



Learning Web Component Development

Web Component 实战

探索 PolymerJS、Mozilla Brick、Bosonic 与 ReactJS 框架

[印] Sandeep Kumar Patel 著
范洪春 邵锋 何语萱 姜天意 译



中国工信出版集团



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

Learning Web Component Development

Web Component 实战

探索PolymerJS、Mozilla Brick、Bosonic与ReactJS框架

【印】Sandeep Kumar Patel 著
范洪春 邵锋 何语萱 姜天意 译

電子工業出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

随着前端技术的发展，组件化和模块化的思想越来越深入人心。早在 2013 年，Web Component 就被写入了 W3C 工作草案中。Google、Facebook、Yahoo、Mozilla 等走在前端的公司均对 Web Component 标准进行了具体实现。本书详细讲解了 Web Component 规范，并且结合具体的类库阐述了所有可以构建 Web Component 规范的关键特性。适用于每一位对 Web Component 规范感兴趣的开发者。

Copyright © 2015 Packt Publishing. First published in the English language under the title 'Learning Web Component Development'.

本书简体中文版专有出版权由 Packt Publishing 授予电子工业出版社。未经许可，不得以任何方式复制或抄袭本书的任何部分。专有出版权受法律保护。

版权贸易合同登记号 图字：01-2015-6065

图书在版编目（CIP）数据

Web Component 实战：探索 PolymerJS、Mozilla Brick、Bosonic 与 ReactJS 框架 /（印）帕特尔（Patel,S.K.）著；范洪春等译. — 北京：电子工业出版社，2015.10

书名原文：Learning Web Component Development

ISBN 978-7-121-27382-7

I. ① W…II. ① 帕…② 范…III. ① 网页制作工具—软件包 IV. ① TP393.092

中国版本图书馆 CIP 数据核字（2015）第 241275 号

责任编辑：张春雨

印 刷：三河市鑫金马印装有限公司

装 订：三河市鑫金马印装有限公司

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：13.5 字数：302.4 千字

版 次：2015 年 10 月第 1 版

印 次：2015 年 10 月第 1 次印刷

定 价：65.00 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至 dbqq@phei.com.cn。

服务热线：（010）88258888。

译者序

“创建一个大型应用的秘诀就是不去创建大型应用，而把应用切分成组件。然后组装测试，由多个组件组合而成一个应用。”

——Justin Meyer, JavaScript MVC 作者

JavaScript 是一门松散类型的动态语言，因而在规范不够完善、社区不够健全的蛮荒时代，开发者们写出的代码也是参差不齐。没有既定规范的代码可读性差，而软件工程所提倡的复用性、可维护性、可扩展性就更是无从谈起。很难说是技术推动了产品的发展，还是快速变化的互联网带来了前端技术的进步。不过最终的结果是前端技术以惊人的速度进行迭代。正如在 JSConf 2015 开发者大会上流行起来的前端摩尔定律：“每 18 ~ 24 个月，前端都会难一倍”。前端模块化方案从无到有，从稍有模式的命名空间到 JavaScript 中的 AMD、CommonJS 模块定义，从浏览器端模块化方案再到依赖编译的 webpack、browserify 构建工具。这一路走来，很多新的标准应运而生，毫无疑问这其中开源社区的贡献巨大。不得不说，ES6 的模块系统很大程度上受到了 Node.js 模块的启发，HTML5 Web API 所引入的 `querySelectorAll` 也一定是在 jQuery 选择器中获得了灵感。

而 Web Component 则带来了 UI 组件化的全新标准。它能够帮助开发者在开发过程中创建出稳定且可复用的组件。Web Component 包括模板元素、HTML Import、Shadow DOM 以及自定义元素。模板元素包含了 HTML、CSS 和 JS，可以取代传统的字符串模板；HTML Import 则可以将外部的 HTML 文档引入到当前页面中，实现了复用；Shadow DOM 则提供了样式和脚本的作用域，是 Web Component 中重要的一环；而自定义元素则可以达到扩充 HTML 元素的目的，意味着开发好一个元素后，处处可用。Web Component 做到了分离关注点，降低了代码的维护成本并且提升了开发效率，可以说是 Web 前端发展的必由之路。

