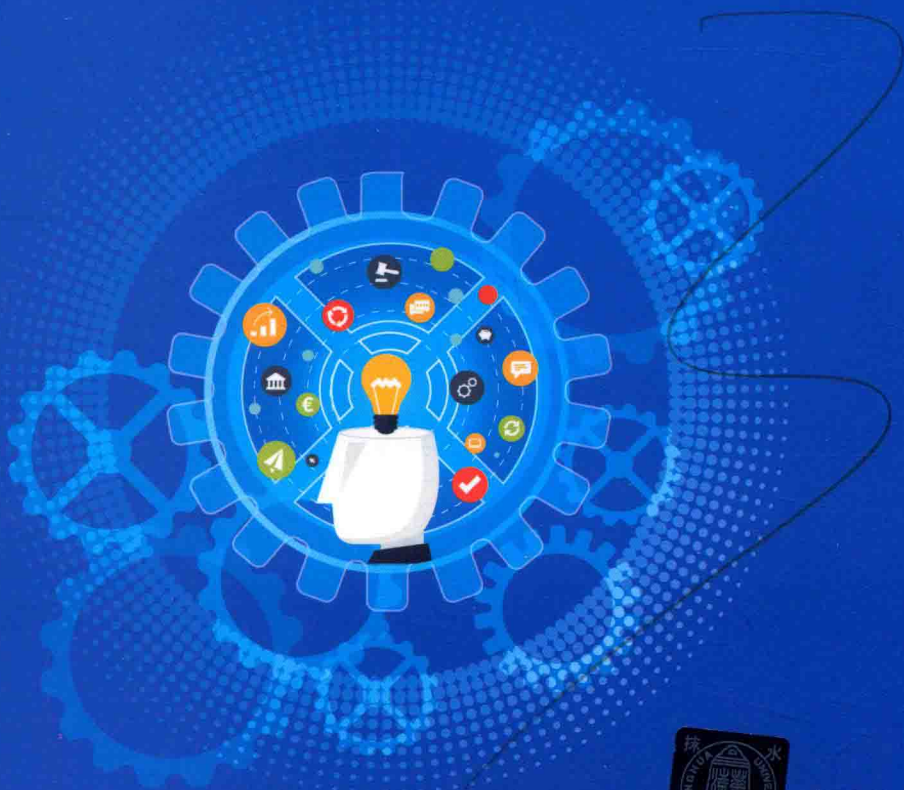


一册在手，快速全面掌握 Django 技巧

基于 Django 的电子商务网站设计

顾翔◎著

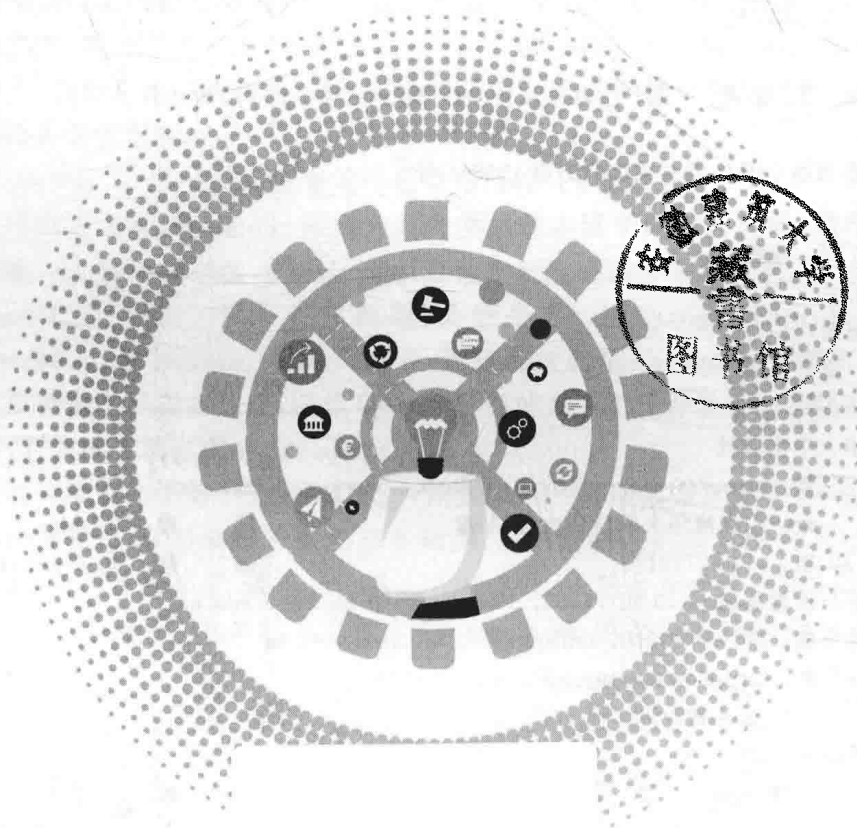
通俗易懂地介绍基于Django框架的Python Web编程框架
通过电子商务网站，提供使用Django框架的案例实战
每一模块均以requests类来对功能进行自动化接口测试，从而达到对产品质量的自检
在线提供本书所有的产品及测试代码



