python

Python 3 自动化软件发布系统

——Django 2 实战

- 基于Python 3.6、Django 2.1以上最新版本开发
- 手把手搭建GitLab、Jenkins、SaltStack平台
- 项目代码存放Github
- 大量使用类视图实现快速开发
- 覆盖Model、View、Form、Mock多个测试用例

陈　刚　王洪军　编著

北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

策划编辑：剧艳婕
封面设计：runsign蓝正设计·赫健

内容简介：

本书以最新的Python 3.6以上版本为编程语言，以Django 2.1以上版本为Web框架，通过一步一步详细讲解，实现一个自动化软件部署系统，并将所有代码开源托管在Github网站上。

书中不但讲解了Python的基础知识、Django的模型、视图、模板之间的关系，而且还通过实战项目，介绍了Django REST Framework的开发步骤、Django Channels实现WebSocket的编程技巧，以及Django TestCase、Mock等测试用例的编写。

除此之外，对于IT公司在DevOps转型过程中涉及的GitLab、Jenkins、SaltStack等CI/CD工具的安装配置，以及自动化软件部署系统与之进行交互的API，都有较深入的讲解。

本书适合运维研发领域的人员，或是有一定Python基础但又想深入学习Python Web开发的广大IT人员阅读参考。

作者简介：

陈刚，有关于大数据、软件开发、电商网络平台、云计算和互联网金融平台等工作经验，现主要从事服务器网络运维及运维开发相关的工作，主攻Docker容器技术及Kubernetes的自动化运维，开发语言以Python为主，且涉及Php、Java和Go语言等。

王洪军，有关于电商平台开发经验，主攻运维开发领域，在Docker容器技术及Kubernetes编排技术上有自己独到的见解。

北京航空航天大学出版社
微信公众号

上架建议：程序设计

ISBN 978-7-5124-3057-0

9 787512 430570 >

定价：119.00元

Python 3 自动化软件发布系统

——Django 2 实战

陈　刚　王洪军　编著

北京航空航天大学出版社

内 容 简 介

本书以最新的 Python 3.6 以上版本为编程语言，以 Django 2.1 以上版本为 Web 框架，通过一步一步详细讲解，实现一个自动化软件部署系统，并将所有代码开源托管在 Github 网站上。

书中不但讲解了 Python 的基础知识、Django 的模型、视图、模板之间的关系，而且还通过实战项目，介绍了 Django REST Framework 的开发步骤、Django Channels 实现 WebSocket 的编程技巧，以及 Django TestCase、Mock 等测试用例的编写。

除此之外，对于 IT 公司在 DevOps 转型过程中涉及的 GitLab、Jenkins、SaltStack 等 CI/CD 工具的安装配置，以及自动化软件部署系统与之进行交互的 API，都有较深入的讲解。

本书适合运维研发领域的人员，或是有一定 Python 基础但又想深入学习 Python Web 开发的广大 IT 人员阅读参考。

图书在版编目(CIP)数据

Python 3 自动化软件发布系统 : Django 2 实战 / 陈刚，王洪军编著. -- 北京 : 北京航空航天大学出版社，2020.1

ISBN 978-7-5124-3057-0

Ⅰ. ①P… Ⅱ. ①陈… ②王… Ⅲ. ①软件工具－程序设计 Ⅳ. ①TP311.561

中国版本图书馆 CIP 数据核字(2019)第 181993 号

版权所有，侵权必究。

Python 3 自动化软件发布系统

——Django 2 实战

陈 刚 王洪军 编著

责任编辑 刘晓明

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) http://www.buaapress.com.cn

发行部电话：(010)82317024 传真：(010)82328026

读者信箱：emsbook@buaacm.com.cn 邮购电话：(010)82316936

三河市华骏印务包装有限公司印装 各地书店经销

*

开本：710×1 000 1/16 印张：46.75 字数：996 千字

2020 年 1 月第 1 版 2020 年 1 月第 1 次印刷 印数：3 000 册

ISBN 978-7-5124-3057-0 定价：119.00 元

若本书有倒页、脱页、缺页等印装质量问题，请与本社发行部联系调换。联系电话：(010)82317024

前　言

正如看电影要关注 IMDB 评分，看书要看亚马逊评分，听音乐要关注 Billboard，在编程的世界里，我们也需要关注 TIOBE(The Importance Of Being Earnest)。

