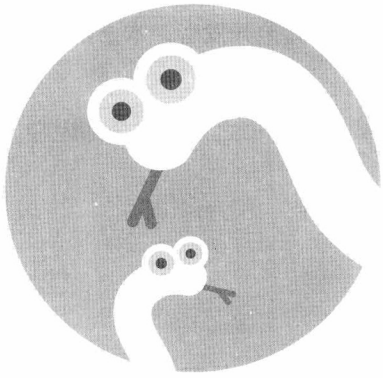
Broadview®
www.broadview.com.cn

来自豆瓣一线开发者的工程实践

Python Web 开发实战

董伟明 著

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn



Python Web 开发实战

董伟明 著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书按照一个 Web 产品从无到有、从简单变复杂、从基础到进阶的过程，多角度、全方位讲述了 Python Web 开发。内容涉及 Web 框架、测试、数据库、消息队列、服务化、持续集成等，把网站工程的全貌展现在读者的眼前，从其中可以了解 Web 工程从开发到上线的完整流程。另外，作者对当前现在正在流行的技术或工具，如 Flask、Celery、Jupyter、Supervisor、SaltStack、Pandas 等都有较为详细的阐述，可作为技术选型时的参考。

对于 Web 开发者、使用 Python 语言的运维工程师和运维开发工程师、想提高 Python 技能的开发者、想了解 Python Web 开发的其他开发者，本书都适合阅读。

图书在版编目（CIP）数据

Python Web 开发实战 / 董伟明著.—北京：电子工业出版社，2016.9
ISBN 978-7-121-29733-5
I. ① P…II. ① 董…III. ① 程序开发工具 – 程序设计 IV. ① TP311.561

中国版本图书馆 CIP 数据核字（2016）第 200131 号

责任编辑：许　艳
印　　刷：北京京科印刷有限公司
装　　订：北京京科印刷有限公司
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：31.5　　字数：616.9 千字
版　　次：2016 年 9 月第 1 版
印　　次：2016 年 9 月第 2 次印刷
定　　价：105.00 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 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。
本书咨询联系方式：（010）51260888-819　faq@phei.com.cn。

推荐序一

作为一名使用 Python 作为主力开发语言十多年的老码农，常常被人要求推荐 Python 相关的图书。经常推荐的都是一些讲解 Python 语言本身的图书，而专注在 Python 的常见应用领域—— Web 开发上的好书，却一直是凤毛麟角。也曾有出版社的朋友约我写一本，但我畏惧写书的艰辛，一直不敢答应。得知伟明的《Python Web 开发实战》一书即将出版，欣慰异常，心想伟明写这个专题实在是再合适不过，必为佳作。读完书稿，果不其然。

由于 Python 具有开发快速、适合多人协作、库丰富、社区成熟等优点，因此是一门非常适合用于 Web 开发的语言。国外的 Youtube、Quora，国内的豆瓣、知乎等，均是以 Python 为主要语言开发的。说起 Python 的 Web 开发，很多人会理解成只要学会某个 Web 框架，能写代码查查数据库，写模板渲染出 HTML，最多再学一下配个 Web Server 把服务启动起来就行，没什么内容。多数 Python 书里“Web 开发”的章节一般也就是讲这些。但其实要完成生产可用的，能够应对一定规模访问量的 Web 系统，Web 开发工程师要学会的远远不止这些。环境搭建、API 设计、网站架构、系统管理、持续集成、服务化、数据处理、并发处理等，这些都是必要的，而且需要付出大量的努力才能掌握的知识。

伟明一直对技术抱有极强的兴趣，也有着优秀的动手能力。我对他的第一印象是从他发给豆瓣的求职信开始的：“目前我给 glances、Salt、tmux-powerline、supervisor、mongo-python-driver、circus、graphite-web、Diamond、autopep8、graph-explorer、pip、Celery 等开源项目贡献过代码，给 Python 标准库 logging 贡献过代码。”能够参与到这么多一线开源软件中的应聘者，确实少见。入职豆瓣后，伟明也表现出了对 Web 开发的深刻理解，很快成为豆瓣多个 Web 产品的主力，并几乎凭一人之力完成了 CODE 项目的私有依赖剥离和开源。

