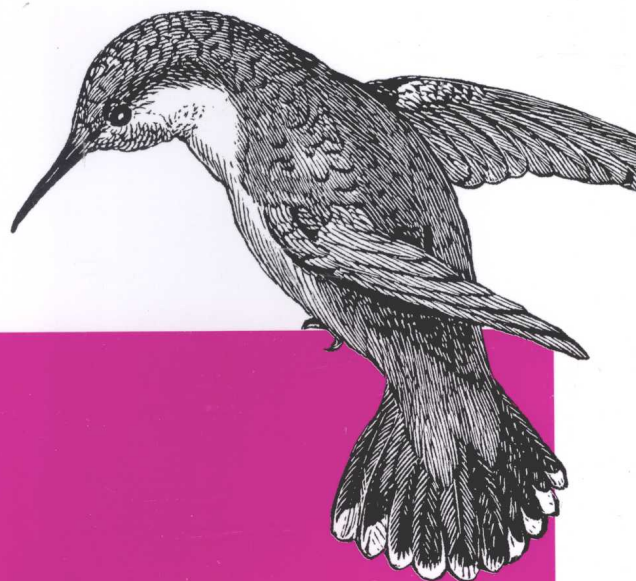
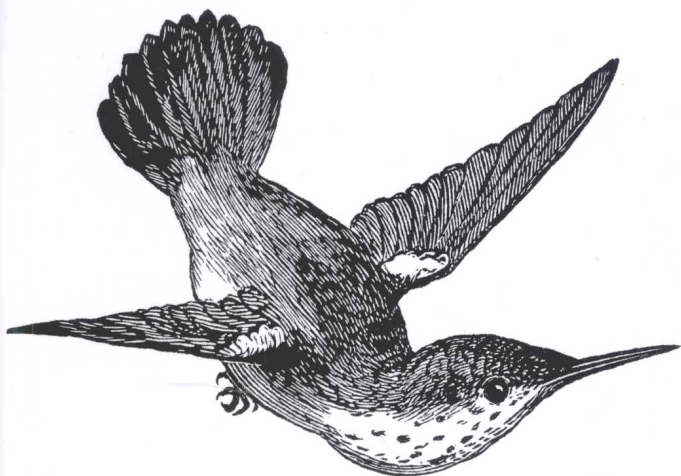


O'REILLY®



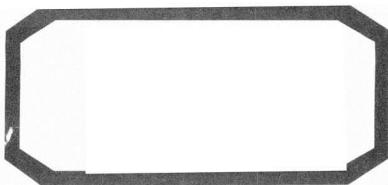
轻量级

Django

Lightweight Django

中国电力出版社

Julia Elman & Mark Lavin 著
侯荣涛 吴磊 译



轻量级 Django

Julia Elman & Mark Lavin 著

侯荣涛 吴磊 译

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'REILLY®

O'Reilly Media, Inc. 授权中国电力出版社出版

中国电力出版社

图书在版编目 (CIP) 数据

轻量级 Django / (美) 茱莉亚·埃尔曼 (Julia Elman), (美) 马克·拉温 (Mark Lavin) 著; 侯荣涛, 吴磊译. —北京: 中国电力出版社, 2016.10

书名原文: Lightweight Django

ISBN 978-7-5123-9396-7

I. ①轻… II. ①茱… ②马… ③侯… ④吴… III. ①软件工具—程序设计 IV. ①TP311.56

中国版本图书馆 CIP 数据核字 (2016) 第 116037 号

北京市版权局著作权合同登记

图字: 01-2016-2777 号

Copyright © 2015 Julia Elman and Mark Lavin, All right reserved.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and China Electric Power Press, 2016. Authorized translation of the English edition, 2015 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2015。

简体中文版由中国电力出版社出版 2016。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。

封面设计 / Ellie Volckhausen, 张健

出版发行 / 中国电力出版社 (<http://www.cepp.sgcc.com.cn>)

地 址 / 北京市东城区北京站西街 19 号 (邮政编码 100005)

经 销 / 全国新华书店

印 刷 / 北京天宇星印刷厂

开 本 / 787 毫米 × 980 毫米 16 开本 14 印张 268 千字

版 次 / 2016 年 10 月第一版 2016 年 10 月第一次印刷

印 数 / 0001 - 3000 册

定 价 / 39.00 元 (册)

