
2. 框架是架构的重要组成部分，架构决定框架的**选型**，框架决定架构的**技术路线**；
3. 架构围绕框架进行一系列的**流程工具建设**，从而形成完善自动的开发体系；



组件框架

开发模式

如何来实现代码的职责分离，MVC，MVVM；形成模块化，组件化；

通讯

有效的进行组件隔离和独立，形成的通信模型；如Flux模式、Vuex、Redux等。

模板

集中的处理数据往HTML的转换过程，代码行、缓存管理、预编译、运算性能、强大语法等各个维度不断追求各种极致；

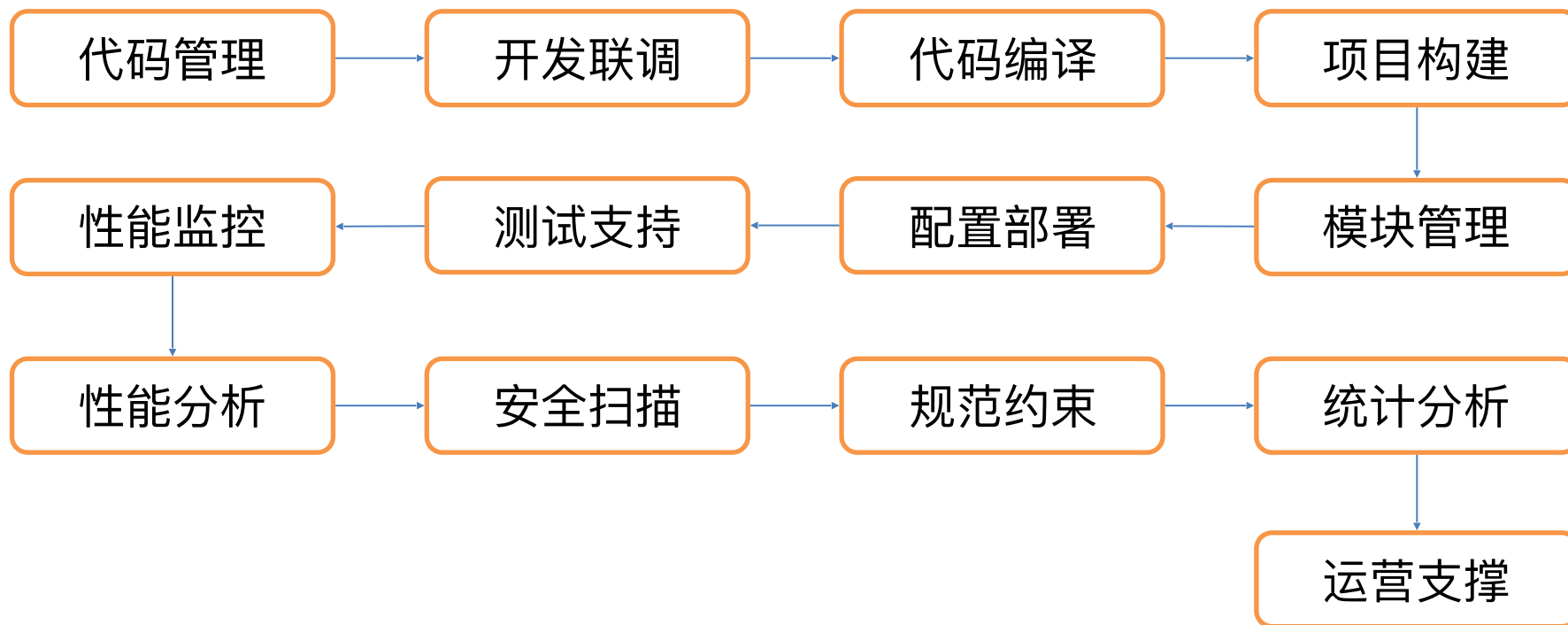
基础类库

为了改善编码生产力，实现DOM 操作、常用函数方法、异步化等,一般引入jQuery、Zepto、Underscor等传统类库，也可以自我封装；



工具平台

工具平台主要就是围绕前端开发流程中的每一步关键节点去建设起来的，工具平台是前端工程化的**主要生产力**；



代码结构

代码结构设计的灵活性在前端架构中的重要性

1. 团队放大数倍，业务复杂数倍，代码管理还能跟的上吗？
2. 如何避免开发人员之间的开发行为耦合？更好并行开发？
3. 如何支撑多特性、多版本的并行？
4. 主路径按功能、按版本还是按渠道？



开发规范

开发规范为了提升开发效率，快速定位问题，统一编程模式、代码风格、

1. HTML语义化、SEO；
2. CSS(SASS、LESS)命名规范、复用和定制、mixins、编码规范；
3. JS规范(TS、ES6)、编码规范；
4. 注释规范、文档规范；
5. 设计思路、模块拆分、结构分层；
6. 监测工具(ESLint、CSSLint)加Codereview



流程边界

明确前端工作内容、职责的定义，前端支持力度；

接口逻辑、页面输出
(nodejs、express、
PHP)

数据业务逻辑
(Ajax、flux、数据驱
动)

交互逻辑
(JS、DOM、动
画)

页面构建
(HTML+CSS
)

前端架构完善

好的架构是进化来的，不是设计来的

最适合自己的，就是最好的



HOW?



框架选型方法

核心就是为了改善编码生产力，对于框架的选型要从三面看

1. 看产品形态

- > 产品特征是强内容还是强交互或者是游戏性；
- > 应用场景，兼容性，平台适配；

2. 看框架的能力

功能是否丰富、社区活跃度、国内社区活跃度如何、文档齐全与否、是否及时更新、测试覆盖率如何、上手难易度如何、框架的生命力

3. 看团队的特性

团队规模、人员梯度（低中高）、稳定性、支持力度、技术储备、成长规划



主流Javascript MVC MVVM框架

Framework	UI Bindings	Composed Views	Web Presentation Layer	Plays Nicely With Others
Backbone.js	✗	✗	✓	✓
SproutCore 1.x	✓	✓	✗	✗
Sammy.js	✗	✗	✓	✓
Spine.js	✗	✗	✓	✓
Cappuccino	✓	✓	✗	✗
Knockout.js	✓	✗	✓	✓
Javascript MVC	✗	✓	✓	✓
Google Web Toolkit	✗	✓	✗	✗
Google Closure	✗	✓	✓	✗
Ember.js	✓	✓	✓	✓
Angular.js	✓	✗	✓	✓
Batman.js	✓	✗	✓	✓



