

# Django

## 企业开发实战

高效Python Web框架指南

胡阳◎著



# Django

## 企业开发实战

高效Python Web框架指南

胡阳◎著



人民邮电出版社  
北京

## 图书在版编目 (C I P) 数据

Django企业开发实战：高效Python Web框架指南 /  
胡阳著. — 北京：人民邮电出版社，2019.2  
(图灵原创)  
ISBN 978-7-115-50689-4

I. ①D… II. ①胡… III. ①软件工具—程序设计—  
指南 IV. ①TP311.561-62

中国版本图书馆CIP数据核字(2019)第008453号

## 内 容 提 要

本书以从零开发一个博客系统为例，介绍 Django 在日常工作中的应用。

本书共分为四部分。第一部分介绍编码之前的准备工作，包括需求分析、Web 开发基础以及选型时 Demo 的练习。第二部分开始正式实现需求，介绍了环境配置、编码规范以及合理的项目结构划分。通过对 Django 各部分 (Model、Form、admin 和 View) 的介绍和使用，我们完成了一个基础的博客系统。第三部分在前面的基础上介绍 Django 第三方插件的使用，通过引入这些插件进一步增强我们的系统。最后一部分也是正式工作中必不可少的部分，包含调试代码、优化系统、压力测试以及自动化等内容。

本书适合 Python Web 开发人员阅读。

- 
- ◆ 著 胡 阳  
责任编辑 王军花  
责任印制 周昇亮
  - ◆ 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>  
北京市艺辉印刷有限公司印刷
  - ◆ 开本：800×1000 1/16  
印张：24.5  
字数：573千字 2019年2月第1版  
印数：1-4 000册 2019年2月北京第1次印刷

---

定价：99.00元

读者服务热线：(010)51095186转600 印装质量热线：(010)81055316

反盗版热线：(010)81055315

广告经营许可证：京东工商广登字 20170147 号

站在巨人的肩上  
**Standing on Shoulders of Giants**



[iTuring.cn](http://iTuring.cn)

站在巨人的肩上  
**Standing on Shoulders of Giants**



iTuring.cn

# 序 一

突然收到作者的邮件邀请,让我给新书写推荐序。由于之前对 the5fire 这个 ID 并没什么印象,所以按照老规矩加微信,一叙旧才知居然是 6 年前 BPUG 结下的善缘(详见 <https://www.the5fire.com/957.html>)。

BPUG(北京 Python 用户组)是 2004 年成立的 CPUG(中国 Python 用户组)的北京分会,从 2004 年到 2007 年共组织有 25 次线下活动,我发起并主持了其中绝大多数线下交流活动。全国各城市也分别成立了 Python 用户组,会不定期组织线下技术交流活动,至今已进行了总计 50 多场 Python 相关的活动。而本书作者就是参加了 2012 年那次 BPUG 活动后,坚定了从 Java 转变为一名 Python 行者的决心,进而积累成本书。

受邀写推荐序,我其实很纠结,因为 Python Web 开发有太多梗可以聊了,但又不好喧宾夺主,所以只能憋住,只说最憋不住的几点。

Django 在 2005 年发布其实也是个巧合,那之前有 all-in-one 的 Web 解决方案,而且异常强大;Zope/Plone 平台如日中天,但是无论学习还是开发部署都太重了。

而从 2003 年开始,堪萨斯城的 World Online 小组在维护一堆报社官网的过程中积累了大量最佳实践,并有意识地整合为快速可定制的 CMS 系统,直到从文档到工具链都成熟时才开源发布。

简单说,Django 和其他框架的不同在于,它是先有成功案例再发布的。这导致一开始,无论技术还是社区,Django 都异常务实。它贯彻了框架本身的职责:沉淀领域工程经验,指导/辅助程序员高效地交付工程。这和以往常见的模型/库(它们专注提供功能)是根本不同的知识凝结形态。可惜到今天,依然有哪种框架更加优秀的争论。刚刚在 CPyUG 列表中爆发的争议点为:

- Flask 集领域最优模块而成;
- 而 Django 大而全,却什么都没做好……

这其实也是新手在选择 Web 开发框架时最容易被误导的。

而本书出版得正是时候,通过长期工程中的 Django 使用体验,来展示为什么一个发布于 13 年前的框架今天依然可以站在 Web 开发一线。

## 背景

早在 2009 年出版的《可爱的 Python》一书中，我就对纷繁的 Python Web 应用框架进行了讨论。那之后，有关 Django 的中文图书并不多，从 z.cn 上也只能查到 5 本。