伟明把他个人多年 Web 开发的经验，以及豆瓣十年来数百名优秀工程师在 Web 开发上最佳实践的积累，凝聚在了《Python Web 开发实战》这本书里，多维度、全面地介绍了 Python Web 开发涉及的各种技术。更难能可贵的是，他还在这本书里留下了非常多关于这些技术

的思索：为什么要使用某个技术？某个需求都有哪些技术可以满足？如何取舍？这种不仅要知其然，还要知其所以然的态度，是工程师快速成长必备的。

这样的一本书能够出版，对于国内的 Python 开发者绝对是个福音。我向所有有兴趣使用 Python 做 Web 开发的开发者们，强烈推荐此书。

洪强宁

爱因互动 CTO
前豆瓣首席架构师
前宜信大数据创新中心首席架构师

推荐序二

这篇序酝酿了好几天，今天终于动笔写下了第一个字。说实话，很久没有看到关于 Python 的好书了，尤其是国人自己的原创书。Web 开发本身就是一件很庞杂的事情，模板渲染、API 的开发、后端的部署，能在一本书中把这些问题都说清楚并不容易。作者伟明与我都出身豆瓣，虽然没有同时期在豆瓣工作过，但豆瓣的 CODE 系统把我们俩联系到了一起。他是 CODE 的三代目，通过 CODE 里一行行 Python 代码，仿佛知道了彼此的心意。感谢伟明把豆瓣的一些工程实践进行了整理和总结，这是本书最宝贵的一点。而本书的精华在于他对各种技术使用场景的探讨：那些库谁都会用，但在什么场景使用，在生产环境中这个库的表现到底如何，则不见得有认真的思考。期望将来还可以看到越来越多这样的图书，祝此书大卖。

清风

SAY CEO

前豆瓣技术总监

推荐序三

一次真诚的倾诉

自从 CPyUG 列表订阅人数超过 10,000 以后，我就认为中文的 Python 学习资料足够多了，而最正确的自学姿势应该从官方文档开始。但是，《Python Web 开发实战》一书又改变了我这一偏见。

有道是："出版是最好的记忆"，伟明亲身证实了这一点。作为一名普通的程序员，只从一个念头出发，独自写出了 500 多页的技术图书，这实在是一件令人敬佩的事。凡是写过书，特别是写过技术图书的人都知道——写书，难的不是写什么，如何写，而是要耐得住寂寞！

在中国生活原本就如此艰辛，无论上学还是工作，周围永远有无数同侪在竞争；而写书几乎是负收入的经济行为，特别是技术图书。当然，图书印刷出来，奉上对家人的感谢，是一种别样的程序员式的浪漫情怀，这种成就感不足为外人道矣。

伟明打动我，让我为他写推荐序，就在于他说自己写书的那个简单的初衷：让公司所有人都知道 Web 开发究竟是什么样的，从而能更好地协同。这其实已经是佛陀流传经文普度众生的大情怀了！

Python 是如此活跃的一种语言，几乎伴随互联网的发生而创立，又伴随互联网的极速发展而繁荣，在 Web 开发领域拥有全栈式的技术生态；又因为脚本语言以及其自身语言的人性化设计，通过 Web 勾联到了几乎所有计算机应用领域，这也导致在特定问题上，Python 总是有一堆解决方案可以选择，而不像其他语言，只有一种方案。但是选择过多，其实也导致了学习成本的增加。

伟明将自身在 Web 领域所有方面的经验提炼后整理成书，本质上是将几十个关联产品的官方文档，结合具体工程经验进行了梳理，给出了领域问题最佳方案的关键思考点和自己的

答案，而更加可贵的是，给出了这些思考点的来源，以及形成过程，即给出了解决各种 Web 领域问题的思维模式。

从前后几个版本的书稿也可以看出，如果没有这本图书的艰苦整理，伟明自己也难以形成这种宏观 + 微观能同时作用的思考模式。所以，我一直认为："输出是更加残酷的输入。"要将纷繁零散的经验，变成他人可以习得的技能，要组织成叙述合理、案例得当、结构清晰的图书，这个过程本身就得对自己的所有积累进行反复的再学习、解析和增补。其中的工作量远远不止这几百页书的内容。