敬告读者

本书封底贴有防伪标签, 刮开涂层可查询真伪
本书如有印装质量问题, 我社发行部负责退换

版权专有 翻印必究

O'Reilly Media, Inc. 介绍

O'Reilly Media 通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自 1978 年开始，O'Reilly 一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly 的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly 为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了 Make 杂志，从而成为 DIY 革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly 的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly 现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版，在线服务或者面授课程，每一项 O'Reilly 的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O'Reilly Radar 博客有口皆碑。”

——Wired

“O'Reilly 凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——Business 2.0

“O'Reilly Conference 是聚集关键思想领袖的绝对典范。”

——CRN

“一本 O'Reilly 的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim 是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照 Yogi Berra 的建议去做了：‘如果你在路遇到岔路口，走小路（岔路）。’回顾过去 Tim 似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

目录

前言	1
绪论	7
第1章 世界上最小的Django项目	11
你好Django.....	11
改进	15
第2章 无状态的Web应用	22
什么是无状态?	22
可复用应用与可组合服务	23
占位图片服务器	23
占位视图	25
创建主页面视图	31
第3章 创建静态站点生成器	38
使用Django创建静态站点	38
什么是快速原型?	39
最初的项目结构	39
修饰页面	41
生成静态内容	52
处理和压缩静态文件	56
生成动态内容	60

第4章 构建REST API	67
Django和REST	67
Scrum板数据图	68
设计API.....	74
测试API.....	82
下一步	96
第5章 使用Backbone.js的	
客户端Django	97
Backbone简述.....	98
设置项目文件	99
连接Backbone到Django.....	104
客户端Backbone路由	106
构建用户认证	110
第6章 单页面Web应用	132
什么是单页面Web应用?	132
发现API.....	133
构建主页	136
sprint详情页面.....	141
CRUD任务	155
第7章 实时Django	165
HTML5实时API	165
在Tornado下使用websocket.....	167
客户端通信.....	175
第8章 Django与Tornado通信	195
从Tornado接收更新	195
改善服务器.....	200
最终的websocket服务器.....	212

前言

自 Django 创建以来，各种各样的开源社区已经构建了很多 Web 框架，比如 JavaScript 社区创建的 Angular.js、Ember.js 和 Backbone.js 之类面向前端的 Web 框架，它们是现代 Web 开发中的先驱。Django 从哪里入手来适应这些框架呢？我们如何将客户端 MVC 框架整合成为当前的 Django 基础架构？

本书讲述如何利用 Django 强大的“自支持”功能体系。本书的目标是引导读者跨越认为 Django 太繁重而无法进行快速应用开发的误区。从创建世界上最小的 Django 应用到构建 RESTful API，本书将引导读者学习如何利用这种流行的 Python Web 框架。

写这本书的目的

我们编写本书主要是出于对 Django 的热爱。我们的社区是令人惊奇的，它有很多资源可以用来学习 Django 和开发应用。尽管我们也喜欢包括官方的 Django 文档在内的一些资源，但这仅仅是对 Django 强大功能的过于关注，而非它的解耦设计。Django 是一个令人满意的框架，它带有很多用于构建 Web 应用的通用程序。在本书中，我们要突出说明如何将这些组件进行分离和间接替换，并从中选出最适合创建应用的一些组件来。同样地，我们想要将 Django 项目和应用的典型框架进行分解。我们的目标是使读者不再提出“我怎样用 Django 完成我的 X 工作”这样的问题，而是提出“Django 能帮我完成 X 工作吗？如果不能，社区有可用的工具吗”这样的问题。

另外，我们想要回答的问题是利用大量客户端实时交互，以及与本地可移动应用匹配的组件构建更多应用时，将 Django 放在网络的什么地方适用。作为框架，用户对 Django 一无所知，这使得一些喜欢 Django 的用户无法得到构建这种应用的指导。我们希望本书能够帮助解决这些问题，想看到 Django 及其社区日益壮大，并在多年后大家也成为它的一员。

本书读者对象

如果你喜欢阅读本书，你很可能是 Django 的中级用户。在学习完 Django 的注册教程以及编写了一些基本 Django 应用后，很想知道下一步该如何做。本书接下来会帮助你基本掌握如何利用 Django 的有效性和简单性。

或者，你可能正在进行一个 Django 项目，很想知道如何将 Backbone.js 整合到你的项目中。本书将引导你进行一些有关整合的最好练习，并为构建内容丰富的 Web 应用提供一个起点。

