

**Broadview**  
www.broadview.com.cn

**[PACKT]** open source\*  
PUBLISHING community experience distilled



**Mastering Flask**

# 深入理解Flask

掌握用PHP创建强大动态Web应用的专项技术

[美] Jack Stouffer 著  
苏丹 译

 中国工信出版集团

 电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
<http://www.phei.com.cn>

Mastering Flask

深入理解Flask

[美] Jack Stouffer 著

苏丹 译

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

Flask 是在 Python 用户中最为流行的 Web 开发框架。本书从一个简单的 Flask 项目入手，由浅入深地探讨了一系列实战问题，包括如何使用 SQLAlchemy 和 Jinja 等工具进行 Web 开发；如何正确地设计扩展性强的 Flask 应用架构和搭建 MVC 环境；对于各种 NoSQL 数据库的特性，何时应该、何时不应该及如何使用它们；通过使用 Flask 扩展快速实现用户的身份系统、RESTful API、NoSQL 查询、后台管理等功能；如何创建自己的扩展；使用 Celery 编写异步任务，使用 py.test 进行单元测试等；最后介绍了如何部署上线，包括使用自己搭建的服务器或使用各种云服务，以及如何权衡和选择这些不同的解决方案。

Copyright © 2015 Packt Publishing. First published in the English language under the title ‘Mastering Flask’.

本书简体中文版专有出版权由 Packt Publishing 授予电子工业出版社。未经许可，不得以任何方式复制或抄袭本书的任何部分。专有出版权受法律保护。

版权贸易合同登记号 图字：01-2015-8407

### 图书在版编目（CIP）数据

深入理解 Flask / (美) 杰克·史都华 (Jack Stouffer) 著；苏丹译. —北京：电子工业出版社，2016.7

书名原文：Mastering Flask

ISBN 978-7-121-29231-6

I. ①深… II. ①杰… ②苏… 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 146735 号

策划编辑：张春雨

责任编辑：徐津平

印 刷：北京中新伟业印刷有限公司

装 订：北京中新伟业印刷有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：787×980 1/16 印张：16 字数：360 千字

版 次：2016 年 7 月第 1 版

印 次：2016 年 7 月第 1 次印刷

定 价：79.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888, 88258888。

质量投诉请发邮件至 [zltz@phei.com.cn](mailto:zltz@phei.com.cn)，盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

本书咨询联系方式：010-51260888-819 [faq@phei.com.cn](mailto:faq@phei.com.cn)。

# 译者序

最近才知道，Flask 在一开始竟然是作为愚人节玩笑诞生的，不过后来大受好评，并成为正式项目。我在最初接触 Flask 的时候，它也只是作为颇具先锋气质的“单文件微框架”而在 Python 世界被人们口口相传。但在很短的时间内，它在 Python Web 开发者中开始流行起来，Flask 社区迅速壮大，为它开发的插件数量也呈爆发式增长。如今，Flask 在事实上已经成为轻量级 Python Web 框架的代言人。不论是对于 Python 初学者，还是对于有若干年经验的 Python 老手，使用 Flask 搭建新项目都是非常有吸引力的选择。

不过即使再优秀的工具，仍然很容易被用错。作者在书中也反复强调了这一点。Flask 提供的是一个最小化的 Web 框架，在此基础上你可以方便地使用你最喜欢的方式，添加你最得心应手的工具，形成一个专属于你和你的项目的高度定制化的强大工具套装。它不会逼你自己造轮子，因为在社区里几乎应有尽有；它也不会逼你用它附赠的轮子，因为它根本就没有准备赠品。对老司机来说，这样的框架可以最大程度地解放战斗力；对初学者来说，又可以利用社区中大量现成的资源很快上手。这样大家便都开心了，毕竟写程序最重要的就是开心。

但问题可能也隐藏在这里面：老手可能会根据自己以往熟悉的方式，把强大的战斗力释放在一些本来或许没有必要重复造轮子的地方；而新手则可能犯更多的错误：比如使用最原始的方式或者来自其他框架的方式搭建程序结构，结果当应用规模膨胀时面临困境；又或者引入了过多没必要引入的 Flask 扩展，导致代码难以测试和维护，稳定性降低，甚至出现冲突。另一个潜在的问题是，庞大的社区带来的好处毋庸多言，但如果大量地使用非官方库，则意味着需要承担社区开发者的水平参差不齐所带来的风险。有个程序员朋友吐槽说好多 Flask 扩展有个共同点，就是文档写得比代码棒。总的来说，如何正确地运用框架给予的高度自由，结合具体项目的需求，在用不用扩展和使用什么扩展之间做出最合理的选择，则是需要一定的经验的。而对于 ORM、NoSQL、云服务等，同样存在用不用、用哪个、如何用的选择难题。

