

Web开发经典丛书

hapi.js之父Eran Hammer作序推荐本书!

HAPI.JS IN ACTION

hapi.js

实战



著
译

博

MANNING



清华大学出版社

Web开发经典丛书

hapi.js实战

[美] Matt Harrison 著

梁 宵 郭美青 翟懿博 译



清华大学出版社

北 京

EISBN:978-1633430211

hapi.js in Action

Matt Harrison

Original English language edition published by Manning Publications, 178 South Hill Drive, Westampton, NJ 08060 USA. Copyright © 2017 by Manning Publications. Simplified Chinese-language edition copyright © 2017 by Tsinghua University Press. All rights reserved.

北京市版权局著作权合同登记号 图字: 01-2017-4219

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

hapi.js实战 / (美) 马特·哈里森(Matt Harrison) 著; 梁宵, 郭美青, 翟懿博 译. —北京: 清华大学出版社, 2017

(Web开发经典丛书)

书名原文: hapi.js in Action

ISBN 978-7-302-47977-2

I. ①h… II. ①马… ②梁… ③郭… ④翟… III. ①JAVA语言—程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2017)第 209163 号

责任编辑: 王 军 韩宏志

封面设计: 牛艳敏

版式设计: 方加青

责任校对: 曹 阳

责任印制: 王静怡

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: 三河市君旺印务有限公司

经 销: 全国新华书店

开 本: 185mm×260mm

印 张: 22 字 数: 521 千字

版 次: 2017 年 9 月第 1 版

印 次: 2017 年 9 月第 1 次印刷

印 数: 1 ~ 3000

定 价: 68.00 元

产品编号: 075002-01

译者序

当 Ryan Dahl 在 GitHub 上发布第一个 Node 版本的时候，也许未想到这将给 Web 开发（尤其是 JavaScript 语言）带来什么。冲出了浏览器沙箱的牢笼，JavaScript 在后端世界大放异彩。全球开发者都张开双臂迎接 Node。Node 的生态系统呈现爆发式扩张，各类工具、框架层出不穷。这使得不借助第三方 Web 服务器和应用服务器构建 Web 应用成为可能，其间涌现了如 Express、Meteor 和 Koa 等优秀的 Node 开发框架。

和众多老牌的 Node 框架相比，hapi 是后起之秀，然而它成功地构建了自己的生态圈。PayPal、Yahoo!、Mozilla 和 Disney 等公司都使用 hapi 构建了自己的 Web 应用，其影响力可见一斑。模块化、以配置为中心和插件化是 hapi 最大的亮点。它的核心仅由少数几个模块构成，其他功能均通过插件扩展来实现，在丰富的功能和无限制的可定制性之间取得了完美平衡。更令人兴奋的是，hapi 经历了沃尔玛在黑色星期五期间史上最大规模的 Node 部署，可以说已经在生产环境中经历了千锤百炼。

本书可能是迄今为止唯一一本介绍 hapi 的入门书籍，由浅入深地全面介绍 hapi 的设计思想与核心功能，并呈现丰富的案例。希望你能从中受益，并享受使用 hapi 带来的快乐，这也是框架名称的由来。

本书由搜狗营销事业部的梁宵、郭美青和翟懿博协作翻译，并由刘建负责最终审校和润色。翻译过程也是学习和总结的过程，我们深感 Node 生态圈的伟大，并为 hapi.js 的设计思想所折服。

感谢我们的家人和伙伴，你们无时无刻不在鼓励着我们。感谢清华大学出版社的李阳老师，没有你的信任和悉心指导，我们不可能在短时间内完成本书的翻译。感谢搜狗营销事业部的高清霞、李剑、李晓清对本书翻译工作的大力支持。

在翻译过程中我们努力做到信达雅，并尽力修正了一些原文的小错误，但由于水平有

限，难免存在疏漏之处，敬请广大读者指正。

谨以此译作献给广大 Node 开发者，相信 Node 的明天会更加美好！

译者

序

任何开发框架的核心思想都是共享模式与代码，我们汲取集体的智慧并依赖彼此的成功。框架是由志同道合的开发者们共享的通用基础设施，用于解决相似的问题。它们也是团队协作和高效沟通的核心组成部分。