哪些人不适合阅读本书？

尽管本书适用许多不同背景的开发人员，但并不能面面俱到。对于那些不具备编写 Python 和 JavaScript 程序能力的人来说，这本书可能很不适合。书中所有的概念和实例都是围绕这些语言编写的，并在每个章节中大量使用。对于那些初次接触 Django 的新手，也不适合。

关于本书示例

书中的每个项目示例都是在快速应用开发的主旨下精心制作的。在每一章，都将学到在项目管理、工具和团队合作支持下创建项目的方法。我们想要读者创建用于自己的项目，并能够根据自己的需要进行定制。一般而言，本书提供的实例代码，都可以在你的程序和文档中进行使用，无需得到我们的许可，除非是更新代码的重要内容。例如，使用本书中的几个代码块编写程序是不需要得到允许的。销售或分发 O'Reilly 书中的实例光盘就需要得到许可。引用本书或本书中的实例代码解决问题不需要得到允许。将本书中的大量实例代码集成到自己产品的文档中需要得到许可。

书中的代码样例可以从下面网站找到：<https://github.com/lightweightdjango/examples>

我们很赞赏对本书的引用，但不要求。通常引用包括标题、作者、出版商和 ISBN。例如：“《*Lightweight Django*》 Julia Elman and Mark Lavin (O'Reilly), Copyright 2015 Julia Elman and Mark Lavin, 978-1-491-94594-0”。

如果发现所使用的代码示例超出了上面的许可范围，可以随时通过邮件与我们联系：

permissions@oreilly.com。

本书结构

第 1 章，世界上最小的 Django 项目。

创建轻便简单的网络应用是本书的核心思想。在这一章，要创建一个可运行的单行文件“Hello World”的 Django 应用。

第 2 章，无状态的 Web 应用。

讲述占位符图像服务的创建方法。第 2 章将引导读者通过创建无状态网络应用来生成占位符图像的 URL。

第 3 章，创建静态站点生成器。

快速构建原型是一门有用的创建和辅助网络应用的技术。我们要通过创建静态网站生成器协助维护团队项目的方式来检验该技术的效果。

第 4 章，构建 REST API。

REST API 是创建丰富的、具有内容相关性的网络应用的重要部分。在这一章我们将利用 Django 静态框架开始构建一个大比例的 Scrum 板应用。

第 5 章，使用 Backbone.js 的客户端 Django。

本章继续讲述在第 4 章中用新构建的 RESTful API 创建的 Backbone.js 应用。介绍创建新 Backbone 应用的每个组件以及如何用 Django 同步该客户端框架。

第 6 章，单页面 Web 应用。

单页面网络应用是一种能够创建丰富客户端网络应用的途径。在本章，我们将回到简单的 Backbone 应用中，不断使这个单页面应用更加健壮。

第 7 章，实时 Django。

这一章讲述创建响应实时交互的网络应用，为用户提供及时满意的服务。为了继续完成先前两章的项目，我们要将一个实时组件添加到使用网络插件和 Tornado 的 Scrum 板上，这是一个用 Python 编写的异步网络库。

第 8 章，Django 和 Tornado 通信。

将强大的 Django 与 Tornado 的健壮特性相结合是创建可变、实时的 Django 应用的重要标志。在本章中，我们将通过集成 Django 的通信能力扩展 Tornado 服务器的功能，来建立一种既安全又可交互的联系。

排版约定

本书使用如下排版约定：

斜体

表示新术语、URL、email 地址、文件名和文件扩展名。

等宽字体 (Constant width)

用于程序列表，以及引用如变量、函数名、数据库、数据类型、环境变量、语句和关键词之类的程序元素。

加粗等宽字体 (Constant width bold)

表示应由用户手动输入的命令或其他文本。

斜体等宽字体 (Constant width italic)

表示要用用户提供的值或上下文确定的值来替换的文本。

所有代码实例运用省略号 (…) 表示先前现实的一些代码已经被跳过，来缩短实例代码或跳到最相关的代码段。



这个符号表示提示或建议。



这个符号表示一般注释。



这个符号表示警告或警示。

致谢

有许多人需要感谢，没有他们的努力这本书是不可能完成的。本书的编辑 Meghan 给了我们很大的支持。

感谢技术评论员 Aymeric Augustin、Jon Banafato、Barbara Shaurette 和 Marie Selvanadin 给我们的评价，无论是表扬还是批评，对我们尽力完成这本书都给予了帮助。也要感谢 Heather Scherer 对本书给予的技术指导。