值得注意的是，就现在而言 Web Component 还是一门新技术，很多浏览器都没有原生实现它。而很多走在前面的公司也以自己的方式实现了这套新标准，其中包括 Google 实现的 PolymerJS、Mozilla 实现的 X-Tag，以及 Facebook 实现的 React。它们或通过 polyfill 实现标准，或通过 Virtual DOM 另辟蹊径。目前来看，不得不说 React 正大放异彩，国内外基于 React 搭建的项目犹如雨后春笋般崛起，这很大程度上得益于它所提出的 UI 状态机、Virtual DOM 及单向数据流。当然，其他两个框架的设计巧思和理念同样值得每位开发者学习。

本书会对这三种实现都做出详细的介绍，同时包含每个框架的周边类库及构建工具，并且附有详细的代码示例。作者在书中详细介绍了 Web Component 的每个细节，并且通过实例深入浅出地讲解了三个基于 Web Component 规范的前端框架及其构建工具。相信读者在读完本书后一定会有所收获，能够开发出自己的 Web Component 项目，或者将其思想用到现有的项目中。

本书的翻译团队由四位一线 Web 开发者组成：

- 范洪春，就职于阿里巴巴数据技术及产品事业部，前端架构方向
- 邵锋，就职于 Splunk 中国研发中心，目前从事大数据可视化方向的开发和研究
- 何语萱，就职于腾讯 ISUX，任 UI 工程师
- 姜天意，就职于阿里巴巴国际事业部，主要从事 Node.js 服务端研发工作

此外，我、语萱和天意同时是知乎专栏《前端外刊评论》的维护者，本书的试读章节也会在第一时间发布到该专栏。也就意味着你在阅读本书时遇到翻译上的任何问题都可以通过专栏与我们取得联系。

最后，感谢博文视点引进了这本书，并给予我们翻译的机会。感谢 Sandeep Kumar Patel 创作了这本书，让更多的开发者了解到这一新标准。感谢我们四个人在这段时间的相互帮助和鼓励。感谢博文视点的编辑对我们信任。感谢我在阿里的两个前端团队的同事在我负责翻译本书期间对我的理解和支持。父母之恩，毋须赘言！

范洪春

2015.10 于杭州

关于作者

Sandeep Kumar Patel 是一位高级 web 工程师，技术博客 www.tutorialsavvy.com 的创始人，该博客自 2012 年创建以来一直被广泛阅读。他在面向对象的 JavaScript 和基于 JSON 的 web 应用开发上，有超过 5 年的经验；曾获 GATE-2005 信息技术 (IT) 资格认证及印度韦洛尔理工大学硕士学位。你可以在 LinkedIn ([http:// www.linkedin.com/in/techblogger](http://www.linkedin.com/in/techblogger)) 上了解到更多关于他的个人信息。DZone 曾在 web 技术出版方向上授予他最具价值博主 (MVB) 奖项。你可以在 <http://www.dzone.com/users/sandeepgi> 查看他的文章。此外，他还获得过 JCG (Java Code Geeks) 颁发的 JCG 徽章。同样，可以在 <http://www.javacodegeeks.com/author/sandeep-kumar-patel/> 查看他的文章。

他的其他书籍如下：

- *Instance GSON*
- *Responsive Web Design with AngularJS*

我要感谢生命中最重要的三个人：一直爱我的爸爸妈妈，Dilip Kumar Patel 和 Sanjukta Patel，当然还有一直支持和鼓励我的妹妹，Sangeeta Patel。由衷感谢 Packt 出版机构，没有他们就不会有这本书。

关于审稿人

Zhelan Chen 毕业于德克萨斯大学达拉斯分校，主修计算机科学技术。她曾在多家公司从事 IT 工作，包括 Java EE、.NET 应用、web 前端及网络服务开发。多年以来，她一直利用业余时间，在达拉斯社区学院教授计算机课程。她曾获得多项 Oracle Java EE 认证。

