

Web开发经典丛书

Chris Fritz作序推荐

Vue.js in Action

Vue.js

开发 实战



[美] 埃里克·汉切特(Erik Hanchett)
本·利斯顿(Benjamin Listwon)

著

任强 邓龚达

译

MANNING



清华大学出版社

Web 开发经典丛书

Vue.js 开发实战

[美] 埃里克·汉切特(Erik Hanchett) 著
本·利斯顿(Benjamin Listwon) 译
任 强 邓龚达



清华大学出版社

北 京

Erik Hanchett, Benjamin Listwon

Vue.js in Action

EISBN: 978-1-61729-462-4

Original English language edition published by Manning Publications, USA (c) 2018 by Manning Publications. Simplified Chinese-language edition copyright (c) 2019 by Tsinghua University Press Limited. All rights reserved.

北京市版权局著作权合同登记号 图字: 01-2018-1493

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

Vue.js 开发实战 / (美)埃里克·汉切特(Erik Hanchett), (美)本·利斯顿(Benjamin Listwon) 著; 任强, 邓龚达 译. —北京: 清华大学出版社, 2019

(Web 开发经典丛书)

书名原文: Vue.js in Action

ISBN 978-7-302-53607-9

I. ①V… II. ①埃… ②本… ③任… ④邓… III. ①网页制作工具—程序设计
IV. ①TP393.092.2

中国版本图书馆 CIP 数据核字(2019)第 174350 号

责任编辑: 王 军

封面设计: 孔祥峰

版式设计: 思创景点

责任校对: 牛艳敏

责任印制: 丛怀宇

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社总机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印装者: 涿州市京南印刷厂

经 销: 全国新华书店

开 本: 170mm×240mm 印 张: 18 字 数: 384 千字

版 次: 2019 年 9 月第 1 版 印 次: 2019 年 9 月第 1 次印刷

定 价: 79.80 元

产品编号: 082441-01

操作说明

组件信息

```
Vue.component('my-component', {
  props: { 组件接收的参数
    myMessage: String,
    product: Object,
    email: {
      type: String,
      required: true,
      default: "test@test.com"
      validator: function (value) {
        返回 true 或 false
      }
    }
  },
  data: function() { 必须是一个函数
    return {
      firstName: 'Vue' ,
      lastName: 'Info'
    }
  },
  methods: { ... }
  computed: ( 当依赖发生变化时, 返回缓存的值
    fullName: function () {
      return this.firstName + ' ' + this.lastName
    }
  )
  watch: { 当 firstName 发生更改时被调用
    firstName: function (value, oldValue) {
      .. }
  },
  components: { 可以在模板中使用的组件
    ButtonComponent, IconComponent
  },
  template: '<span>{{ myMessage }}</span>',
}) 也可以使用反引号实现多行
```

自定义事件

使用 prop 将数据传递到子组件, 使用自定义事件将数据传递到父元素。

设置监听器到组件上, 监听器在父组件中:

```
<button-counter v-on:incrementThis="incVal">
```

父组件内:

```
methods: (
  incVal: function (toAdd) { . . . }
```

```
}
```

button-counter 内:

```
this.$emit('incrementThis', 5) 数据被发送给父组件
                                自定义事件名称
```

生命周期钩子

```
beforeCreate beforeUpdate
created       updated
beforeMount   beforeDestroy
mounted       destroyed
```

使用单插槽

组件模板:

```
<div>
  <h2>I'm a title</h2>
  <slot>
    //Only displayed if no content
  </slot>
</div>
```

带插槽数据的组件示例:

```
<my-component>
  <p>This will go in the slot</p>
</my-component>
```

多插槽

组件模板:

```
<div class="container">
  <header>
    <slot name="header"></slot>
  </header>
  <main>
    <slot>Default content is here</slot>
  </main>
  <footer>
    <slot name="footer"></slot>
  </footer>
</div>
```

带插槽数据的组件示例:

```
<app-layout>
<h1 slot="header">Page title</h1>
<p>the main content.</p>
<p slot="footer">Contact info</p>
</app-layout>
```