清华大学出版社

基于 Django 的电子商务网站设计

顾翔◎著



清华大学出版社
北京

内 容 简 介

本书是一本介绍如何基于 Django 框架开发网站的书籍,分 4 章。第 1 章是 Python、Django 发展历史与概要介绍和安装方法,以及 HTTP 的基础知识;第 2 章详细介绍了 Django 基本知识,包括如何启动 Django 服务、Hello World 程序、HttpRequest 与 HttpResponse 对象、setting.py 的配置、cookie 和 session、Django 的 MTV 开发模式框架、Django 的模型与数据库的管理、Django 的视图管理、Django 的模板管理以及基于 Python Requests 类数据驱动的 HTTP 接口测试;第 3 章以电子商务网站为例,介绍电子商务网站的需求、电子商务网站数据 Model 设计以及用户信息、商品信息、购物车、送货地址、订单、电子支付模块,最后介绍如何建立自定义的错误页面;第 4 章在第 3 章的基础上介绍如何构建安全的网站,分别是密码的加密、防止 CSRF 和 XSS 的攻击、权限操作的漏洞以及防止 SQL 注入。

本书可作为准备使用 Django 框架开发网站、学习接口测试的软件工作人员的学习用书,也可作为在校本科生和研究生的参考用书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

基于 Django 的电子商务网站设计/顾翔著. —北京:清华大学出版社,2018
ISBN 978-7-302-50512-9

I. ①基… II. ①顾… 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(2018)第 139861 号

责任编辑:白立军 常建丽

封面设计:杨玉兰

责任校对:焦丽丽

责任印制:沈 露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京嘉实印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:14.25

字 数:349 千字

版 次:2018 年 10 月第 1 版

印 次:2018 年 10 月第 1 次印刷

定 价:45.00 元

产品编号:078376-01

在百度百科中对 Python 是这样介绍的：

“Python，是一种面向对象的解释型计算机程序设计语言，由荷兰人 Guido van Rossum 于 1989 年发明，第一个公开发行人版发行于 1991 年。

Python 是纯粹的自由软件，源代码和解释器 CPython 遵循 GPL (GNU General Public License) 协议。Python 语法简洁清晰，特色之一是强制用空白符 (white space) 作为语句缩进。Python 具有丰富和强大的库，它常被昵称为胶水语言，能够把用其他语言制作的各种模块 (尤其是 C/C++) 很轻松地联结在一起。常见的一种应用情形是，使用 Python 快速生成程序的原型 (有时甚至是程序的最终界面)，然后对其中有特别要求的部分用更合适的语言改写，例如，3D 游戏中的图形渲染模块，性能要求特别高，就可以用 C/C++ 重写，而后封装为 Python 可以调用的扩展类库。需要注意的是，在使用扩展类库的时候，可能需要考虑平台问题，因为某些扩展类库可能不提供跨平台的实现。