大约在 Node.js 概念兴起的同时，从一个小工具集开始，逐渐演化出 hapi.js 框架。从大家的经验中，我们学到了更多东西，构建了更大规模的产品系统，扩充了工程师团队，增加了协作复杂性，这个框架不断进化来反映这些问题，并在内部把它们消化了。

hapi.js 社区的文档是非常好的参考资料，但是它不会（也不能）包含大量的知识点和经验，这些需要从多年的实践中获得。这便是《hapi.js 实战》的切入点——提供大量知识点，这是迄今为止提供给那些少数吃螃蟹的人为数不多的参考资料。

作为 hapi 的核心项目维护者，Matt Harrison 促使内部人员的专业技能和社区领导力有机结合。核心维护者的大部分工作包括与新加入的和有经验的开发者互动，解答疑问和调研问题。正是这些覆盖了成百上千个问题和疑问的经验和工作使得本书弥足珍贵。

进入 hapi.js 之门并非难事。你可以只用几个方法在几小时之内搭建起一个可运行的 Web 应用。然而，hapi.js 是一个基于最佳实践不断演进的框架，使用自有术语。从服务器、连接、插件和路由，到领域、扩展和方案，这个框架能从很多方面使你的生活变得更简单，使你的产品变得更出色。但如果你不知道它们的存在，它们也无能为力了。

不管对于哪门技术，从新手成长为专家都需要时间。Matt 用本书将大量非文档化的零散知识组织成一篇容易学习的记叙文，新手和专家都会从中受益。

当开始（或继续）体验 hapi.js 框架时，你将加入一个忙碌而庞大的应用开发者社区，他们共同致力于开发一流的工具和模式。这些工具已经支持一系列引人瞩目的产品和服务。从支付服务、在线零售店到云和音乐服务，如今你很可能已经在使用至少一个基于 hapi.js 的应用了。我们或许会在未来使用到你创建的应用。

有本书在手，你已经向成为 hapi.js 的专家迈出了第一步。无论你利用其中的经验去构建创新型产品，创建自己的开源组件去分享给他人，还是作为一名积极的贡献者加入 hapi.js 社区，我都希望你会从框架的使用中找到乐趣并获得力量。它是由数十名开源开发者用爱与奉献精神打造的，期望每天都能让应用开发变得更加美好。

它叫 hapi 是有原因的。

Eran Hammer

Hapi.js 的创建者

自序

在我 16 岁的时候，偷偷地把小伙伴的口头禅录制到闪存里，乐此不疲。我在 Photoshop 中把花哨的图案切成小块，并在 Dreamweaver 中用 HTML 表格整合网站。

我花了很长的时间远离 Web，训练自己成为一名建筑师。但我总是对构建网站念念不忘，用自己的代码创造新奇的玩意儿，让它在屏幕上呈现并运行。于是在毕业的时候，我做了一个 180 度的转变——我决心回归 Web。

但情况已经发生了变化！我们有了 CSS、Web 标准和 JavaScript，而 PHP 是当时炙手可热的服务器技术。我补习了所能学到的一切，包括 WordPress、Joomla、Zend、CodeIgniter 以及当时所有的大型框架和内容管理系统。我做了一些相当像样的网站，并找到了一份初级开发者的工作。事实上，我不知道自己真正在做什么。我对正在发生的事情的本质所知甚少。这些框架做了很多不可思议的事情……而且它们太大了，我觉得完全迷失了。

当我发现 Node.js 时，情况开始改变了。我从编写一些小脚本开始，比如启动一个小型 HTTP 服务器，为浏览器提供访问硬盘文件的服务。没有花哨的东西，没有重量级框架，只有几行简单的 JavaScript 代码，和我在浏览器中了解的一样。我知道每行代码在做什么。代码和内部运作之间没有距离，这让我更加满意。

此后，我逐渐理解了每一件事情。我过去使用的那些所有重量级框架只不过是模糊 Web 开发简单美丽的初心。我发誓如果不了解框架的运行机制，就永远不会在工作中使用它。

与 Node 世界中的其他人一样，我很自然地开始使用 Express 开发 Web 应用。那时 Express 对于开发小型项目很好用。然而随着项目规模越来越大，越来越正规，我不断地遇到问题。比如在面临扩充开发者队伍、多团队协作和开发大型应用时，使用 Express 是件痛苦的事情。拼凑几个自己的依赖就能使用 Node 开发正规应用，我开始怀疑这种想法