主流Javascript MVC MVVM框架

UI Bindings (UI绑定) :

作者想说的不仅仅是模板，而是想谈一种在底层模型出现变化时，视图层能够自动相应地更新的陈述性方法。一旦您用过了支持UI Binding的框架（例如Flex）就很难放手回头了。

Composed Views (模块化视图) :

与所有的软件开发者一样，作者也喜欢编写模块化、可重用的代码。基于这样的原因，当给UI编程的时候，作者喜欢使用视图的方法来创作（个人更偏好在模板层时使用），当然这样也就需要拥有足够丰富的视图组件来支持。关于这一点有一个可重用的页面小工具的范例。

Web Presentation Layer (web表示层) :

我们是在为web编写程序，最不想要的就是Native风格的小工具；但是也没有什么理由来为一个web框架来创建它自己的布局管理器。HTML和CSS是目前解决样式与布局的最好的方法，他们被这样应用着，框架也应该以这一点为核心。

Play Nicely With Others (兼容，友好) :

不得不承认，jQuery是十分犀利的。作者不喜欢那种绑定着一个sub-par jQuery副本的框架，而直接推荐使用jQuery的那种框架才是作者需要的



英文原文: <http://codebrief.com/2012/01/the-top-10-javascript-mvc-frameworks-reviewed/>

主流Javascript MVC MVVM框架

[Backbone.js](#)

Backbone.js是web最火的框架，如果不了解它将寸步难行，众多知名品牌均支持该框架，令人印象深刻，京东到家、服装定制等；

优点：强大的社区，还有大量的实力支持。例如它本身就较多地使用了Underscore.js（也是一个强大的框架）

缺点：抽象功能不够强，以及一些需要的功能还没实现。整个框架十分轻量级，产出的结果是一大堆引用文件和样板：而且应用的规模越大这一点就会越明显。



主流Javascript MVC MVVM框架

[Ember.js](#)

Ember（之前是SproutCore 2.0）是竞争者中的新丁。它是一个尝试：从SproutCore2.0中抽取分离其核心feature并转变成为一个更加紧凑的模型框架，更加适合web。

优点：特别丰富的模板系统，拥有可编写的视图和UI绑定

缺点：由于太新，文档跟不上



主流Javascript MVC MVVM框架

[Angular.js](#)

Angular是在目前非常流行的框架，由Googler开发，包含了很多有趣的设计选择。

优点：关于模板的范围和控制器的设计考虑的很周到。具有依赖注入系统，支持丰富的UI绑定语法，从而使得过滤和转换这样的工作开销很小；

缺点：代码库很不健全，也不够模块化。视图也不够模块化（关于这点在[Batman.js](#)的缺陷中讨论的更加细致）

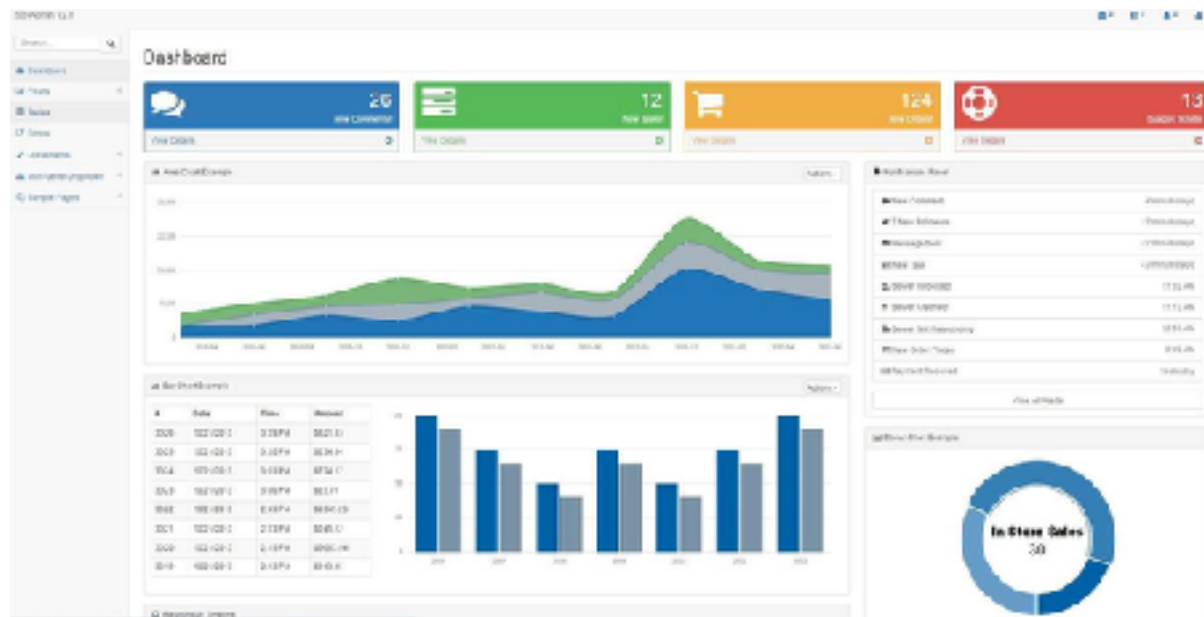


主流UI框架

Bootstrap

Bootstrap是由twitter推出的Web前端UI框架，它由Twitter的设计师Mark Otto和Jacob Thornton合作开发,是一个CSS/HTML框架。它使用了最新的浏览器技术，Bootstrap提供了时尚的排版样式，表单，buttons，表格，网格系统等等。