TIOBE 编程社区索引是编程语言流行度的指标。索引每月更新一次。评级基于全球技术工程师、课程和第三方供应商的数量。流行的搜索引擎，如谷歌、必应、雅虎、维基百科、亚马逊、YouTube 和百度，用于计算评级。

流行，意味着应用广泛，职位需求多，更容易获取编程资源，从这种维度上讲，其更值得学习。

2018 年 9 月，Python 首次进入 TIOBE 指数前三(长期霸榜前两名的为 C 和 Java)。

2019 年 1 月，TIOBE 宣布：Python 成为 2018 年度编程语言(第一次是在 2010 年)。Python 已成为当今大学中最常被教授的首选语言，在统计、AI 编程、脚本编写、系统测试等领域均排名第一。此外，Python 还在 Web 编程和科学计算领域处于领先地位。

Python 语言即将结束从版本 2 到 3 的不兼容漫长升级(2020 年 1 月 1 日，官方不再提供对 Python 2 版本的升级支持)，且核心开发人员已表示，Python 4 只是 Python 3.10 之后的平稳升级版本。可以预见，Python 这条“大蟒蛇”，还会继续在计算机编程的各个领域里兴风作浪，勇往直前，成为一条“贪食蛇”。

所以，对于一个 IT 编程人员来说，不是要不要学 Python 的问题，而是何时开始学的问题。本书即可引领你进入 Python 的 Web 开发的世界！

本书定位

笔者希望带领读者学习五个方面的知识：① 了解 Python 基础知识；② 清楚 Django 开发的流程；③ 学习 Django 常见第三方模板的开发；④ 实现一个自动化软件部署系统(分为两个主要模板，自动化部署项目名称为 Manabe，其实时日志读取模块的项目名为 mablog)；⑤ 安装配置 DevOps 中 CI/CD(Continuous Integration/Continuous Deployment——持续集成/交付)领域经常使用的开源工具。

这五个方面，市面上都有很多专著专门讲解其中某一方面的知识，而想把这五个方面的所有知识点都清楚、透彻地做全面讲解，显然是不太现实的。

所以，为了使读者能切实学习到在运维研发这一工作岗位上要求的知识，本书中对这五个知识模板做了不同程度的强弱处理，主线始终围绕实战型的自动化部署系统。

Python 基础知识：这个方面，现在市面上已有很多比较好的入门书籍，网络上也有不少好的开源书籍及连载，所以笔者只点明主要的知识点，以示例代码的方式，快速讲解自动化部署系统中涉及的编码知识。这方面没有涉及的内容，希望读者在日常工作中，不断地汲取更多的知识，让自己的内功修炼不断地升级。

Django 开发套路：作为一个基于 Python 的 Web 开发框架，Django 开发中，既会涉及前端 HTML、CSS、JS 的知识，也会涉及后端的 MTV 的知识（M——Model 模型、T——Template 模板、V——View 视图），这方面的内容也是蛮多的。我们在组织本书内容时，会以一个比较简短的 BBS 原型功能为主线，讲解 Django 的开发实现方式。更深的知识点，网上可以参照的书籍不是很多（印象中，西文的 Packtpub 出版的 *Django 2 by Example*、图灵翻译电子书《精通 Django》这两本入门不错），所以需要读者再仔细阅读 Django 的官方文档。幸好，官网已开始有中文文档了。

Django 第三方模板使用：这方面的知识，主要涉及的是 Django REST Framework 和 Django Channels。一个讲 REST API 开发，一个讲 WebSocket 实时应用，都是现代 Web 开发不得不会的技能。在讲解这些知识时，以笔者个人的学习经历来看，单纯地讲解，或是简单地介绍示例代码，都不会让读者能有多深的理解。所以，我们将结合具体的自动化部署系统代码，把这两者的应用讲明白。

