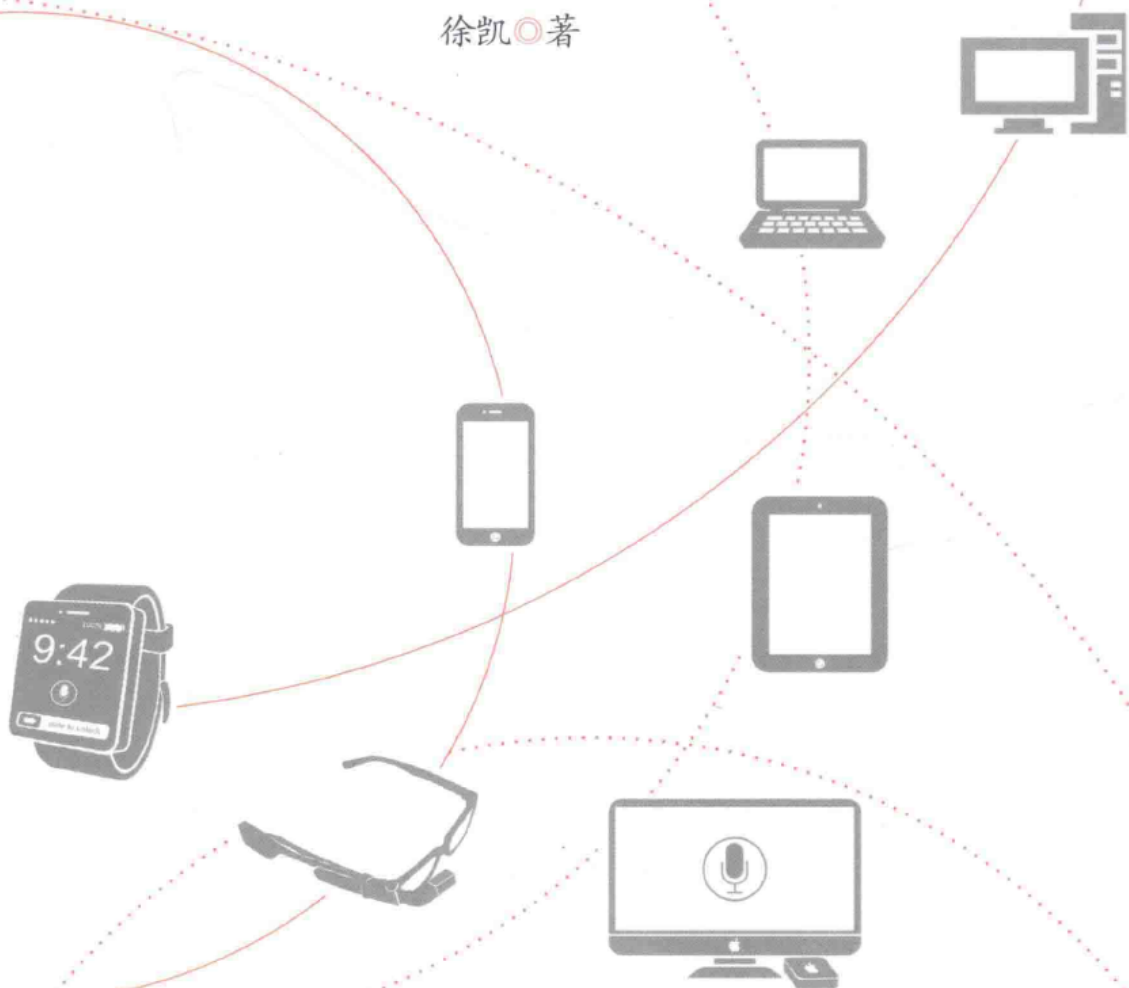


跨终端Web

徐凯◎著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

跨终端Web

徐凯◎著



电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

移动互联网不可阻挡地进入了我们的生活。作者将自己在百度和天猫期间的跨终端 Web 的开发实践转化为书中的技术方案和实现,呈现给各位读者。第 1 章提出了跨终端 Web 的概念以及实现跨终端 Web 的多重途径,第 2 章主要介绍 Mobile Web 的技术基础,第 3~7 章是全书的核心,按照开发流程组织逐步讲解了实现跨终端 Web 所需要的各类技术基础设施,第 8 章主要介绍了 Hybrid App 的发展历程、实现细节以及成熟的框架,第 9 章介绍的跨终端存储方案(Storage)是作者曾经的冠军作品,第 10 章完整介绍了如何通过脚本录制和回放来实现跨终端动作同步。