官方网址: <http://getbootstrap.com/>

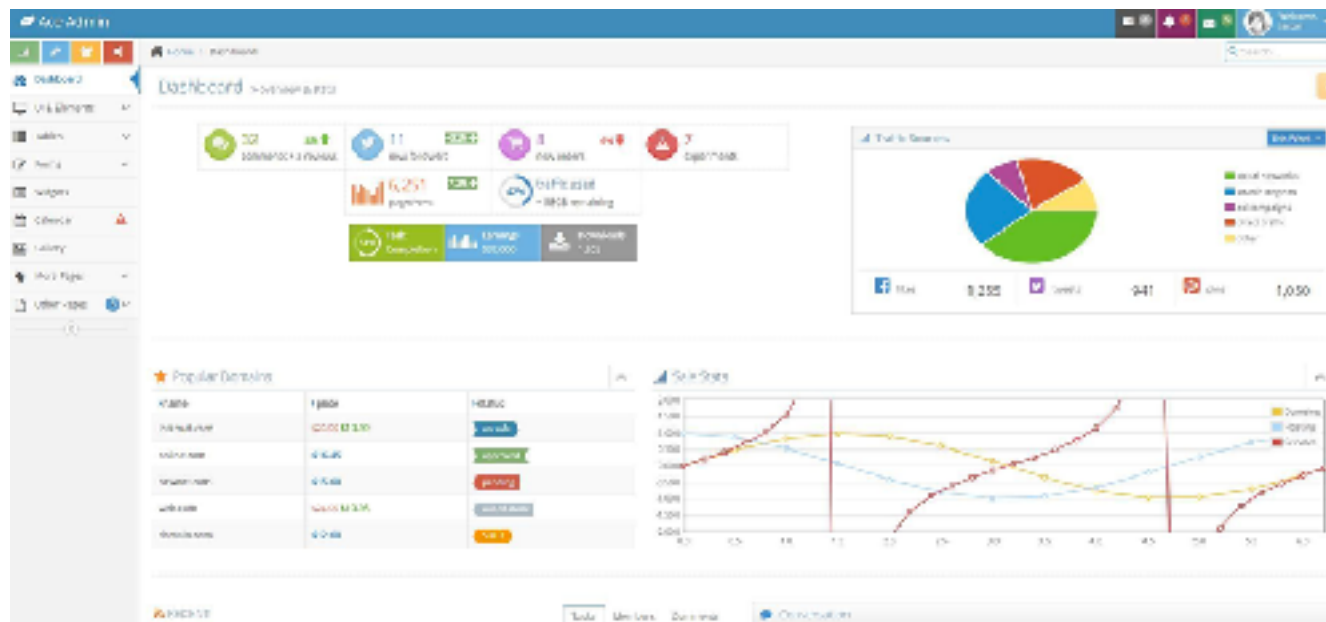


主流UI框架

Ace Admin

响应式Bootstrap网站后台管理系统模板ace admin，非常不错的轻量级易用的admin后台管理系统，基于Bootstrap3，拥有强大的功能组件以及UI组件，基本能满足后台管理系统的需求，而且能根据不同设备适配显示，而且还有四个主题可以切换。

官方地址：<http://ace.jeka.by/>

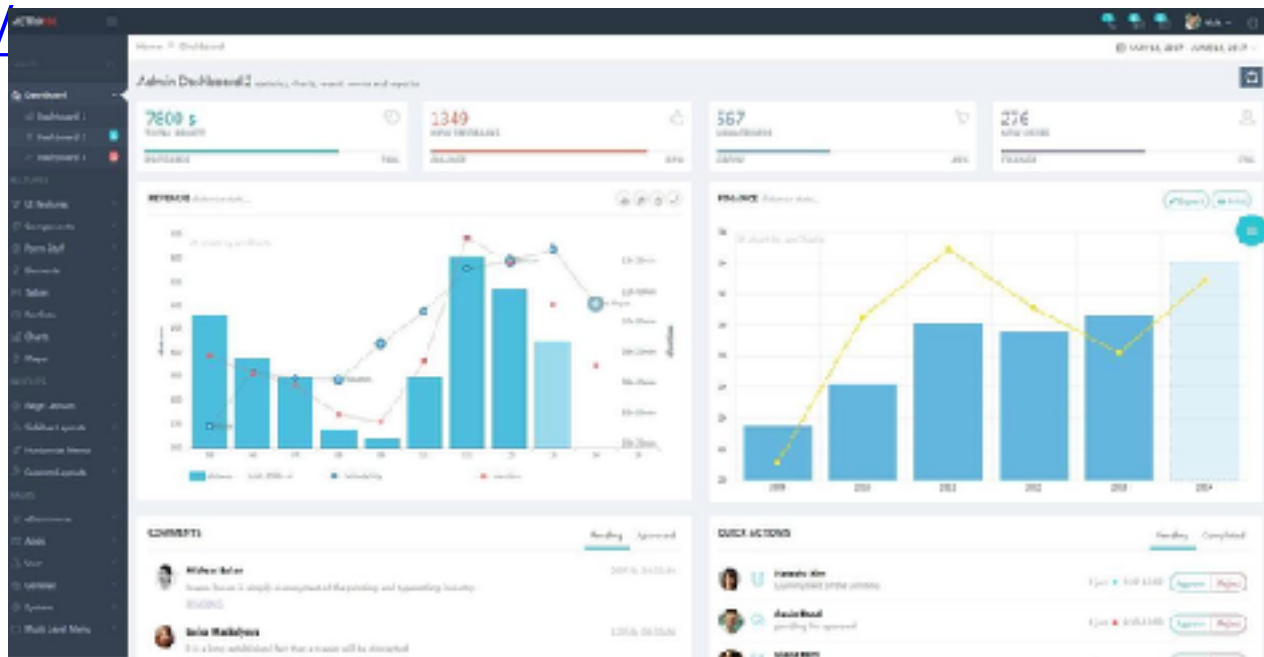


主流UI框架

Metronic

Metronic 是一套精美的响应式后台管理模板，基于强大的 Twitter Bootstrap 框架实现。Metronic 拥有简洁优雅的 Metro UI 风格界面，6 种颜色可选，76 个模板页面，包括图表、表格、地图、消息中心、监控面板等后台管理项目所需的各种组件。

