Broadview®
www.broadview.com.cn

WEEX
跨平台开发实战

向治洪 / 著

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

WEEX
跨平台开发实战

向治洪 / 著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

近年来，伴随着大前端和移动跨平台技术的兴起，移动应用的开发手段越来越多，常见的移动跨平台技术有 React Native、WEEX 和 Flutter 等。WEEX 是由阿里巴巴研发的一套移动跨平台技术框架，目的是解决移动应用开发过程中频繁发版和多端研发的问题。

本书是一本系统介绍 WEEX 跨平台应用开发的书籍，涵盖了 WEEX 开发的方方面面，主要由基础知识、高级应用开发和项目实战三部分组成。第一部分重点介绍 WEEX 开发的基础知识，后两部分则重点介绍 WEEX 开发的进阶知识和项目实战。

本书是一本 WEEX 入门与实战类书籍，适合有一定前端开发基础或者移动端开发基础的读者阅读。因此，无论你是前端开发者，还是移动端开发者，都可以通过对本书的学习来掌握移动跨平台应用开发的技能。

图书在版编目（CIP）数据

WEEX 跨平台开发实战 / 向治洪著. —北京：电子工业出版社，2019.9
ISBN 978-7-121-36895-0

Ⅰ. ①W… Ⅱ. ①向… Ⅲ. ①移动终端－应用程序－程序设计 Ⅳ. ①TN929.53

中国版本图书馆 CIP 数据核字（2019）第 123434 号

责任编辑：刘恩惠
印　　刷：三河市良远印务有限公司
装　　订：三河市良远印务有限公司
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：20　字数：448 千字
版　　次：2019 年 9 月第 1 版
印　　次：2019 年 9 月第 1 次印刷
定　　价：79.00 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）88254888，88258888。

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

本书咨询联系方式：（010）51260888-819，faq@phei.com.cn。

前言

近年来，伴随着大前端概念的提出和兴起，移动端和前端的边界变得越来越模糊，一大批移动跨平台开发框架和模式涌现出来。从早期的 PhoneGap、Inoic 等 Hybrid 技术，到现在耳熟能详的 React Native、WEEX 和 Flutter 等跨平台技术，无不体现着移动端开发的前端化。

作为阿里巴巴开源的一套移动跨平台技术框架，WEEX 框架最初是为了解决移动开发过程中频繁发版和多端研发的问题而开发的。具体来说，使用 WEEX 提供的跨平台开发技术，开发者可以很方便地使用 Web 前端技术来构建高性能、可扩展的原生性能体验，并支持在 Android、iOS 和 Web 等多平台上进行部署。

作为目前主流的跨平台技术框架之一，WEEX 项目使用 Vue.js 进行编写，对于熟悉 Web 前端开发的开发者来说，其是一个不错的选择。在性能和项目迭代方面，WEEX 与 PhoneGap、Inoic 等 Hybrid 技术相比也有一定的优势。

不过由于种种原因，WEEX 的社区生态并不是很完善，也没有一本系统介绍 WEEX 的书籍。基于对跨平台技术的热爱，以及积累的一些 WEEX 项目实战经验，笔者思量再三决定对 WEEX 框架进行系统的梳理，并将其整理成书。

“路漫漫其修远兮，吾将上下而求索。”通过对 WEEX 技术的学习和本书的写作，笔者深刻地意识到“学无止境”的含义。如果本书对你学习 WEEX 有所帮助和启发，笔者将不胜欣慰。

如何阅读本书

本书共分为 9 章，章节概要如下。

第 1 章～第 4 章：这 4 章属于 WEEX 入门与基础部分。这部分内容主要包括 WEEX 简介、WEEX 环境搭建、WEEX 基础知识以及 WEEX 开发常用的组件和模块等内容。同时，本部分内容配备了大量的实例，通过这部分内容的学习，读者将会对 WEEX 有一个基本的认识。

第 5 章～第 8 章：这 4 章属于 WEEX 进阶部分。这部分内容主要由讲述 Rax、Vue.js、BindingX

和 WEEX Eros 的章节组成，主要是介绍 WEEX 开发中的一些进阶知识。同时，为了加快 WEEX 的开发效率，建议开发者直接使用 WEEX Eros 和 weexplus 等 WEEX 脚手架。

第 9 章：这一章属于 WEEX 项目实战部分。这部分讲述了 WEEX 项目实战的内容，是对 WEEX 基础知识的综合运用。通过此部分的知识讲解，读者将会对 WEEX 有一个全面的认识。

希望通过本书的讲解，读者可以对 WEEX 技术有一个全面的了解，并能够使用它进行移动跨平台项目的开发。

适读人群

本书是一本关于 WEEX 入门与实战的书籍，适合前端开发者和移动 Android/iOS 开发者阅读。因此，不管你是一线移动开发工程师，还是有志于从事移动开发的前端开发者，都可以通过学习本书来获取移动跨平台开发的技能。

读者服务

轻松注册成为博文视点社区用户（www.broadview.com.cn），扫码直达本书页面。

- **提交勘误**：您对书中内容的修改意见可在 提交勘误 处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- **交流互动**：在页面下方 读者评论 处留下您的疑问或观点，与我们和其他读者一同学习交流。