是不是太天真了。也许像 Zend 和 ASP.NET 这样的大型重量级框架才适于做那样的事情，它们带有大量的类和抽象，那才是复杂项目所需要的吗？

大约在那个时候我找到了 hapi。这些在沃尔玛的家伙们竟敢在黑色星期五用一个 Node.js 应用去支撑世界最大零售商的移动流量。他们是一个庞大的团队，拥有大量用户，并且他们的系统很复杂。看来他们在使用 Express 时遇到了和我一样的挫折，并开发了自己的框架。最令人振奋的是，他们把它开源了。也许可以使用 Node 开发企业级应用了。我必须试一试！

切换到 hapi 对我来说是快速而无痛的。我喜欢简单的配置驱动的 API 和强大的插件系统。我的代码比以前整洁多了。还有，它即快速又安全。我决定一直使用 hapi 了。我开始在它的 GitHub 项目上贡献代码，改善文档，并撰写博客为 hapi 布道。

我发现社区成员相当热心，乐于助人，响应问题和修复 bug 都很快。这个项目一直处于一个非常健康的状态。只发布测试覆盖率为 100% 的代码，这条原则显示了他们对质量的承诺。

当 Manning 出版社联系我撰写一本关于 hapi 的书籍时，我对前景有一些担忧，但是我知道我会答应的。hapi 需要一本书，而我责无旁贷。很荣幸能与你分享我从 hapi 中学到的一切。在经历了差不多 18 个月的写作后，我比以前更了解这个框架了，而我的观点一如既往。我相信它在丰富的功能和无限的可定制性之间取得了完美平衡，但依然能够简单地启动和运行，而且不妨碍你去开发复杂的应用。

致 谢

说来惭愧，只有我的名字出现在本书的封面上。事实上，这是很多人士共同工作和努力的结晶。我一个人无法完全胜任这项任务。每位作者在致谢中都会这样说，这并不是出于谦卑，这是事实。

首先，我要感谢开发编辑 Susanna Kline，我与她在 Skype 上交流了很久。我们无所不谈，从天气到航班，再到牛仔(她来自德州)，偶尔谈谈本书。Susanna 有无限的能力，没有她的好建议和见解，这会是一本没那么厚的完全不同的书。同时感谢 Cynthia Kane 在第 1 章中作为编辑对我的帮助。

非常感谢 Michael Stephens，他是我在 Manning 出版社的第一位联系人。感谢你 Mike，在我们第一次聊天时对我的鼓励，并相信我能胜任这个项目。起初你给我的信心支撑着我走到了最后。

感谢 Corbin Collins，他的建议非常宝贵，保证了本书具有很高的质量和准确性。感谢 Nickie Buckner 的火眼金睛和对原稿的技术审查，他在文本中挑出了非常细小的错误。同样非常感谢 Elizabeth Martin，他在出版的最后阶段收集了很多问题。下面的审稿人慷慨地付出了他们的时间并改进了本书：Davide Fiorentino lo Regio、Earl Bingham、Gavin Whyte、Gonzalo Huerta-Canepa、Jeff Smith、Jeroen Benckhuijsen、Jerry Tan、Jonathan Whittington、Margriet 和 Nikander Bruggeman、Matt Hernandez、Nick McGinness、Philippe Charrière、Ryan Pulling、Stephen Byrne 以及 Thomas Peklak。

同时感谢所有 hapi 贡献者在最开始阶段创建了 hapi，使用它工作和编程是一件非常愉悦的事情。没有你们卓越的工作，就不会有本书的问世。特别感谢 Eran Hammer 创建了 hapi 项目，以及合作阅读本书的原稿，并为我们撰写了精彩的序言。

感谢我的女友 Yi Ching，一直支持我并相信我的能力，即使我没那么自信。

感谢我的父母 Gail 和 David，在我寻找自我的道路上赐予我关爱、鼓励和资源，即使在看上去无路可走的时候。

前言

《hapi.js 实战》是为完全的新手准备的入门书籍。开篇以简单实用的示例使你逐步适应，然后在理论知识的基础上，深度覆盖 hapi 中所有的重要功能、各种各样的插件和庞大生态系统中的模块。阅读本书后，你将可以得心应手地构建可维护的、快速的、安全的生产环境 Web 应用。

本书面向的读者

任何对构建网站、API、单页面服务器或用 JavaScript 编写的联网 HTTP 服务感兴趣的人应该阅读本书。无论你是否在职业生涯中使用过 Node.js，还是刚刚开始，本书都将给你呈现一个 360 度视角的 hapi 世界。