2017 年 7 月 20 日，IEEE 发布编程语言排行榜：Python 高居首位。”

在许多欧美国家，Python 已经成为各大学的基本教学语言。另外，随着大数据与人工智能 (AI) 技术的兴起，Python 语言在这方面也表现得非常出色。

Python 有以下几个 Web 开发框架，分别是 Flask、Django、Tornado、Bottle、web.py、web2py 及 Quixote。就本人而言，我最喜欢的是 Django 这个框架，虽然掌握这个框架需要学习很多知识，但是 Django 的目的是为了让开发者能够快速地开发一个网站，它提供了很多模块，如 admin。

作者认为刚开始学习 Django 框架，只需要掌握一些最基本的知识，不需要一开始就学习全部知识，只要掌握了这些基本知识，再根据自己的需求学习其他高深的知识，就会变得更加容易了，这正是书写本书的目的：让读者在最短的时间内尽快掌握 Django 框架。建议：如果没有接触过 Django 框架，甚至没有接触过 Web 开发，就可以首选本书。

本书第 1 章是 Python、Django 发展历史与概要的介绍和安装方法以及 HTTP 的基础知识，但是作者在这里没有对 Python 语言进行介绍，如果没有 Python 语言的任何基础，建议通过其他渠道学习掌握 Python 语言后再来学习此书。

第 2 章介绍 Django 基本知识。这里对 Django 的基本知识介绍得比较全面，读者可能看不懂 2.7 节到 2.9 节介绍的知识，只要了解一些概念就可以了，经常使用到的知识会在第 3 章中结合案例进行详细介绍。

第3章结合电子商务网站项目,通过用户信息、商品信息、购物车、送货地址、订单和电子支付这6个模块进行详细介绍。对于其中的每个子模块,通过如何设计 url.py、如何开发 view.py、模板的设计、接口测试用例的设计以及接口测试代码的书写进行介绍。这里特别需要指出的是,随着软件质量在软件研发中的地位越来越高,并且随着迭代快速响应用户需求的普及,自动化测试显得越来越重要。但是,单元测试代码的繁多以及基于 GUI 的自动化测试受界面影响很大的原因,这些测试都没能很好地普及,而基于单元测试与基于 GUI 的自动化测试之间的接口测试在业界却越来越普及。又由于 Python 提供的 Requests 类能够更好地配合接口测试的开发,所以作者在本书中对接口测试的技术和实现方法进行了详细描述。作者也是一边书写本书一边书写电子商务网站和接口测试代码,每次程序结构发生变化,都会运行一下以前写好的接口测试代码,以保证新的修改没有影响以前的功能。另外,接口测试的运行速度远远快于基于 GUI 的自动化测试。本章包含的接口测试用例共 49 条,全部运行,耗时仅为 2.709 499s。

第4章在第3章的基础上介绍如何构建一个安全的网站,分别是密码的加密、防止 CSRF 和 XSS 的攻击、权限操作的漏洞以及防止 SQL 注入。由于第3章对程序进行了很好的封装,所以在这里对产品和测试代码的修改变得更加简单了。正如在本书中作者所写的那样,我们不可能一开始就能书写一个易于维护的好代码,这需要在书写代码的过程中不断地优化。现在敏捷技术提倡研发过程的迭代优化,同样,在书写代码时也需要进行迭代优化。

本书的产品和测试代码放在网站 <https://github.com/xianggu625/ebusiness> 上,欢迎下载。

另外,在书写后期,巴特尔、金鑫、刘中秋、任荣哲、万巧、杨军军、叶微及赵院娇对本书文稿进行了校验,在此表示真心的谢意,同时也感谢家人的鼓励与协助,没有你们的支持,本书是不可能在如此短的时间内完成的。