在我看来，在面向业务的开发中，深入二字的含义并不是指黑科技和奇技淫巧，而是具有预见性的解决方案选择。基于对当下项目需求的准确认知，采用最适合当前情况

的不多不少的架构设计，在避免提前优化的同时对未来做好充分准备，对测试部署等流程的意义的深刻理解，才是深入和进阶，也正是本书的重点所在。实际上本书超纲的地方有点多，探讨的问题已经远不止 Flask 了。

这是我翻译的第 1 本书。翻译这件事果然是比较特别的体验，跟写作和编程都很不一样。翻译一整本书对于像我这样的拖延症患者来说，是个极大的挑战，因为即使每次翻译一大段，进度条也几乎没有移动的意思；但如果拖下去，它就绝对不会结束。拖到最后时刻然后突击解决的常规办法已经不奏效了，无论如何，经过漫长的修炼，终于还是完成了。感谢张春雨编辑邀请我翻译本书，感谢 Cookie 对我不务正业的理解和支持，感谢公孙欠扁等朋友帮我解决一些英文问题。

水平所限，翻译中必定存在问题，请读者见谅。我在豆瓣、Twitter、微博等处的 ID 都是 su27，读后若有困惑或批评意见，欢迎找我讨论、吐槽。

祝每位读者都能从本书中获益。珍爱生命，我用 Python！

苏丹

2016 年 6 月

# 关于作者

**Jack Stouffer** 是一名有数年 Web 应用开发经验的程序员。他在两年前将自己所有的项目都切换到了 Flask，目前在位于密歇根州奥本山的 Apollo America 工作，使用 Python、Flask 和 JavaScript 编写企业内部的业务工具。Jack 是开源技术的信徒与支持者，他在 GitHub 上发布以他所推荐的最佳实践原则编写的 Flask 范例时，该范例立即成为线上最受欢迎的 Flask 代码库。Jack 还曾经担任由 Packt Publishing 出版的 *Flask Framework Cookbook* 一书的审稿人。

---

首先，我想感谢 Apollo America 的两位经理：Peter Stouffer 和 Louise Laing，他们给了我十分弹性的工作时间来写作本书。若非如此，我不可能在有限的时间内完成本书。其次，我想感谢 Riddhi Tuljapurkar，他是本书的内容编辑，也是我在 Packt Publishing 的联系人。他在本书交稿日将到之时，仍然给了我额外的时间，并对书稿提出了宝贵的意见。然后，我想感谢 Llewellyn Rozario，他给了我写作本书的机会。最后，我想感谢五位技术审稿人：Ayun Park、Ben Chaimberg、Nidal Alhariri、Rotem Yaari 和 Pedro Baumann，他们付出了珍贵的时间和精力，使本书的最终版本与初稿相比，在质量上得到了极大提升。

---

# 关于技术审稿人

**Nidal Alhariri** 是一名企业家、全栈计算机程序员和技术顾问。他是 **Enferno Framework**（一个基于 Flask 的 Python Web 框架）的作者。

他也是 **Level 09 Studios** 的创始人和 CTO。**Level 09 Studios** 是一家网络与软件开发代理公司，坐落于迪拜和柏林，为各种背景的客户提供服务，包括国际组织、酒店、高档楼盘、卫星电视、出版社及很多其他客户。

他作为很多软件应用和网络系统背后的推动力量已接近 12 年，具有按时、按预算交付项目的出色记录，对其服务满意的客户数不胜数，同时他也跟开发者社区有着深厚的联系。

**Pedro Alejandro Baumann** 是 **Athelas Perú** 的共同创始人、心理治疗师、系统管理员及自学成才的程序员。他是 **Athelas** 的后端开发工程师和系统管理员。他喜欢学习新技术，醉心于 Python，喜欢在闲暇时钻研一些开源项目（他几乎把 **ZURB foundation** 添加到了每个需要它的 Flask 项目中）。他坚持不懈地折腾各种各样的项目，最近在开发一个向家长和教师报告视频游戏真相的项目。另外，他还在为秘鲁的一个编程博客写作（**De Código y Algo más**）。

