

Vue.js 前端开发

快速入门与专业应用

陈陆扬 著



学以致用

以实际项目为立足点，
拒绝纸上谈兵，
帮助你快速上手！

应用广泛

摸透Vue.js，无论是
桌面端还是移动端，
让你从容面对！

求职必备

国内互联网公司已广泛
使用Vue.js开发，
再不学就晚了！



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

Vue.js 前端开发

快速入门与专业应用

陈陆扬 著



人民邮电出版社
北京

图书在版编目 (C I P) 数据

Vue.js 前端开发快速入门与专业应用 / 陈陆扬著
— 北京 : 人民邮电出版社, 2017. 2
ISBN 978-7-115-44493-6

I. ①V… II. ①陈… III. ①三维动画软件 IV.
①TP391.414

中国版本图书馆CIP数据核字(2017)第004640号

内 容 提 要

本书主要介绍 Vue.js 的使用方法和在实际项目中的运用, 它既可以在一个页面中单独使用, 也可以将整站都构建成单页面应用。为了便于理解, 本书会从传统的开发角度切入, 先从数据渲染、事件绑定等方面介绍在 Vue.js 中的使用方法, 然后渐进到 Vue.js 自身的特性, 例如数据绑定、过滤器、指令以及最重要的组件部分。除了框架用法外, 本书还介绍了和 Vue.js 相关的重要插件和构建工具, 这些工具有助于帮助用户构建一个完整的单页面应用, 而不仅仅是停留在个人 DEMO 阶段的试验品。而对于复杂项目, Vue.js 也提供了对应的状态管理工具 Vuex, 降低项目的开发和维护成本。鉴于完稿前, Vue.js 2.0 已正式发布完毕, 本书也在相关用法上对比了 1.0 和 2.0 的区别, 并补充了 render 函数和服务端渲染等特性。

本书适用于尚未接触过 MVVM 类前端框架的开发者, 或者初步接触 Vue.js 的开发者, 以及实际应用 Vue.js 开发项目的开发者。

-
- ◆ 著 陈陆扬
责任编辑 赵 轩
责任印制 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
固安县铭成印刷有限公司印刷
 - ◆ 开本: 720×960 1/16
印张: 12.75
字数: 299 千字 2017 年 2 月第 1 版
印数: 1-3 000 册 2017 年 2 月河北第 1 次印刷
-

定价: 45.00 元

读者服务热线: (010) 81055410 印装质量热线: (010) 81055316
反盗版热线: (010) 81055315

前言

PREFACE

近年来，前端框架的发展依旧火热，新的框架和名词依旧不停地出现在开发者眼前，而且开发模式也产生了一定的变化。目前来看，前端 MVVM 框架的出现给开发者带来了不小的便利，其中的代表就有 Angular.js、React.js 以及本书中将要介绍的 Vue.js。这些框架的产生使得开发者能从过去手动维护 DOM 状态的繁琐操作中解脱出来，尽可能地让 DOM 的更新操作是自动的，状态变化的时候就自动更新到正确的状态。不过，新框架的引入不可避免的就是学习成本的增加以及框架普及性的问题，相对于 Angular.js 和 React.js 而言，Vue.js 的学习曲线则比较平稳。目前在 GitHub 上已经获得了超过 30000 的 star，成为了时下无论从实用性还是普遍性来说都是可靠的 MVVM 框架选择之一。

首次听说 Vue.js 的时候，都是介绍说体积小、适合移动端、使用简单，等等。但一开始对于新框架我一直持观望态度，毕竟前端框架更新太快，而且这又是个个人项目，仅由作者尤雨溪一人维护，不像 Angular.js 和 React.js 那样有公司支持。后来为了解决一个移动端的项目，我才正式接触了 Vue.js。由于项目本身天然存在组件这个概念，并且需要在手机上运行，调研后觉得应该没有比 Vue.js 更适合的工具了。在使用过程中，逐渐体会到了 Vue.js 的便利，数据绑定及组件系统对于提高开发效率和代码复用性来说都有相当大的帮助，并且初期对线上项目使用这种新框架的顾虑也渐渐消除了，即使随着后期复杂度的增加也并没有对项目的开发和维护成本造成影响。