对于本书的内容和产品以及测试代码如果有什么问题,读者可以加我的微信:xianggu0625,另外,我的个人网站是 <http://www.3testing.com>,下面的二维码是我的微信公众号,欢迎各位扫描关注。



顾 翔

2018年2月于上海

第 1 章 Python、Django 和 HTTP	1
1.1 Python 语言	1
1.1.1 Python 语言概述	1
1.1.2 Python 的安装	3
1.2 Django 框架	4
1.2.1 Django 介绍	4
1.2.2 Django 的安装	5
1.3 HTTP 概述	6
1.3.1 HTTP 的工作原理	7
1.3.2 HTTP 的请求	8
1.3.3 HTTP 的应答	9
1.3.4 HTTP 的连接性	13
1.3.5 HTTP 的无状态	15
第 2 章 Django 基本知识	17
2.1 启动 Django 服务	17
2.2 Hello World 程序	21
2.2.1 直接打印显示内容	21
2.2.2 通过文件模板显示内容	22
2.2.3 文件模板参数	22
2.3 获取参数	23
2.3.1 通过 GET 方式获取	23
2.3.2 通过 POST 方式获取	23
2.4 HttpRequest 对象与 HttpResponse 对象	27
2.4.1 HttpRequest 对象	27
2.4.2 HttpResponse 对象	28
2.5 setting.py 的配置	29
2.5.1 中间件介绍	29
2.5.2 其他配置介绍	30

2.5.3	自定义静态文件	38
2.5.4	案例	39
2.6	session 和 cookie	42
2.6.1	session	43
2.6.2	cookie	45
2.6.3	Django 的用户登录和注册机制	47
2.7	Django 的 MTV 开发模式框架	49
2.8	Django 的模型与数据库的管理	50
2.8.1	Django 的数据库	50
2.8.2	Django 的模型	51
2.8.3	Django 的后台管理	55
2.8.4	Django 如何对数据库进行操作	57
2.9	Django 的视图管理	61
2.9.1	urls.py 中路径的定义	61
2.9.2	方法中显示内容	63
2.9.3	处理表单	63
2.9.4	分页功能	65
2.10	Django 的模板管理	66
2.10.1	变量的使用	66
2.10.2	标签的使用	66
2.10.3	过滤器的使用	72
2.11	基于 Python Requests 类数据驱动的 HTTP 接口测试	75
2.11.1	测试金字塔	75
2.11.2	unittest	76
2.11.3	requests 对象的介绍与使用	80
2.11.4	数据驱动的自动化接口测试	85
2.11.5	进一步优化	89
第 3 章 电子商务网站的实现		92
3.1	需求描述	92
3.1.1	用户信息模块	92
3.1.2	商品信息模块	92
3.1.3	购物车模块	92
3.1.4	送货地址模块	93
3.1.5	订单模块	93
3.1.6	订单支付模块	93
3.2	数据 Model 设计	93
3.3	用户信息模块	95
3.3.1	用户注册	96

3.3.2	用户登录	104
3.3.3	用户信息显示	109
3.3.4	用户登录密码的修改	130
3.4	商品信息模块	134
3.4.1	商品信息的维护	135
3.4.2	商品概要信息的分页显示	137
3.4.3	商品信息的模糊查询	144
3.4.4	商品信息的详情显示	146
3.5	购物车模块	149
3.5.1	把商品放入购物车	150
3.5.2	查看购物车中的商品	155
3.5.3	修改购物车中的商品数量	161
3.5.4	删除购物车中的某种商品	164
3.5.5	删除购物车内所有的商品	165
3.6	送货地址模块	167
3.6.1	送货地址的添加与显示	167
3.6.2	送货地址的修改	175
3.6.3	送货地址的删除	179
3.7	订单模块	181
3.7.1	总订单的生成和显示	182
3.7.2	查看所有订单	192
3.7.3	删除订单	197
3.8	电子支付模块	200
3.9	建立自定义的错误页面	200
第 4 章	构建安全的网站	205
4.1	密码的加密	205
4.2	防止 CSRF 攻击	206
4.2.1	CSRF 攻击介绍	206
4.2.2	Django 是如何防范 CSRF 攻击的	207
4.2.3	针对 CSRF 防御接口测试代码的调整	208
4.3	权限操作的漏洞	212
4.4	防止 XSS 攻击	218
4.5	防止 SQL 注入	218
参考文献		220