页面入口：*http://www.broadview.com.cn/36895*

目录

第1章 WEEX简介

近年来，伴随着“大前端”概念的提出和兴起，移动端和前端的边界变得越来越模糊，一大批移动跨平台开发框架和模式涌现出来。从早期的 PhoneGap、Inoic 等 Hybrid 混合技术，到现在耳熟能详的 React Native、WEEX 和 Flutter 等跨平台技术，无不体现着移动端开发的前端化。

2016 年 6 月，阿里巴巴开源了 WEEX 移动跨平台框架。WEEX 在语法上是使用 Vue.js 编写的，更加贴近 Web 前端开发，在性能和快速迭代方面，相比其他框架也有一定的优势。

1.1 WEEX 概述

WEEX 是由阿里巴巴研发的一套移动跨平台技术框架，最初是为了解决移动开发过程中频繁发版和多端研发的问题而开发的。使用 WEEX 提供的跨平台技术，开发者可以很方便地使用 Web 技术来构建具有可扩展的原生性能体验的应用，并支持在 Android、iOS、YunOS 和 Web 等多平台上部署。具体来说，当在项目中集成 WeexSDK 之后，就可以使用 JavaScript（JS）和主流的前端框架来开发移动应用了。

同时，WEEX 框架的结构是解耦的，渲染引擎与语法层分离，也不依赖任何特定的前端框架，目前，开发者可以使用 Vue.js 和 Rax 两个前端框架来进行 WEEX 页面开发。同时，WEEX 的另一个主要目标是跟进流行的 Web 开发技术并将其与原生开发技术相结合，实现开发效率和运行性能的高度统一。

1.1.1 原生平台与 Web 平台的差异

WEEX 作为一个跨平台解决方案，致力于使用 Web 开发技术来构建 Android、iOS 和 Web 跨平台应用。也就是说，WEEX 不仅可以运行在 Web 环境中，还可以运行在 Android 和 iOS 等客户端环境中。不过，尽管 WEEX 致力于打造三端统一的跨平台应用，并尽可能保持多平台的

一致性，但原生客户端平台和 Web 平台之间天生存在的平台差异决定了客户端应用和 Web 应用存在一定的区别，具体体现在开发方式和应用体验上。

WEEX 不支持 DOM 操作

DOM（文档对象模型）是 Web 编程中的概念，即 HTML 和 XML 文档的编程接口。WEEX 的运行环境是以原生应用为主的，在 Android 和 iOS 环境中使用原生组件执行界面渲染，而不使用 DOM 元素，所以任何涉及 DOM 元素的操作都不被 WEEX 支持。

同时，WEEX 标签支持事件绑定操作，不过和浏览器捕捉及触发事件的方式不同，WEEX 中的事件是由原生组件捕捉并触发的，并且事件的属性也和 Web 平台中的有差异。

WEEX 不支持 BOM 操作

BOM（浏览器对象模型）是浏览器为 JavaScript 提供的接口，由于 WEEX 在移动客户端运行时并不需要浏览器环境，所以 WEEX 也不支持浏览器提供的 BOM 接口以及 BOM 接口提供的 API。

由于 WEEX 并未提供浏览器中的 window 对象和 screen 对象，并且不支持使用全局变量，因此想要获取设备的屏幕或环境信息，可以使用 WXEnvironment 变量。同时，WEEX 也没有提供面向浏览器的 history、location 和 navigator 等对象，如果要管理和操作视图之间的跳转，需要借助 WEEX 提供的 navigator 模块来实现。

WEEX 支持调用移动原生 API

WEEX 能够调用移动原生 API，主要通过注册、调用模块来实现。虽然 WEEX 中内置了一些通用的组件和模块，如 clipboard、navigator 和 storage 等，不过，WEEX 内置的原生模块往往非常有限，为了保持框架的通用性，此时可以通过 WEEX 提供的横向扩展能力来扩展原生模块。

1.1.2 设计理念

为了让读者更全面地了解 WEEX 框架，以及在合适的场景中使用 WEEX 进行跨平台应用开发，本节会从 WEEX 的设计理念入手，详细介绍 WEEX 的设计哲学和优势。

性能为王

性能对于 WEEX 来说，是最为重要的核心价值，也是 WEEX 区别于传统的基于 WebView 的 Hybrid 框架的重要特性之一。和 React Native 的机制一样，WEEX 的虚拟 DOM 机制使用 JavaScript 上下文来维护页面布局，并通过向移动终端发送规范化的渲染指令，进而调用

Android、iOS 的原生渲染引擎来渲染界面。相比于浏览器环境通过渲染系统实现渲染的方式，WEEX 是通过原生系统 UI 体系来达到更佳的性能和用户体验的。

同时，WEEX 也采取了多种手段来优化性能体验，包括优化 JavaScript 与 Native 端的通信频率和通信量，使用二进制的方式降低单次通信的耗时等。未来还会通过跨平台内核将 DOM 管理移至原生层实现，以彻底解决原生平台与 JavaScript 层之间进行异步通信带来的成本问题，从多个维度提升 WEEX 引擎的性能。