更加奇妙的是，在没有这部书稿前，其实我们并不熟悉，只是在社区列表中见过邮箱名而已。但是，有了独有的知识成果后，伟明就有了立场，也有了动机和理由，邀请我以及类似洪教授/Limodou 这些中国 Python 学习者的前辈来评点和审核书稿，获得直接的联系，即人脉。

所以，我在郑重推荐此书的内容之外，更加倡议大家向伟明学习——敢于写书，通过真诚的技术图书总结自己的过去，获得更好的未来，帮助更多的 Pythonista。

Zoom.Quiet（大妈）

优视眼动科技 CTO
Python 中文社区创始人之一及管理员
OBP 及蟒营工程设计者兼主持人

Zoom.Quiet（**大妈**）

从 2002 年开始接触 Python，积极推广 Pythonic，筹办了自 2012 年起连续四届 PyCon 中国大会，编撰有《可爱的 Python》等图书。作为大家熟知的社区"大妈"，主持了 OSTC 2015"程序媛专场"，坐实了这一称号，得到广大程序员认可。

推荐序四

说起来给《Python Web 开发实战》一书写序还真是很突然。2016 年 5 月 30 日，我突然被拉到了一个微信群里，正觉得纳闷的时候，看到群里 Zoom.Quiet 的介绍，才知道是怎么回事。原来《Python Web 开发实战》已经基本成书，让大家看一看。对于本书的作者董伟明，我们没有在线下交流过，但是对 Python 的热爱时不时地会把大家通过某种方式吸引到一起。

这是一本原创图书，从书名来看是和 Web 相关的，而 Web 领域正好和我的兴趣以及平时的工作相关。作为一个开源 Web 框架的开发者，自然对 Web 开发的内容比较感兴趣，借由此书正好可以了解一下别人是如何理解以及如何实践 Web 开发的，更何况作者还是豆瓣的工程师，因此对书的内容还是有一些期待。

经过一番阅读之后，我与伟明交流了一些看法，他给予了详细的解释与说明，我对他的写作思路也有了一些了解。对 Web 开发的理解其实可以有很多角度，比如，从开发者的角度，这就会更多从具体的功能实现、框架使用来看待；从运维者的角度，会更多地从部署、维护、平台的角度来理解；从测试及质量的角度，会关心代码的测试性及代码审查；从框架开发者的角度，就要了解 Web 开发涉及哪些领域，每一领域应该用什么技术与工具来组织，不同领域又如何通过某些框架来有机地结合在一起。仅凭一本书，想完全满足所有人的需求是非常困难的。

阅读本书，我最大的感受就是：全和新。

全指的是内容覆盖面较广。原本我以为作者会主要讲 Flask 框架的开发，但其实 Flask 框架在本书中的比重并不大，反而是与 Web 相关的开发技术的介绍占了大部分的篇幅，甚至也包含了部署以及 Python 本身的一些特性和工具。对此我也有疑问，并向作者咨询。Web 开发的概念其实太大了，不同的角度可以有不同的理解。比如我们常说的 Web 框架，其实绝大部分都只涉及展示相关的开发，所以应该更精确地称之为 Web 展示框架或 Web 应用框

架。但是它很有可能依赖底层的批处理、大数据处理等技术，这些虽然不能算纯粹的 Web 技术，但是却可以放在 Web 开发这一概念下。因此如果把每一块与 Web 开发相关的内容都写出来，那么本书的厚度就可想而知了。所以作者是从个人实践的角度出发，把他所理解的与 Web 开发相关的技术尽可能全地，并且尽可能用更多的实例来讲述。之所以我会有“全”这个感受，因为书的内容涉及了 Web 框架、Ajax 的前后端交互、测试、数据库、数据分析、服务化、部署、系统管理、常用工具等内容，有点百科全书的意味。