第1章

Python、Django 和 HTTP

Django 是基于 Python 语言的 Web 开发框架,所以要学习好 Django,首先要有基本的 Python 开发技巧,以及要了解 HTTP 的基本知识。本章首先介绍 Python 语言及其安装(Python 语法不在本书中介绍,读者可以查找其他书籍阅读),然后介绍 Django 知识及其安装,最后简单地介绍 HTTP。

1.1 Python 语言

1.1.1 Python 语言概述

在介绍 Python 之前,先来欣赏一下 Python 禅歌(读者可以在 Python 编译窗口中输入 `import this` 获得)。

英文原版

The Zen of Python, by Tim Peters

Beautiful is better than ugly.

Explicit is better than implicit.

Simple is better than complex.

Complex is better than complicated.

Flat is better than nested.

Sparse is better than dense.

Readability counts.

Special cases aren't special enough to break the rules.

Although practicality beats purity.

Errors should never pass silently.

Unless explicitly silenced.

In the face of ambiguity, refuse the temptation to guess.

There should be one—and preferably only one—obvious way to do it.

Although that way may not be obvious at first unless you're Dutch.

Now is better than never.

Although never is often better than * right * now.

If the implementation is hard to explain, it's a bad idea.

If the implementation is easy to explain, it may be a good idea.

Namespaces are one honking great idea — let's do more of those!

中文翻译版及解释

优美胜于丑陋

明了胜于晦涩

简洁胜于复杂

复杂胜于凌乱

扁平胜于嵌套

间隔胜于紧凑

可读性很重要

即便假借特例的实用性之名,也不可违背这些规则

不要包容所有错误,除非你确定需要这样做

当存在多种可能时,不要尝试去猜测

而是尽量找一种,最好是唯一一种明显的解决方案

虽然这并不容易,因为你不是 Python 之父

做也许好过不做,但不假思索就动手做还不如不做

如果你无法向人描述你的方案,那肯定不是一个好方案;反之亦然

命名空间是一种绝妙的理念,我们应当多加利用

Python 语言诞生于 20 世纪 90 年代初,它已被逐渐广泛地应用于系统管理任务的处理以及 Web 编程。Python 由于其易理解性、易读性以及简洁性,并且有近 30 年的历史,以及对云计算、大数据与人工智能(AI)开发(仅次于 R 语言)有很好的支持,因此越来越受到大众的喜欢。业界有种说法:“Java 十行代码,用 Python 一行代码就可以实现”,这句话说得虽然有些夸张,但是“Java 三到四行代码,用 Python 一行代码就可以实现”是完全没有问题的。

Python 的创始人是 Guido van Rossum^①,在 1989 年圣诞节期间,住在阿姆斯特丹,为了打发圣诞节的无聊时光,决定开发一个新的脚本解释程序,作为 ABC 语言的一种继承。Guido 选用 Python(大蟒蛇的意思)作为该编程语言的名字,是因为他是一个名为 Monty Python 喜剧团体的爱好者。

