Atitit 2015 技术趋势与没落技术

目录

[1.1. 离线优先Web应用程序（Offline first web applications） 1](#_Toc14450)

[1.2. 依照产品而不是项目（products rather than projects）的 思路进行思考。 2](#_Toc17714)

[1.3. Postman（getpostman.com/features）是一个在Chrome 中使用的REST客户端插件 2](#_Toc2393)

[1.4. PlantUML（ 2](#_Toc20957)

[1.5. NoPSD运动想要抓住的，是“即时设计”这个在视觉设计中重 要和实用的概念。 3](#_Toc31955)

[1.6. “产品优 于项目” 3](#_Toc25629)

[1.7. Css bem 3](#_Toc21678)

[1.8. HTML5 iFrame来沙盒化 3](#_Toc13021)

[1.9. 幂等过滤器 4](#_Toc24345)

[1.10. “只积 累”是一种设计策略，数据删除通过撤回而不是更新Accumulate-only data 4](#_Toc17086)

[1.11. 双 因素认证。 基于时间的一次性密码（TOTP）是这个领域的标 准算法 4](#_Toc26803)

[1.12. Enlive是基于Clojure 的模板框架，她彻底分离程序代码和HTML标记，用CSS选择 器查找并替换文档片段 4](#_Toc9818)

[1.13. 。本地存储同步 local storage sync 5](#_Toc14399)

[1.14. Jackrabbit 3，是一个可扩展和高性能的层次化内容存储系统， 5](#_Toc21940)

[஡వӑԁ఩Ҭ (Text-it-as-a-service) 5](#_Toc10456)

[/ 5](#_Toc32278)

[1.15. IndexedDB 6](#_Toc24275)

[1.16. Nashorn 是一个面向 Java 8 发布的全新 JavaScript 引擎 6](#_Toc30183)

[1.17. Retrofit过将一个 REST API 转化为 Java 接口， 6](#_Toc13496)

[2. 没落技术 6](#_Toc7241)

[2.1. gitflow 6](#_Toc10489)

[2.2. Application Servers 46. OSGi 47. SPDY 6](#_Toc27254)

[2.3. 团队应该克制他们的互联网级 规模的情结而倾向于简单可行的解决方案。 7](#_Toc6724)

## 离线优先Web应用程序（Offline first web applications）

提 供了基于缓存和更新机制来设计 Web 应用离线访问的能力。 它的实现需要在DOM中设定一个标志来检查接入设备是否在 线，离线则访问本地存储，在线则同步数据。现在所有的主 流浏览器都支持离线模式，通过显示的指定 HTML 属性来使 本地信息可访问，同时启动如 HTML，CSS，Javascript 或其 他资源的下载和缓存。当前已经有一些工具使离线优先应用 的实现变的简单，如Hoodie（hood.ie）， CouchDB（couchdb.apache.org），不仅如此它们还提供 与本地部署的本地存储应用的集成能力。 大多数软件开发的心智模型都是做项目，在不同的档期内进 行计划、执行和交付。敏捷开发极大的挑战了这种模型，通 过与开发过程同时进行的持续需求发现，代替了预先的需求 确定。精益创业的技术，如观察需求的A/B测试 （martinfowler.com/bliki/ObservedRequirement.html）， 进一步削弱了这种心态。我们认为，大多数的软件开发工作 应该遵循精益企业的引领（info.thoughtworks.com/leanenterprise-book.html），将自己定义为构建支持业务流程 的产品。这样的产品并没有所谓的最终交付，更多的是一个 探索如何更好的支持和优化业务流程的过程，只要业务依然 有价值就会不断持续下去。基于这些理由，我们提倡企业组 织

## 依照产品而不是项目（products rather than projects）的 思路进行思考。

## Postman（getpostman.com/features）是一个在Chrome 中使用的REST客户端插件

，通过Postman，你可以创 建请求并且分析服务器端返回的信息。这个工具在开发新 的 API或者实现对于已有API的客户端访问代码时非常有 用。Postman支持OAuth1和OAuth2，并且对于返回的 JSON和XML数据都会进行排版。通过使用Postman，你可 以查看你通过Postman之前发起过的请求，并且可以非常 友好的编辑测试数据去测试API在不同请求下的返回。同 时，虽然我们不鼓励录屏式的测试方法，但是Postman提 供了一系列的拓展允许我们将它作为跑测试的工具。