本书讲解深入浅出,通俗易懂,适合有一定 PC Web 基础,希望迅速了解 Mobile Web,致力于 PC 和 Mobile Web 技术融合的读者。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

跨终端 Web / 徐凯著. —北京:电子工业出版社, 2014.6
ISBN 978-7-121-23345-6

I. ①跨… II. ①徐… III. ①网页制作工具 IV. ①TP393.092

中国版本图书馆 CIP 数据核字(2014)第 110704 号

策划编辑:张春雨

责任编辑:付 睿

印 刷:北京丰源印刷厂

装 订:三河市鹏成印业有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编:100036

开 本:787×980 1/16 印张:15.25 字数:270 千字

版 次:2014 年 6 月第 1 版

印 次:2014 年 6 月第 1 次印刷

印 数:4000 册 定价:55.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件到 dbqq@phei.com.cn。

服务热线:(010) 88258888。

献给我的父母。

献给李玉北，我的第一位导师。

献给三七，我的第二位导师。

献给舒文，我在天猫的师兄。

献给所有百度和天猫时期的战友们。

推荐序

2012 年年底，我从云 OS 运营团队转岗到天猫前端团队，之前和手机硬件厂商、Mobile OS 及 App 市场的合作经历让我重新思考前端的未来和电子商务的技术方向。除了在天猫内部实践基于移动优先（Mobile First）的设计研发外，还和 InfoQ 合作把 Qcon 北京 2013 的前端议题定为“跨终端 Web”，于是就有了采访稿《从可编程到跨终端——QCon 北京 2013 “跨终端的 Web” 专题出品人鄢学鹏专访》¹，阐述当时的认知和想法。现在“跨终端 Web”已经诞生一年半了，鬼道同学把这些想法和实践都编写成书实在让人惊喜和敬佩，我有幸成为这个成就的见证人，并再次有机会和大家谈谈跨终端 Web 的初心和思考。

1991 年诞生了 World Wide Web、HTTP 和 HTML，1995 年诞生了 JavaScript、Java 和 PHP，1996 年诞生了 CSS，1998 年 CSS 2.1 正式发布，1999 年 CDN 诞生且 HTTP 1.1 和 HTML 4.01 正式发布，2000 年 ECMA-262 3rd 正式发布，2001 年 IE 6 发布。作为 Web 开发者和前端工程师的我们应该对这些大事件相当不陌生，但如果你仔细看看这些年份就会发现我们所使用的绝大部分技术和方案都比 IE 6 老。在那个时期没有真正意义的前端，常见情况是美工切图，后端嵌套模板，也没有所谓的前端架构，前后端的发布机制几乎一样，那就是门户盛行的时代，Yahoo、新浪、搜狐和网易当道。

IE 6 是第一轮浏览器大战的胜者，垄断式的胜利直接导致 Web 基础设施至少 10 年的缓慢发展，标准的意义也就不那么明显了。其实 1994 年就有了 W3C，但民间为了解决 W3C 面对浏览器厂商不给力的混乱状况于 1998 成立了 WaSP（The Web

¹ <http://www.infoq.com/cn/news/2013/01/qcon-beijing-web-interview>

Standards Project), 其中一个创始人于 2003 年发表了影响力巨大的著作《Design with The Standards (网站重构)》, 当然在中国产生了估计作者都没有想到的额外影响, 那就是出现了重构工程师。2004 年 Firefox 1.0、Gmail 和 Google Suggest 发布, 2005 年 AJAX 横空出世和 Google Maps 发布, 2006 年 jQuery 和 YUI 发布。这些事件的到来显示出了低成本的 Web 在跨浏览器上的巨大威力和人机交互的显著改善, 导致当时互联网 99% 以上内容都是使用 Web 来呈现的, 所以对于当时绝大部分人而言, WWW 就是互联网。巨大的需求导致 Web 开发的规模化和专业分工, 这就是前端工程师开始出现并大量集结于以 Web 业务为主的互联网公司的原因。

我们可以称这个时代为前端工程师 1.0, 比如在 Google、Facebook、Yahoo、阿里巴巴、百度、腾讯等公司中, 前端团队小有规模且以 Web 开发为主, 此时的前端工程师们大量使用 HTML、CSS、JavaScript 开发基于桌面浏览器的网页和应用。这个时代的前端工程师主要具备 3 个方面的能力, 一是跨浏览器的兼容能力, 需要理解渐进增强和优雅退化的思想, 深入 Web 标准并结合环境数据制定 GBS (Graded Browser Support 浏览器分级标准); 二是富交互 Web 的开发能力, 能够基于, 甚至开发 JavaScript 库去实现人和机器的复杂互动; 三是性能优化能力, 实践表明前端占 Web 性能 80%, 类似 CDN 和按需加载等各种性能优化方案深度地影响了前端架构和发布机制。