即使你熟悉 hapi，毫无疑问在本书中还会找到新的有用资料。有经验的 hapi 开发者可以参考本书来了解如何使用更深奥的功能，这些例子将对你有所帮助。

本书的组织结构

本书被分为 3 个部分，共 11 章。

第 I 部分“入门”不会磨磨蹭蹭。它会很快带你编写代码。

- 第 1 章“hapi 简介”便是如此。你会了解到 hapi 对什么有益，以及一些用于创建应用的关键组成部分。
- 第 2 章“构建 API”收集一些需求，并帮助你构建一个功能完整的 hapi 应用。
- 第 3 章“构建网站”研究一些更加面向前端的 hapi 功能，比如静态内容服务。

第 II 部分“扩展工具箱”让你在第 I 部分的基础知识之上，对关键组成部分有更深入的理解。我还混合了一些内容，教你一些额外的超能力，比如验证和缓存。

- 第 4 章“深入理解路由和 handler”深入探讨 hapi 应用中两个通用的核心组件。
- 第 5 章“理解请求和响应”研究一个请求的生命周期以及如何修改它。

- 第 6 章“使用 Joi 验证”教你如何使用具有表现力的强大 Joi 库来锁定 API，对抗恶意的数据输入。
- 第 7 章“使用插件构建模块化应用”展示如何扩展 hapi 以及如何把应用拆分成小的可维护的插件包。
- 第 8 章“充分利用缓存”教你如何利用浏览器和服务端端的缓存来增加应用负载。

第 III 部分“创建健壮的应用”讲述如何确保你的应用是安全的、经过严格测试的、远离 bug 的。

- 第 9 章“身份验证和安全”研究了验证用户身份的多种方式和一些通用的安全漏洞防范技术。
- 第 10 章“使用 Lab、Code 和 server.inject() 进行测试”教你书写简单强大的测试去探查应用的每个角落。
- 第 11 章“投入生产环境及更多相关内容”帮助将你的应用投入生产环境，并提供一些出错时的建议和技术。

有两个附录。附录 A 提供了补充信息，包括下载安装 Node 和 npm。附录 B 介绍了版本号，包含了对本书用到的包的说明。

关于代码

本书包含了很多源代码示例，有带编号的代码清单，也有行内的普通文本。在两种例子中，源代码用等宽字体格式化以区别于普通文字。有时代码为粗体，用来高亮显示特别重要或与周边讨论有关的代码。

多数情况下，源代码被重新格式化了；我们加入了折行和修正的缩进以适应书中可用的页面空间。在极少数情况下，这样还不够，代码清单包含了续行符 `\`。此外，当代码以文本描述时，源代码中的注释通常会从代码清单中移除。

所有例子的源代码和本书中的代码清单都可以在 GitHub(<https://github.com/mtharrison/hapi.js-in-action>) 上找到。代码被层次化地组织起来，以匹配书中的章节标题。这样设计是为了在你学习任何章节时都能方便地查找代码。

另外，可以访问 www.tupwk.com.cn/downpage，输入本书的书名或中文 ISBN 下载代码；也可直接扫描封底的二维码下载。

本书和 GitHub 上的代码样例仅工作在 Node.js 4.0.0 以上版本，因为用到了一些 ES2015 的属性，比如 `let`、`const` 和箭头函数。

作者在线

购买《hapi.js 实战》时，能够免费访问由 Manning 出版社运营的私有论坛，在这里你可以对本书进行评论，提出技术问题，并获得作者和其他用户的帮助。要访问和订阅这个论坛，请去 <https://www.manning.com/books/hapi-js-in-action>。这个页面提供了注册后如何进入论坛的信息，指出什么样的帮助有用，介绍论坛的行为规范。

Manning 出版社对读者的承诺是提供一个地点，在这里各位读者之间，读者与作者之间可以碰撞出思想的火花。我们不承诺作者现身的次数，他们对该书论坛的贡献是自愿的（并且是无酬劳的）。我们建议你尽量问他一些具有挑战性的问题，激发他的兴趣。

只要本书还在销售中，作者在线论坛和前面讨论的存档就会在出版社的网站上开放。

其他在线资源

- 如果发现任何关于 hapi 使用方法或整体项目的问题，可在讨论区仓库 <https://github.com/hapijs/discuss> 提交 issues。
- 在 Stack Overflow 上也有 hapi.js 的标签 (<http://stackoverflow.com/questions/tagged/hapijs>)。

