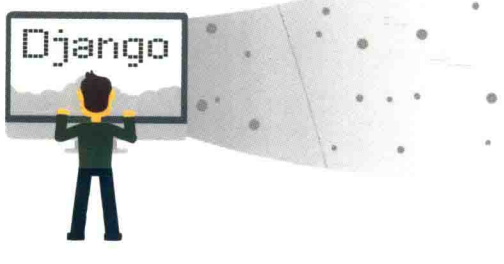
TURING 图灵原创

Django 企业开发实战

高效Python Web框架指南



胡阳◎著

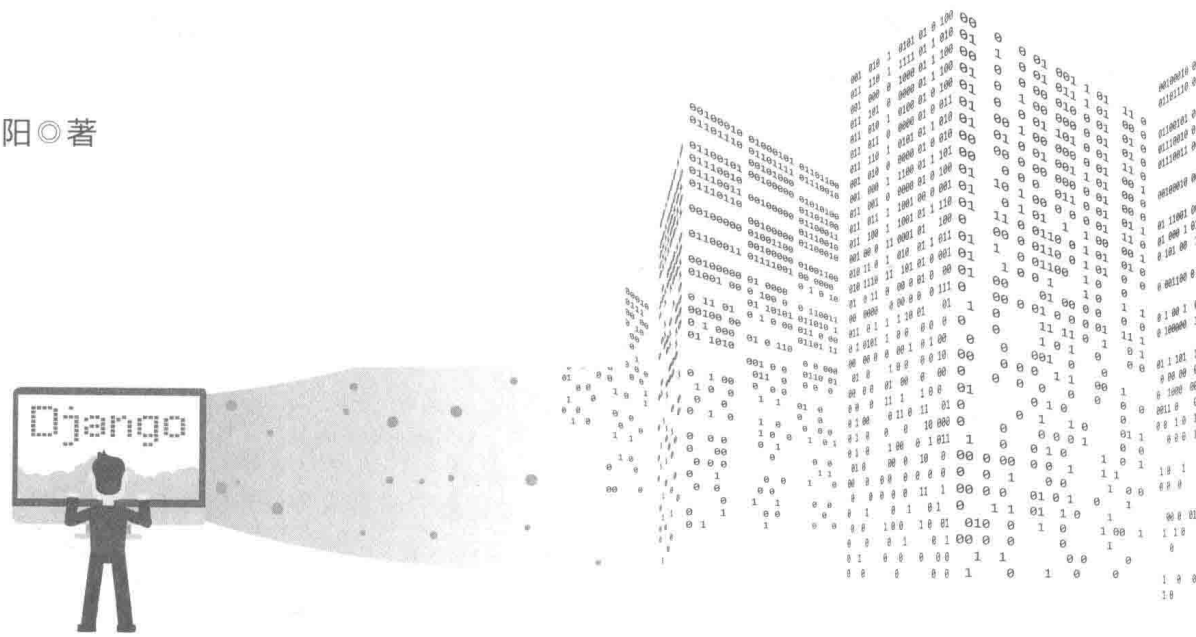
中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵原创

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（CIP）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

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？

答案可能很囧：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技术活动。）

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 世界中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。

Zoom.Quiet，uSEE.tech CTO

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，让团队协助变得更容易。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的心得，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 开发转行到 Python Web 开发，到今天已经有 7 年多了。一开始，我就是从 SSH（Structs+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“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 的最新版本已经是 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/57fHzlRWE）。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。

谁应该来读这本书

本书面向所有对 Python Web 开发感兴趣的人，无论你是刚开始学习编程的学生，还是刚入职场的程序员，抑或是有多年编程经验、打算转换自己技能栈的老程序员。

本书的目标读者概括如下：

- 学完 Python 基础、想要继续学习 Web 开发的人；
- 想要使用 Django 快速开发日常业务系统（商业 Web 项目、自动化运维系统等）的人；
- 有其他语言 Web 开发经验、想要转到 Python Web 开发的人。

哪些公司或产品在使用 Django

基于 Django 开发的比较有名的产品在 https://www.djangoproject.com/start/overview/ 上可以看到，其中有很多我们耳熟能详的产品，比如 Instagram、Disqus、Pinterest、Mozilla、Sentry 和 Open Stack 等。

国内使用 Django 的公司或产品有搜狐、奇虎 360、豆瓣、今日头条、妙手医生、闪银奇异、蚂蚁金服等。

生态

Django 之所以被广泛应用，除了本身提供了完备的功能之外，也得益于它的成熟生态。这一框架本身没有覆盖到的功能，一般都会有优秀的第三方插件来补足，比如用来做 RESTful 接口的 django-rest-framework、用来调试性能的 django-debug-toolbar 和用来搜索的 django-haystack 等。更多的第三方插件，可以在 https://djangopackages.org/上查看。

学习曲线

相对于 Flask、web.py 和 Bottle 这一类微型框架来说，Django 上手确实有点复杂，但也并不难，因为官网的新手指导写得很清晰。在众多框架中，Django 的文档算是相当优秀和完整的了。

你需要花费更多的时间来学习 Django，这是因为它提供的内容远多于其他框架。刚开始，可能会觉得很多地方不明白，但是等你熟悉了之后就会发现，在 Web 应用开发上，Django 已经提供了很完备的支持，比如登录认证、权限管理、admin 管理后台、缓存系统、常见的 Web 安全防御等。Django 每一层或者模块所提供的功能都很清晰，比如什么样的需求在哪一层来处理，或者在哪个模块中处理。

