

# Ember.js

实战

## Ember.js IN ACTION

[挪] Joachim Haagen Skeie 著  
卢俊祥 译



# Ember.js 实战

Ember.js  
IN ACTION



[挪] Joachim Haagen Skeie 著  
卢俊祥 译

人民邮电出版社  
北京

## 图书在版编目 (C I P) 数据

Ember.js 实战 / (挪) 斯基 (Skeie, J.H.) 著 ; 卢俊祥译. — 北京 : 人民邮电出版社, 2015.7

书名原文: Ember.js in Action

ISBN 978-7-115-39001-1

I. ①E… II. ①斯… ②卢… III. ①JAVA语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2015)第094405号

## 版权声明

Original English language edition, entitled *Ember.js in Action* by Joachim Haagen Skeie published by Manning Publications Co., 209 Bruce Park Avenue, Greenwich, CT 06830. Copyright ©2014 by Manning Publications Co.

Simplified Chinese-language edition copyright ©2015 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由 Manning Publications Co. 授权人民邮电出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

版权所有, 侵权必究。

- 
- ◆ 著 [挪] Joachim Haagen Skeie  
译 卢俊祥  
责任编辑 杨海玲  
责任印制 张佳莹 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>  
北京艺辉印刷有限公司印刷
  - ◆ 开本: 800×1000 1/16  
印张: 14.75  
字数: 321千字 2015年7月第1版  
印数: 1-2500册 2015年7月北京第1次印刷

著作权合同登记号 图字: 01-2014-7055号

定价: 49.00元

读者服务热线: (010) 81055410 印装质量热线: (010) 81055316

反盗版热线: (010) 81055315

# 内容提要

---

Ember.js 号称是“雄心勃勃”的 JavaScript MVC 框架、现代 JavaScript MVC 框架的一个代表，是构造如单页面应用等现代 Web 应用程序的新型 Web 端开发框架。本书深入介绍了这一框架的方方面面。

全书分为三个部分。第一部分为基础内容，共 4 章，引导读者对 Ember.js 有个概括性认识，并掌握其基础知识点，此部分包含两个示例程序——注意事项应用程序与博客应用程序；第二部分包括第 5 章到第 8 章，涉及 Ember.js 实战的几个关注点，如 Ember Data、自定义组件以及测试；第三部分为第 9 章到第 11 章，讨论 Ember.js 的高级主题，如认证、运行循环和打包部署。

本书对于 Web 开发者掌握 Ember.js 非常有价值。阅读本书需要读者具备一定的 JavaScript 开发经验。

## 译者简介

---

卢俊祥 译者，书迷；关注 Web 技术趋势，热衷 App 开发、Web 开发、数据分析、架构设计以及各类编程语言；陈氏太极拳五十六式爱好者；佛禅人生，缘散缘聚。

# 前言

---

2006年起,我已经以某种方式进行 Web 应用开发。我开始为挪威最大的零售商开发 Web 应用,先是使用 JavaServer Pages (JSP) 技术,之后换成 JavaServer Faces (JSF)。当时这些技术很不错,并能让使用者达到使用目的。在那时 (Ajax 流行之前), HTTP 的请求-响应周期要求将大多数处理逻辑放在服务器端,服务器端在每次请求中传递所有标记、脚本和样式表给浏览器。

虽然以服务器端方式开发 Web 应用也能行之有效,但状态问题总是绕不开的。由于服务器端要求记住所有登录用户,管理状态很快就成为一个棘手而占用大量内存的任务。要如何处理用户打开多个标签页并在彼此间进行切换?跨多个(虚拟)机器延展服务的时候如何持久化会话数据?如果在服务器端存储用户状态,那又如何以一致方式方便地进行水平扩展?处理起来真不容易。

当我开始从事开源项目 Montric (那时叫 EurekaJ),我很快决定,如果想要在不借助独立的会话缓存的情况下水平扩展应用,我就得掌握一个具备良好前景和流行度的 JavaScript 框架。