操作说明

表达式

```
<div id="app">
  <p>I have a {{ widget }}</p>
  <p>{{ widget + 's' }}</p>
  <p>{{ isWorking ? 'YES' : 'NO' }}</p>
  <p>{{ widget.getSalePrice() }}</p>
</div>
```

DIRECTIVES

指令

基于真值插入/移除元素:

```
<p v-if="inStock">{{ widget }}</p>
<p v-else-if="onOrder"> . . .</p>
<h1 v-else> . . .</h1>
```

通过元素的 displayCSS 属性来控制:

```
<h1 v-show="ok">Hello World!</h1>
```

双向数据绑定:

```
<input v-model="firstName" />
```

v-model.lazy=". . ." 更改后同步输入
v-model.number=". . ." 始终返回数值
v-model.trim=". . ." 去除空格

列表渲染

```
<li v-for="item in items" :key="item.id">
  {{ item }}
</li>
```

指定 key

通过位置访问数组:

```
<li v-for="(item, index) in items"> . . .
```

遍历对象:

```
<li v-for="(value, key) in object"> . . .
```

在组件上使用 v-for:

```
<my-item v-for="item in products"
:products="item" :key="item.id">
```

绑定

```
<a v-bind:href="url"> . . .</a>
```

缩写

→ <a :href="url">...

true 或 false 将添加或移除元素特性:

```
<button :disabled="isButtonDisabled">...
```

当 isActive 为真时, active 样式类会出现:

```
<div :class="{ active: isActive }"> . . .
```

样式 color 被设置为 activeColor:

```
<div :style="{ color: activeColor }">
```

Action/事件

单击后调用组件的 add 方法:

```
<button v-on:click="add"> . . .
```

缩写

→ <button @click="add"> . . .

可以传递参数:

```
<button @click="add(widget)"> . . .
```

防止页面被重载:

```
<form @submit.prevent="add"> . . .
```

只触发一次:

```
<img @mouseover.once="show"> . . .
```

.stop 停止所有事件的传播

.self 只有当 event.target 是元素自身时才触发

键盘回车的示例:

```
<input @keyup.enter="submit">
```

译者序

随着互联网的蓬勃发展，前端技术也百花齐放，出现了许多优秀的、应用广泛的前端框架。选择哪个前端框架，是入门者到专家所有层次开发人员都需要思考的问题。本书虽然并不对众多前端框架做比较，也不帮你做决策；但如果你想初探甚至深入学习 Vue.js，本书会提供你需要的所有帮助。本书不仅深入浅出地讲解 Vue.js 的相关概念和技术，还准备了一系列小而全的示例来帮助你融会贯通。

非常荣幸清华大学出版社能给我翻译本书的机会，李阳编辑是指引我进入翻译领域的老师。编辑老师们一如既往、不厌其烦地帮助并指导我们，他们为本书的翻译付出了很多心血，没有他们，本书不可能顺利付梓。感谢同事邓龚达欣然接受我的邀请，与我合作翻译本书，没有他，我无法在这么短时间内完成本书的翻译。另外感谢我的同事叶盛飞和周骅，他们校对了部分重要章节，并给予我许多宝贵的指导建议。

最后，希望你能通过阅读本书提升 Vue.js 技能，创建出更多丰富多彩的应用程序。

任 强

2019年3月31日

序 言

前端 Web 开发已经变得异常复杂。如果你从未使用过现代的 JavaScript 框架，构建你的第一个只显示“Hello”的应用程序可能需要一个星期！这听起来可能很荒谬，我也认为如此。问题是，大多数框架都假定你已对终端、JavaScript 高级技术以及诸如 Node 包管理器(Node Package Manager, NPM)、Babel、Webpack 等工具有一定的了解，甚至知道更多。

令人振奋的是，Vue 并没有这样的束缚。我们称之为“渐进式”JavaScript 框架，因为它既可以向下扩展，也可以向上扩展。如果你的应用程序很简单，你可以像使用 jQuery 那样使用 Vue：通过放入一个<script>标签。但是，随着技能的提升和需求的增多，Vue 也会随着你的成长而变得更加强大和高效。