官方地址：<http://www.metronic.com/>

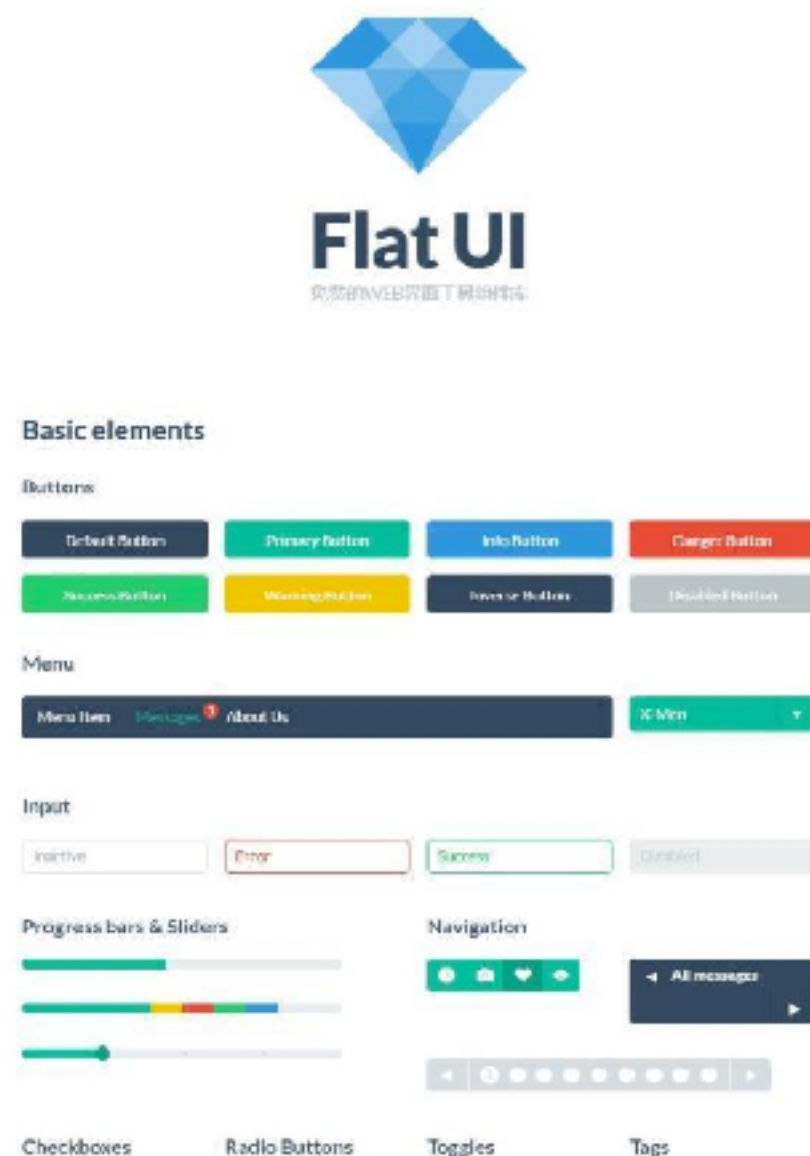


主流UI框架

Flat UI

Flat UI是一套精美的扁平风格 UI 工具包，基于 Twitter Bootstrap 实现。这套界面工具包含许多基本的和复杂的 UI 部件，例如按钮，输入框，组合按钮，复选框，单选按钮，标签，菜单，进度条和滑块，导航元素等等。

官方网址: <http://www.bootcss.com/p/flat-ui/>

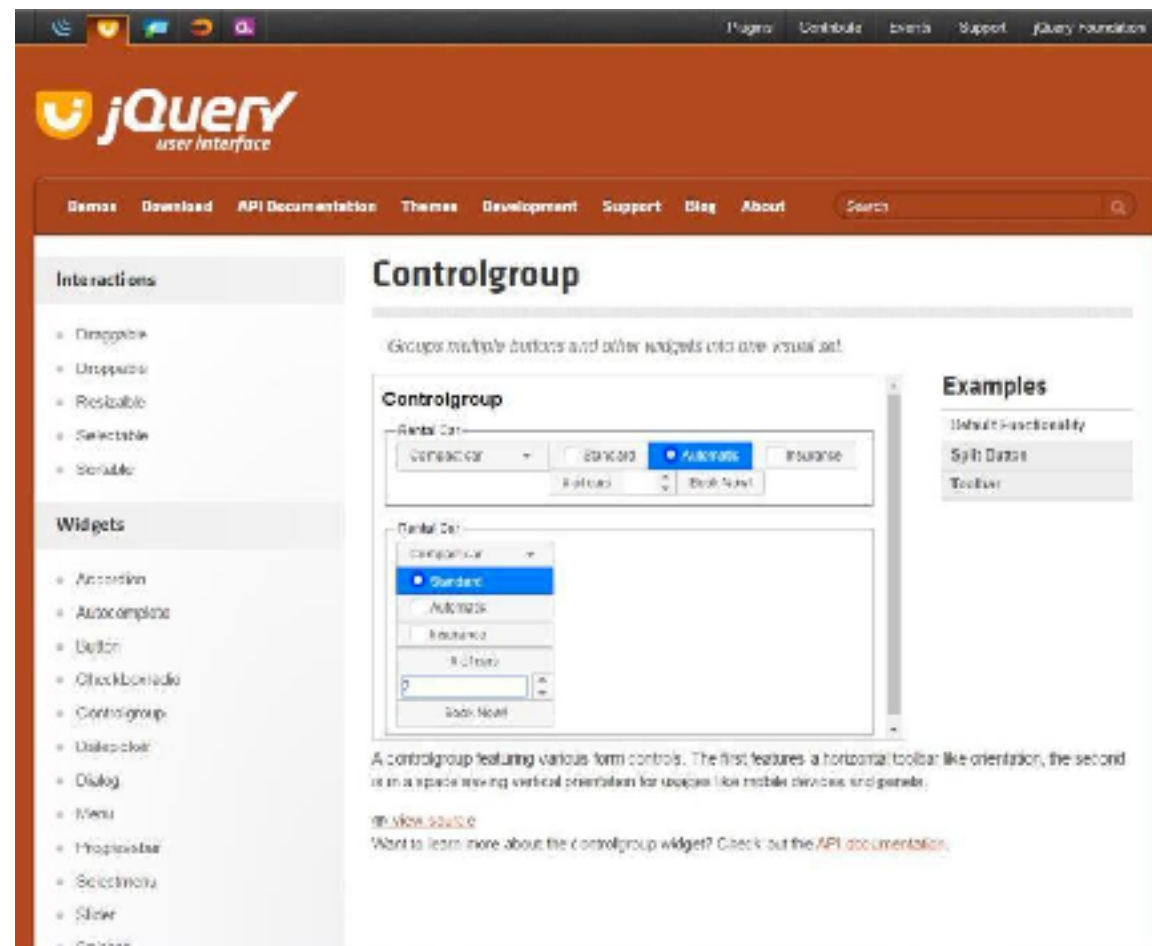


主流UI框架

jQuery UI

jQuery UI是一款基于jQuery的开源 Javascript框架，jQuery UI框架主要提供了用户交互、动画、特效和可更换主题的可视控件，让开发者可以更方便地实现网页交互界面，jQuery UI的整个框架比较庞大，但你也可以根据自己需要的功能生成适合自己的框架底层。