## PlantUML（

在我们的经验中，相比其他办法而言，通过在一开始创建大 量详尽的设计图表来开发软件系统，并不是什么好的选择。 但用图表来表达系统中格外复杂和难懂的部分却往往是个好 的想法，何况UML本身也已经提供了若干实用的、众所周知 的图表。我们喜欢用PlantUML（plantuml.sourceforge.net） 来创建这些图表，因为它让我们以清晰的文字形式来表达图 表背后的意图，而不用再去摆弄那些“过度”的图形化工具。 同时，文字形式的表达方式还支持版本管理，并且可以和源 代码存放在一起。

## NoPSD运动想要抓住的，是“即时设计”这个在视觉设计中重 要和实用的概念。

你不需要预先设计好整个应用程序或者每一个 界面元素，而是使用尽可能轻量的工具按需设计。我们已经看到 简单易学的工具被更多的使用，比方说Sketch，纸和笔也同样 有更多回归的趋势（尤其是与一个现有强大的数字风格指南配 合使用的时候）。由于为屏幕设计时平面模型的限制，使用类 似于Invision、FramerJS和Origam这样的工具，或者是简单的 HTML/CSS加上一点Javascript来创造不同的保真效果的原型， 也已经成为沟通设计意图时逐渐常见的很有用的方式。

## “产品优 于项目”

我们长期坚信将软件开发看做一个项目（即在有限的时间和预算 内交付一些东西）的观点并不适合现代的商业模式了。软件开发 的主要努力应该用在使软件成为一个支撑和反思业务并持续演进 的产品。在其所支撑的业务失效以前，这样的努力都不应停止。 根据我们在自己以及外部项目中的观察，我们确信这种“产品优 于项目”的方法对除特殊情况外的所有情况都有效

## Css bem

调试CSS问题是非常痛苦的，有多少次你不得不从成千上万的 样式覆盖中找出问题所在。为减少这种痛楚，很多团队引入了各 种各样的代码规范(guidelines)，比如避免级联与覆盖，使用真 正需要的(opt-in)样式，并强调有意义的命名。BEM(代表Block, Element,Modifier)是一种简明的CSS命名规范，它能把你的 CSS变得更语义化和结构化。通过使用BEM，可以更容易的理 解CSS规则是怎么影响一个元素的，并且更重要的是理解CSS规

## HTML5 iFrame来沙盒化

现代的网页往往含有过多的来自第三方的JavaScript部件和代码 片段，这对安全和性能都会产生负面影响。虽然我们仍在等待基 于Web组件的JavaScript完全隔离方案，我们的团队已经从使用 HTML5 iFrame来沙盒化不可信的JavaScript中受益匪浅。

## 幂等过滤器

现在有很多服务，尤其是传统的服务，在编写时都假设请求仅会 发生一次。做为网络本身而言，这很难保证。幂等过滤器是个简 单的组件，他通过检查重复请求并保证发给服务器的请求只有一 技术 接上页 次。过滤器应当做为装饰器叠加在现有的服务调用上，并且它应 该仅做这一件事情。

## “只积 累”是一种设计策略，数据删除通过撤回而不是更新Accumulate-only data

不可变的数据结构正在变得越来越受欢迎，函数式语言，如 Clojure 和 Scala 默认就提供不变性。不变性使得代码更容易编 写、阅读和理解。在数据库层，采用“只积累”的数据存储也给 予了一些同样的好处，并且使审计和历史查询更为简单。实现 © November 2015, ThoughtWorks, Inc. All Rights Reserved. TECHNOLOGY RADAR NOVEMBER 2015 | 7 技术 接上页 选择各不相同，从使用 Datomic 这样特定的积累数据存储，到 在传统数据库中采用简单的“只追加，不更新”的手段。“只积 累”是一种设计策略，数据删除通过撤回而不是更新，而“只增 加”只是一种具体的实现技术

## 双 因素认证。 基于时间的一次性密码（TOTP）是这个领域的标 准算法

以我们建议采用可显著提高安全性的双 因素认证。基于时间的一次性密码（TOTP）是这个领域的标 准算法，服务器端的实现简单直接，在谷歌和微软的智能手机 中也免费提供这样的认证应用