Vue 不仅是由计算机科学家建立的，而且是由设计师、教育工作者和更多以人为中心的行业的其他人建立的。因此，我们的文档、指南和开发工具都是世界一流的。Vue 的使用经验与其性能、可靠性和多用途对我们同样重要。

Erik 把以人为本的精神注入本书。首先，本书非常直观。许多详细的插图和带注释的屏幕截图在真实开发人员的工作流程中牢固地奠定了示例的基础。因此，实际上你可以学习如何使用浏览器和 Vue 的开发工具来确认正在学习的内容，更重要的是，在出现问题时进行故障排除。

对于那些在前端开发、JavaScript 甚至编程方面没有很强背景的人，Erik 还仔细解释了理解 Vue 在做什么以及为什么这样做的基本概念。这与他以项目为中心引入新特性的方法相结合，意味着这本书对于初出茅庐的开发人员来说是理想的，他们希望通过 Vue 作为自己的第一个现代前端框架来扩展技能。

——Chris Fritz, Vue 核心团队成员和文档管理员

作者简介



Erik Hanchett 是一位拥有超过 10 年开发经验的 Web 开发人员。他是 *Ember.js Cookbook* (Packt Publishing, 2016) 的作者, 也是 YouTuber(<http://erik.video>) 和博主(<http://programwitherik.com>)。他运行了一个邮件列表, 通过 <https://goo.gl/UmemSS> 为 JavaScript 开发人员提供提示和技巧。在工作或编码之余, 他会和妻子 Susan 以及两个孩子 Wyatt 和 Vivian 共度时光。

致 谢

首先，同时也是最重要的，我要感谢我的妻子 Susan，因为没有她的帮助，这本书永远不会完成。我要感谢我的儿子 Wyatt 和女儿 Vivian。他们是我如此努力工作的动力源泉。我还要感谢所有的评论家、Vue.js in Action 论坛的成员，以及任何有助于对这本书给予反馈的人。你们提供的帮助使得这本书远比我一个人所能做到的要好得多。另外，Chris Fritz，感谢你写了一篇精彩的序言。最后，我衷心感谢 Vue.js 社区、Evan You 以及所有让 Vue.js 成为如此卓越框架的人。

—Erik Hanchett

首先，我要向我的妻子 Kiffen 表示最诚挚的感谢，感谢她对我的支持和鼓励，不仅感谢让我参与这项工作，还感谢她对我们生活各个方面的支持和鼓励。对我们的儿子 Leo，我们家族宇宙中心的明星，我要感谢你无限的微笑、拥抱和欢呼。感谢他们的鼓励、理解和支持。我衷心感谢 Manning 的编辑团队。关于 Erik，没有他这本书就不会有生命，我深表感谢；我祝福你一切顺利。最后，感谢 Evan You 和所有为 Vue.js 做出贡献的人，感谢他们汇集大量的软件并组成一个伟大的社区。我很荣幸能成为这个社区的一员。

—Benjamin Listwon

我们都应该感谢我们的技术校对员 Jay Kelkar，以及一直以来提供反馈的所有审稿人，包括 Alex Miller、Alexey Galiullin、Chris Coppenbarger、Clive Harber、Darko Bozhinovski、Ferit Topcu、Harro Lissenberg、Jan Pieter Herweijer、Jesper Petersen、Laura Steadman、Marko Letic、Paulo Nuin、Philippe Charriere、Rohit Sharma、Ronald Borman、Ryan Harvey、Ryan Huber、Sander Zegveld、Ubaldo Pescatore 和 Vittorio Marino。

前 言

2017 年年初，在 Benjamin Listwon 因个人原因退出之后，我有机会撰写这本书。我最近刚从雷诺内华达大学获得了工商管理硕士学位，自从我出版上一本书 *Ember.js Cookbook* (Pact Publishing, 2016) 以来已经整整一年了。我已经开始了我的 YouTube 频道，和 Erik 合作，我花了大部分时间试图找出如何更好地为我的小规模但人数不断增长的观众录制节目教程。大约在这段时间，我开始在 Vue.js 上制作一个电影系列，并从我的观众那里得到了积极反馈。这让我更想探索 Vue.js。