虽然今日市面上大部分前端工程师招聘条件还是基于前端工程师 1.0 的, 但我们的环境自 2007 年 iPhone 发布起就在悄然却快速地变革, 我们可能是家里唯一使用电脑的那个人, 而手机在吃饭、走路、约会、上厕所等时都已经成为不可或缺的陪伴。当我还在云 OS 做运营时, 便有一个令人震惊的发现: 人类发展五千年到现在的最大共同点可能是每人都有移动智能设备。人们已经处于各种各样碎片化的智能移动设备上, 这些设备有触摸屏、相机、麦克风、陀螺仪、加速计等特性, 使得人同机器交互的方式发生了巨大的革命, 更重要的是它们都和网络连接, 没有了离线状态。于是, 当一个人在大街上看到奇闻异事随手用手机拍下来并发到微博上, TA 的朋友可能在手机、平板电脑或普通电脑的微博上看到并转发, 也可能被 TA 的朋友转到微信或 IM 等, TA 朋友的朋友同样可能在手机、平板电脑或普通电脑的另一种软件上看到这个信息, 甚至信息被电视媒体或平面媒体发现并造成更大范围的传播, 更多 TA 不认识的电视或平面媒体通过扫描报道时的二维码又找到 TA 的微博, 这一切

可能在瞬间完成。这个小场景告诉我们人和信息是在智能终端（Phone、Pad、Desktop、TV 等设备）、软件终端（浏览器、SNS、IM 等应用）和传统终端（电视、广播、平面等媒体）间交叉流动的，这是一个去中心的网状结构，也是互联网的本质特性，所以跨终端不仅仅是跨越设备（device），更是跨越人机交互的场景入口（end）。

面对这种终端碎片化的潮流，前端工程师怎么办？解决方案就是基于最重要的前端开发思想渐进增强和优雅退化得出的移动优先：一是毫无疑问绝大部分用户已经或正在成为智能设备用户，我们要为 80% 的目标用户服务；二是专注于核心业务需求，人的本性、业务本质和商业模式本质基本上不会随着终端改变而改变，所以相同业务在手机、平板、桌面和电视上呈现的本质和商业模式不会有不同，小屏幕终端是我们重新思考业务本质和核心人机交互流程的机会，其挖掘出的本质会改变其他终端；三是针对未来人机交互，现在移动设备引领人机交互的变革潮流，通过必备特性虚拟或增强现实，并逐步引入到桌面和电视等设备中。所以，选择哪个具体技术方案，是响应式 Web，还是服务端响应式 Web（URL 不变，通过服务端在不同设备展现不同模板），还是多个 URL Web（不同的终端，不同的 URL），还是 Hybrid 应用（Native 和 Web 混合使用，比如 iOS App Store 一直就是引用壳里面套个网站），还是 Native 应用，是依据业务本质、人性需求和人机环境趋势来综合判断的。由于 Web 的本质特性就是低成本跨平台但对设备先进特性支持不够，而 Native 应用能够充分利用硬件先进特性但受限于系统平台，导致开发者没有发布能力，所以大多数情况下，可以把跨终端 Web 作为默认选择。

在跨终端的时代，渐进增强和优雅退化依旧是最重要的前端开发思想，但前端工程师们不能像以前一样仅仅固守在 Web 上，Web 和客户端应用的融合已经成为必然，Web 从页面（page）到应用（application）反映了人机交互革命带来新的体验趋势，客户端和动画开发成为了前端工程师的基本功。移动优先的跨终端解决方案核心是一套数据有多个高品质低成本展现方式，这促使前后端分离成为必需，前端工程师不仅仅要关心客户端环境也要关心服务端环境，所以 GBS 需要升级到 GTE（Graded Target Environments，分级目标环境），工程师更关注端到端的数据接口约定（比如正文中的 IF 就是这种思想下的一种实践方案），这会彻底改变前端的开发方式、架构和发布机制，也会导致前端团队快速膨胀直到前后端比率趋于一致。这是一个巨