本书主要从我自身的学习和开发经验出发，介绍了 Vue.js 的基础用法和特性，包括 Vue.js 的一些插件用法，用于解决客户端路由和大规模状态管理，以及打包发布等构建工具，便于正式用于线上环境。

最后，感谢 Vue.js 作者尤雨溪先生为前端开发者提供了这款优秀的框架，使得开发者能够更好地应对项目复杂度；也感谢人民邮电出版社的大力支持，写书的过程的确对人是一种折磨和考验；最后感谢每天早上 4 点多就开始叫我起床的两只猫，它们对本书的进度的确起到了很好的推动作用。

目 录

Contents

第1章 Vue.js简介	
1.1 Vue.js是什么	1
1.2 为什么要用Vue.js	2
1.3 Vue.js的Hello world	2
第2章 基础特性	
2.1 实例及选项	5
2.1.1 模板	6
2.1.2 数据	7
2.1.3 方法	9
2.1.4 生命周期	10
2.2 数据绑定	12
2.2.1 数据绑定语法	13
2.2.2 计算属性	17
2.2.3 表单控件	18
2.2.4 Class与Style绑定	21
2.3 模板渲染	22
2.3.1 前后端渲染对比	23
2.3.2 条件渲染	23
2.3.3 列表渲染	25
2.3.4 template标签用法	27
2.4 事件绑定与监听	28
2.4.1 方法及内联语句处理器	28
2.4.2 修饰符	29
2.4.3 与传统事件绑定的区别	30
2.5 Vue.extend()	31
第3章 指令	
3.1 内置指令	32
3.1.1 v-bind	32
3.1.2 v-model	33
3.1.3 v-if/v-else/v-show	33
3.1.4 v-for	33
3.1.5 v-on	34
3.1.6 v-text	34
3.1.7 v-HTML	35
3.1.8 v-el	35
3.1.9 v-ref	35
3.1.10 v-pre	36
3.1.11 v-cloak	36
3.1.12 v-once	37
3.2 自定义指令基础	37
3.2.1 指令的注册	37
3.2.2 指令的定义对象	37
3.2.3 指令实例属性	39
3.2.4 元素指令	40
3.3 指令的高级选项	41
3.3.1 params	41
3.3.2 deep	42
3.3.3 twoWay	43
3.3.4 acceptStatement	44
3.3.5 terminal	44
3.3.6 priority	45
3.4 指令在Vue.js 2.0中的变化	46
3.4.1 新的钩子函数	46
3.4.2 钩子函数实例和参数变化	46
3.4.3 update函数触发变化	47
3.4.4 参数binding对象	47
第4章 过滤器	
4.1 过滤器注册	48
4.2 双向过滤器	49
4.3 动态参数	50
4.4 过滤器在Vue.js 2.0中的变化	51
第5章 过渡	
5.1 CSS过渡	52
5.1.1 CSS过渡的用法	52
5.1.2 CSS过渡钩子函数	54
5.1.3 显示声明过渡类型	57
5.1.4 自定义过渡类名	57