首先，我会倾听 Evan You (Vue.js 的创建者) 及其框架路线图。然后我看了无数的 YouTube 教程和其他创建者的视频。我访问了在线论坛和 Facebook 群组，看看人们在谈论什么。无论我到哪里，人们都对 Vue.js 框架的可能性感到兴奋。这使我想探索写这本书的可能性。

经过深思熟虑，并且与我妻子谈话后，我决定去做这件事。幸运的是，Benjamin 为我打下很好的基础，所以我可以在此基础上进行创作。在接下来的 10 个月里，我花了无数的夜晚和周末研究、测试和写作。

我希望我能告诉你写这本书多么容易，或者说我没有遇到任何问题。但事情并没有按计划进行。我遇到了个人挫折，错过了最后期限，遇到了写作障碍，如果这还不够的话，我最终不得不在 Vue.js 做了一次更新之后进行重大修改。

尽管如此，我还是为这本书感到骄傲。每次挫折，我都被激励加倍努力。我决心以最高质量完成这本书。

谢谢读者，非常感谢你购买了本书。我真的希望它能帮助你学习 Vue.js。如果真的帮到了你，请告诉我。你可以在 @ErikCH 上发推特，在 erik@programwitherik.com 上发邮件给我，或者在 <https://goo.gl/UmemSS> 上加入我的邮件列表！再次感谢！

关于本书

在你学习如何编写 Vue.js 应用程序之前，让我们先介绍一下你应该了解的一些事项。

在本书中，我们将介绍你所需了解的所有知识来让你熟悉 Vue.js。本书的目标是为你提供所需的知识，因此你可以毫不犹豫地跳到任何 Vue.js 应用程序中。

在为本书做研究时，我无数次听到官方的 Vue.js 指南是学习 Vue.js 的最佳资源。虽然官方指南很棒，但我强烈建议你在学习 Vue.js 时将它们作为附加参考，但它们并不涵盖所有内容，并且它们并不完美。本书涵盖的内容超越官方指南，并且书中的示例更容易理解、可关联，因此你可以更轻松地将这些概念应用到你自己的项目中。我认为某个主题超出了本书的讨论范围，或者说不够重要，因此我添加了一份参考资料，你可以在官方指南中了解更多相关内容。

本书可以有几种不同的阅读方式。你可以从前到后阅读。在这种情况下，你将获得 Vue.js 所提供技能的全部广度。或者你也可以使用本书作为参考手册来查找你需要掌握更多信息的那些概念。无论哪种方式都是可以接受的。

在本书的后面部分，我们将转换为使用构建系统创建 Vue.js 应用程序。别担心，附录 A 中包含如何使用 Vue.js 构建工具 Vue-CLI。Vue-CLI 最重要的好处之一是，它可以帮助我们创建更复杂的 Vue.js 应用程序，而不必担心构建或转换我们的代码。

贯穿本书，我们将创建一个 Vue.js 宠物商店应用程序。某些章节比其他章节更多地使用这个宠物商店示例应用程序。我刻意这么做，这样你可以很容易地掌握一个概念，而无须了解它如何与这个宠物商店示例应用程序共存。但是，那些喜欢用真实应用程序学习的人仍然有选择的权利。

读者对象

本书适合任何有兴趣学习 Vue.js 并具有 JavaScript、HTML 和 CSS 经验的人。我不希望你对此有太多了解，但了解基础知识，如数组、变量、循环和 HTML 标签，将会有所帮助。至于 CSS，我们将使用 Bootstrap 3，它是一个 CSS 库。但是，你不需要了解 Bootstrap 的任何内容以跟随示例。它只是用于帮助构建样式。

在本书的前面部分，我使用 ECMAScript 2015(也称为 ES6)介绍了示例代码。在开始阅读本书之前，最好先查看一下。在大多数情况下，我只使用一些 ES6 功能，例