大的挑战，也是一个前所未有的机会，更是时势的要求。我把这个时代叫作前端工程师 2.0。

这篇推荐序被我拖延了很久很久，结尾之时恰逢鬼道同学入职天猫一周年，就拿老乔的话来 zhuangbility 一下：“你如果出色地完成了某件事，那你应该再做一些其他的精彩事儿。不要在前一件事上徘徊太久，想想接下来该做什么。”这应该是最好的礼物之一。

三七

2014.4.8

前言

移动互联网不可阻挡地进入了我们的生活。笔者将自己在百度和天猫期间的跨终端 Web 的开发实践转化为书中的技术方案和实现，呈现给各位读者。

本书大纲来自于笔者 2013 年 7 月在 D2 上的主题分享“移动优先的跨终端 Web”²，2013 年 11 月 W3CTECH 2013 上做了第二次分享³。

面向对象

本书适合有一定 PC Web 基础，希望迅速了解 Mobile Web，致力于 PC 和 Mobile Web 技术融合的读者。

本书的第 1 章、第 3 章、第 5 章面向所有的读者，即使您对 Web 技术毫无了解，仍可以顺利阅读下来。本书其他章节假定您对 PC Web 前端技术已经有所了解，知道 HTML、CSS、JavaScript 这些名词的含义，并且动手写过一页。

章节简介

第 1 章阐述了移动互联网的现状和遇到的问题，提出了跨终端 Web 的概念以及实现跨终端 Web 的多重途径，破除了“唯 Media Query 论”，并引入移动优先的概念，丰富跨终端 Web 的内核。

第 2 章从对比 PC Web 的角度介绍了 Mobile Web 的技术基础，并对远程调试、兼容

² <http://luics.github.io/demo/d2/>

³ <http://luics.github.io/demo/cew-w3ctech-1311/>

在线视频：<http://www.imooc.com/course/video/?mid=535>

性等开发问题做了详尽的阐述。

第 3~7 章是全书的核心，按照开发流程组织，逐步讲解了实现跨终端 Web 所需要的各类技术基础设施，包括：

- 基准，给出了调试和测试过程中的基准，确定调试和测试的范围。
- 检测，探讨如何构造一个为全站服务的终端属性检测工具。
- 接口，探索实现流程复用的途径及端到端的接口规范（IF）。
- 定位，分别介绍了基于 Hash 和基于 History API 的跨终端定位方案。
- 预览，介绍了实现跨终端预览的方案，并给出一个简化的实现。

第 8 章介绍了 Hybrid App 的发展历程、实现细节以及成熟的框架，并详细分析了以传感器为核心的 Device API。

第 9 章的跨终端存储方案（Storage）是笔者参加“2013 Kissy Gallery 组件大赛”时的冠军作品。

第 10 章完整介绍了如何通过脚本录制和回放来实现跨终端动作同步。

感谢

感谢三七（鄢学鹏），“跨终端 Web”这个概念最早由三七提出，感谢三七为本书写序以及在本书写作过程中给予的支持。三七是位充满智慧的前端架构师和优秀的管理者，从他那得到的启示远不止跨终端 Web，在此一并谢过。

感谢舒文（舒文亮），他为本书第 4 章“检测”写了初稿。舒文带领跨公司的团队实现了服务于全阿里的多终端检测（MED）系统。

感谢惜朋（沙峰），他为本书第 10 章“动作同步”撰写了“脚本录制和回放”的初稿，并实现了其中的 Demo。

感谢闭月（吴燕云）和蔡伦（盛柳钦），他们为本书第 7 章“预览”提供了 Focus 截图和服务端代码。闭月和蔡伦的家庭新成员即将诞生，在此祝福小生命为他们的生活带来更多的欢乐。

感谢 IF 项目虚拟团队的蒋壮（王跃乐）、劳谦（董松洁），他们分别为 IF 在测试和后端领域内的扩展做了开拓性的工作。

鬼道 于 2014-02-23

目录

1 跨终端 Web.....	1
1.1 终端 VS.设备	1
1.2 一个贯穿全书的例子	2
1.3 后续章节	3
1.4 移动优先	4
1.4.1 移动流量暴增	4
1.4.2 聚焦业务本质	5
1.4.3 人机交互扩展	7
1.4.4 再说书名	8
1.5 不只是响应式	8
1.5.1 响应式	8
1.5.2 多站点	13
1.5.3 多模板	13
1.5.4 多平台	15
1.6 解决方案	16
2 Mobile Web	17
2.1 HTML5	18
2.2 HTML	19
2.2.1 移动页面模板	19
2.2.2 Viewport	21
2.2.3 touch-icon	26
2.2.4 其他	27
2.3 触屏事件	27
2.3.1 触屏事件一览	27