Krzysztof Cislo 是一名软件开发者，10 年来一直专注于 web 应用架构。他的 web 开发之旅开始于使用 PHP 语言建站。与此同时，他也保持对前端语言的了解，包括 JavaScript、HTML 和 CSS。不久之后，又完全转向服务端技术，Java 在很长一段时间内都是他最主要的兴趣。然而，最近他又对前端技术产生了兴趣，正在考虑重返 JavaScript 世界。

Lars Kappert (<http://webpro.nl>) 是一名来自荷兰的前端架构师和开发主管。他专注于架构、解决方案、性能、工具及 web 站点和应用的开发。他的工作和很多核心 web 技术密切相关，包括 HTML5、NodeJS 和 CSS。他是 GitHub (<https://github.com/webpro>) 上活跃的开源开发者，也会将文章发布到 Mideum (<https://medium.com/@webprolific>)、Smashing Mag (<https://www.smashingmagazine.com/author/lars-kappert/>) 等。你可以在 Twitter 上 @webprolific (<https://twitter.com/webprolific>) 关注他。

Sebastian Metzger 毕业于德国埃朗根-纽伦堡大学，主修信息系统技术。他在很大的银行和小型创业公司都工作过，并且于 2014 年成立了自己的软件开发及技术咨询公司。早在 2011 年，他就已经借助谷歌 Web 工具箱 (GWT) 使用 Java 语言开发单页面应用。现在，他已经步入了基于 NodeJS 和 AngularJS 的全栈 JavaScript 开发之路。他借助 Google Polymer 于 2014 年创建了他自己的第一个 Web Component 应用，可以在 <http://foodtrack.de> 查看到。你一定要在 <http://sebastianmetzger.com> 看一下他的公司主页，因为他经常在这里针对一些新技术和趋势写博客。

前言

欢迎你来到《Web Component 实战》。如果你想学习和理解 W3C 提出的 Web Component 规范，使用 Polymer、Bosonic、Mozilla Brick 及 ReactJS 开发自己的 Web Component，那这本书一定很适合你。它提供了一个系统化的方法来构建响应式 web 应用程序。这本书会阐述所有可以帮助构建 Web Component 规范的关键特性，并附有所需的完整源代码。

本书概览

第 1 章 Web Component 简介

本章主要介绍 Web Component，囊括了创建 Web Component 单元的详细说明。

第 2 章 Polymer 简介

本章主要介绍关于 Google 的 Polymer 类库。解释了该类库的架构体系。同样也探索了 core 和 paper 元素。

第 3 章 使用 Polymer 开发 Web Component

本章主要介绍关于如何使用 PolymerJS 开发自己的 Web Component。同样也提供了使用该类库创建自定义组件的详细指南。

第 4 章 探索 Web Component 开发工具 Bosonic

本章侧重于讲解 Bosnic 工具。阐述了如何使用这些工具创建自定义组件。

第 5 章 使用 Mozilla Brick 开发 Web Component

在本章你会接触到 Mozilla Brick 类库，囊括了 Brick 类库的简明介绍。同样也包含了使用 Brick 创建各类组件的代码示例。

第 6 章 使用 ReactJS 开发 Web Component

本章侧重于讲解 ReactJS，解释了什么是 React 方式。包含了使用 ReactJS 创建 Web Component 的代码示例。

附录 Web Component 参考文献

列出了进一步学习 Web Component 可能用到的站点和论坛。

阅读本书可能用到的工具

本书所需的工具和类库如下。

- WebStorm IDE
- 最新的 Chrome
- GIT
- npm
- Bower
- Yeoman
- Grunt

面向读者

本书适用于有意向学习和开发 Web Component 的开发者，对于那些想要学习市场上各类框架来开发 Web Component 的开发者同样适用。当然，这本书对于每一个有兴趣深入了解 Web Component，想要开发属于自己的自定义组件的开发者也很适用。

本书约定

在本书中，会有不同的文本样式用于区分不同类别的信息。下面是这些样式的部分示例，并且解释了它们的含义。

文本段中的代码、数据表名、文件夹名、文件名、文件扩展名、路径名、URL、用户输入的文本及 Twitter 账号会按照下面的方式展示：

“我们可以通过使用 `include` 指令来包含其他的上下文。”