如箭头函数和 ES6 导入。当我们进行这种转变时，我会在书中警告你。

学习路线图

本书内容分为三部分，每部分都建立在前一部分的基础之上。第 I 部分是了解 Vue.js 的关键。在第 1 章和第 2 章中，我们将创建第一个 Vue.js 应用程序。我们将看看 Vue.js 实例是什么及其与应用程序的关系。

在第 II 部分(第 3~9 章)，我们将深入了解 Vue.js 的几个最重要的部分。第 I 部分更多是 Vue.js 的开胃菜，而第 II 部分是主菜。你将学习如何创建 Vue.js 应用程序的复杂性。我们将首先学习反应模型，然后将创建一个宠物商店应用程序，我们将在本书的其余部分使用它。

我们将添加表单和输入框，还将介绍如何使用 Vue.js 强大的指令绑定信息，然后查看条件、循环和表单。

第 6 章和第 7 章非常重要。我们将学习如何使用组件将 Vue.js 应用程序分成几个逻辑部分，我们将先看一下创建 Vue.js 应用程序所需的构建工具。

第 7 章还将介绍路由。在前面的章节中，我们使用简单的条件来导航应用程序。通过添加路由，我们可以正确地在应用程序中移动并在路由之间传递信息。

第 8 章介绍使用 Vue.js 可以执行的强大动画和转场。这些功能都融入了语言，是你应该掌握的很好功能。

在第 9 章中，我们将学习如何使用 Mixin 和自定义指令轻松扩展 Vue 而不用重复自己。

第 III 部分是关于建模数据、使用 API 和测试的全部内容。在第 10 章和第 11 章中，我们首先深入探讨 Vue 的状态管理系统 Vuex。然后我们将看看如何开始与后端服务器进行通信，我们将了解有关服务器端呈现框架 Nuxt.js 的更多信息。

第 12 章专门用于测试。在任何专业环境中，你都需要了解测试，我们将介绍你必须了解的基本要素。

书籍论坛

购买本书后就可以免费访问由 Manning Publications 运营的私人网络论坛，可以在该论坛上对本书发表评论，提出技术问题，并从作者和其他用户那里获得帮助。要访问该论坛，可访问 <https://forums.manning.com/forums/vue-js-in-action>。还可以访问 <https://forums.manning.com/forums/about>，了解有关 Manning 论坛和行为规则的更多信息。

源代码

本书包含许多源代码示例，包括编号列表和内联普通文本。在这两种情况下，源代码都以这样的固定等宽字体格式化，以与普通文本分开。有时使用粗体突出显示已从本章前面的步骤中更改的代码，例如当新功能被添加到现有代码行时。

在许多情况下，源代码已经重新格式化；我们添加了换行符并重写缩进以适应书中可用的页面空间。在极少数情况下，如果这还不够，列表将包括行继续标记(►)。此外，当使用文本描述代码时，源代码中的注释通常已从代码清单中删除。代码注释伴随着许多代码清单，从而突出了重要的概念。

本书的源代码可从出版商的网站(www.manning.com/books/vue-js-in-action)和我的个人 GitHub 存储库(<https://github.com/ErikCH/VuejsInActionCode>)下载，也可通过扫描封底的二维码下载。你还可以在附录 A 中找到有关下载代码和设置编程环境的更多说明。

在阅读本书时，你会注意到我经常将源代码拆分为单独的文件。我已将完成的文件和每章中的分隔文件都包含在源代码中，因此你可以按照这些步骤进行操作。如果你在代码中发现错误，请随意通过 pull request 发送到我的 GitHub 存储库。我会维护该存储库，并在自述(README)文件中发表任何更新的评论。

软件要求

为简单起见，本书中的所有代码都适用于任何现代浏览器。我已经在 Firefox 58、Chrome 65 和 Microsoft Edge 15 上亲自测试过。我不建议尝试在旧浏览器上运行我的任何应用程序，因为你肯定会遇到问题。Vue.js 本身不支持 IE 8 及以下版本。你必须具有符合 ECMAScript 5 标准的浏览器。