我评估了多个框架,并用 Cappuccino ([www.cappuccino-project.org](http://www.cappuccino-project.org)) 和 SproutCore (<http://sproutcore.com>) 这两个框架搭建了原型。虽然 Cappuccino 的工具比较完整,而且提供了更细致好看的用户界面,我还是选择了一个 SproutCore,因为它可以让我使用既有技术积累,它还承诺可以跟第三方库方便结合。SproutCore 提供了更强大的视图,其组件方式可以最终组建出一个完整功能的 Web 应用。以组件为基础,使用服务器端框架,这些特性让我对 SproutCore 倍感亲切。但经历了最初的喜悦之后,我发现将第三方库集成到 SproutCore 并不简单。

随着 SproutCore 开发团队被收购以及该框架放慢了发展脚步, SproutCore 社区开始发生了变化, SproutCore 2.0 版进展顺利,但老版本与新版本间的裂痕却在扩大,最终, SproutCore 产生了一个新的分支—— Ember.js。

Ember.js 依托能够提供优良 Web 体验的技术而打造,它能够帮助开发者使用既有技能组

合开发 JavaScript 应用程序。Ember.js 并不抽象或隐藏 JavaScript、HTML 或者 CSS 的实现细节，反而与时俱进地充分利用这些技术。

不用说，我肯定是跟随 Ember.js，并且决定用 Ember.js 重写 EurekaJ 的前端部分。在此期间，我把项目改名为 Montric (<http://montric.no>)。从那时起，我就一直使用 Ember.js。在 v1.0.0 预览版发布期间，Ember.js 社区的发展也经历了起伏。那时频繁调整 API，感觉每周都在重新考虑和审视概念，但随着问题被逐步解决，思路也越来越明晰。在预览版时就决定写一本全面介绍 Ember.js 的图书，但对于这个想法我确实还有顾虑。

在我写作过程中，Ember.js v1.2.0 也发布了，API 变得稳定了，整个项目健康发展。今天，Ember.js 已经成为一个了不起的框架，能够帮助你创建极具挑战的 Web 应用。

# 致谢

---

至完稿时，本书是最全面且最结合实战的 Ember 书籍，学习本书是有一定挑战性的。在历经打击和困难的写作过程中，我也获益颇多。

感谢 Manning 团队出版这本 Ember.js 书籍，让我们有机会开启一段 Ember 学习之旅。我还要特别感谢策划编辑 Susanna Kline 容忍我一次次延期以及在 Skype 上无数次地发问，并始终耐心给予反馈，帮助我提高。同时，感谢文字编辑团队——Lianna Wlasiuk 与 Rosalie Donlon、Sharon Wilkey、Teresa Wilson——他们修订了全书大量拼写及语法错误。感谢也要送给校对员 Melody Dolab、排版员 Marija Tudor，以及项目经理 Mary Piergies 和 Kevin Sullivan。

审稿人确保了各个阶段目标的实现，我要对他们的工作表示感谢，感谢 Benoît Benedetti、Chetan Shenoy、Dineth Mendis、Jean-Christopher Remy、Leo Cassarani、Marius Butuc、Michael Angelo、Oren Zeev-Ben-Mordehai、Philippe Charrière、Richard Harriman 以及 Rob MacEachern。最后，感谢技术审校 Deepak Vohra 在送印前的认真复审。

我还要向我美丽的妻子 Lene、两个总给我惊喜的孩子 Nicolas 和 Aurora 致以特别而崇高的谢意！Lene 的支持和理解对我的写作是如此重要，要知道，写作得占用大量晚上和周末的空暇时间。当把时间优先花在其他事情上的时候，家庭带给你的安全感和幸福感至关重要。



# 关于本书

---

Ember.js 是一个最具雄心的 JavaScript Web 应用框架。随着 v1.0.0 正式版的发布，经过不到两年的发展，API 已经稳定了下来，项目也有序推进，并很快又推出了 v1.1.0 和 v1.2.0 两个版本。

构建一个庞大而有雄心的 Web 应用是个挑战。Ember.js 的应运而生是因为创建者们希望开发出一种框架，能够简化并标准化 Web 应用开发方式。本书的出发点是通过实例来讲解 Ember.js 的特性及精彩之处。