自动化软件部署系统：作为本书的主线，这部分内容是绝对要作为重点讲解的，不但会在每章说明设计的思路，还会有代码段详细讲解主要代码的作用，以及每个功能设计完成之后都有对应的网页截图。书中贴的代码段都有 Github 上对应的文件名称，并且附有行号。应特别注意的是，因为书中有的代码段是选取 Github 上文件中的代码片段，所以行号主要是作为代码解释之用，并不对应于真正的文件行号。一定要特别注意这个规则！另外，自动化软件部署系统的前端 UI，我们选用的是网上免费开源的 H-ui 的管理框架，这部分内容放在附录里单独讲解。

CI/CD 的工具：这是我们自动化部署系统的周边配套工具，主要包括 GitLab、Jenkins、SaltStack 三个工具。没有这些工具，我们的系统就跑不起来。而没有自动化系统，这些工具却可以通过手工操作实现。所以，这部分内容，也是作为一个运维研发人员必须要熟悉的（但运维研发不需要对这些工具做日常运维和功能管理，这原则上是同部门其他同事的职责）。我们对这部分内容，会讲解其安装、配置、日常管理。同时，由于我们的自动化部署系统需要对接这些工具的 API 接口，所以，对于这三个工具的 API 也会做系统调用范围内的讲解。

本书内容

第1章：首先以简短完整的示例代码，从Python基本数据类型开始，系统地讲解了Python这种编程语言中的数据类型、顺序、分支选择语句、循环语句。然后，对函数的类的知识也做了梳理。因为自动化软件部署系统中使用了线程池，所以在本章，进程池和线程池的并发编程也会涉及到。本章接下来讲解的内容，就是Python的模板化及测试方面的内容。这也是我们写代码时必须要了解的。需要说明的是，本书除了在第1章讲解Python的测试内容外，还比较全面地讲解了Django的测试。Django部分的测试内容，没有单独成章节，而是作为一些章节的补充，放置于最后面。Model测试、View测试、Form测试、Mock测试、API测试、Channels连接测试，这些测试用例，都会讲解到。

第2章：从本章开始，重点讲Django这个Web开发框架的内容，包括后面的自动化软件部署系统，也是在Django框架基础之上实现的。在本章中，先讲解了关于Web和HTTP协议的常识，HTML、CSS、JS代码的作用；之后，在Web开发中引入Django介绍。在介绍Django框架时，以循序渐进的方式，逐步引入Django的视图、URL路由、模板和模型数据库的学习。这个学习过程，不是一下就全部推向读者，而是进行了知识点隔离，在进行前面知识讲解的同时，不会涉及后面知识点的内容。这种讲解方法，我们认为是一种更有效的学习方式。在了解了Django的主要知识之后，本章还会介绍Django的表单及开箱即用的后台管理界面，这些知识，可以让读者更快速、更规范地进行Django的Web开发。

第3章：本章主要介绍自动化软件部署系统的工作流，它也是一种Devops的运作流程。有了这个流程模型，在进行后面的代码编写时，才能既见树木，又见森林。在本章开始，写了几个简单的开发示例代码（主要是Java），并进行了经典的编译打包和部署操作。这个手工操作的过程，会在后面用自动化的方法来实现。在有了示例代码之后，本章接着讲解如何将示例代码使用GitLab来管理源代码，使用Jenkins来自动编译源代码生成软件包，使用SaltStack来远程拉取软件包并进行启停部署。本章最后，还讲解了GitLab及Jenkins的主要API接口，这些接口，我们会在后面的开发时，使用第三方功能模块进行配置和调用。

第4章：从本章开始，正式进入软件自动化部署系统的实战开发。在本章中，我们规划好此项目的目录结构，使用Django命令建好相关的App应用，实现全面数据库的设计。在设计每个数据表时，都会讲解这个数据表的作用、每个字段的意义。更重要的是，再结合第2章的知识点，使用Django orm为每个数据表生成一批模拟数据。有了这些数据，就可以更深入地理解orm技巧，更深入地理解部署系统数据库，并且后面开发的每一个UI网页，都有充足的演示数据。