---

我想感谢我的妻子 **Lucia**，她一直鼓励我成为一名程序员并以此为生。我也想感谢我的同事，他们愿意让我尝试自己当时喜欢的任何技术来进行工作。

---

**Ben Chaimberg** 是芝加哥大学的一名学生，自儿时被父亲教编程以来，他便从未停止过编程。目前他正在开发一个 Flask 的全栈 Web 应用，同时也在努力完成数学系的学业。

---

我想感谢父母长期以来的支持和全心投入，感谢 Max 的友情与耐心，感谢 Kono 女士的教导与关怀。

---

**Ayun Park** 是一名 19 岁的韩国软件工程师，从 2011 年开始用 Python 编程。我们可以在 <http://parkayun.kr> 上找到他。

**Rotem Yaari** 是一名以色列软件工程师，从 2001 年开始用 Python 开发软件。他积极参与了好几个开源 Python 项目，比如 Slash 测试框架、Weber Flask webapp 框架、Flask-Loopback 及其他项目。

在过去的 8 年里，Rotem 一直在数据存储方面的创业公司中进行后端基础设施的开发工作。他的专长为使用 Web 服务和高可用性基础设施进行复杂的软件集成。



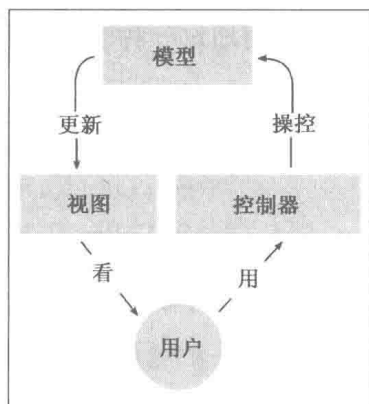
# 前言

Flask 是一个基于 Python 的 Web 框架，它的设计目的是提供 Web 开发所需要的最小功能子集。Flask 与别的框架（尤其是采用其他编程语言的框架）的不同之处在于：它没有绑定诸如数据库查询或者表单处理等功能库，以及它们所组成的整个生态系统。它倾向于对这些功能的实现方式不做任何限定。

这种系统带来的主要好处是：开发者能够使用他们想用的任何方式和工具，去设计他们的应用架构。另外，对于一些常见的通用功能，Flask 没有以某种特定方式去实现，这意味着在 Flask 中，对标准库的使用会比在其他框架里多得多，这保证了通用功能的稳定性及针对其他 Python 程序员的可读性。由于 Flask 社区十分庞大，所以该社区提供的添加通用功能的方式非常多。本书的重点之一，就是介绍扩展 Flask 的方法，了解它们如何帮助我们避免重复造轮子。这些扩展的妙处在于，如果不需要那些额外的功能，就不用引入它们，这样我们的应用才会保持简洁、轻量。

这种系统的主要弊端是，大部分新的 Flask 用户不知道如何正确地规划大型应用的结构，最后弄出一大堆让人难以理解和维护的代码。所以本书的另一个重点是如何为 Flask 应用创建模型/视图/控制器（MVC）架构。

MVC 体系在最初被发明时，主要用来设计桌面应用的用户界面。它将数据处理（model）、交互逻辑（controller）和用户界面（view）隔离成三层不同的组件。



这三层组件的隔离，使得程序员不用为每个网页重新实现一遍相同的功能，而是只要重用那些代码。比如，若是没有把处理数据的代码拆分到单独的函数中，那么我们不得不在每个页面的渲染函数里，把同样的数据库连接代码和 SQL 查询代码都写一遍。本书中大量的调查研究资料及 Web 应用开发中各种踩坑的第 1 手痛苦经验，使得本书成为目前关于 Flask 的最详尽、实用的资料。真诚地希望你会喜欢它。

## 本书涵盖的内容

第 1 章，入门，帮助读者应用 Python 项目的最佳实践方式，并搭建一个 Flask 开发环境。读者会得到一个十分基础的 Flask 应用框架，在整本书中都会用到。

第 2 章，使用 SQLAlchemy 创建数据模型，介绍如何把 Python 的数据库工具 SQLAlchemy 与 Flask 结合起来使用，为用户的数据库建立一套面向对象的编程接口。