## 路线图

本书内容分为以下三个部分。

- 第一部分——Ember.js 基础。
- 第二部分——创建雄心勃勃的真实 Web 应用。
- 第三部分——高级 Ember.js 主题。

第一部分通过简单、独立的例子介绍 Ember.js 核心特性以及应用这些特性应具备的条件。

- 第 1 章介绍 Ember.js 及其背景，以及它的 Web 应用开发特性。读者将了解 Ember.js 的基本概念和技术。
- 第 2 章以第 1 章为基础，进一步介绍 Ember.js 核心特性。这一章会介绍绑定、计算属性、观察者模式以及 Ember.js 对象模型。
- 第 3 章是 Ember 路由器专题，路由器负责整合应用的各个部分。
- 第 4 章介绍 Ember.js 应用开发首选模板库 Handlebars.js。这一章将在 Ember.js 应用程序的表格功能中使用 Handlebars.js 特性，同时还将介绍 Ember.js 添加到 Handlebars.js 的 Ember.js-specific 插件。

第二部分会结合案例展开阐述，并介绍后续大多数章节将用到的 Montric 库。这一部分

会深入探讨 Web 应用开发的难点：如何与服务器端高效交互、编写自定义组件以及测试 Ember.js 应用。

- 第 5 章研究如何使用第三方 Ember Data 库跟服务器端通信（基于 Ember Data beta 2 版本）。在演示如何自定义 Ember Data 以适配既有服务器端 API 之前，我们先来研究服务器端 API 和 Ember.js 应用在使用 Ember Data 时的一些相应处理。
- 第 6 章会演示不依赖框架的情况下如何与服务器端交互。此外，还将一步步演示如何搭建一个完整的 CRUD 数据层。
- 第 7 章专门讲解如何自定义组件，这是 Ember.js 在后期加入的特性。通过 Ember.js 组件，可以创建原子级别的独立组件，从而在自己应用或更复杂组件构建过程中复用。
- 第 8 章讲述如何测试 Ember.js 应用程序。可以通过 QUnit 和 PhantomJS 来构建可行的测试方案。

第三部分将进一步深入高级 Ember.js 主题，并讨论其他服务和工具，为应用开发提供便利，并加深读者对 Ember.js 的理解。

- 第 9 章讲述如何通过第三方认证系统创建认证与授权功能。本章以开源解决方案 Mozilla Persona 为实现基础。
- 第 10 章借助 Backburner.js 库以更轻松的方式使用 Ember。这个后台引擎可以让你更好地驾驭 Ember.js 应用程序，并确保应用视图实时更新的同时还具备高效性能。
- 第 11 章介绍在代码不断膨胀的情况下如何组织 Ember.js 应用程序，以及在发布准备阶段如何创建、装配和打包应用程序。

## 阅读对象

本书立足于帮助读者成为一名熟练而高效的 Ember.js 开发者。有赖于读者的背景，使用如 Ember.js 这样的框架开发 JavaScript 应用程序有可能要面对较大困难。本书帮助读者快速掌握 Ember.js 概念并熟悉 Ember.js 技术及应用结构，本书适合 Ember.js 新手和专业开发者阅读。

作为前提条件，本书假定读者已经熟练掌握 JavaScript 语言，并对 jQuery 相关知识有一定了解。

## 代码约定及下载

本书排版约定如下。

- 楷体用于表示专业术语。
- Courier 字体用于代码样例、元素、属性、方法名、类、接口及其他标识符。
- 在代码清单里，以及强调重要概念时，会采用代码注释的方式。

- 限于版面，一些较长的代码行会折断到下一行。因此，在需要时代码清单里会出现续行标志（↵）。

本书包含了许多代码片段和源代码。第一部分的源代码可以通过本书 GitHub 页获取，或者到 Manning 出版社网站 [www.manning.com/Ember.jsinAction](http://www.manning.com/Ember.jsinAction) 下载 zip 格式的源代码压缩包。第二部分和第三部分的内容以 Montric 源代码为基础，Montric 源代码可以通过 GitHub 获取。

由于本书案例是实际的项目，本书写成时项目代码应该已经发生了变化。考虑到这一点，本书使用以下链接给出写作时的代码版本。

- 第 1 章和第 2 章：<https://github.com/joachimhs/Ember.js-in-Action-Source/tree/master/chapter1/notes>。
- 第 3 章：<https://github.com/joachimhs/Ember.js-in-Action-Source/tree/master/chapter3/blog>。
- 第 5 章、第 7 章~第 9 章和第 11 章：<https://github.com/joachimhs/Montric/tree/Ember.js-in-Action-Branch>。
- 第 6 章：<https://github.com/joachimhs/EmberFestWebsite/tree/Ember.js-in-Action-branch>。