官方网址: <http://jqueryui.com/>



主流UI框架

jQuery UI Bootstrap

jQuery UI Bootstrap 是一个jQuery UI主题。它受到Twitter项目的启发，将其slickness引入jQuery UI Widgets中。它是jQuery UI和Bootstrap的集成，它是Bootstrap样式的，因此外观比较漂亮，同时它拥有jQuery UI的控件功能，这也方便开发者快速地创建一个网页控件。官方网站：<https://github.com/jquery-ui-bootstrap/jquery-ui-bootstrap/>



主流UI框架

BUI

BUI她是基于jQuery,兼容KISSY的UI类库，专致于解决后台系统的框架方案，BUI提供了丰富的DPL含有强大的控件库对业务做了精细的分析。

官网：<http://www.builive.com/>



主流工程化工具



Grunt

- + 易于上手
- + 通过配置实现更高控制水平
- + 发展历史更久，社区规模更大，插件选项更多
- 非流式文件I/O使其速度较慢



Gulp

- + 需要配置的部分较少
- + 处理速度更快
- + 流式机制，允许异步文件处理
- + 代码编写需求更低
- API较为有限

主流工程化工具

WEBPACK

1、代码拆分

两种组织模块依赖的方式，同步和异步。异步依赖作为分割点，形成一个新的块；

2、Loader

本身只能处理原生的 JavaScript 模块，但是 loader 转换器可以将各种类型的资源转换成 JavaScript 模块；

3、智能解析

有一个智能解析器，可处理任何第三方库，无论它们的模块形式是 CommonJS、普通的 JS 文件、ES6

4、插件系统

丰富的插件系统，来满足各式各样的需求

5、快速运行



主流JS模块规范

CommonJS

优点:

- 服务器端模块便于重用

- NPM中已经有将近20万个可以使用模块包

- 简单并容易使用

缺点:

- 同步的模块加载方式不适合在浏览器环境中，同步意味着阻塞加载，浏览器资源是异步加载的

- 不能非阻塞的并行加载多个模块

实现:

- 服务器端的 [Node.js](#)

- [Browserify](#)，浏览器端的 CommonJS 实现，可以使用 NPM 的模块，[modules-webmake](#)，类似Browserify，还不如 Browserify 灵活

- AMD-RequireJS、CMD-SeaJS



主流JS模块规范

ES6

ES6 标准增加了 JavaScript 语言层面的模块体系定义。[ES6 模块](#)的设计思想，是尽可能的**静态化**，使得编译时就能确定模块的依赖关系，以及输入和输出的变量。CommonJS 、AMD 模块，都只能在运行时确定这些东西。

优点：

- 容易进行静态分析

- 面向未来的 EcmaScript 标准

缺点：

- 原生浏览器端还没有实现该标准

- 全新的命令字，新版的 Node.js才支持

实现：

- [Babel](#)