第5章：本章讲解从代码上实现自动化用户管理功能。其主要功能是用户的注册、登录、退出、修改密码和邮箱、忘记密码时的找回、登录时的验证码实现等。由于

这些内容涉及了前端和后端开发的知识融合，所以在本章开始，会先讲解 Web 开发中 Cookie 及 Session 的知识。然后，再系统地讲解 Django 内置的用户管理功能。Manabe 项目的用户管理功能，就是在结合 Django 内置功能及自定义功能之上实现的。在实现用户管理功能的相关代码时，我们会按照表单代码、视图代码、模板代码及路由注册代码这四大步骤（有的功能块可能没有表单内容），细致地讲解代码的主要内容。

第 6 章：在本章中，主要讲解自动化软件部署系统中应用和服务器的录入。有了这两个功能，再加上合适的权限管理，就能让不同的用户参与到系统的建设中来，让运维研发人员专注于系统的开发和功能的完善，而系统数据的准备性和扩展性就留给不同的部门分别负责。在技术上，本章系统地讲解了 Django 开发技术中的类视图（class based view）、自定义的四级用户权限设计、zTree 的前端库用于实现权限树形展示、Select 2 前端库用于多用户的方便选择；在最后，还示范了 Django 中的 Model 测试、View 测试及 Form 测试用例的编写。

第 7 章：在本章，实现了发布单的新建、软件编译功能。其中，在编译软件时，调用了 Python 的第三方库操作 Jenkins 的 API 接口，有了这个功能之后，发布系统就可以将 Jenkins 隐藏于幕后，减轻一般研发人员亲自操作 Jenkins 的负担。

第 8 章：环境流转，是我们在实际工作中遇到的一个功能。可能有的公司不需要，但知道如何实现这一功能，相信对读者会有一定的启发。一个发布系统有了环境流转功能，并对它进行权限管理，就可以将开发、测试、运维三种角色串连起来，形成自动化部署的流水线操作。

第 9 章：在本章，实现了自动化部署的核心功能——软件部署。同样，我们会以视图、网页模板、URL 路由的方式进行层层推进实现。对三个最重要的函数，deploy_cmd()、deploy()、cmd_run()的代码，也进行了细致的讲解。其中，涉及到调用 saltstack api 的内容，也进行了相应的封装。在发布过程中，涉及的服务器和应用的数据表字段变化，都有细致入微的实现。在实现发布的同时，服务启停作为一个附加而常用的功能，也可以进行功能上的实现。当然，这些都是基于第 6 章的权限管理机制来进行控制的。本章末尾，发布的历史记录功能，也都一一实现。

第 10 章：在本章，我们使用了 Django Channels 这样一个较新的 WebSocket 技术，解决了软件发布过程中的实时日志读取的问题。为了示范分模板的调用，在本章中，我们启动了一个名为 mablog 的新的 Django 项目。这个项目的主要目的有两个：一是实现写入日志的 API，供第 9 章的软件发布时调用；二是使用 WebSocket 的实时双向连接推送技术，把日志作为消费者，在一个连接中实时地从服务器推向浏览器。相对于间隔循环的 Ajax，它的优势可以说是相当明显的。但由于这是一个新的异步技术实现，需要读者在熟悉了 Django 的常规同步编程之后，才能继续本章的学习。在本章接下来的部分，讲解如何使用 Daphne 在生产中部署 mablog 项目，以及如何编写 Channels 连接的测试用例。

第 11 章：本章主要讲解如何使用 Django REST Framework(DRF)进行 RESTful API 接口的开发。在本章中，首先讲述了 DRF 的安装配置，以及为了安全而设计的 Token 认证实现。然后，基于 DRF 的 Viewset 集成快速开发功能，实现了用户 App 应用，以及服务器的 API 接口开发。其间，涉及到的序列化技术、外键引用技术、Token 生成及获取、URL 注册规则都有实战级的演示。最后，还使用了 Requests 库进行 Post、Get 请求测试，介绍了如何进行 DRF 的测试用例的编写。