尽管要保持案例的真实有效，但我还是会努力通过文字来阐述 Ember.js。虽然在独立而优秀的案例中使用 Montric 源代码可能会偶尔给读者带来挑战，但也让本书内容更深入。此外，书中案例的更新变化也能够让读者了解 Ember.js 的发展历程。

## 作者在线

购买本书的读者可以免费访问 Manning 出版社维护的专用论坛，在论坛里读者可以评论本书、提出技术问题并得到作者和其他开发者的帮助。要访问论坛并订阅信息请访问 [www.manning.com/Ember.jsinAction](http://www.manning.com/Ember.jsinAction)。该页面提供了注册后如何访问论坛的指引、各种帮助信息以及论坛行为准则。

Manning 以尽责的态度提供一个读者间、读者与作者间互动的空间。Manning 无法承诺作者的参与程度，其对论坛的贡献基于自愿而免费的原则。我们建议你尽量向作者提一些富有挑战性的问题，以保持作者的热情！

只要书籍得以出版，你就可以通过出版社网站访问作者在线论坛以及早期的讨论归档内容。

## 作者介绍

Joachim Haagen Skeie 是一位自由职业者，其供职于自己的公司 Haagen Software AS。致力于开发 Montric（一款开源的应用程序性能监控工具）和 Conticious（一款开源的主要用于

Ember.js 富互联网应用的 CMS API) 的同时,他还是一名独立咨询师、Ember.js 和 RaspberryPi 课程讲师。Montric 和 Conticious 的前端部分基于 Ember.js 技术,后端使用 Java 技术。

Joachim Haagen Skeie 从 2006 年开始,就从事各种规模的 Web 应用开发工作,主要使用 Java 和 Ember.js 技术。他和妻子、孩子一起居住在挪威首都奥斯陆。

## 关于封面图画

---

本书封面图画的标题是“巴黎的戏剧导演”。这幅图出自 19 世纪法国出版的西尔万·马雷夏尔 (Sylvain Maréchal) 地域服饰习俗四卷本摘要。其每幅图画皆由手工精心绘制和着色而成。丰富多样的马雷夏尔作品集让我们强烈感受到 200 年前世界上城镇与地域在人文上的巨大差异。那时候人们生活封闭，所操语种、方言各不相同。无论是在街道还是乡野，通过各式穿戴，就能够很容易地识别人们的职业、身份，以及他们来自何方。

自那时起着装发生了变化，同时那个年代丰富的地域多样性也逐渐褪去。现在要分辨出不同大陆的居民已非常困难，更不用说城镇或地域的区别了。也许我们已用文化多样性来装扮个人生活的多样化——当然是从更加多样而快节奏的科技生活的角度。

有时候从某个角度很难描述一本计算机书籍，Manning 参考两个世纪前地域生活的丰富多样性来设计图书封面，借此表达对计算机领域开创精神的赞誉，并通过马雷夏尔的图画将时光带回到历史。

# 目录

## 第一部分 Ember.js 基础

### 1

#### 第 1 章 发力雄心勃勃的 Web 应用 2

- 1.1 Ember.js 适用场景 2
- 1.2 从静态页面到 Ajax、再到全功能 Web 应用 4
  - 1.2.1 异步 Web 应用的兴起 5
  - 1.2.2 Ember.js 模型 6
- 1.3 Ember.js 概览 6
  - 1.3.1 Ember.js 特性 7
  - 1.3.2 Ember.js 应用程序结构 7
- 1.4 第一个 Ember.js 应用程序：记事本应用 9
  - 1.4.1 记事本应用开发起步 11
  - 1.4.2 创建命名空间与路由器 13
  - 1.4.3 定义应用程序路由 14
  - 1.4.4 创建并列出事项 16
  - 1.4.5 选择并查看单条事项 19
  - 1.4.6 删除事项 23
- 1.5 小结 28