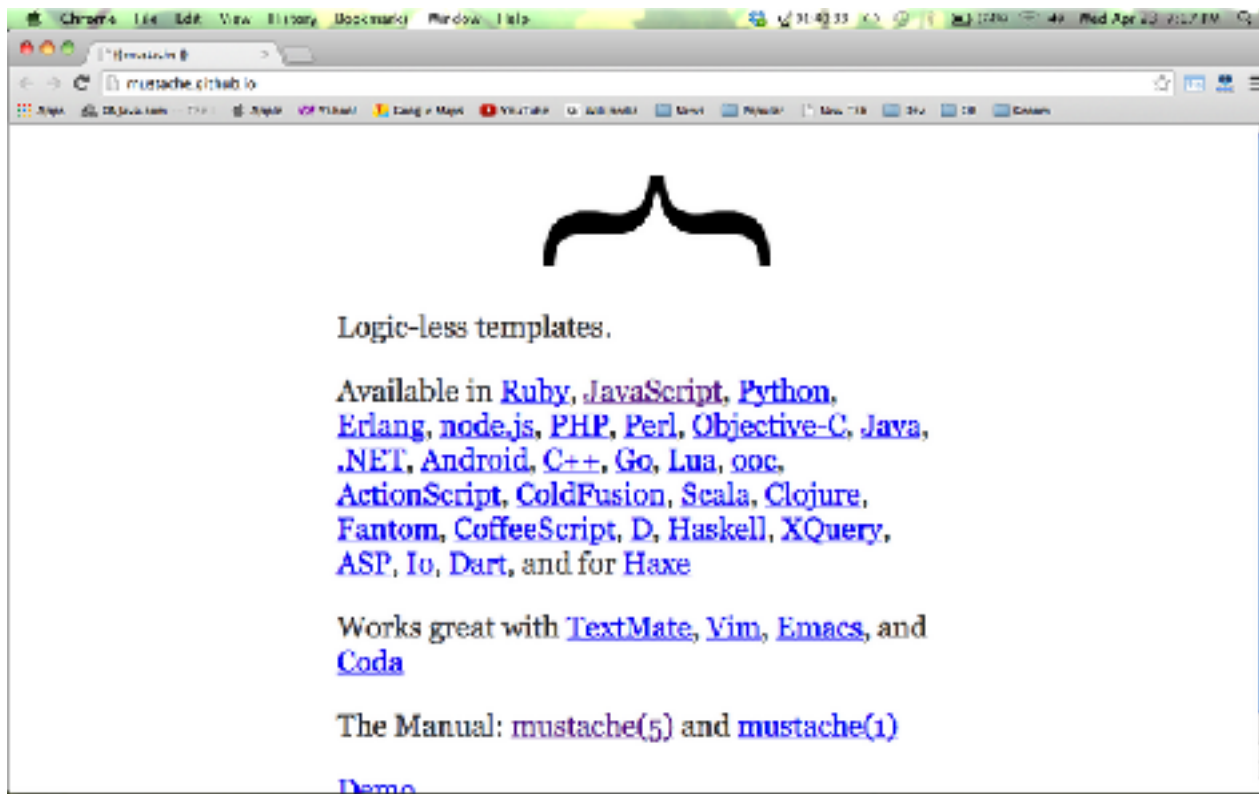
主流工程化思路



JS模版引擎

Mustache

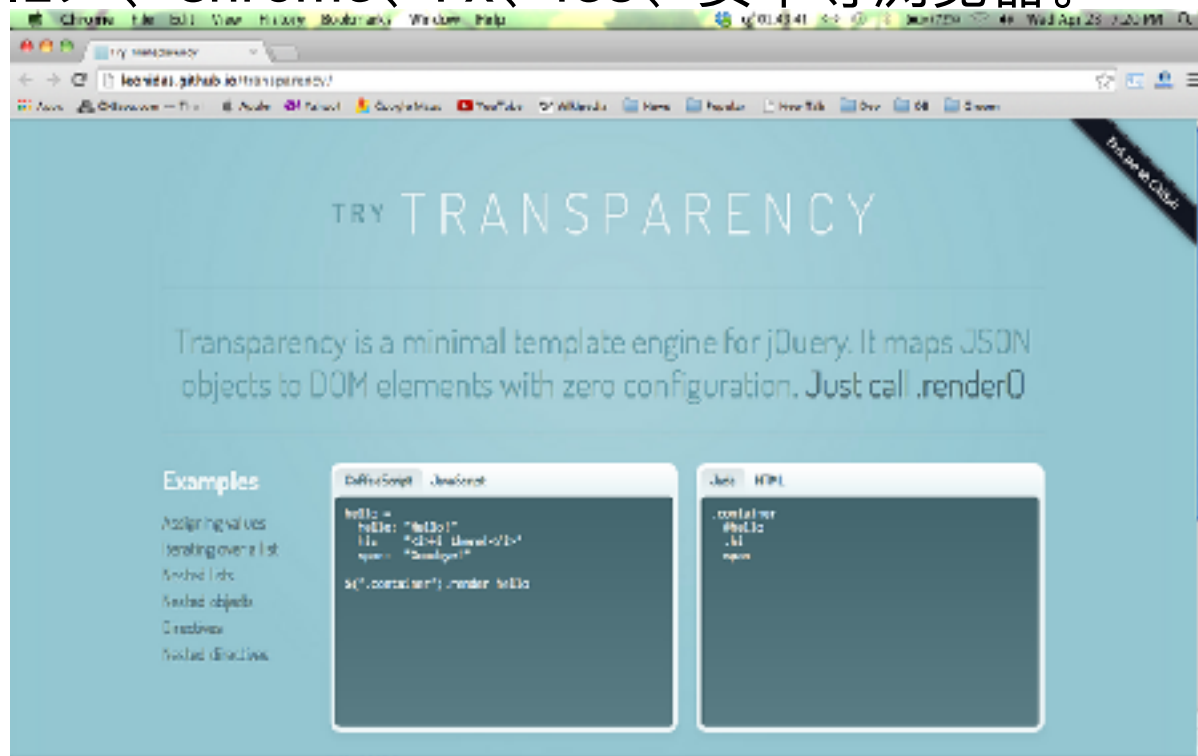
Mustache是一个logic-less（无逻辑或轻逻辑）语法模板。可以用于组织HTML、配置文件、源代码在内的任何东西。Mustache使用JavaScript对象的值，用来扩展模板代码中的大括号标签。



JS模版引擎

Transparency

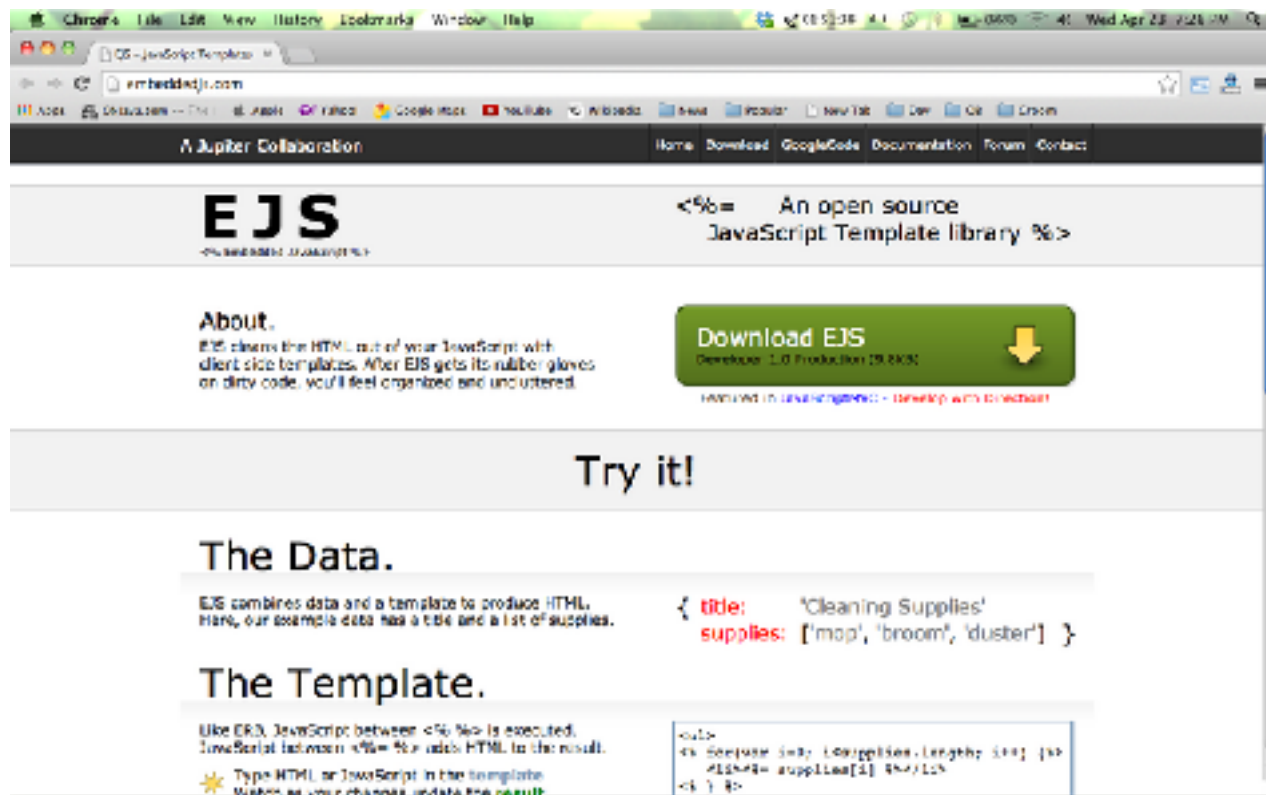
Transparency是一个强大的客户端模板引擎，用来将数据绑定到Web页面的BOM结构中。其模板无需特殊格式，直接完全符合HTML。直接使用JavaScript逻辑，无需新学特殊的“模板语言”。兼容IE9+、Chrome、Fx、iOS、安卓等浏览器。