为什么说“新”，因为书中讲的许多东西都是现在正在流行的技术或工具，像 Flask、Celery、Jupyter、Supervisor、SaltStack、Pandas 等。其中有些我还是第一次接触，说明作者平时接触的内容的确非常丰富，同时也结合了豆瓣的一些具体的实例，这样会更有借鉴意义。

全书的难度不是很大，内容广泛全面，不过因为篇幅所限，对于前端的技术介绍得不多，有些章节可能描述也不是太细。不过前端技术虽然也算是 Web 开发技术，但是与 Python 的关系就不那么紧密了，本书毕竟是一本 Python 相关的书，所以涉及不多也是正常的。而且许多具体的技术本身内容都很丰富，也绝不是短篇幅可以说清楚的，所以反而有个基础性的介绍，在需要时自行学习可能更好。因此本书比较适合对于 Web 开发有一定了解，但是希望了解更多 Python Web 开发技术的读者。

非常感谢作者把自己的经验分享给大家。

李迎辉

Python 开源资深行者
Python-CN 邮件列表创建人
UliPad 和 Uliweb 作者

业界热评

本书从 Python 开发开始，循序渐进，把网站工程的全貌展现在读者的眼前，是了解 Web 工程从开发到上线完整流程的绝佳参考书籍。同时书中的很多实例取自豆瓣工程开发团队的实际工作，对于想了解豆瓣内部技术实现的朋友，也有很大的参考价值。

邢霖（CNBorn）
前豆瓣东西技术负责人

开卷有益，已经很久没有看到原创的有价值的 Python Web 开发书籍了。很多刚进入 Python 世界的人，想要在 Web 开发上有更多的发展，但却不知如何往下学习。伟明的这本书提供了一个非常好的“知识地图”，书中涉及了 Python Web 开发的方方面面。与此同时，对于那些已经在 Web 开发上积累了一些经验，想要更进一步学习的人来说，这本书也能让你收获满满。我阅读完书稿也有了不少收获。书中涉及的知识点非常多，任何一个点都可以单独写成一本书。作者根据自己的经验积累，提炼出干货，略去了基础的部分，这对于读者来说也是幸事，不然你可能得抱一个大部头的书回去了。最后需要说的是，在 Web 开发的道路上，这本书是不错的进阶指南。

胡阳（the5fire）
Python 程序员
目前就职于手机搜狐网
任资深开发工程师
负责 m.sohu.com 网站的前后端开发和维护

董伟明是我见过的实践和执行能力超群的工程师。这本书从开发环境的搭建，Web 框架的使用，到最后的持续集成和 Python 的进阶用法，无一不是他多年的实际工程经验总结，十分宝贵。如果你刚开始学习 Python，这本书能给你展示 Python 的方方面面，让你可以快速

进入实际的 Web 工程的开发。如果你已经使用 Python 多年，这本书也能让你学习到 Python 的很多使用技巧。

姚钢强（acmerfight）
知乎工程师

这本书非常全面地介绍了使用 Python 进行 Web 开发的方方面面，既有 Web 框架、缓存、消息队列、并发处理的场景介绍和技术选型，又有开发流程、质量保证的丰富实战经验。作者通过非常细致的 Step by Step 教程，一步一步揭开了 Web 开发的神秘面纱，不管你有没有 Web 开发基础，相信都能从这本书中获益良多。

蔡斌（VeryCB）
DeepDevelop 工程师
前豆瓣条目组技术负责人

本书适合有一定 Python 和 Web 开发基础的用户。书中没有对语言基础的讲解，更多的是对 Web 方面的专注。内容很丰富，基本上覆盖日常 Web 项目开发中涉及的各个层级，对相关概念和原理的描述十分详尽，而每个示例代码都进行了分段解释，清晰明了。

正如书名，整本书都是作者对实际 Web 项目中大量实战经验的总结，绝非纸上谈兵。相信通过阅读该书可以帮助开发者规避掉大量项目中的“坑”，构建出更高性能、更稳定的 Web 项目。