### 2

#### 第 2 章 Ember.js 风格 29

- 2.1 绑定对象 30
- 2.2 自动更新模板 33
- 2.3 计算属性 35
- 2.4 观察者模式 38
- 2.5 Ember.js 对象模型 39
- 2.6 Ember.js 实现各层间数据

同步 41

2.7 小结 42

### 3

#### 第 3 章 使用 Ember.js 路由器融合应用结构 43

- 3.1 Ember.js 实战博客介绍 44
- 3.2 服务器端模型-视图-控制器模式的困境 46
  - 3.2.1 Ember MVC 模式 47
  - 3.2.2 将各层组合起来 48
- 3.3 Ember 路由器：Ember.js 的状态图 49
- 3.4 Ember.js 实战博客第一部分：博客索引页 52
  - 3.4.1 创建博客路由器 52
  - 3.4.2 添加视图和模板 55
  - 3.4.3 显示博客文章列表 56
- 3.5 Ember.js 实战博客第二部分：添加博客文章路由 59
- 3.6 依赖注入与 Ember 容器 65
  - 3.6.1 使用 needs 属性连接控制器 65
  - 3.6.2 通过 Ember 容器连接对象 66
- 3.7 小结 68

### 4

#### 第 4 章 通过 Handlebars.js 自动更新模板 69

- 4.1 模板是什么 69

- 4.1.1 简单表达式 70
- 4.1.2 块表达式 71
- 4.2 内置块表达式 73
  - 4.2.1 if 及 if-else 块表达式 73
  - 4.2.2 {{unless}}块表达式 74
  - 4.2.3 {{with}}块表达式 74
  - 4.2.4 Handlebars.js 的注释 75
- 4.3 结合使用 Handlebars.js 与 Ember.js 75
  - 4.3.1 在 index.html 中定义模板 76
  - 4.3.2 直接在 Ember.TEMPLATES 散列中定义模板 77
  - 4.3.3 创建 Handlebars.js 模板支持的 Ember.js 视图 77
- 4.4 Ember.js 提供的 Handlebars.js 表达式 78
  - 4.4.1 {{view}}表达式 79
  - 4.4.2 {{bind-attr}}表达式 80
  - 4.4.3 {{action}}表达式 81
  - 4.4.4 {{outlet}}表达式 83
  - 4.4.5 {{unbound}}表达式 84
  - 4.4.6 {{partial}}表达式 84
  - 4.4.7 {{link-to}}表达式 85
  - 4.4.8 {{render}}表达式 85
  - 4.4.9 {{control}}表达式 86
  - 4.4.10 {{input}}和{{textarea}}表达式 87
  - 4.4.11 {{yield}}表达式 88
- 4.5 创建自己的表达式 88
- 4.6 小结 89

## 第二部分 创建雄心勃勃的真实 Web 应用

### 第 5 章 获取数据: 使用 Ember Data 与服务端交互 92

- 5.1 将 Ember Data 用作应用缓存 93
  - 5.1.1 定义 Ember Data 模型 93
  - 5.1.2 标识映射的 Ember Data 94
  - 5.1.3 模型对象间的关联 96
  - 5.1.4 模型状态和事件 97
  - 5.1.5 与后台通信 99

- 5.2 初尝 Ember Data 100
  - 5.2.1 从模型中获取数据 101
  - 5.2.2 在模型中指定关联关系 101

### 5.3 Ember Data 模型的关联 103

- 5.3.1 了解 Ember Data 模型的关联关系 103
- 5.3.2 Ember Data 端数据加载 106
- 5.4 自定义适配器和序列化器 108
  - 5.4.1 编写自定义适配器, 但保留默认的序列化器 109
  - 5.4.2 编写自定义适配器和序列化器代码 111
  - 5.4.3 自定义 URL 113
- 5.5 小结 113

## 6 第 6 章 绕过 Ember Data 与服务端交互 115

### 6.1 Ember Fest 介绍 115

- 6.1.1 了解应用程序的路由器 117
- 6.1.2 使用 model() 钩子函数获取数据 117
- 6.1.3 实现标识映射 118
- 6.2 获取数据 119
  - 6.2.1 通过 find() 函数返回指定演讲数据 120
  - 6.2.2 通过 findAll() 函数获取所有演讲数据 121
  - 6.2.3 实现 Emberfest.Talk 模型类 121