对所有开源的开发人员和提供者表示感谢，他们无穷无尽的努力为本书提供了使用和写作所需的各种工具。

感谢早期版本的读者，他们给了我们完成工作的机会，使我们能够解决敲字和格式缺陷等问题，并提供了反馈，从而使错误得到纠正。

Julia

我非常感谢我极其可爱的家庭和亲密的朋友，他们对我写作本书给予了一贯的支持。感谢我的丈夫 Andrew，对我能力的信任和漫长与崎岖的写作过程中的一贯鼓舞和坚定的支持。感谢我的母亲 Katherine，她使我超越了自己的能力。感谢我的继父 Tom，他教我如何使用无线电钻为我的汽车更换机油，为我灌输了艰苦工作的价值。感谢我的哥哥 Alex 和姐姐 Elizabeth，他们一直作为旁观者为我鼓劲。感谢我最好的朋友 Jenny，她给了我永恒的爱和终生的友谊。

还要对我的极好的合作者 Mark 的卓越才能和友谊表示感谢。他是我曾经合作过的最有才能的开发者之一。我们并肩完成了这本书，我不能想象与其他人合作撰写这本书会是怎样一个过程。

我还要非常感谢 Python 社区和在我的事业上给我灵感、鼓励和指导的特殊成员：James Bennett、Sean Bleier、Nathan Borrer、Colin Copeland、Matt Croydon、Katie Cunningham、Selena Deckelmann、Jacob Kaplan-Moss、Jessica McKellar、Jesse Noller、Christian Metts、Lynn Root、Caleb Smith、Paul Smith、Karen Tracey、Malcolm Tredinnick、Ben Turner 和 Simon Willison。

Mark

首先，这本书离不开来自我的家庭的爱和支持。我妻子 Beth 和女儿 Kennedy 忍受我长时间工作，以及由于负担的加重而导致的脾气暴躁。还要感谢我的哥哥 Matt，他给了我

很深的见解和早期的反馈。感谢我的父母和我的哥哥 James，他们给了我无尽的支持。

感谢我的合作者 Julia。我们之间是在友谊和互相尊重中进行合作的。我将永远珍惜比我们合作成果建立的更加伟大的东西。

最后，感谢我在 Cactus 小组的同事们对我及时的支持、反馈和鼓励。

绪论

在开始之前，本章将对完成本书中的例子所需的知识和软件做一个概要介绍。

Python

本书面向的是具有 Python 经验的开发者，使用的是 Python 3。特别提醒，书中所有的示例都在 Python 3.3 和 Python 3.4 中得到了测试。尽管我们并不推荐，但对 Python 足够熟悉的读者也可以在需要的时候进行代码转换，然后在 Python 2.7 下完成这些例子。如果想要查看有关 Python 新版本中的特性以及使用于系统的安装指南，请访问：<https://www.python.org/downloads/>。

假定已经在本地开发设备上安装了 Python，并知道如何编辑 Python 文件和运行这些文件。尽管有些系统或者安装可能需要使用 Python3 或 Python 的完整版，如 Python 3.3、Python 3.4，但本书在命令行中调用 Python 时，将使用 `python` 命令。同样的，在安装新的程序包时，因为有些安装版本需要使用 `pip3` 命令，因此本书的示例将使用 `pip` 命令。针对本书和一般的 Python 开发，我们推荐使用 `virtualenv` 为每个项目创建一个隔离的 Python 环境。没有这样的隔离环境而通过 `pip` 安装程序包时，可能会需要访问根目录或计算机系统管理员的权限。假设没有创建隔离环境，就需要在 `pip` 命令前加上 `sudo` 前缀或一些用来获取这些权限的其他命令，但这些前缀都未在示例中写出。

Python 程序包

在开始本书之前，唯一需要安装的 Python 程序包是 Django。书中所有示例都是在 Django 1.8.3 下编写和测试的。建议使用 `pip` 进行安装：


```
hostname $ pip install Django==1.8.3
```



直到 2015 年 8 月, Django 1.8.3 仍然处于发布的候选阶段。如果早先的版本无法使用, 可以用 `pip https://github.com/django/django/archive/stable/1.8.x.zip` 命令从开发分支中安装 1.8 的先行版。