## Enlive是基于Clojure 的模板框架，她彻底分离程序代码和HTML标记，用CSS选择 器查找并替换文档片段

Mustache或FreeMarker等模板框架都把代码和标记混杂在 同一个文件中以实现复杂的动态内容。Enlive是基于Clojure 的模板框架，她彻底分离程序代码和HTML标记，用CSS选择 器查找并替换文档片段。Enlive展示了函数式编程的威力，用 一组简单的、可组合的函数作用于通用的抽象结构来实现复杂 行为。我们使用Clojure的团队认为Enlive是个既实用又直接 的工具。

## 。本地存储同步 local storage sync

当我们实现单页应用 (single-page applications) 时，“离线使 用”这个问题迟早都会浮出水面。鉴于很难正确的将离线模式加 入一个现有的应用中，“离线优先”的思想已经成为了开发单页 应用的一种趋势。本地存储同步 local storage sync 就是一种我 们曾经成功运用过的重要实现技术。使用这种技术，面向用户的 代码将不再发送请求到后端系统，而是仅仅从本地存储 (local storage) 中获取数据。并由一个后台服务对本地存储和后 端系统进行同步，通常，会调用某种形式的 REST API。

拥抱“数据湖”的理念就是要去除因缺少 ETL 开发人员或是堆 积了太多前台数据模型设计而造成的瓶颈。它旨在帮助开发人员 以敏捷的模式，基于他们需要数据的时机和方式 (在合理的限度 内)，来创建他们自己的数据处理管道。这也与我们评价很高的 另一模型 —— 技术运维 (DevOps) 模型有很多相同点。

## Jackrabbit 3，是一个可扩展和高性能的层次化内容存储系统，

Jackrabbit Oak (http://jackrabbit.apache.org/oak/)，之前叫 做 Jackrabbit 3，是一个可扩展和高性能的层次化内容存储系统， 可用于内容管理系统的基础。其存储方式除了使用文件系统之 外，也支持 MongoDB 和 RDMS，其更适合于大数据量的场景。 虽然它使用 Java 实现，但是可以通过类似于 JCR 这样的标准从 各种平台很方便地对其进行访问。

஡వӑԁ఩Ҭ (Text-it-as-a-service)

/Rapidpro (http://rapidpro.io/) 为业务提供轻松地创建或修改复杂的短 信服务应用的能力，同时无需开发人员的大力支持。相对于 USSD 会话来说，文字信息的成本更低。这提供了一个更加经济 实惠的方式，来针对功能手机 (feature phones) 构建具有伸缩性 的应用程序。这在我们的项目中已经看到了成功的案例。流程非 常容易构建；行为也可以在诸如发送短消息、邮件，甚至是调用 外部 API 等任意时点被触发 。

## IndexedDB

随着单页面应用和离线优先 (RɞLQH\_x0010\_ȴUVW) 的开发方式变得更加 可行和普遍，在 web 浏览器中持久化数据的需求与日剧增。本 地存储 (Local Storage) 非常容易使用，而且得到了大多数浏览器 的支持。而对于更加复杂的使用场景，IndexedDB 就能派上用 场了。尽管有时候 IndexedDB 不失为一个好的解决方案，但是 我们仍然建议只有在绝对必要的情况下再使用它，这是因为使用 它会增加复杂性，而且其 API 也多少有些笨拙。我们在对 LocalForage (https://github.com/mozilla/localForage) 框架的 使用中也获得了良好的体验，它在那些各

## Nashorn 是一个面向 Java 8 发布的全新 JavaScript 引擎

。当我 们需要在浏览器和服务器执行完全相同的代码时——例如做校验 或者数据迁移 —— 1DVKRUQ就是 Java 世界里的可选方案之一， 虽然 Java 应用中很少会遇到这种边缘场景。尽管可以通过 1RGH或 Avatar 来使1DVKRUQ承载整个应用，但是我们还不确 定这是不是一种好的实现。 5HWURȴW(KWWS\_x001D\_\_x0012\_\_x0012\_VTXDUHJLWKXELR\_x0012\_UHWURȴW\_x0012\_)

## **Retrofit**过将一个 REST API 转化为 Java 接口，

为在 Android 项目上构建 HTTP 客户端 提供了一

# 没落技术

## gitflow

## Application Servers 46. OSGi 47. SPDY

## 团队应该克制他们的互联网级 规模的情结而倾向于简单可行的解决方案。