ABC 语言是由 Guido 参加设计的一种教学语言。Guido 认为 ABC 语言非常优美以及

^① Guido van Rossum(吉多·范·罗苏姆)1989 年在荷兰的国家数学和计算机科学研究院(Centrum voor Wiskunde en Informatica, CWI)创立了 Python 语言。1991 年初,Python 发布了第一个公开发版。Guido 原居荷兰,1995 年移居美国,并遇到了他现在的妻子。2003 年年初,Guido 和他的家人,包括他 2001 年出生的儿子 Orlijn 一直居住在华盛顿州北弗吉尼亚郊区。随后他们搬迁到硅谷,2005 年开始他就职于 Google 公司,其中有一半时间是花在 Python 上,现在 Guido 在为 Dropbox 工作。——百度百科

非常强大,是专门为那些非专业程序员而设计的。但是,ABC 语言并没有取得最后的成功,Guido 认为主要是 ABC 语言的非开放性造成的,所以 Guido 决心在 Python 中避免这个错误。同时,他还想在 ABC 语言中实现他想过但是没有实现的东西。

Python 在 Guido 手中诞生了,可以说,Python 是从 ABC 语言发展起来的,主要受到 Modula-3(另一种相当强大的语言,为小型团体所设计)的影响,并且结合了 UNIX Shell 和 C 的习惯。

Python 已经成为最受欢迎的程序设计语言之一。2011 年 1 月,它被 TIOBE 编程语言排行榜评为 2010 年度语言。2004 年以后,Python 的使用率呈线性增长。在 2017 年 4 月份 TIOBE 编程语言排行榜中,Python 语言仅次于 Java、C、C++ 和 C#,位于第五位。

由于 Python 语言的简洁性、易读性以及可扩展性,在国外用 Python 做科学计算的研究机构日益增多,一些知名大学已经采用 Python 来教授程序设计课程。例如,卡耐基梅隆大学的编程基础、麻省理工学院的计算机科学及编程导论都是使用 Python 语言来讲授的。众多开源的科学计算软件包都提供了 Python 的调用接口。再如,著名的计算机视觉库 OpenCV、三维可视化库 VTK、医学图像处理库 ITK。而 Python 专用的科学计算扩展库就更多了,例如,Java 语言的三个十分经典的科学计算扩展库: NumPy、SciPy 和 matplotlib,它们分别为 Python 提供了快速数组处理、数值运算以及绘图功能。因此,Python 语言及其众多的扩展库构成的开发环境十分适合工程技术、科研人员处理实验数据、制作图表,甚至开发科学计算应用程序。

1.1.2 Python 的安装

目前,市场上呈现 Python 2. x 系列与 Python 3. x 系列共存的现象。读者可以安装 Python 2. x 系列或者 Python 3. x 系列。如果开发的目的是基于原有 Python 2. x 系列产品的维护,建议选择 Python 2. x 系列;如果目的是开发一个完全新的产品,那么建议选择 Python 3. x 系列。作者写这本书的时候,Python 的最高版本是 3. 6,但是作者觉得 Python 3. 6 还是不成熟,所以本书选择的版本是 Python 3. 5。

Python 工具的官方下载地址是 <http://www.python.org/download>。

Python 下载完毕以后,务必要配置好环境变量(本书全部基于 Windows 开发环境进行介绍)。

图 1-1 是 PYTHON_HOME 变量的配置,变量值为安装 Python 的文件路径,在 Python 3. 5 中默认为 C:\Users\<Your_ID>\AppData\Local\Programs\Python\Python35\。