第 3 章，通过模板创建视图，讲解如何使用 Flask 的模板系统 Jinja，在 SQLAlchemy 模型的支持下动态地创建 HTML 页面。

第 4 章，使用蓝图创建控制器，介绍如何使用 Flask 的蓝图（blueprint）功能来组织视图，同时避免代码冗余。

第 5 章，进阶的应用结构，结合在以上 4 章中学到的知识，详细讲解应该如何组织代码，以创建可维护性和可测试性都更好的应用结构。

第 6 章，保护应用安全，讲解如何使用各种 Flask 扩展，为每个视图添加基于权限的登录系统。

第 7 章，在 Flask 中使用 NoSQL 数据库，讲解什么是 NoSQL 数据库，以及在应用需要更强大的功能时，应该怎样将 NoSQL 数据库与之整合。

第 8 章，构建 RESTful API，介绍如何通过一种安全和易用的方式，把应用数据库中的数据提供给第三方使用。

第 9 章，使用 Celery 创建异步任务，讲解如何把成本高或耗时长任务挪到后台执行，让它们不拖慢应用。

第 10 章，有用的 Flask 扩展，讲解如何利用一些广受欢迎的扩展，让应用跑得更快、增加更多的功能，以及让调试变得方便、易行。

第 11 章，构建你自己的扩展，了解 Flask 扩展是如何工作的，以及怎样编写扩展。

第 12 章，测试 Flask 应用，讲解如何通过应用增加单元测试和用户界面测试，以保障程序质量，减少错误代码的数量。

第 13 章，部署 Flask 应用，介绍如何把已经完成的应用从开发环境中拿出来，部署到线上服务器中。

## 阅读本书前需要做的准备

阅读本书前，我们只需准备自己喜欢的一个文本编辑器和一个网页浏览器。另外，需要在机器上安装 Python。

Windows、Mac OS X 和 Linux 用户都可以很容易地学习本书的内容。

## 本书的目标读者

本书是为那些对 Flask 有一些了解，但希望把对 Flask 的理解程度从略懂提升到精通的 Web 开发者所准备的。

## 惯例

在本书中，我们会发现存在很多不同的文本格式，这些文本格式用于传达不同的信息。下面是一些不同的文本格式的例子，以及对这些文本格式所表达含义的解释。

文本中的代码关键词、数据库表名、文件夹名、文件名、文件扩展名、路径名、示例 URL、用户输入和 Twitter 用户名会这样显示：

“`first()` 和 `all()` 方法返回一个值，并且终止整个链式调用。”

一个代码块会这样显示：

```
class User(db.Model):
    id = db.Column(db.Integer(), primary_key=True)
    username = db.Column(db.String(255))
    password = db.Column(db.String(255))
    posts = db.relationship(
        'Post',
        backref='user',
        lazy='dynamic'
    )
```

当我们想让代码块中的某些部分吸引我们的注意力时，可以将相关的行或者内容以粗体显示：

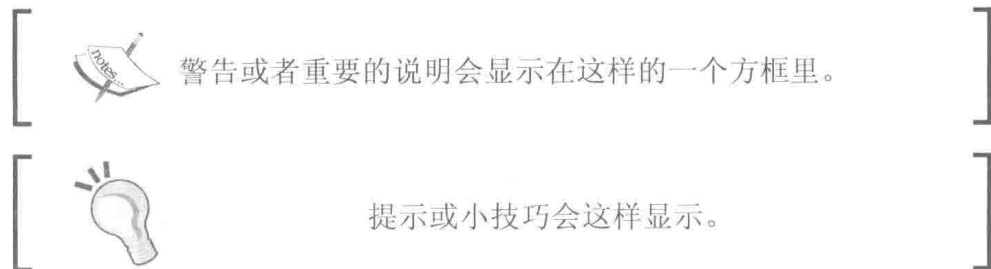
```
from flask.ext.sqlalchemy import SQLAlchemy

app = Flask(__name__)
app.config.from_object(DevConfig)
db = SQLAlchemy(app)
```

所有命令行的输入或输出会这样显示：

```
$ python manage.py db init
```

新引入的概念和重要的关键词以黑体显示。在屏幕上看到的文本，比如在菜单或者对话框中出现的文字，也会这样显示：“单击另一个写着**下载 Bootstrap** 的按钮，就会开始下载一个 Zip 文件。”