代码块会遵循下面的方式：

```

<!DOCTYPE html>
<html>
  <head lang="en">
    <meta charset="UTF-8">
    <title>Web Component: template support</title>
  </head>
  <body></body>
</html>

```

如果想强调代码块的某一特定部分，会加粗相应的行或者项：

```

<body>
  <div id="container"></div>
  <template id="aTemplate">
    <h1>Template is activated using importNode method.</h1>
  </template>
</script>

```

所有的输入或者输出命令行遵照下面的方式书写：

```
npm install --save b-sortable
```

新的术语和关键字加粗显示。在屏幕上出现的单词，比如在菜单或者弹窗中，会以类似这样的文本出现：“现在，单击 **Delete SayHello Element**，可以从 DOM 树中移除该元素”。



弹窗中会有类似这样的警示或者重要提示。



类似这样的小提示或小技巧。

读者反馈

我们永远欢迎读者的反馈。告诉我们你觉得这本书怎么样——喜欢或者不喜欢什么。读者反馈对我们来说非常珍贵，它会帮助我们改进，相信你会从中受益的。

仅仅需要将反馈内容通过 broadview@phei.com.cn 发送给我们，并且在邮件的主题中指明书籍的名称。

如果你在哪一方面有专长并且对撰写或者贡献一本书很有兴趣，可以在 <http://www.broadview.com.cn> 参阅我们的作者向导。

示例代码下载

可以在博文视点官网 (<http://www.broadview.com.cn/>) 下载到本书及其他博文视点书籍对应的示例代码。具体步骤为：① 在首页点击进入“下载”专区；② 通过书名、ISBN 号或作者名中任意一项找到本书；③ 按照提示完成下载即可。

本书勘误

尽管我们已经尽全力确保内容的准确性，但错误还是可能会出现。如果你在我国的书籍中发现了错误（或许是文本或者代码中的错误），如果能将其告知我们，感激不尽。这样，你可以挽回其他读者的沮丧心情并且帮助我们提升这本书的后续版本。如果有任何勘误信息，请访问 <http://www.broadview.com.cn>，找到本书并按提示提交勘误信息。一旦你的勘误信息被核实，你的提交将被接收并且勘误信息会被上传到我们站点或者被添加到这本书新印次的版本中。

版权声明

对于所有的媒体来说，互联网上的盗版现象都是一个持续存在的问题。我们会谨慎地保护我们的版权和许可。如果你在互联网上发现以任何形式非法复制我们的书籍进行传播，请立即提供给我们网址或者站点名以方便我们进行补救。请通过 broadview@phei.com.cn 联系我们，并附上涉嫌盗版材料的链接。

非常感激你在保护我们的作者上所做出的帮助，这也潜在地帮助我们从而为你提供更有价值的内容。

目录

前言	xiii
第 1 章 Web Component 简介	1
Web Component 是什么?	1
Web Component 的使用动机和困境	2
Web Component 架构	2
模板元素	3
模板元素的细节	3
检测是否支持模板特性	4
延迟加载的模板	6
激活模板	7
HTML Import	11
HTML Import 特性检测	11
访问引入的文档	13
HTML Import 的事件	15
Shadow DOM	16
Shadow DOM 特性检测	17
Shadow tree	19
自定义元素	24
自定义元素特性检测	24
自定义元素的开发	26
示例: 自定义元素	29

节点分布	31
内容插入点	31
Shadow 插入点	33
为 Web Component 添加样式	35
构建时钟组件.....	39
时钟模板.....	39
时钟元素注册脚本.....	40
组件的使用	41
X-Tag.....	43
X-Tag 元素生命周期	43
开发 X-Tag 自定义元素	44
Polymer	48
Mozilla Brick.....	48
ReactJS	48
Bosonic	48
总结.....	49
 第 2 章 Polymer 简介	50
什么是 Polymer?	50
安装和配置 Polymer.....	50
下载 ZIP 文件	51
使用 GIT 克隆.....	51
使用 Bower.....	52
PolymerJS 结构.....	53
包含 polyfill 的 Web Component	54
Polymer 库	56
元素.....	56
Core 元素	56
core-tooltip 元素.....	62
paper 元素.....	64
Material Design	65
Polymer Designer 工具.....	72