在本书的前几章中，我使用了一些 ES6 特性。你需要使用现代 Web 浏览器来运行这些示例。

我们将在本书中创建的宠物商店应用程序将在移动浏览器上运行。然而，宠物商店应用程序并未针对移动设备进行优化，因此我建议你在台式计算机上运行这些示例。

你不必担心操作系统。如果 Web 浏览器运行了，就应该没问题。实际上没有其他要求。

在线资源

正如我之前提到的，当你使用本书中的示例时，Vue.js 官方指南非常适合用作参考资料。你可以在 <https://vuejs.org/v2/guide/> 上找到指南。它们在不断更新。

在 GitHub 页面 <https://github.com/vuejs/awesome-vue> 上有一个与 Vue.js 相关的精彩内容的首选列表。在这里，你可以找到 Vue.js 播客(Podcast)、其他 Vue.js 资源、第三方库，甚至是使用 Vue.js 的公司的链接。强烈建议你查看一下。

Vue.js 社区规模庞大且不断发展。与其他 Vue.js 开发人员交流的最佳地点之一是 <https://forum.vuejs.org/> 上的官方 Vue.js 论坛。在这里可以讨论或获得任何关于 Vue 的帮助。

如果你正在寻找更多视频教程，我的频道——YouTube 上的 <http://erik.video>——涵盖了大量有关 Vue.js 和 JavaScript 的信息。

更多信息

这本书中覆盖了大量的材料。如果你遇到了困难，或者你需要帮助，请不要犹豫，与我联系。如果我无法帮助你，至少我会指出正确的方向。别害羞。你会发现 Vue.js 社区中的人们对初学者是很友好的。

此外，在阅读本书时，请尝试采用你学到的几个概念并自行实施。最好的学习方法之一就是实践。例如，尝试创建自己的电子商务网站，而不是参照宠物商店应用程序。使用本书作为指导，以确保你不会陷入困境。