### 6.3 数据持久化 124

- 6.3.1 通过 createRecord() 函数提交新的演讲 124
- 6.3.2 通过 updateRecord() 函数修改演讲数据 127
- 6.3.3 通过 delete() 函数删除演讲数据 130

### 6.4 小结 131

## 7 第 7 章 编写自定义组件 133

- 7.1 关于 Ember 自定义组件 134

- 7.2 实现可选列表 134
  - 7.2.1 定义 selectable-list 组件 137
  - 7.2.2 selectable-list-item 组件 138
  - 7.2.3 delete-modal 组件 140
  - 7.2.4 通过已完成的三个组件删除具体项 141
- 7.3 实现树形菜单 144
  - 7.3.1 树形菜单的数据模型 144
  - 7.3.2 定义 tree-menu 组件 145
  - 7.3.3 定义 tree-menu-item 和 tree-menu-node 组件 145
  - 7.3.4 单选功能支持 147
- 7.4 小结 150

## 8 第 8 章 测试 Ember.js 应用程序 152

- 8.1 使用 QUnit 和 PhantomJS 进行单元测试 153
  - 8.1.1 QUnit 介绍 154
  - 8.1.2 使用 PhantomJS 在命令行执行测试 156
  - 8.1.3 集成 QUnit 和 PhantomJS 157
- 8.2 使用 QUnit 编写简单的 Ember.js 单元测试 160
- 8.3 集成测试 164
  - 8.3.1 Sinon 介绍 165
  - 8.3.2 添加新告警信息的集成测试 166
- 8.4 通过 Ember. Instrumentation 进行性能测试 168
- 8.5 小结 169

## 第三部分 高级 Ember.js 主题

### 9 第 9 章 使用 Mozilla Persona 进行认证 172

- 9.1 集成第三方认证系统 173
  - 9.1.1 首次登录及注册 173
  - 9.1.2 通过第三方认证提供者登录 Montric 177

- 9.2 通过 HTTP cookie 登录用户 180
- 9.3 小结 183

## 10 第 10 章 Ember.js 运行循环——Backburner.js 185

- 10.1 什么是运行循环 185
  - 10.1.1 Ember.js TodoMVC 应用程序介绍 186
  - 10.1.2 解释 Ember.js 运行循环 187
- 10.2 在运行循环框架下执行代码 191
  - 10.2.1 在当前运行循环中执行代码 191
  - 10.2.2 在下一个运行循环中执行代码 192
  - 10.2.3 在后续运行循环中执行代码 193
  - 10.2.4 在指定队列执行代码 194
  - 10.2.5 通过运行循环执行重复任务 196
- 10.3 小结 197

## 第 11 章 打包与部署 198

- 11.1 理解 JavaScript 应用程序打包和装配 198
  - 11.1.1 选择目录结构 199
  - 11.1.2 组织自己编写的源代码 200
  - 11.1.3 组织非 JavaScript 资源 202
  - 11.1.4 Ember.js 应用程序装配过程 203
- 11.2 使用构建工具 Grunt.js 204
  - 11.2.1 为 Montric 应用程序引导 Grunt.js 构建系统 205
  - 11.2.2 连接 JavaScript 代码 206
  - 11.2.3 抽取插件配置代码到单独文件 208
  - 11.2.4 Lint 常见错误 209
  - 11.2.5 预编译 Handlebars 模板 211
  - 11.2.6 压缩源文件 214
  - 11.2.7 Grunt.js 的优缺点 217
- 11.3 小结 218



# 第一部分

## Ember.js 基础

JavaScript MVC 框架 Ember.js 用于组织大型 Web 应用代码结构。与其他流行的 JavaScript 应用框架相比，它具有更完整的 MVC 模式特征，并包含创建新一代 Web 应用所需特性。它自信满满，严格依赖约定优于配置的设计范式来构造应用程序。

由于包含大量特性及应用约定，Ember.js 的学习曲线比较陡峭。本书第一部分包含 4 章内容，帮你尽快找到 Ember.js 开发的感觉，并确保你从一开始就能有所成。

前两章重点介绍 Ember.js 常用核心特性。第 3 章介绍 Ember 路由器，第 4 章聚焦于为 Ember 程序员遴选的模板库 Handlebars.js。