作者简介

Matt Harrison 是一位自由职业的 Web 开发者和顾问。他是 hapi.js 的核心贡献者，高产的博客作者，也是一位活跃的 Node.js 社区成员。此前，他曾是一名建筑师。他喜欢吃拉面，喝吉尼斯黑啤。

关于封面插图

《hapi.js 实战》的封面人物是 Sauvage de la Nouvelle Zeelande(一位新西兰原住民)。这幅插画取自一本名为 *Costumes de Différents Pays* 的多国服饰画集，由 Jacques Grasset de Saint-Sauveur(1757–1810) 创作，1797 年在法国出版。每幅插画都经过手工精绘并着色。

Grasset de Saint-Sauveur 丰富生动的画集让我们穿越时空，看到 200 年前文化色彩浓郁的世界各地的城镇和区域。那时的人们彼此隔离，说着不同的方言和语言。在街道或乡村，仅从他们的服饰，就能轻易地分辨出他们住在哪里以及他们在生活中的职业或身份。

自那时起，服饰习俗已经改变了，当时如此丰富多彩的地区多样性已经逐渐消失。如今通常很难区分来自不同大陆的居民。也许，尝试从乐观的角度来看，文化和视觉上的多姿多彩已经蜕变成更加多样化的个人生活——或者说更丰富以及有趣的现代科技生活。曾几何时，很难区分出计算机图书之间的差别，Manning 出版社通过封面推动了计算机出版行业的创新，这些封面来自两个世纪前特色鲜明的地区性生活，它们通过 Grasset de Saint-Sauveur 的画作复活了。

目 录

第 I 部分 入门

第 1 章 hapi 简介	3
1.1 hapi 是什么	4
1.1.1 hapi 的特色	6
1.1.2 hapi 是哪类框架	8
1.2 hapi 的组成部分	11
1.2.1 服务器	13
1.2.2 连接	13
1.2.3 路由	13
1.2.4 handler	13
1.2.5 插件	13
1.3 何时应该 (不该) 使用 hapi	14
1.3.1 何时应该使用 hapi	14
1.3.2 何时不应该使用 hapi	15
1.4 hapi 的运作方式	15
1.4.1 安装 hapi	15
1.4.2 创建服务器	16
1.4.3 添加路由	16
1.4.4 注册插件	17
1.4.5 运行 hapi	18
1.5 获得帮助	18
1.5.1 hapi.js 网站	19
1.5.2 Make Me hapi	19
1.5.3 GitHub	19
1.5.4 IRC	19
1.5.5 Stack Overflow	20

1.5.6 阅读代码	20
1.6 小结	20

第 2 章 构建 API

2.1 设计 API	21
2.1.1 你应该接受这个任务	21
2.1.2 收集需求	22
2.1.3 设计 API 接口	22
2.2 准备工作	23
2.2.1 工作目录	23
2.2.2 准备数据库和样本数据	23
2.2.3 sqlite3 node 模块	24
2.3 获取和搜索食谱	25
2.3.1 server.route() 介绍	25
2.3.2 路由 handler	26
2.3.3 接口 A: 获取所有食谱	28
2.3.4 接口 A: 搜索食谱	30
2.3.5 接口 B: 获取单一食谱	31
2.4 编写可维护的代码	32
2.4.1 模块化路由	32
2.4.2 用好 server.bind(): 设置 handler 中的上下文	33
2.4.3 模块化 handler	35
2.5 身份验证	37
2.5.1 模式和策略	37
2.5.2 实现不记名 token 身份验证	38
2.5.3 使用用户凭据	40

2.6	食谱创建和标星	40
2.6.1	测试接口	40
2.6.2	接口 C: 创建食谱	41
2.7	小结	44

第 3 章 构建网站

3.1	DinDin 网站	45
3.1.1	网站的样子	45
3.1.2	网站是如何运作的	47
3.1.3	设置	47
3.2	网页和静态内容服务	49
3.2.1	静态文件服务	49
3.2.2	整个目录服务	51
3.2.3	server.views(): 使用 Handlebars 动态渲染视图	53
3.2.4	DRY 视图: 布局和片段	57
3.3	使用外部 API	60
3.3.1	使用 Wreck: 调用 API	60
3.3.2	动态主页	62
3.3.3	食谱详情页	62
3.3.4	视图 helper	65
3.4	管理登录和用户会话	67
3.4.1	hapi-auth-cookie 插件	67
3.4.2	表单	69
3.4.3	实现登录	71
3.4.4	创建食谱	75
3.4.5	实现注销	78
3.5	小结	79