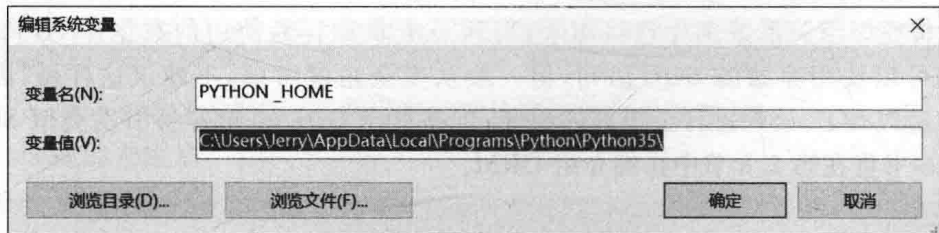


图 1-1 PYTHON_HOME 变量的配置

图 1-2 是在 PATH 中增加的两个参数,设置 %PYTHON_HOME% 是为了可以在任意路径下运行 Python 命令;设置 %PYTHON_HOME%\Scripts\是为了可以在任意路径下运行 %PYTHON_HOME%\Scripts\ 路径下的命令,如 pip 或 pip3。

```
%PYTHON_HOME%
%PYTHON_HOME%\Scripts\
```

图 1-2 PATH 中的配置

1.2 Django 框架

1.2.1 Django 介绍

1. Django 概况

Django 项目是一个 Python 语言定制框架,它源自一个在线新闻 Web 站点,于 2005 年以开源的形式被释放出来。Django 框架的核心组件如下。

- (1) 用于创建模型的对象关系映射。
- (2) 为最终用户设计完美的管理界面。
- (3) 一流的 URL 设计。
- (4) 设计者友好的模板语言。
- (5) 缓存系统。

Django 是用 Python 语言开发的一个开源的 Web 开发框架 (Open Source Web Framework, OSWF),它鼓励快速开发,并遵循 MVC 设计理念。Django 遵守 BSD 版权^①,初次发布于 2005 年 7 月,并于 2008 年 9 月发布了第一个正式版本 1.0。

Django 根据比利时的爵士音乐家 Django Reinhardt 命名,他是一个吉普赛人,主要以演奏吉它为主,还演奏过小提琴等。

由于 Django 在近年来的迅速发展,应用越来越广泛,被著名 IT 开发杂志 *SD Times*^② 评选为 2013 *SD Times* 100,位列“API、库和框架”分类第六位,被认为是该领域的佼佼者。

2. Django 的设计理念

Django 的主要目的是简便、快速地开发数据库驱动的网站。它强调代码的复用以及多个组件可以很方便地以“插件”形式服务于整个框架。Django 有许多功能强大的第三方插件,甚至可以很方便地开发出自己的工具包,这使得 Django 具有很强的可扩展性。Django 还强调快速开发和 DRY (Do Not Repeat Yourself) 的原则。

Django 基于 MVC 的设计十分优美。

(1) 对象关系映射 (Object-Relational Mapping, ORM): 以 Python 类形式定义数据模型,ORM 将模型与关系数据库连接起来,得到一个非常容易使用的数据库 API。虽然在 Django 中可以使用原始的 SQL 语句,但一般从安全角度考虑,不建议这样做,因为一是 Django 已经对 SQL 语句进行了很好的封装;二是显示 SQL 语句容易引发类似 SQL 注入的威胁。本书将在第 2.8 节中详细介绍 ORM。

^① BSD (Berkeley Software Distribution, 伯克利软件套件) 是 UNIX 的衍生系统,1977—1995 年由加州大学伯克利分校开发和发布。——百度百科

^② *SD Times* 即《软件开发时代》杂志。

(2) URL 分配: 使用正则表达式匹配 URL, 就可以设计任意的 URL。本书将在 2.9.1 节进行详细介绍。

(3) 模板系统: Django 提供强大而可扩展的模板语言, 它可以分隔设计、内容和 Python 代码, 并且具有可继承性。本书将在 2.10 节进行详细介绍。