但 Django 从发布之初，就一直是最活跃和发展最好的 Web 框架。在其他大型框架逐渐不活跃后，它以其绝对丰富的产品线以及生态环境，成为 Python 世界中商业网站开发的首选框架。但是，为何对应图书出版这么不活跃？

答案可能很囿：因为 Django 官方文档太完备了，几乎一切问题从文档中都可自行解决，几乎没有什么特殊知识点需要用图书来解决。此外，Django 社区也极其活跃和友好。甚至对于想入门的妹子们，都有对应的 Django Girls 国际品牌活动。每年会组织上千场全天免费的培训活动来吸引有一定基础的程序媛从其他技术栈迁移到 Django 世界。（对了，在 PyCon 大会上，Guido 老爹还专门点名赞扬了这一针对萌妹子服务的活动。）

## 触动点

作者的立意很清晰：

降低学习 Django 时的心智负担，主要针对想学习 Django 但又无从下手，或者看了很久文档能完成新手教程，但想自己开发一套系统时却无从开始的人。

即使官方文档将技术问题详尽陈列在前了，上手具体工程时依然存在问题。

简单来说，官方文档就像学校传统计算机教育或是几乎所有在线编程课程一样：只说了语言/框架本身，并没有涉及如何编程或者怎样进行工程实践。

即使读者确定自己需要的知识点都在官方文档中，也根本无从判定应该从哪里开始，以及在不短的时间里以什么路径从头构建一个稳定可持续发展的 Web 应用系统。这也是作者为什么在正式构建博客之前，用了三分之一的篇幅详细阐述了 Django 商用工程的前期筹备以及技术栈。读者只有穿过 PM、UML、WSGI、Flask、Tornado、DBA、IDE、virtualenv 等一系列成熟技术后，才能开始构建真正可用的博客系统原型。否则，用 5 分钟完成的原型在有点儿规模的协同过程中，一定会崩溃。

说回开头提及的那场争议，经过几十封邮件的交战后，有位知名不具的老司机总结了开源世界的共识或者规则，具体如下。

选择最适合自己的工具组成工具链。如果这个工具链不能适合几个场景，那么这几个点是不是很痛？不痛就可以忍。如果很痛，是否可以通过第三方补丁/插件改进？可以的话就用上补丁/插件，然后承担一定的非标准代价。如果没有补丁/插件，是否可以通过修改解决？如果可以自己改，然后看是不是可以放出去。如果很痛并且没有第三方插件，自己修改代价又很高，那么重新选择适合自己的工具链。如果真的没有合适的工具链，那么无非是以下两种情况。

- 用户数很多：恭喜，你撞到了一片蓝海，即发现有个场景用户数很多（或者可以预见会很多），竟然还没合适的解决方案，这绝对是一个创业的好机会。
- 用户数很少：恭喜，你发现了一个无人来过的狭小领域，攻克不得不解决的问题很可能是你将来的核心竞争力。

但是大多数人在面对软件系统的时候，实际上是以闭源软件的思路来解决问题的。

- 把这个问题给我解决了！
- 要多少钱？

这种思路其实无可厚非。每家公司都要集中解决自己的核心问题，而不是先解决非相关问题，例如框架。但是如果将这个思路用到开源社区中，就不大好使，这时就需要有对应的商业公司来解决两者间的鸿沟/商机。MySQL 和 Django 当年都不约而同地选择成立基金会，用商业来维持社区发展，就是意识到开源带来的活性可以令自己活得更久、更好、更强。本书作者也一样，公开自己的经验不仅可以加深自己的理解，还可以回馈 Django 这一伟大社区。

其实“降低学习 XXX 时的心智负担”这类通用解决方案是有的，就是每一位自学成功的开发者都具备的习惯。

以最小可用作品为核心的持续迭代，即在第一个 42 分钟就完成构建并发布一个几乎没有功能但是可运行的 Django 网站，然后持续迭代，在一直可运行的基础上一点一点追加功能，在这个过程中保持开发→调试→部署→运营的循环过程。接着要确保任何一个新知识点可以立即并入运行中的功能网站，这就可以依据一个健康的开发过程持续积累知识树，同时有可视、可用、可发布、可演示、可追踪……活的真实项目来印证。

当然，这样一来，整个图书的结构就不得不包含大量重复叙述，无法形成流畅的阅读体验。我期待作者在后续电子版本中能以这种实战自学的过程为线索来写。另外，这也是 2008 年我提出的 PythoniCamp 自学模型，用于专门构建对初学者友好的课程。配合这本对有工程经验者更加友好的图书，形成 Django 实效系列图书/课程。