第 II 部分 扩展工具箱

第 4 章 深入理解路由和 handler

4.1	深入理解路由	83
4.1.1	hapi 的路由: 路由的排序和 冲突处理	83
4.1.2	路由方法	84

4.1.3	参数化路径	85
4.1.4	hapi 如何选取路由	88
4.2	构建自定义 handler	90
4.2.1	国际化例子	91
4.2.2	解析 Accept-Language header	92
4.2.3	第一个实现	93
4.2.4	再次简化	94
4.3	服务器方法	96
4.4	路由先决条件	99
4.4.1	异步 JavaScript 的并发问题	99
4.4.2	指定路由先决条件	101
4.4.3	使用带有先决条件的服务器 方法	102
4.4.4	多重串行先决条件	103
4.4.5	并发先决条件: 并行地运行 任务	105
4.5	管理文件上传	107
4.5.1	使用数据输出: 把文件内容 读入内存	108
4.5.2	使用流输出: 以流的方式获取 文件	109
4.5.3	使用文件输出: 把文件存储到 磁盘	110
4.5.4	额外的 payload 设置	111
4.4	小结	111

第 5 章 理解请求和响应

5.1	request 对象和生命周期	113
5.1.1	什么是 request 对象	113
5.1.2	请求的生命周期	115
5.1.3	扩展点	118
5.1.4	应该使用哪个扩展点?	121
5.2	reply 接口和 response 对象	121
5.2.1	什么是 reply 接口?	121
5.2.2	reply() 的有效参数	123

5.2.3 response 对象	124	6.4.5 在表单中渲染错误	165
5.2.4 使用流来响应	126	6.4.6 表单提交成功后的重定向	167
5.3 处理错误	128	6.5 小结	168
5.3.1 程序员错误和操作错误	129	第 7 章 使用插件构建模块化应用	169
5.3.2 HTTP 状态码	129	7.1 插件思想	169
5.3.3 介绍 Boom: 创建 HTTP		7.1.1 插件的定义	171
友好的错误	131	7.1.2 插件的作用	172
5.3.4 网站友好的 HTML 错误		7.1.3 把所有东西放进插件	174
页面	132	7.1.4 Pingoo 应用	174
5.4 小结	136	7.2 创建和加载插件	176
第 6 章 使用 Joi 验证	139	7.2.1 创建插件	176
6.1 介绍 Joi	140	7.2.2 使用 server.register() 加载	
6.1.1 Joi 的工作方式	140	插件	179
6.1.2 一个简单例子: 验证标量		7.2.3 插件依赖	180
类型	141	7.2.4 使用选项配置插件	182
6.1.3 一个更复杂的例子: 验证一个		7.3 使用 Glue 组合插件	186
复合类型	142	7.3.1 什么是 Glue?	186
6.2 掌握 Joi	144	7.3.2 创建一个清单	187
6.2.1 了解 API	145	7.3.3 使用 Confidence 工具实现智	
6.2.2 Joi.assert() 和 Joi.validate()	146	能配置	190
6.2.3 Joi 中的类型转换	146	7.4 插件通信	193
6.2.4 abortEarly 选项	147	7.4.1 全局的服务器配置	193
6.2.5 探索 Joi 错误	148	7.4.2 通过 server.expose() 在插件中	
6.3 hapi 中的验证	150	对外公开属性	195
6.3.1 使用 Joi 进行输入验证	150	7.4.3 使用事件系统	196
6.3.2 验证 payload	152	7.5 小结	200
6.3.3 验证响应	155	第 8 章 充分利用缓存	201
6.3.4 使用 failAction 自定义验证		8.1 客户端缓存	202
响应	156	8.1.1 手动设置 header	203
6.4 整合: 使用 hapi 和 Joi 进行 Web		8.1.2 在配置中设置缓存策略	203
表单验证	157	8.1.3 重新验证和 ETag	204
6.4.1 如何工作	158	8.2 介绍 Catbox: 一个多策略的对象	
6.4.2 创建骨架	159	缓存库	207
6.4.3 创建路由和视图	160	8.2.1 什么是 Catbox	208
6.4.4 添加验证	163		