JS模版引擎

Embeddedjs

EJS以类似PHP的JS/HTML通过标签混排的形式，帮助开发者将JavaScript和HTML部分有效分离。



JS模版引擎

DoTjs

最快和简洁的JavaScript模板引擎，同时用于Node.js和浏览器。



JS模版引擎

Handlebarsjs

一套语义化模板引擎。兼容Mustache

T.js

一个用简单的JavaScript数据结构去渲染表现html/xml内容的模板引擎

Dustjs

一套同时可用于浏览器或Node.js的异步模板引擎。

Nunjucks

Nunjucks是一套富功能的模板引擎。模板语言功能强大，支持块继承、自动转义、宏、异步控制等功能。

artTemplate

一个简单高效地JavaScript模板引擎，相比于jsRender这种复杂的模板引擎，artTemplate就显得很“娇小”。



JS类库和工具

对于Web设计人员和开发人员来说，手里有一套便利的、最新的工具可供使用是极其重要的。Web设计和开发的趋势变化极快，所以拥有最新的工具是必不可少的。

了解参考：

1. <https://baijiahao.baidu.com/s?id=1577800175291788714&wfr=spider&for=pc>
2. <https://medium.com/javascript-scene/top-javascript-frameworks-topics-to-learn-in-2017-700a397b711>
3. <http://inspiretrends.com/best-javascript-libraries-and-tools/>
4. <https://codegeekz.com/javascript-libraries/>
5. <https://www.douban.com/note/193867172/>
6. <http://blog.csdn.net/laszloyu/article/details/4075502>
7. <http://blog.csdn.net/minjun0901/article/details/6612023>



代码规范

ESLint是一个用来识别 ECMAScript 并且按照规则给出报告的代码检测工具，使用它可以避免低级错误和统一代码的风格。

ESLint被设计为完全可配置的，主要有两种方式配置ESLint：

1. 在注释中配置：使用JavaScript注释直接把配置嵌入到JS文件中。
2. 配置文件：使用下面任一的文件来为全部的目录和它的子目录指定配置信息。
 - a) javascript：使用.eslintrc.js文件并导出一个包含配置的对象。
 - b) YAML：.eslintrc.yaml或者.eslintrc.yml
 - c) JSON：.eslintrc.json，并且此文件允许使用JS形式的注释
 - d) 废弃的用法：.eslintrc，此文件可以是JSON或者YAML
 - e) package.json：在package.json文件中创建eslintConfig属性，所有的配置包含在此属性中。



代码规范

[stylelint](#)

基于 Javascript 的CSS审查工具，它易于扩展，支持最新的 CSS 语法，也理解类似 CSS 的语法。此外，因为它是基于 JavaScript，所以比起 Ruby 开发的 [scss-lint](#) 速度更快。

由 [PostCSS](#) 提供技术支持，所以它也可以理解 PostCSS 解析的语法，比如 SCSS。

参考阅读： <http://www.jianshu.com/p/2bcdce1dc8d4>

参考阅读： <http://www.w3cplus.com/workflow/How-to-lint-your-css-with-stylelint.html>

内部分享： [《在项目中启用 ESLint 与 Stylelint 检查》](#)



FE's Future?



前端新技术框架-PWAs

1. PWAs (Progressive Web Apps, 渐进式网页应用) 是由谷歌提出的新一代 Web 应用概念, 旨在提供可靠、快速、类似 Native 应用的服务方案。

2. 在安全、性能和体验三个方面都有很大提升;

3. PWA 本质上是 Web App, 借助一些新技术也具备了 Native App 的一些特性, 兼具 Web App 和 Native App 的优点;

<https://pwa.rocks> 有诸多PWA的实践应用

京东项目: 京东摄影M版

内部分享: [《PWA-开发如App般体验的web应用》](#)



前端新颖技术框架-PWAs

1. Service Worker & Cache Storage : 能够显著提高应用加载速度、甚至让 web 应用可以在离线环境使用;
2. App Manifest : 描述 web 应用元数据 (metadata)、让 web 应用能够像原生应用一样被添加到主屏、全屏执行;
3. Notifications & Push API : 提高 web 应用与操作系统集成能力, 让 web 应用能在未被激活时发起推送通知;



前端新颖技术框架-小程序

1. 小程序是什么？

和微信强关联的一个APP入口，实现用户用完即走的新用户体验。小程序的入口在微信-发现-小程序，在此可以查看您已查看过的小程序及推荐的附近小程序。

2. 小程序类似原生 Native 的实现方式，使页面渲染速度大大提升，用户体验上优于移动 WEB 页面，而开发成本又远小于 APP。

3. 独立的开发者工具，集成编辑、调试、部署于一体，大大方便前端开发。

京东项目：图书小程序、移动充值、京东E卡、小程序生成平台

内部分享：[《从0构建小程序》](#) [《小程序快速入门》](#)



前端新颖技术框架-RN

- ReactNative: Facebook推出的基于React.js的移动应用开发框架, 结合了Web应用和Native应用的优势, 可以使用统一的框架来开发原生应用 (Android/iOS)。2015年正式成为开源项目。
- WEEX: 阿里手淘推出的基于Vue.js的移动应用开发框架。架构借鉴了ReactNative的设计, 完善了Web端开发。2016年中成为开源项目。
- JDReact: 京东基于ReactNative开发的框架, 优化RN核心库性能及稳定性, 对京东业务API及公共组件进行了封装, 实现了Web端扩展, 完整支持三端开发。并集成到了京东客户端
- 京东项目: 排行榜、京东拍卖、京东摄影等
- 内部分享: [《JDReact平台介绍》](#) [《JDReact开发常见问题》](#)