强烈推荐从事 Web 开发的 Pythoner 阅读。

Spawnris
腾讯工程师

前言

为什么写这本书

2011 年，我还在一家互联网商务公司做一枚小小的运维工程师，那时公司的运维使用的语言主要是 Shell。我本来是一个网络运维，后来在工作中开始接触 Shell，学了 2 个月之后，感觉可以应付各种需求，虽然程序运行得很慢，但作为一个工作不久且不是 IT 相关专业毕业的运维人员，我还是有点沾沾自喜。这种情况只持续了 2 个多月，由于公司高速发展，一个新入职的同事打破了我的美梦。

这位新同事入职的第一件工作就是对接各个业务部门的日志需求。很快就发生了一件让我特别震撼的事情：同样的一个日志需求，我使尽浑身解数用 20 行 Shell 写好，运行一次要 20 分钟，而这位同事使用 Perl 语言的脚本只用了 4 行，运行 3 分钟就完成了。可以想象我当时的感受。这是我第一次了解到选择正确的工具和方法的重要性。我抑制不住地告诉当时的领导悦秋：我要学一门运维使用的高级语言！

悦秋特别郑重地告诉我：一定要学 Python。而在 2011 年，Python 还只是一门小众语言，在 BAT 等大公司招聘时仅作为一些职位的附属要求。回想至此，非常感谢悦秋让我选择了正确的路，否则我现在可能只能写关于 Perl 语言实践方面的书了。

从运维到运维开发，再到豆瓣做产品开发（也就是 Web 开发），一路走来我发现，自己走了很多弯路，没有人告诉我什么是对的，什么是错的，该怎么做选择。这些都得自己花时间去琢磨和验证，有时候从 Google、StackOverflow 搜索答案，或者在 GitHub 直接看源码获得灵感甚至正确答案，而涉及职业规划、该学什么、怎么学，这样的问题除了悟性大部分就是靠直觉了。

从买书看基本语法和拷贝别人的代码开始学习 Python，很早我就开始努力让代码符合 PEP8，尽量让代码写得 Pythonic（这点很关键，未来就不再需要费力改正学习过程中留下的坏习惯了）。能用 Python 完成日常工作之后，我开始研究和寻找各种 Python 高级玩法、黑魔法。这个时候我还是在不断买书、看书、看之前买的书、看一些技术博客来巩固和补充自己的知识体系。所谓“技术”中的“术”也就到这里了。

有了“术”还远远不够，还需要有实际的经验，以及在正确的时机使用正确的工具和方法，这是“技”。“技”是一套分析并解决问题的思路，要想提高“技”，除了个人的领悟，最重要的是靠大量的实践，有时候我们称之为“造轮子”。关键是在造的过程中得思考，比如什么时候该抽象了？这个轮子和竞品相比有什么优势？技术选型上为什么要使用 XX？

使用 Python 会遇到这样的问题：什么时候该用多进程？怎样提高代码执行效率？Flask 为什么流行？曾经遇到一个冷门的 Celery 的 Bug，当时使用谷歌没有找到解决方案，甚至解决的思路也没有，怎么办？我开始翻阅 PEP 文档，阅读优秀开源项目的源码，还把 Python 标准库模块中的代码全部过了一遍，收获颇丰。同时我还会根据工作中遇到的问题，给开源项目和 Python 提一些 Issue，后来还给它们提交 Patch，用自己微薄的能力，让社区变得好一点点。

虽然过了而立之年，我还在不间断地更新博客（dongwm.com），希望给其他开发者带来帮助和灵感。当许艳编辑找到我时，双方一聊，发现对国内开发者而言实战类 Python Web 开发方面的书确实不多，我顿时觉得可以以自己多年的工作经验积累写一本，为女儿两岁生日送上一份不一样的礼物。作为一个做过运维，现在做后端，却经常写前端程序的人来说，我了解产品的整个过程，是适合写一本这样的书的。写这本书的意义还在于，将自己这几年在使用 Python 进行 Web 开发中对各方面知识的理解和积累的经验进行梳理和总结，让更多人受益，同时对自己也是一种成长，也算是对国内的 Python 环境做出个人的贡献了。