第 12 章：作为本书的最后一章，首先实现了 Django 中的 Logging 日志记录功能，接着使用百度 echarts 技术，实现了一个简单的发布单数据编译曲线图和柱状图。用户可以在此基础上，实现更多的系统统计和图表功能。Django 的生产环境部署，也是一个必不可少的学习内容。在本章，我们示例了在生产环境中推荐的 uwsgi、gunicorn 的安装及运行，以及前端 Nginx 代码的配置。同时，为了对接新兴的 Docker 容器技术，在本章还实现了 Nginx 镜像，以及支持 Uwsgi、Gunicorn、Channels 模板的镜像。最后，介绍了如何使用 Coverage 这个工具，实现 Django 开发中的代码覆盖率测试报告。

附录 1：详细讲解了 Python 3.6 在 Windows 系统下的安装、Django 2.1 的安装配置、Pycharm 开发 IDE 的基本配置使用。对于日常开发中的 Python 虚拟环境管理——当前流行的基于内置 Venv 模板及 Pipenv 都有所涉及。

附录 2：现在，很多公司的代码管理平台都从 Svn 切换到 GitLab。这个附录，就讲解了 GitLab 的安装、服务启停和常用功能的使用，如新增项目及项目组、新增用户及授权。

附录 3：Jenkins 在中小企业几乎已成为 CI 工具的代名词，我们的自动化部署系统也是调用 Jenkins 的 API 来生成软件包的。在本附录中，首先演示了如何使用 Docker 容器来快速安装一个 Jenkins 示例服务器。然后，进行初步的管理配置。接着，介绍了 Jenkins 新版中的 Pipeline 的语法入门。有了这些基础知识，就能理解自动化部署系统中的 Jenkins 方面的内容了。

附录 4：在自动化部署系统中，我们使用了一个名为 H-ui 的开源前端管理框架，用于快速生成所有网页的模板。在本附录中，重点讲解了这个 H-ui 前端框架的主要组件。然后，将自动化部署系统的前端网页，按子模板形式进行了划分。有了这些基础知识，在看正文时，前端代码部分就很好地对接起来了。除此之外，对于实践开发过程中使用的三个主要前端库——jQuery、zTree、Select 2，也进行了简单的讲解。

附录 5：这是本书最后一个附录，讲解了目前最流行的企业 Docker 仓库管理工具——Harbor。其安装、配置、用户管理、项目管理过程，都以实践的形式一一呈现。最后，还进行了 Docker 上传及下载的测试。读者可以结合第 12 章的几个镜像，学习如何将企业内容实现 Docker 镜像的管理。这对于现在的运维及运维研发人员，都可以说是一个基本技能了，故在最后，涉及到了块的内容。

代码规范

本书涉及的代码有三种，下面一一说明其显示规范。

1. 命令行

书中的命令行，以 BKB 小五号字体，加粗强调显示。如下示例：

```
python manage.py runserver
```

2. 命令输出

命令的输出，以 BKB 小五号字体，加粗强调显示。如下示例：

```
Performing system checks...

System check identified no issues (0 silenced).
September 15, 2018 - 22:07:46
Django version 2.1, using settings 'manabe.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

3. 程序代码

程序代码的显示，首先会分行，然后给出此代码所有 Github 的 URL 链接地址。此地址以 BKB 小五号字体、加粗、斜体强调显示。紧接着给出代码内容，每一行代码左边都会有连续的行号，整个代码块以浅灰色为底。代码内容如果强调不为完整文件或内容，会有“…”省略号出现。如果代码内容强调为一个完整的类或函数或文件，则不会涉及“…”省略号。总之，一切最终代码，以 Github 上的为准。

这里需要再次强调的是，代码块中的行号，是为了在接续的代码解释中方便解释之用，并不代表此代码块在 Github 文件中真正的行号。这样做的原因是，我们常常需要将 Github 中的一个代码文件，在不同的页面拆成不同的代码块来解释。如果特别强调行号的对应关系，反而会让读者在阅读书中的每一个代码块时，有不连贯的感觉；而且在代码解释中，也会有支离破碎的数字，因而，我们会采用每个代码块自我连续的行号处理方案。这是读者在看本书代码和 Github 上代码时，需要特别注意的地方。如下示例：

https://github.com/aguncn/manabe/blob/master/manabe/appinput/views.py

```
01  class AppInputListView(ListView):
02      template_name = 'appinput/list_appinput.html'
03      paginate_by = 10
04
05      def get_queryset(self):
06          if self.request.GET.get('search_pk'):
```