5.2 JavaScript过渡	58	7.1.8 路由配置项	95
5.2.1 Velocity.js	58	7.1.9 route钩子函数	96
5.2.2 JavaScript过渡使用	58	7.1.10 路由实例属性及方法	99
5.3 过渡系统在Vue.js 2.0中的变化	59	7.1.11 vue-router 2.0 的变化	100
5.3.1 用法变化	59	7.2 Vue-resource	104
5.3.2 类名变化	60	7.2.1 引用方式	104
5.3.3 钩子函数变化	61	7.2.2 使用方式	105
5.3.4 transition-group	63	7.2.3 \$http的api方法和选项参数	105
第6章 组件		7.2.4 拦截器	107
6.1 组件注册	65	7.2.5 \$resource用法	107
6.1.1 全局注册	65	7.2.6 封装Service层	109
6.1.2 局部注册	66	7.3 Vue-devtools	109
6.1.3 注册语法糖	67	7.3.1 安装方式	110
6.2 组件选项	67	7.3.2 使用效果	110
6.2.1 组件选项中与Vue选项的区别	67	第8章 Vue.js工程实例	
6.2.2 组件Props	68	8.1 准备工作	111
6.3 组件间通信	71	8.1.1 webpack	111
6.3.1 直接访问	71	8.1.2 vue-loader	113
6.3.2 自定义事件监听	72	8.2 目录结构	115
6.3.3 自定义事件触发机制	72	8.3 前端开发	117
6.3.4 子组件索引	75	8.4 后端联调	122
6.4 内容分发	76	8.5 部署上线	124
6.4.1 基础用法	76	8.5.1 生成线上文件	124
6.4.2 编译作用域	77	8.5.2 nginx	125
6.4.3 默认slot	78	8.5.3 gitlab	126
6.4.4 slot属性相同	79	8.5.4 jenkins	127
6.4.5 Modal实例	79	第9章 状态管理：Vuex	
6.5 动态组件	82	9.1 概述	130
6.5.1 基础用法	82	9.2 简单实例	131
6.5.2 keep-alive	83	9.2.1 所需组件	131
6.5.3 activate 钩子函数	84	9.2.2 创建并注入store	132
6.6 Vue.js 2.0中的变化	85	9.2.3 创建action及组件调用方式	133
6.6.1 event	85	9.2.4 创建mutation	134
6.6.2 keep-alive	86	9.2.5 组件获取state	135
6.6.3 slot	87	9.3 严格模式	137
6.6.4 refs	87	9.4 中间件	137
第7章 Vue.js常用插件		9.4.1 快照	138
7.1 Vue-router	88	9.4.2 logger	138
7.1.1 引用方式	88	9.5 表单处理	139
7.1.2 基本用法	89	9.6 目录结构	142
7.1.3 嵌套路由	90	9.6.1 简单项目	142
7.1.4 路由匹配	92	9.6.2 大型项目	143
7.1.5 具名路由	93	9.7 实例	145
7.1.6 路由对象	93	9.7.1 state结构	146
7.1.7 v-link	94	9.7.2 actions.js	148

9.7.3	app.js	148	10.5.3	Switch	172
9.7.4	组件结构	148	10.5.4	Slider	172
9.7.5	base组件	152	10.5.5	wxc-tabbar	174
9.7.6	展示结果	154	10.5.6	wxc-navpage	175
9.8	Vue.js 2.0的变化	155	10.6	Weex内置模块	176
9.8.1	State	155	10.6.1	dom	176
9.8.2	Getters	156	10.6.2	steam	177
9.8.3	Mutations	157	10.6.3	modal	178
9.8.4	Actions	157	10.6.4	animation	179
9.8.5	Modules	159	10.6.5	webview	180
第10章	跨平台开发: Weex		10.6.6	navigator	181
10.1	Weex简介	161	10.6.7	storage	182
10.2	Weex安装	162	第11章	Vue.js 2.0新特性	
10.2.1	ios环境安装	162	11.1	Render函数	183
10.2.2	android环境安装	163	11.1.1	createElement用法	184
10.2.3	web端运行	164	11.1.2	使用案例	185
10.3	Weex实例与运行	164	11.1.3	函数化组件	186
10.4	Weex基础语法	168	11.1.4	JSX	187
10.4.1	数据绑定	168	11.2	服务端渲染	188
10.4.2	事件绑定	169	11.2.1	vue-server-renderer	188
10.4.3	模板逻辑	169	11.2.2	简单实例	189
10.5	Weex内置组件	170	11.2.3	缓存和流式响应	191
10.5.1	scroller	170	11.2.4	SPA实例	193
10.5.2	list	171			

第

章

Vue.js简介

近几年，互联网前端行业发展得依旧迅猛，涌现出了很多优秀的框架，同时这些框架也正在逐渐改变我们传统的前端开发方式。Google 的 AngularJS、Facebook 的 ReactJS，这些前端 MVC (MVVM) 框架的出现和组件化开发的普及和规范化，既改变了原有的开发思维和方式，也使得前端开发者加快脚步，更新自己的知识结构。2014 年 2 月，原 Google 员工尤雨溪公开发布了自己的前端库——Vue.js，时至今日，Vue.js 在 GitHub 上已经收获超过 30000star，而且也有越来越多的开发者在实际的生产环境中运用它。