谁应该看本书

虽然语言只是工具，但是阅读本书还是需要有一定的 Python 基础，如果你还没学过 Python，那就先学习一段时间再来阅读本书，收获会更多。

本书主要面向如下 4 类人群：

- Web 开发者。
- 使用 Python 语言的运维工程师和运维开发工程师。

- 想提高 Python 技能的开发者。
- 想了解 Python Web 开发的其他开发者。

为什么值得看

本人阅读过大量和 Python 有关的纸质书和开源图书，渐渐学到了很多控制自己“剁手”买书的方法。我来分析一下为什么你值得拥有本书。:)

为什么要买书来看？我认为不外乎两个原因：有趣和能学到东西。技术书肯定不会太有趣，那么最重要的就是能学到东西。市面上 Python 相关的书相当多，但是有些内容陈旧或者不符合国情，或者并非开发第一线的人所写或者翻译，这样的书显然价值就要打一些折扣；其次是同质化严重，偏入门级别，我个人认为市面上关于 Python 入门或者教授语法知识的书不少，而再深入一点的就很匮乏了。

本书有几个特点：第一，使用了当前主流和前瞻性的技术，如 Docker、Ubuntu 16.04 LTS、Cython、CFFI、Py.test、asyncio、IPython 5.0 LTS 等，书中一部分内容是在 Python 3 下完成的。本书中全部工具都使用当前最新版，能保证在相当长的时间内书中的内容都不会过时；第二，笔者在国内应用 Python 最大的豆瓣网做产品开发，一直在第一线写代码，大量例子和经验都来自实际工作；第三，笔者非常关注 GitHub 和 Python 社区，会第一时间了解到新的趋势和思想，并在书中体现。举个例子，代码检查工具 pep8 已经在 Guido van Rossum 的要求下改名为 pycodestyle 了。

叔本华在《人生的智慧》中说过一段话，大意是“人要么庸俗，要么孤独”。笔者认为这个道理在阅读上面也成立：读什么样的书，就会逐渐成为什么样的人。本书提供了很多笔者在其他书中没有看到过的思考方式和 Python 的用法，这也是本书存在的意义。

本书的特别用法

本书中有些效果可能会让读者迷惑，在这里解释一下。

1. 交互模式下的效果。本书在交互模式下完成的例子都使用了 IPython，效果如下：

   ```
   In : template.render(name='Xiao Ming')
   Out: u'Hello Xiao Ming!'
   ```

   其中 `In :` 是输入，`Out:` 是输出。

2. 终端提示符。本书绝大多数的终端提示符没有使用 `#` 或者 `$`，而使用了 `>`。

本书的组织方式和阅读建议

笔者和身边的一些朋友交流过，大多数人买书来看，基本上都是看到书中讲到了自己一直不太懂的知识点，或者感兴趣的话题。因此，在写作本书时，笔者有意让每章相对独立。你可以选择跳着看，当然更推荐从第 1 章一直看到最后一章，因为本书是按照一个 Web 产品从无到有、从简单变复杂、从基础到进阶的过程编排的。