## 下载示例代码

你可以从 <http://www.broadview.com.cn> 下载所有已购买的博文视点书籍的示例代码文件。

## 勘误表

虽然我们已尽力谨慎地确保内容的准确性，但错误仍然存在。如果你发现了书中的错误，包括正文和代码中的错误，请告诉我们，我们会非常感激。这样，你不仅帮助了其他读者，也帮助我们改进后续的出版。如发现任何勘误，可以在博文视点网站相应图书的页面提交勘误信息。一旦你找到的错误被证实，你提交的信息就会被接受，我们的网站也会发布这些勘误信息。你可以随时浏览图书页面，查看已发布的勘误信息。

# 目录

第 1 章 入门 .....	1
使用 Git 进行版本控制 .....	1
安装 Git .....	1
Git 基础 .....	2
使用 pip 管理 Python 包 .....	4
在 Windows 上安装 pip .....	4
在 Mac OS X 和 Linux 上安装 Python 包管理器 pip .....	5
pip 基本操作 .....	6
用 virtualenv 的沙盒管理依赖 .....	7
virtualenv 基础 .....	7
开始我们的项目 .....	8
使用 Flask Script .....	9
总结 .....	10
第 2 章 使用 SQLAlchemy 创建数据模型 .....	11
设置 SQLAlchemy .....	11
Python 安装包 .....	12
Flask SQLAlchemy .....	12
我们的第 1 个数据模型 .....	13
创建 user 表 .....	15

CRUD	16
新增数据	16
读取数据	16
修改数据	19
删除数据	19
数据模型之间的关联	20
一对多	20
多对多	23
SQLAlchemy 会话对象的方便之处	25
使用 Alembic 进行数据库迁移	26
总结	28
<b>第 3 章 通过模板创建视图</b>	<b>29</b>
Jinja 的语法	29
过滤器	30
注释	34
if 语句	34
循环	35
宏	36
Flask 特有的变量和函数	37
创建视图	38
视图函数	38
编写和继承模板	41
Flask WTFForms	45
WTFForms 基础	46
自定义检验器	48

发布评论 .....	48
总结 .....	51
<b>第 4 章 使用蓝图创建控制器 .....</b>	<b>52</b>
请求的构建和销毁，以及全局变量 .....	52
错误页面 .....	53
使用类描述视图 .....	53
方法视图 .....	55
蓝图 .....	55
总结 .....	57
<b>第 5 章 进阶的应用结构 .....</b>	<b>58</b>
模块项目 .....	58
重构代码 .....	59
应用的工厂模式 .....	62
总结 .....	64
<b>第 6 章 保护应用安全 .....</b>	<b>65</b>
准备工作 .....	65
修改用户模型 .....	65
创建表单 .....	67
创建视图 .....	71
社交网络登录 .....	76
使用会话 .....	84
使用 Flask Login .....	87
用户角色 .....	90
总结 .....	95

第 7 章 在 Flask 中使用 NoSQL 数据库 .....	96
NoSQL 数据库的种类 .....	96
键值数据库 .....	97
文档数据库 .....	97
列式数据库 .....	98
基于图的数据库 .....	100
关系型数据库与 NoSQL 的比较 .....	101
关系型数据库的优势 .....	102
NoSQL 数据库的优势 .....	103
在什么情况下用什么数据库 .....	104
在 Flask 中使用 MongoDB .....	105
安装 MongoDB .....	106
配置 MongoEngine .....	106
定义文档 .....	107
CRUD .....	112
NoSQL 中的关联关系 .....	117
利用 NoSQL 的强大能力 .....	120
总结 .....	126
第 8 章 构建 RESTful API .....	127
REST 是什么 .....	127
构建 RESTful Flask API .....	130
GET 请求 .....	131
格式化输出 .....	132
请求中的参数 .....	135



POST 请求	138
身份认证	140
PUT 请求	143
DELETE 请求	146
总结	147
<b>第 9 章 使用 Celery 编写异步任务</b>	<b>148</b>
Celery 是什么	148
配置 Celery 和 RabbitMQ	150
在 Celery 中创建任务	151
运行 Celery 任务	154
Celery 工作流	155
监控 Celery	160
在 Flower 中通过网页进行监控	161
创建一个提醒应用	161
生成每周摘要	163
总结	167
<b>第 10 章 有用的 Flask 扩展</b>	<b>168</b>
Flask Script	168
Flask Debug Toolbar	170
Flask Cache	173
缓存函数和视图	174
缓存带参数的函数	175
缓存带有查询参数的路径	177
使用 Redis 作为缓存后端	177