本书主要以 Vue.js 1.0.26 版本为基准进行说明，Vue.js 2.0 版本与之不同的地方，会在对应章节中说明。

1.1 Vue.js是什么

单独来讲，Vue.js 被定义成一个用来开发 Web 界面的前端库，是个非常轻量级的工具。Vue.js 本身具有响应式编程和组件化的特点。

所谓响应式编程，即为保持状态和视图的同步，这个在大多数前端 MV* (MVC/MVVM/MVW) 框架，不管是早期的 backbone.js 还是现在 AngularJS 都对这一特性进行了实现（也称之为数据绑定），但这几者的实现方式和使用方式都不相同。相比而言，Vue.js 使用起来更为简单，也无需引入太多的新概念，声明实例 `new Vue({ data : data })` 后自然对 data 里面的数据进行了视图上的绑定。修改 data 的数据，视图中对应数据也会随之更改。

Vue.js 的组件化理念和 ReactJS 异曲同工——“一切都是组件”，可以将任意封装好的代

码注册成标签，例如：`Vue.component('example', Example)`，可以在模板中以 `<example></example>` 的形式调用。如果组件抽象得合理，这在很大程度上能减少重复开发，而且配合 Vue.js 的周边工具 `vue-loader`，我们可以将一个组件的 CSS、HTML 和 js 都写在一个文件里，做到模块化的开发。

除此之外，Vue.js 也可以和一些周边工具配合起来，例如 `vue-router` 和 `vue-resource`，支持了路由和异步请求，这样就满足了开发单页面应用的基本条件。

1.2 为什么要用Vue.js

相比较 Angularjs 和 ReactJS，Vue.js 一直以轻量级，易上手被称道。MVVM 的开发模式也使前端从原先的 DOM 操作中解放出来，我们不再需要在维护视图和数据的统一上花大量的时间，只需要关注于 data 的变化，代码变得更加容易维护。虽然社区和插件并没有一些老牌的开源项目那么丰富，但满足日常的开发是没有问题的。Vue.js 2.0 也已经发布了 beta 版本，渲染层基于一个轻量级的 virtual-DOM 实现，在大多数场景下初始化渲染速度和内存消耗都提升了 2~4 倍。而阿里也开源了 weex(可以理解成 ReactJS-Native 和 ReacJS 的关系)，这也意味着 Vue.js 在移动端有了更多的可能性。

不过，对于为什么要选择使用一个框架，都需要建立在一定的项目基础上。如果脱离实际情况我们来谈某个框架的优劣，以及是否采用这种框架，我觉得是不够严谨的。

作为新兴的前端框架，Vue.js 也抛弃了对 IE8 的支持，在移动端支持到 Android 4.2+ 和 iOS 7+。所以如果你在一家比较传统，还需要支持 IE6 的公司的话，你或许就可以考虑其他的解决方案了（或者说说服你的老板）。另外，在传统的前后端混合（通过后端模板引擎渲染）的项目中，Vue.js 也会受到一定的限制，Vue 实例只能和后端模板文件混合在一起，获取的数据也需要依赖于后端的渲染，这在处理一些 JSON 对象和数组的时候会有些麻烦。

理想状态下，我们能直接在前后端分离的新项目中使用 Vue.js 最合适。这能最大程度发挥 Vue.js 的优势和特性，熟悉后能极大的提升我们的开发效率以及代码的复用率。尤其是移动浏览器上，Vue.js 压缩后只有 18KB，而且没有其他的依赖。

1.3 Vue.js的Hello world

现在来看一下我们第一个 Vue.js 项目，按照传统，我们来写一个 Hello World。

首先，引入 Vue.js 的方式有很多，你可以采用直接使用 CDN，例如：

```
<script src='http://cdnjs.cloudflare.com/ajax/libs/vue/1.0.26/vue.min.js'></script>
```

也可以通过 NPM 进行安装：

```
npm install vue // 最新稳定版本
```

正确引入 Vue.js 之后，我们在 HTML 文件中的内容为：

```
<div id="app">
```

欢迎访问：电子书学习和下载网站 (<https://www.shgis.com>)

文档名称：《Vue.js前端开发 快速入门与专业应用》陈陆扬 著.pdf

请登录 <https://shgis.com/post/3392.html> 下载完整文档。

手机端请扫码查看：