要了解更多关于 Django 新版本中的特性, 请访问 <https://docs.djangoproject.com/en/dev/releases/1.8/>。要查看其他安装指令, 也可以参阅 Django 安装指南。

在讲解各章的过程中, 还要安装其他额外的软件包。第 1、2、3 章中的项目是相互独立的, 可以在单独的虚拟环境中处理, 只需部署 Django 即可。第 4 章到第 8 章包含一个大的项目, 在这些章节中需要使用同一个虚拟环境。

Web 开发

由于 Django 是一个 Web 框架, 本书假设读者具备了基本的 HTML 和 CSS 知识, 因此 JavaScript 的例子更加深些, 所需的相关知识会在接下来的段落中详细说明。如果对 HTTP 协议, 尤其是各种 HTTP 动词 (GET、POST、PUT、DELETE 等) 的用法和作用有一定理解, 则会很有帮助。

JavaScript

本书后面的章节会大量使用 JavaScript。需要熟练编写 JavaScript/jQuery。有 DOM 操作和使用 jQuery 调用 Ajax 经验的开发者, 可以学做 Backbone.js 的示例。如果熟悉如 Angular.js、Ember.js 或 Knockout.js 等其他的客户端架构, 那就更超前了。本书并非是 Backbone.js 的权威指南。如果对 JavaScript 开发, 尤其是 Backbone.js 的 MVC 结构不熟悉, 推荐阅读一些 O'Reilly 的书:

- JavaScript: The Definitive Guide, by David Flanagan
- JavaScript: The Good Parts, by Douglas Crockford
- JavaScript Patterns, by Stoyan Stefanov
- Speaking JavaScript, by Axel Rauschmayer
- Developing Backbone.js Applications, by Addy Osmani

浏览器支持

本书中的示例使用了相对新的 HTML5 和 CSS3 的 API，因此希望使用现代的浏览器。在以下浏览器之前的版本并未经过全面的测试，可能不支持示例中用到的技术：

- IE 10+
- Firefox 28.0+
- Chrome 33.0+

应当熟悉如何在喜欢的浏览器中使用开发工具来修改潜在的错误，查看网络请求，以及使用 JavaScript 控制台。

附加软件

我们会在后面的章节中用到两种流行的数据库：PostgreSQL 和 Redis。在需要的章节中会给出简要的安装指导，但有关读者所使用系统的更完整的指南请查看官方文档。

PostgreSQL 是一个开源的关系型数据库系统，在 Django 社区中有着广泛的支持。所有 Django 支持的 PostgreSQL 版本在本书中都可以使用。Django 1.8 支持 PostgreSQL 9.0 及以上版本。

Redis 是一个开源的键值对缓存。本书中使用了 Redis 的 pub/sub 特性，因此需要 2.0 或更高版。

世界上最小的 Django 项目

我们中有多少人是从官方的注册教程中开启 Django 使用之旅的？对许多人来说这似乎只是一个过程，但作为一个介绍 Django 的入门就显得很困难了。我们需要运行各种命令、生成多个文件，甚至很难分辨出什么是项目，什么是应用。对于想要使用 Django 来创建应用的新手来说，他们会觉得选择一个 Web 框架有点太“重量级”了。要怎样才能让我们轻便、简单地开始，消除恐惧呢？

让我们来花点时间探询一下启动 Django 项目的推荐任务吧。通常创建新的项目要使用 `startproject` 命令。这个命令没有什么神奇的地方，只是创建一些文件和目录而已。

尽管 `startproject` 命令是一个很有用的工具，但却不是启动 Django 项目所必须的。可以根据需求，自由地设计自己的项目。对于大型项目，开发者会得益于 `startproject` 命令所提供的代码组织。然而，不应该因为这一命令的方便使用而忽略了它的工作和作用。

在本章中，我们将利用 Django 的基本生成模块设计一个创建简单项目的示例。这个轻量级的“Hello World”项目将使用单个文件的方法来创建一个简单的 Django 应用。

你好 Django

用一种新语言或者新框架创建一个“Hello World”示例是第一个通常项目。我们在 Flask 社区中看到这个简单的开始示例，以此来说明这是怎样一个轻量级的微型框架。

在本章中，我们将从一个 `hello.py` 文件开始。这一文件将包含运行 Django 项目所需的所有代码。为了得到能完整运行的项目，我们需要创建一个为根目录 URL 提供服务的视图，以及配置 Django 环境所需的设置。