```
07                search_pk = self.request.GET.get('search_pk')
08                return App.objects.filter(
09                    Q(name__icontains=search_pk) |
10                    Q(package_name__icontains=search_pk))
11            return App.objects.all()
12
13        def get_context_data(self, **kwargs):
14            context = super().get_context_data(**kwargs)
15            context['now'] = timezone.now()
16            context['is_admin_group'] = is_admin_group(self.request.user)
17            context['current_page'] = "appinput-list"
18            context['current_page_name'] = "App应用列表"
19            query_string = self.request.META.get('QUERY_STRING')
20            if 'page' in query_string:
21                query_list = query_string.split('&')
22                query_list = [elem for elem in query_list if not elem.startswith('page')]
23                query_string = '?' + "&".join(query_list) + '&'
24            elif query_string is not None:
25                query_string = '?' + query_string + '&'
26            context['current_url'] = query_string
27            return context
```

代码解释：

第2行：template_name，指明此视图要渲染的模板是appinput/list_appinput.html。

第3行：paginate_by：指明每页显示10个App服务。更多的服务列表，以分页的方式呈献。

第5～11行：get_queryset方法，此方法，定义了……

读者对象

本书不是一本细致讲解Python的入门书，所以希望读者具有一定的Python基础，有过使用Python开发脚本或是其他应用的经验，至少，也应该看过网上几篇入门版的教程。*A Byte of Python*（《简明Python教程》）这本入门级教程，值得推荐。

另外，也希望读者能有一点运维经验，对Linux系统的操作较熟悉。

能满足以上两点要求的IT从业人员，都可以是本书的读者对象。



致　谢

首先，感谢北京航空航天大学出版社的剧艳婕编辑，是她让我们有机会系统地总结自己的技能，并能高效地把本书推送到读者面前。

==========

本书写作过程，适逢家中新增二宝。妻子潘永日夜操心，孝云大姐不辞辛劳；我则陪伴甚少，无以为报。遥想家母魏恩枢，艰难时代，自学缝纫起家，养大我兄弟姐妹四人，个中强韧，不言而喻。本书如果有小小成就，要归功于这天下无私而伟大的母爱！

陈　刚

==========

感谢亲人，感谢朋友！

王洪军

==========

感谢公司领导和同事对我们平时工作的支持和理解。

在写作本书时，两位作者参考了网上的大量文档，由于这些文档相互引用的情况比较多，所以本书并未能逐个列举出所有参考资料。在此一一感谢这些朋友，是他们，让这个 IT 世界更美好。

纠　　错

限于作者能力有限，书中错误在所难免。如果读者发现其中错误，欢迎发送到作者邮箱 aguncn@163.com。请以 xxx 章节 xx 页的内容或代码有错误为标题。

陈　刚　王洪军

2019 年 7 月

目　录

第 1 章　Python 基础 …… 1

1.1　Python 基本数据类型的常用操作 …… 5
1.1.1　数字(Number)——人生不能只会做减法 …… 6
1.1.2　字符串(String)——一入编码深似海 …… 8
1.1.3　列表(List)——古龙之七种武器 …… 10
1.1.4　元组(Tuple)——一颗不变心 …… 14
1.1.5　字典(Dictionary)——键值 CP …… 15
1.1.6　集合(Set)——我们的集合无悖论 …… 16
1.2　顺序、分支及循环语句 …… 17