我们先来大致了解一下这个过程。

- **第 1 章** 首先回答两个问题："为什么应该选择 Python 作为 Web 开发语言"和"选择 Python 2 还是 Python 3"，然后介绍 Python 主流的 Web 框架，并为如何选择给出建议。
- **第 2 章** 帮助读者跑起来一个包含本书所讲内容的 Ubuntu 环境，读者可以直接在里面运行书中的例子。限于篇幅，如果想要了解环境搭建的整个过程以及笔者做这些选择的理由，可以在本书源代码项目中的 setup.md 文件中获取，短链接地址为 http://bit.ly/29N4Pqv。接着将展开介绍 Python 的包管理和虚拟环境相关的内容。通过学习这章，读者对 Python 生态环境会有一定了解。
- **第 3 章** 先从最简单的 Flask 例子开始，学习一些 Flask 相关的知识，接着学习 Jinja2 和 Mako 模板（Mako 在豆瓣的使用非常广泛），使用 MySQL，最后学以致用，从零开始完成一个相对复杂、在豆瓣有类似功能的文件托管服务。这个项目贯穿本书，在之后的章节中会对它继续扩展。
- **第 4 章** 这一章是 Flask 的进阶，包含了大量的 Flask 扩展的使用，还介绍了信号机制和 Werkzeug 的使用。到这里读者对 Flask 和 Web 开发已经入门，可以根据自己的想法自己做一些应用了。
- **第 5 章** 现在 Web 端应用对交互的要求很高，移动应用对后端的 API 需求也非常多，需要很好的异构通信方式，本章将介绍笔者对 REST 的理解，并提出一些设计 API 的注意事项，最后通过 jQuery 和 fetch 实现使用 Ajax 的例子，让读者了解如何让前后端通信。
- **第 6 章** 我们已经有了有实际业务逻辑的 Web 应用，可是用户还不能访问，本章将介绍如何选择应用服务器，用主流的方式在生产环境中运行这个应用。之前应用中只是使用了 MySQL，在实际的网站应用中，缓存、键值对数据库、NoSQL 数据库都是主流的解决方案，本章将一一介绍为什么要用这种技术以及怎么用。最后作为总结，笔者根据自己的实际经验绘制一张大型网站的架构图，并详细介绍其中模式选

择的理由和经验。

- **第 7 章** 在第 6 章，Web 应用已经运行起来，用户也可以访问了。但是如下问题也随之出现：
  - 应用依赖多个服务，如 MySQL、Redis 等，这些服务器在新环境中的部署是有顺序要求的，而且程序要保证一直在运行状态。
  - 上线过程不能自动化。每次上线都要手动执行很多命令，既耗时又容易出错。
  - 希望能及时了解和分析服务器和应用的运行状态。

  看完本章相信你就可以知道对应的解决方案了。
- **第 8 章** Web 应用运行良好，可是应用的质量还没有保证，如何在上线之前发现更多的 Bug 的需求变得越来越迫切。本章将介绍主流的测试方法，并用一个 GitHub 项目实现持续集成。
- **第 9 章** 前面介绍的是 Web 应用必备的内容，从本章开始介绍一些进阶的内容。消息队列能带来更好的用户体验，本章将介绍豆瓣用到的消息队列工具 Beanstalkd，以及 Celery 推荐的消息队列 RabbitMQ。如果 Web 产品有大量的定时任务或者其他异步任务，就可以使用 Python 界最知名的 Celery 解决，本书将从浅入深让读者熟悉 Celery 原理和使用方法，最后分享笔者使用的进阶实践。
- **第 10 章** 现在各个大公司都在谈服务化，通过这几年的改造和实践，大家都有自己的一套服务化方案，豆瓣也不例外。本章将告诉读者为什么要服务化、豆瓣的服务化设计，以及使用开源的 Thrift 改造文件托管服务。
- **第 11 章** 笔者在工作中经常要给各个业务方提供数据支持，如日志统计分析、数据报表。本章将演示如何使用纯 Python 代码在单个服务器上利用多核实现 MapReduce 功能，还详细讲解豆瓣工程师都在用的 DPark，包含安装、环境配置、使用和框架化分析 UV & PV；接着将展示几个笔者在实际工作中遇到过的数据报表需求，并讲解如何用 Pandas 做数据可视化。
- **第 12 章** 这一章将详细介绍 IPython 和 Jupyter Notebook 这两个工具，并分享其在豆瓣对应的实践。除此之外，还列出笔者日常用来排错和调试的工具，包括了解 Linux 服务器的相关情况、性能测试、分析 Python 程序性能瓶颈三个方面。
- **第 13 章** Web 开发日常也会有一些并发编程工作，所以本章以抓取微信公众号文章为主线，分别使用多线程、多进程、Gevent、Future 和 asyncio 这 5 种编程方式完成不同阶段的爬取任务，也深入地分析在它们之间如何选择。
- **第 14 章** Python 进阶并不只针对 Web 开发人员，对于所有 Python 开发者都有意义。