使用设计工具开发	73
获取 GitHub 令牌	73
开发 E-mail 订阅表单	74
Yeoman Polymer 生成器	75
Polymer 生成器命令	76
Polymer 应用生成器	77
Polymer 元素生成器	77
Polymer 种子生成器	78
Polymer Github 页面生成器	78
总结	78
 第 3 章 使用 Polymer 开发 Web Component	79
PolymerJS 的 ready 事件	79
Polymer 表达式	80
Polymer 模板的自动绑定	82
过滤器表达式	86
内建的过滤器表达式	86
自定义过滤器表达式	89
全局的过滤器表达式	91
开发 Polymer 自定义元素	92
定义自定义元素	93
定义元素属性	93
定义默认属性	94
定义公共属性和方法	94
发布属性	95
定义生命周期方法	95
注册自定义元素	97
开发一个自定义元素示例	97
扩展自定义元素	99
Polymer 方法	101
Polymer 的 mixin 方法	101
Polymer 的 import 方法	103
Polymer 的 waitingFor 方法	104

Polymer 的 forceReady 方法.....	106
异步任务执行.....	107
开发一个数显时钟.....	109
使用 Yeoman.....	111
Yeoman 元素生成器.....	111
Yeoman 种子生成器.....	114
Yeoman GitHub 页面生成器.....	115
为生产环境下使用 vulcanize 做准备.....	116
安装 vulcanize.....	116
运行 vulcanize 进程.....	116
总结.....	117

第 4 章 探索 Web Component 开发工具 Bosonic..... 118

Bosonic 是什么?	118
浏览器支持情况	119
配置 Bosonic.....	119
Bosonic 包.....	119
内建元素	120
b-sortable 元素.....	120
b-toggle-button 元素	122
开发自定义元素	124
第 1 步 创建 red-message 元素目录.....	124
第 2 步 进入到 red-message 目录.....	125
第 3 步 为 <red-message> 生成一个框架.....	125
第 4 步 验证目录结构.....	125
第 5 步 定义 <red-message> 元素的代码.....	126
第 6 步 修改 index.html 示例文件.....	127
第 7 步 使用 Grunt 生成分发文件.....	127
第 8 步 运行 index.html 文件.....	130
Bosonic 生命周期	130
生命周期示例	130
开发一个数显时钟	136
总结.....	140

第 5 章 使用 Mozilla Brick 开发 Web Component	141
Brick 库是什么?	141
Mozilla Brick 1.0	142
Mozilla Brick 2.0	142
安装 Mozilla Brick	143
配置 Mozilla Brick	144
内置组件	145
brick-calendar 元素	145
brick-flipbox 元素	145
brick-deck 元素	148
brick-tabbar 元素	151
brick-action 元素	154
brick-menu 元素	157
X-Tag 库	158
使用 X-Tag 来开发一个数显时钟	159
总结	164
 第 6 章 使用 ReactJS 开发 Web Component	 165
走近 React	165
Flux 架构	165
Flux 的关键特性	166
安装 ReactJS	167
配置 ReactJS	168
使用 ReactJS	168
JSX 是什么	170
使用 JSX 创建自定义组件	170
ReactJS 行内样式	172
ReactJS 事件处理	174
使用非 DOM 属性	176
ReactJS 组件生命周期	179
ReactJS 初始化	179
ReactJS 存在期	180
ReactJS 销毁期	181

ReactJS 生命周期示例	181
有状态的自定义组件	184
生产环境预编译 JSX	186
JSX 文件监听	190
使用 ReactJS 开发一个数显时钟	190
第 1 步 开发脚本定义数显时钟组件生命周期	190
第 2 步 定义数显时钟组件的 CSS 样式	191
调试 ReactJS	192
总结	194
 附录 Web Component 参考文献	 195