1.2.1　顺序编程——知所先后，则近道矣 …… 18
1.2.2　分支语句——《交叉小径的花园》 …… 19
1.2.3　循环——《恐怖游轮》 …… 21
1.3　函数和类 …… 22
1.3.1　函数——好莱坞原则 …… 22
1.3.2　类——有没有对象，都累 …… 25
1.4　并发编程 …… 28
1.4.1　进程池示例——《低俗小说》 …… 28
1.4.2　线程池示例 …… 30
1.5　模块化 …… 31
1.6　测　试 …… 34
1.7　小　结 …… 39

第 2 章　Django 基础 …… 42

2.1　Web 及 HTTP …… 42
2.1.1　Web 简介 …… 42
2.1.2　HTTP 协议 …… 43
2.1.3　HTTP 协议方法 …… 44

2.2 HTML、CSS、JavaScript …… 45
2.2.1 HTML …… 45
2.2.2 CSS …… 45
2.2.3 JS(JavaScript) …… 46
2.2.4 Python、Django 是怎么和 Web 搭上关系的 …… 46
2.3 Django 简介及 Web 实现 …… 48
2.3.1 Django 简介 …… 48
2.3.2 Python 的 Web 服务器 …… 49
2.3.3 Django 模块实现 Web 服务器 …… 49
2.4 Django 新建项目应用及运行机制 …… 50
2.4.1 新建 Django 项目应用 …… 50
2.4.2 Django 目录及文件分析 …… 51
2.4.3 Django 框架的工作机制 …… 53
2.5 Django 视图 …… 55
2.5.1 基于函数的视图 …… 55
2.5.2 Django 的 HttpRequest 对象 …… 58
2.5.3 Django HttpRequest 对象中的 QueryDict 对象 …… 59
2.5.4 Django 的 HttpResponse 对象 …… 59
2.6 Django 路由 URL …… 61
2.6.1 UrlConf 简介 …… 61
2.6.2 UrlConf 的 Urlpatterns …… 62
2.6.3 UrlConf 的路由分发 …… 63
2.6.4 UrlConf 的反向解析 …… 64
2.6.5 UrlConf 的命名空间 …… 65
2.7 Django 模板 Template …… 66
2.7.1 Django Template Language 简介 …… 67
2.7.2 Django Template 加载配置及基本使用 …… 69
2.7.3 Django Template 的 Render 快捷使用 …… 71
2.8 Django 模型 Model …… 72
2.8.1 Model ORM …… 73
2.8.2 Model 示例 …… 73
2.8.3 ORM 常用 Field 及属性 …… 77
2.8.4 Django Shell 操作 ORM …… 79
2.8.5 函数视图操作 ORM(显示 board 列表) …… 81
2.8.6 函数视图操作 ORM(显示指定 board 的 topic 列表) …… 83
2.8.7 函数视图操作 ORM(新增 topic) …… 85
2.8.8 函数视图操作 ORM(指定 board 的 topic 内容) …… 88
2.8.9 类视图操作 ORM(显示 board 列表) …… 90
2.8.10 Model 的底层数据库连接配置 …… 92

2.9　Django 表单 Form …… 93
2.10　Django 后台管理 Admin …… 95
2.10.1　Admin 界面登录 …… 95
2.10.2　应用 Model 的注册 …… 96
2.10.3　Model Admin 自定义管理界面 …… 98
2.11　Django 测试 …… 99
2.12　小　结 …… 100

第 3 章　自动化软件发布工作流 …… 101

3.1　示例项目 …… 102
3.1.1　编写示例代码 …… 102
3.1.2　编译项目 …… 105
3.1.3　手工运行 …… 106
3.1.4　其他示例代码 …… 107
3.2　使用 GitLab 保存源代码 …… 109
3.2.1　建立用户和项目组 …… 109
3.2.2　建立项目 …… 113
3.2.3　将本地代码推送到 GitLab 中 …… 115
3.3　使用 Jenkins 编译项目 …… 120
3.4　使用 Nginx 作为软件仓库 …… 125
3.4.1　Jenkins 和 Nginx 服务器之间免密码登录 …… 126
3.4.2　安装并配置 Nginx 服务器 …… 128
3.4.3　安装 Jenkins 插件 …… 128
3.4.4　配置 Jenkins Pipeline …… 129
3.4.5　验　证 …… 130
3.5　使用 SaltStack 推送服务端脚本 …… 132
3.5.1　Salt Master 及 Salt Minion 的安装 …… 132