最后一件事：学得开心。要有创意，做一些很炫酷的事情。如果你这样做了，一定要在 Twitter 上通过 @ErikCH 联络我！

目 录

第 I 部分 初识 Vue.js

第 1 章 Vue.js 介绍.....3

1.1 站在巨人的肩膀上.....4

1.1.1 MVC 模式.....4

1.1.2 MVVM 模式.....6

1.1.3 什么是反应式应用程序..7

1.1.4 JavaScript 计算器.....7

1.1.5 Vue 计算器.....10

1.1.6 JavaScript 和 Vue 的 差别.....11

1.1.7 Vue 如何促进 MVVM 和响应性.....12

1.2 使用 Vue.js 的理由.....12

1.3 展望未来.....14

1.4 本章小结.....14

第 2 章 Vue 实例.....15

2.1 我们的第一个应用程序.....16

2.1.1 Vue 根实例.....16

2.1.2 确保应用程序可以 运行.....18

2.1.3 在视图中显示内容.....20

2.1.4 检查 Vue 中的属性.....22

2.2 Vue 生命周期.....23

2.2.1 添加生命周期钩子.....24

2.2.2 探索生命周期代码.....25

2.2.3 是否保留生命周期 代码.....27

2.3 显示商品.....27

2.3.1 定义商品数据.....27

2.3.2 添加商品视图标签.....28

2.4 运用输出过滤器.....31

2.4.1 编写过滤器函数.....31

2.4.2 将过滤器添加到我们的 标签并测试不同的值...32

2.5 练习题.....33

2.6 本章小结.....34

第 II 部分 视图与视图模型

第 3 章 增加交互性.....37

3.1 购物车数据, 从添加一个 数组开始.....38

3.2 绑定到 DOM 事件.....39

3.2.1 事件绑定基础.....39

3.2.2 将事件绑定到 Add to cart 按钮.....39

3.3 添加购物车件数按钮并 计数.....42

3.3.1 何时使用计算属性.....42

3.3.2 使用计算属性检查更新 事件.....43

3.3.3 显示购物车商品计数 及测试.....47

3.4 让我们的按钮具备用户直 观功能.....49

3.4.1	密切关注库存	49	5.1.1	用 v-if 添加剩余的商品数量	80
3.4.2	使用计算属性和库存 ..	50	5.1.2	使用 v-else 和 v-else-if 添加更多消息	82
3.4.3	指令 v-show 的基础知识	51	5.2	循环商品	83
3.4.4	使用 v-if 和 v-else 显示被禁用的按钮	52	5.2.1	使用 v-for 范围循环增加星级评分	84
3.4.5	添加 Adding the cart 按钮用于切换	54	5.2.2	将 HTML 类绑定到星级评分	85
3.4.6	使用 v-if 显示结账页面	55	5.2.3	设置商品	87
3.4.7	对比 v-show 与 v-if/v-else	56	5.2.4	从 product.json 文件导入商品	89
3.5	练习题	57	5.2.5	使用 v-for 指令重构应用程序	91
3.6	本章小结	57	5.3	排序记录	95
第 4 章	表单与输入框	59	5.4	练习题	97
4.1	使用 v-model 绑定	60	5.5	本章小结	97
4.2	关于值绑定	68	第 6 章	使用组件	99
4.2.1	绑定值到复选框	68	6.1	什么是组件	100
4.2.2	使用值绑定和单选按钮	69	6.1.1	创建组件	100
4.2.3	学习 v-for 指令	71	6.1.2	全局注册	101
4.2.4	没有可选 key 的 v-for 指令	73	6.1.3	局部注册	102
4.3	通过应用程序学习修饰符	74	6.2	组件之间的关系	103
4.3.1	使用 .number 修饰符 ..	75	6.3	使用 props 传递数据	104
4.3.2	修剪输入值	76	6.3.1	字面量 props	104
4.3.3	v-model 的 .lazy 修饰符	78	6.3.2	动态 props	105
4.4	练习题	78	6.3.3	props 验证	108
4.5	本章小结	78	6.4	定义模板组件	111
第 5 章	条件语句、循环和列表	79	6.4.1	使用内联模板字符串	111
5.1	显示可用的库存信息	80	6.4.2	text/x-template 脚本元素	112
			6.4.3	使用单文件组件	113

6.5 使用自定义事件.....	114	7.8 练习题.....	153
6.5.1 监听事件.....	114	7.9 本章小结.....	153
6.5.2 使用.sync 修改子 属性.....	116	第 8 章 转场和动画.....	155
6.6 练习题.....	117	8.1 转场基础.....	155
6.7 本章小结.....	117	8.2 动画基础.....	160
第 7 章 高级组件和路由.....	119	8.3 JavaScript 钩子.....	161
7.1 使用插槽.....	120	8.4 组件的转场.....	164
7.2 具名插槽.....	123	8.5 更新宠物商店应用 程序.....	167
7.3 作用域插槽.....	125	8.5.1 在宠物商店应用程序 中添加转场.....	167
7.4 创建动态组件应用程序.....	127	8.5.2 在宠物商店应用程序 中加入动画.....	168
7.5 设置异步组件.....	129	8.6 练习题.....	171
7.6 使用 Vue-CLI 转换宠物 商店应用程序.....	131	8.7 本章小结.....	171
7.6.1 使用 Vue-CLI 新建 应用程序.....	132	第 9 章 扩展 Vue.....	173
7.6.2 设置路由.....	134	9.1 用 Mixin 实现功 能复用.....	174
7.6.3 将 CSS、Bootstrap 和 axios 添加到应用程 序中.....	135	9.2 通过示例学习自定义 指令.....	179
7.6.4 设置组件.....	137	9.3 render 函数和 JSX.....	184
7.6.5 创建 Form 组件.....	139	9.3.1 render 函数示例.....	185
7.6.6 添加 Main 组件.....	140	9.3.2 JSX 示例.....	188
7.7 路由.....	143	9.4 练习题.....	192
7.7.1 添加带参数的商品 路由.....	143	9.5 本章小结.....	192
7.7.2 设置带标签的 router-link.....	146	第 III 部分 数据建模、API 调用和测试	
7.7.3 设置带样式的 router-link.....	148	第 10 章 Vuex.....	195
7.7.4 添加子编辑路由.....	149	10.1 Vuex 的优势.....	196
7.7.5 使用重定向和通 配符.....	151	10.2 Vuex 状态与 mutation.....	197
		10.3 getter 和 action.....	201