Django 一开始的学习曲线有点陡，然后是平缓上升的。先陡主要是因为新手需要一下子接受

很多东西，但是随着后面不断使用和了解，你会发现，学习所耗费的时间完全值得。你可以更快地做出完善的系统，这会是一笔很划算的投资。

本书目的

本书的初衷前面也提到了，就是把我知道的关于软件开发的知识、实际项目开发中总结到的经验，以一个 Blog 系统为例写出来，让后来者可以参考我的经验快速成长，同时也可以搭建出自己的试验场。

本书内容

本书共分为四部分，分别为：初入江湖、正式开发、第三方插件的使用、上线前的准备及线上问题排查。

每一部分着重介绍一个大的主题：第一部分介绍正式进入编码之前的一些工作；第二部分在第一部分的基础上开始实现需求，进行编码，直到完成需求；第三部分重点介绍我们常用的第三方插件，如 xadmin、django-autocomplete-light 和 django-rest-framework 等；最后一部分介绍调试、优化、自动化部署以及压力测试等内容。

每一部分中的每一章专注介绍一个比较小的主题，像第 5 章主要介绍模型中涉及的内容，比如各种字段的介绍、`QuerySet` 的使用和优化以及 ORM 的概念等。

每一章的内容完成后，代码都可以正常运行，这样你可以快速看到运行结果。此外，每一章也都是基于前一章的结果来进行开发的。

本书对应的源码[①]放在 https://github.com/the5fire/typeidea，其中每一个分支对应每一章最后的源码。比如，第 5 章的源码是 book/05-initproject 分支下的代码，其他的以此类推。

即使 GitHub 上提供了源码，本书还是呈现出了完整的代码，建议读者根据内容进行编写，同时也要注意书中提到的一些需要自己独立完成的作业。

为什么选择一个 Blog 系统

选择写一个 Blog 系统，是因为绝大部分人对这个系统/业务都很熟悉。实际上，在工作中我们经常需要开发一些从来没接触过的业务，所以把业务落地（从开发到部署上线）比起曾经开发过什么系统更为重要。

这里选择一个大家都熟悉的业务，是为了降低大家理解业务的难度。用一句话来说的话，那就是：降低学习 Django 时的心智负担。

① 本书源码也可从图灵社区（iTuring.cn）本书主页免费注册下载。

“理解”这个词很重要。无论是设计语言、设计框架、设计项目结构，还是设计产品流程，都需要重视这个词。此外，也需要注意降低用户（团队）的心智负担。关于“心智负担”这个词，这里不再做过多解释，只强调一点：对于正式项目开发来说，降低开发人员维护代码时的心智负担，就是减少复杂度、降低风险的方式。

另外一个原因前面也提到过，每个开发者都需要有自己的“试验场”，这个观点我在书中也会多次提到。技术的世界变化非常快，比方说今天我在公司项目的技术选型上使用了 Django 1.11，那意味着在后续的维护过程中不太可能会升级到 Django 2.0 或者更高的版本。因为在成本上不允许这么做，但你会因此就放弃对新技术的追求和实践吗？当然不会，我们可以通过其他方式来获得新技术的实践经验，进而在某个时刻把这些经验引入到商业项目中来。

所以，即便现在有各种博客平台和公众号平台供你写下自己的技术总结、思考，你可能也需要一个能够随时实践新技术的“试验场”，来让你获得工作之外的经验点。

致谢

感谢我的父母，没有他们，也就没有我。

感谢搜狐，我在这个平台上学会了很多，收获了很多。

感谢一起共事过的同事们，跟你们一起编码、讨论的经历，催生了这本书。

感谢 Adrian Holovaty 和 Simon Willison 创造了这个优秀的框架。

感谢尹吉峰、董伟明、李者璈（Manjusaka）对本书的审校、建议和短评，让我能进一步完善本书内容。

特别感谢大妈（Zoom.Quiet）和清风老师给本书写推荐序，感谢他们给我提的各种建议（关于图书编写、编程规范、编码细节等非常实用的建议）。也非常感激大妈 2012 年组织的 BPUG，会上清风老师和其他众位前辈的分享对当时刚转入 Python 开发的我来说影响重大。

最后，感谢我的爱人和女儿。没有我爱人的支持以及对我和女儿的细心照料，我就不会有时间来完成工作之外的其他兴趣（如写书、写博客、录视频等）。感谢我的女儿给我带来不一样的世界，她每一天的成长都是肉眼可见的。对我而言，她的学习能力是我需要借鉴的，这不断督促我去学习更多新东西。她的成长是对时间很好的展现，这让我对时间有了更直观和深刻的认识。

勘误和反馈

鉴于我的经历和视野有限，书中错误之处在所难免，敬请读者指正。欢迎到 GitHub 上给我提问题，网址是 https://github.com/the5fire/django-practice-book，或者发邮件到我的邮箱 thefivefire@gmail.com（注明：《Django 企业开发实战》勘误），这里先行谢过。

如果你对本书的内容或者组织方式有任何疑问，也欢迎到我的博客（https://www.the5fire.com）上或者通过微信公众号（Python 程序员杂谈）给我留言。

目　录

第一部分　初入江湖

第二部分 正式开发