前端新颖技术框架-RN

JDReact平台 微信小程序 Weex HTML5

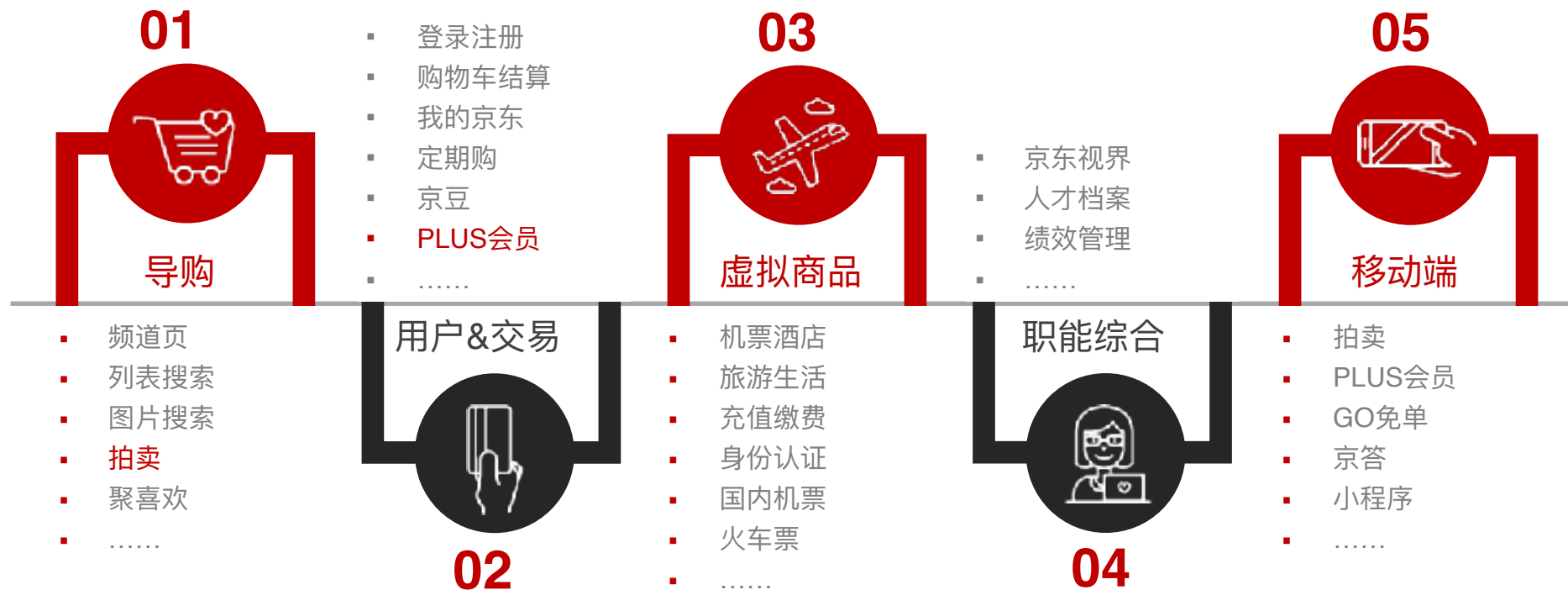
支持的平台	Android / iOS / Web	微信支持的平台	Android / iOS / Web	Web
技术门槛	低，基于React.js	低，类似JS语法	低，基于H5和Vue.js	低，Html5 JS
开发框架	JDReact SDK	小程序SDK	Weex SDK	JQuery
框架能力	京东样式组件、导航、多媒体、京东网络、感应器、京东业务能力等	组件、样式、导航、多媒体、网络、罗盘、感应器等	组件、样式、导航、网络、动画、存储等	没有原生调用能力和京东业务能力
用户体验	用户体验好，接近原生	用户体验好，优于H5页面	用户体验好，接近原生	用户体验差
业务部署流程	JDReact CMS平台支持线上灰度发布和更新能力	小程序打包上传，审核通过微信内发布	目前没有部署支持	实时部署
社区及文档	有Facebook的官方支持，京东内部也有JDReact平台的社区/文档支持	腾讯及微信社区支持	阿里手淘支持，目前还不太普及	完善



JDC FE?



JDC前端项目概括



JDC前端开发方式

页面构建

模块(widget)拆分、分工协作、模拟数据、页面重构、拼合页面；
Legos、JDF、FEZ、Athena

UI交互

JS交互逻辑、UI组件(JDFUI+定制组件)

代码规范

ESLint+CSSLint检测、codeview

走查调整

视觉走查、产品走查

测试联调

接口联调、测试、优化



JDC前端沉淀积累

- ▣ BLOG
- ▣ 分享
- ▣ 课程
- ▣ 专利建设

专业发展



- 智能构建平台
- Widget协同开发平台
- 静态文件引用查询系统
- 商城基础数据智能模拟服务
- 京东精选商详编辑系统

创新



JDC前端技术架构

C端：jQuery+Seajs+Sass+JDFUI+JDF

B端：Vue+Vuex+JDF

M端：

- [主]Vue+Vuex+ES6+Sass+webpack

- [部]React+Redux+ES6+Sass+webpack

- [传]Requirejs+Backbone+Zepto + (fastclick.js+Hammer.js+iScroll5.js)

APP：ReactNative+JDReact+ES6+FLex



性能调优

减少请求数

- ① 合并样式、脚本文件 ② 合并背景图片 ③ CSS3图标、Icon Font

降低请求量

- ① 开启GZip
- ② 优化静态资源，jQuery->Zepto、阉割IScroll、去除冗余代码
- ③ 图片无损压缩 ④ 图片延迟加载 ④ 图片降质 ⑤ 减少Cookie携带

很多时候，也会采用类似“时间换空间、空间换时间”的做法，比如：

- ① 缓存为王，对更新较缓慢的资源&接口做缓存（浏览器缓存、localsorage、application cache这个坑多）
- ② 按需加载，先加载主要资源，其余资源延迟加载，对非首屏资源滚动加载
- ③ fake页技术，将页面最初需要显示Html&Css内联，在页面所需资源加载结束前至少可看，理想情况是index.html下载结束即展示（2G 5S内）
- ④ CDN



知识体系



知识体系

移动性能优化技能图谱



谢谢！