3.5.2　通过 Salt Master 远程执行脚本命令 …… 134
3.5.3　Salt-API 配置 …… 137
3.5.4　实现远程拉取软件、启停服务的脚本范例 …… 146
3.5.5　测　试 …… 151
3.6　Jenkins REST API 使用讲解 …… 153
3.6.1　Jenkins 原生 API 的获取 …… 154
3.6.2　Python-Jenkins 库的安装 …… 159
3.6.3　Python-Jenkins 的常用方式 …… 160
3.6.4　封装一个 Python 脚本，实现自动化编译软件 …… 163
3.7　GitLab REST API 使用讲解 …… 167
3.7.1　Python-GitLab 的安装、配置 …… 167
3.7.2　Python-GitLab 常用功能使用 …… 169

3.7.3 使用 Python-GitLab 获取 Zep-Backend-Java 文件列表 …… 174
3.8 小 结 …… 175

第 4 章 自动化发布的数据库模型 …… 176

4.1 功能展示 …… 176
4.1.1 用户管理 …… 176
4.1.2 应用 App 的管理 …… 176
4.1.3 服务器的管理 …… 178
4.1.4 应用权限的管理 …… 178
4.1.5 发布单的新建及软件包编译 …… 179
4.1.6 环境流转 …… 179
4.1.7 软件发布 …… 180
4.1.8 软件流转及发布历史 …… 181
4.2 新建项目及 App …… 181
4.2.1 新建目录，通过 Pip 安装相关模块 …… 181
4.2.2 新建项目及相关 App …… 181
4.2.3 生成数据及管理员密码 …… 182
4.2.4 启动 Django 服务并验证 …… 184
4.2.5 与 PyCharm 集成 …… 185
4.3 调整文件内容 …… 186
4.3.1 中文及时区 …… 186
4.3.2 App 注册 …… 187
4.3.3 URL 路由调整 …… 189
4.4 Django Models 的抽象基类 …… 190
4.5 应用数据表 …… 192
4.5.1 models.py 文件内容 …… 192
4.5.2 将应用数据表迁移进数据库 …… 196
4.5.3 生成模拟数据 …… 198
4.6 环境数据表 …… 204
4.6.1 models.py 文件内容 …… 204
4.6.2 将环境数据表迁移进数据库 …… 204
4.6.3 生成模拟数据 …… 206
4.7 服务器数据表 …… 207
4.7.1 models.py 文件内容 …… 207
4.7.2 将服务器数据表迁移进数据库 …… 209
4.7.3 生成模拟数据 …… 211
4.8 发布单状态数据表 …… 212
4.8.1 models.py 文件内容 …… 213
4.8.2 将发布状态数据表迁移进数据库 …… 213

4.8.3　生成模拟数据 …… 215
4.9　发布单数据表 …… 216
4.9.1　models.py 文件内容 …… 216
4.9.2　将发布单数据表迁移进数据库 …… 220
4.9.3　生成模拟数据 …… 221
4.10　权限管理数据表 …… 224
4.10.1　models.py 文件内容 …… 225
4.10.2　将权限数据表迁移进数据库 …… 226
4.10.3　生成模拟数据 …… 228
4.11　历史记录数据表 …… 231
4.11.1　models.py 文件内容 …… 231
4.11.2　将历史数据表迁移进数据库 …… 232
4.12　理解 Django Migrate(数据迁移) …… 233
4.12.1　Migrate 原理 …… 234
4.12.2　理解更新 models.py 文件的原理 …… 236
4.12.3　重置 migration …… 238
4.13　小　结 …… 239

第 5 章　用户注册登录及密码管理 …… 241

5.1　用户管理简介 …… 241
5.2　Web 开发中的 Cookie 及 Session …… 242
5.2.1　Cookie …… 242
5.2.2　Session …… 243
5.3　中间件(Middleware)及预安装(INSTALLED_APPS) …… 245
5.3.1　Django 框架中的 Middleware …… 245
5.3.2　Django 框架中的 INSTALLED_APPS …… 248
5.4　用户管理 …… 249
5.4.1　用户注册 …… 250
5.4.2　用户认证 …… 250
5.4.3　用户登录 …… 250
5.4.4　用户退出 …… 251
5.4.5　修改密码 …… 251
5.5　用户组管理 …… 252
5.6　Django 权限功能 …… 253
5.6.1　权限管理简介 …… 253
5.6.2　用户权限 …… 254
5.6.3　用户组权限 …… 256
5.7　Manabe 用户注册 …… 256
5.7.1　用户注册表单 …… 257