交互与体验

提升交互体验一直是 WEEX 不断追求的目标。与 React Native 等跨平台技术方案不同，WEEX 希望在 Android、iOS 及 Mobile Web 等终端上具有完全一致的表现，因此 WEEX 在内置组件的设计上充分考虑到终端表现的一致性，并为 WEEX 开发者提供一致的交互体验。

为了满足这一需求，WEEX 借助 GCanvas 等组件增强了框架的 2D 及 3D 渲染能力，为高性能渲染场景提供了可能；同时，借助 WEEX 的 Expression Binding 交互理念，用户与应用交互时的体验得以提升。对于前端开发中列表性能较差的问题，WEEX 提供了基于模板和数据分离的 recycle list 组件，大大提升了列表的渲染和交互性能。

更高的开发效率

易用和高效，是 WEEX 不断致力提升的方向。从一开始，WEEX 的设计理念就是面向前端开发技术栈，利用前端技术在开发体验和效率上的优势，为开发者提供接近前端开发体验的开发环境。

基于此，开发者可以使用 JavaScript、CSS 和 HTML 技术来进行 WEEX 开发。WEEX 的内置组件则通过 HTML 标签提供给开发者，API 也通过 JavaScript 对象提供。同时，WEEX 还支持 Vue.js DSL，因此开发者可以利用 Vue.js 提供的强大易用的开发范式来进行 WEEX 应用开发。

易于扩展

易于扩展是衡量一个框架好坏的重要特性之一，因此从一开始 WEEX 就注重易于扩展方面的设计。基于 WEEX 的实现特点，WEEX 在框架层面提供了模块和组件两种扩展方式。

其中，模块用于扩展无 UI 的基础功能接口，而组件则用于扩展包含 UI 的界面组件，开发者可以根据需要，选择不同的扩展方式实现需求。

1.1.3 WEEX 工作原理

作为一套前端跨平台技术框架，WEEX 建立了一套源码转换以及原生平台与 JavaScript 通信的机制。WEEX 表面上是一个客户端框架，但实际上它串联起了从本地开发、云端部署到分

发的整个链路。

具体来说，整个链路的串联过程是这样的：在开发阶段编写一个.we文件，然后使用WEEX提供的weex-toolkit转换工具将.we文件转换为JSBundle，并将生成的JSBundle部署到云端，最后通过网络请求或预下发的方式将其加载至用户的移动应用客户端；当集成了WeexSDK的客户端接收到JSBundle文件后，再调用本地的JavaScript引擎来执行相应的JSBundle，并将执行过程中产生的各种命令发送到原生平台进行界面渲染、数据存储、网络通信以及用户交互响应。WEEX的整个工作流程图如图1-1所示。

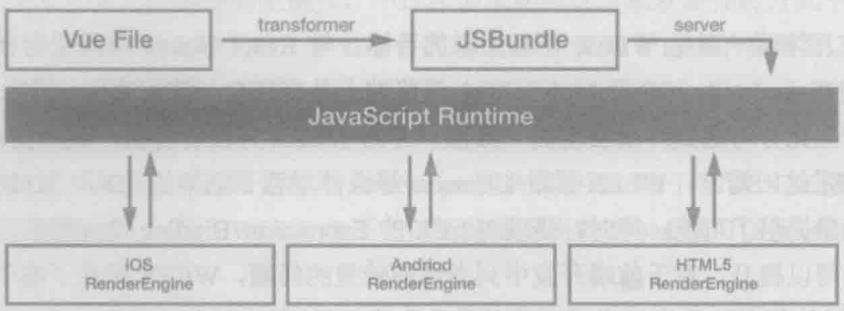

图1-1 WEEX的整个工作流程图

由图1-1可知，WEEX框架中最核心的部分是JavaScript Runtime。具体来说就是，当需要执行渲染操作时，在iOS环境下选择基于JavaScriptCore的iOS系统提供的JSContext，在Android环境下使用基于JavaScriptCore的JavaScript引擎。

当JSBundle从服务器端下载完成之后，WEEX在Android、iOS和Web端会运行一个JavaScript引擎来执行JSBundle，同时向各终端的渲染层发送渲染指令，并调度客户端的渲染引擎实现视图渲染、事件绑定和处理用户交互等操作。

由于Android、iOS和HTML5等终端最终使用的是原生的渲染引擎，也就是说使用同一套代码在不同终端上展示的样式是相同的，并且WEEX使用原生引擎渲染的是原生的组件，所以在性能上要比传统的WebView方案好很多。

当然，尽管WEEX已经提供了开发者所需要的最常用的组件和模块，但面对丰富多样的移动应用研发需求，这些常用基础组件还是远远不能满足开发的需要，因此WEEX提供了灵活自由的扩展能力，开发者可以根据自身的情况定做属于自己客户端的组件和模块，从而丰富WEEX生态。

1.2 移动跨平台技术剖析

“得移动端者得天下”，移动端取代PC端，成了互联网行业最大的流量分发入口，因此不