2.3.2 通用触屏事件	28
2.4 调试	31
2.4.1 远程调试	31
2.4.2 设备调试	43
2.5 兼容性	44
2.5.1 OS 版本碎片化	44
2.5.2 国内的特殊情况	46
2.5.3 WebView	46
2.5.4 更多工具	46
2.6 文档	48
3 基准	51
3.1 GBS	51
3.2 MGBS	53
3.2.1 准备	53
3.2.2 操作系统分级	54
3.2.3 屏幕分辨率分级	65
3.2.4 浏览器分级	71
3.2.5 MGBS	73
3.3 GTE	75
3.3.1 分层设计	76
3.3.2 核心层	76
3.3.3 数据层	78
4 检测	81
4.1 终端	81
4.1.1 什么是终端	81
4.1.2 分类	82
4.2 终端检测	82
4.2.1 场景	82
4.2.2 原理	83
4.2.3 实现	85
4.3 遗留问题	86
4.3.1 硬件信息	86

4.3.2 更精准的终端检测	86
5 接口	87
5.1 跨终端流程复用	87
5.1.1 示例 1	87
5.1.2 示例 2	88
5.2 IF	89
5.2.1 始于一次重构	90
5.2.2 新的环境	95
5.2.3 模型	95
5.2.4 解决方案	96
5.2.5 架构	96
5.2.6 路线图	99
5.3 if-spec 2.0	102
5.3.1 JSON Schema	102
5.3.2 Demo	109
5.3.3 meta	113
5.3.4 if-spec 1.0	114
5.4 if-mock 2.0	116
5.5 if-guide 2.0	118
5.6 总结	122
6 定位	125
6.1 定位	126
6.1.1 Hash	126
6.1.2 History API	127
6.1.3 视图定位	129
6.2 数据	129
7 预览	131
7.1 客户端	132
7.2 服务端	133
7.3 示例	136

8 Hybrid App	139
8.1 Hybrid 简史	139
8.1.1 背景	139
8.1.2 简史	140
8.1.3 现状	142
8.2 Hybrid 技术	144
8.2.1 Native 调用 Web	144
8.2.2 Web 调用 Native	144
8.2.3 Bridge	146
8.3 Hybrid 框架	150
8.3.1 PhoneGap	151
8.3.2 Titanium	152
8.4 Device API	153
8.4.1 动作传感器	156
8.4.2 环境传感器	158
8.4.3 音频	159
8.4.4 视频	160
8.5 小结	160
9 存储	161
9.1 状态持久化	162
9.2 技术方案	163
9.2.1 整体方案	163
9.2.2 跨终端存储方案	164
9.2.3 跨域通信方案	166
9.2.4 安全性	168
9.2.5 遗留问题	168
9.3 使用	168
9.3.1 实例化	169
9.3.2 set/get	169
9.3.3 remove/clear	170
9.3.4 推荐命名	170

10 动作同步	171
10.1 原理	171
10.1.1 案例	171
10.1.2 动作同步	172
10.2 实现	173
10.2.1 Selenium	173
10.2.2 脚本录制和回放	174
附录 A GBS	183
附录 B JSON Schema Core	189
附录 C JSON Schema Validation	201
附录 D if-spec 2.0	221
作者简介	225

跨终端 Web

1.1 终端 VS.设备

当移动页面只运行于独立浏览器中时，“终端”和“设备”（图 1-1）这两个名词在多数场景下是可以相互替换的。现在移动页面不仅仅运行于独立浏览器中，也运行于众多 App 的 WebView¹中。终端和设备的概念出现了微妙的变化：页面可以运行于同一设备的不同终端下。本书取名为“跨终端”而不是“跨设备”也正是适应了这个变化。

“跨终端”中的“跨”强调了在多个终端上的信息传递、相互协作等动态内涵。跨终端希望从移动优先的理念出发，从设计、产品、开发都能够先做出移动版本再层层累加，这是理念上的“跨”；另一方面，对于用户而言，看到的页面地址永远都只有一个，不同终端间的差异应该对用户透明，这是现实的“跨”。



图 1-1 移动设备

¹ Android 中称为 WebView，iOS 中称为 UIWebView，后续章节简称为 WebView；WebView 就好比是嵌入 App 的迷你浏览器，实际上这个迷你浏览器和大部分独立浏览器一样都使用 WebKit 内核。