综上，作者从大学时代开始独自挣扎了近 10 年，终于通过个人努力完成赛道升级，进入更大的平台，开始更高、更复杂、更强技术的求索。本书是一名靠谱工程师的私人经验集成，作者通过样例仓库、私人博客、公众号等渠道，期待和读者持续交流。本书的内容也将随着读者的加入持续增订，期待慢慢变成 Django 世界上最可靠的企业工程入门资料库。而购买本书，将成为你加入这一丰饶世界的门票。



## 序 二

Python 拥有众多 Web 开发框架，但这里面我最喜欢的无疑就是 Django。首先，它提供了完整的项目组织实践，让团队协作变得更容易。其次，它提供了丰富的功能。关于这一点，可能很多人会提出质疑，因为他们喜欢更轻的框架。但是实际上每个项目都是复杂的，使用比较轻的框架时，最后还会安装一堆质量参差不齐的第三方库，这比使用 Django 还要臃肿。并且，到目前为止，它是组织最完善的 Python 框架，成立了自己的基金会，这为其长远发展提供了保障。另外，经过多年的发展，诸如 Instagram 等知名的互联网公司都在使用 Django。可以说，Django 没有明显的 bug，用在自己的项目中也更为放心，同时性能方面的优化方案也有更多的参考依据。

Django 有诸多优点，并且功能也比较多。市面上相关的英文书有很多，但中文书寥寥无几，所以看到有人写这方面的书，我感到很开心。在成书的过程中，本书作者与我有过邮件沟通和线下交流。作者在工作中深度使用 Django，书里凝结了他日常工作中心得，内容更接地气，甚至可以直接应用到自己的工作中。

最后用一句比较俗的话结尾：预祝本书大卖。

清风，连续创业者

# 前言

自 2011 年，我从 Java Web 开发转到 Python Web 开发，到今天已经有 7 年多了。一开始，我就是从 SSH ( Struts+Spring+Hibernate ) 框架转到 Django 的。当时的第一感觉是这玩意儿太轻便了，比 SSH 好用太多了。但熟悉了 Python 社区之后，我发现 Django 已然算是 Python 社区中一个比较重的框架了。说它重的主要原因在于它的定位是企业级开发框架，或者说全功能的 Web 开发框架，也因为它的“通过更少的代码，更快速地开发 Web 应用”的口号。

在使用 Django 的几年中，我和同事们做了  $N$  多个系统，有对内的，比如 CAS 权限管理后台、新闻系统后台、电影内容系统后台、游戏内容管理后台、投票系统后台和运营相关系统等；也有对外的，比如游戏中心 H5 Web 版、用户留言系统、H5 活动页系统和交易相关系统等。

我们开发过的所有系统，无论是从开发速度还是上线后新功能的迭代，Django 都能很好地满足需求。所以我就在考虑，能不能总结出来一些对大家有帮助的东西，主要面向想学习 Django 但又无从下手，或者看了很久文档，虽能完成新手教程，但想自己开发一套系统时却无从下手的人。这就是我写这本书的第一个原因。

第二个原因是，我从 Java 转到 Python 之后，是通过 *The Django Book* 一书的在线翻译版学习 Django 的。对于快速学习来说，中文版无疑是最好的选择。当时 Django 的版本是 1.3，而那本书使用的版本是 1.1。虽说当时 Django 1.1 没有太大的版本差异，但是时至今日，Django 的最新版已经是 2.0 了，整体结构以及功能都有了很大的变化，我发现依然有人在看那本书入门。因此，应该有新的教程出现，帮助初学者快速入门。

第三个原因是，我在 2012 年很幸运地通过了面试，从小公司跳槽到搜狐，但入职第一年的工作内容让我意识到在上家公司一年左右使用 Django 的经历并没有给我增加太多实战经验。我在 2012 年的年终总结里写道：

主要是因为之前的工作都是开发阶段的，用的也都是 Django 的皮毛，对部署和环境配置等都没有概念。

所以我希望为那些刚开始学习 Django 的人提供比较完整的内容，并且这些内容是从实际工作中总结出来的。

基于上述三个原因，也就有了这本书和之前录过的一套视频 ( <https://st.h5.xiaoe-tech.com/st/57fHzIRWE> )。希望我的经验对你有所帮助，使你不至于在使用了 Django 一年后还处于我当时的状态。

欢迎访问：电子书学习和下载网站 (<https://www.shgis.com>)

文档名称：《Django企业开发实战》胡阳.pdf

请登录 <https://shgis.com/post/3357.html> 下载完整文档。

手机端请扫码查看：