(4) 表单处理: 可以方便地生成各种表单模型, 实现表单的有效性检验。可以方便地从定义的模式实例生成相应的表单。本书将在 3.3 节进行详细介绍。

(5) Cache 系统: 可以挂在内存缓冲或其他的框架实现超级缓冲。

(6) 会话(session): 用户登录与权限检查, 快速开发用户会话功能。本书将在 2.6 节进行详细介绍。

(7) 国际化: 内置国际化系统, 方便开发出多种语言的网站。

(8) 自动化的管理界面: 不需要用大量的工作对后台内容进行维护。Django 自带一个 Admin Site, 类似于后台管理系统。

3. 工作原理

(1) 用 `manage.py runserver` 启动 Django 服务器。

(2) 同时载入同一目录下的 `settings.py`。该文件包含了项目中的配置信息, 如 `URLCONF` 等, 其中最重要的配置就是 `ROOT_URLCONF`, 它告诉 Django 哪个 Python 模块应该用作本站的 `URLCONF`, 如图 1-3 所示。

```
53 ROOT_URLCONF = 'ebusiness.urls'
```

图 1-3 settings.py 中的 ROOT_URLCONF

(3) 当访问 URL 的时候, Django 会根据 `ROOT_URLCONF` 的设置来装载 `URLCONF`。

(4) 然后按顺序逐个匹配 `URLCONF` 里的 `URLpatterns`。如果找到, 则会调用相关联的视图方法, 并把 `HttpRequest` 对象作为第一个参数(通常是 `request`)。

(5) 最后, 该 `view` 方法负责返回一个 `HttpResponse` 对象。

Django 的工作原理如图 1-4 所示。

1.2.2 Django 的安装

安装完毕 Python, 接下来安装 Django, 安装 Django 有以下 4 种方法。

1. 利用 pip 安装

由于 1.1.2 节中已经在 `PATH` 变量中增加了 `%PYTHON_HOME%\Scripts\` 项, 所以可以在任意路径下运行如下命令。

```
>pip install django[==version]
```

`[==version]` 可以不书写, 不书写表示默认安装的是最新版本。

另外, 卸载的方法是:

```
>pip uninstall django
```

2. 利用 tar.gz 安装

首先下载 `gz` 包, 如 `Django-1.10.3.tar.gz` 文件, 其中 1.10.3 是 Django 的版本号, 然后

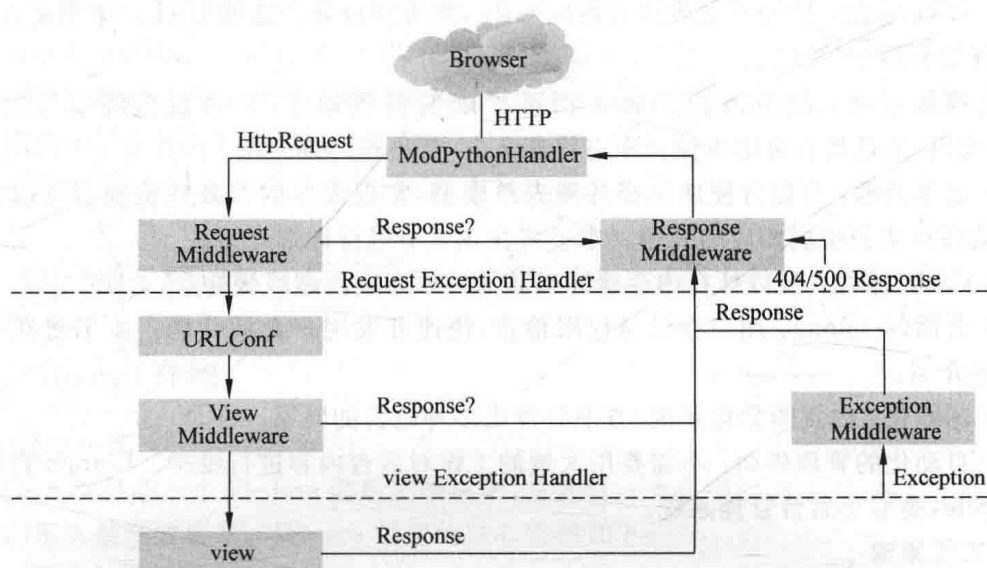


图 1-4 Django 的工作原理

进入目录内运行如下命令。

```
...>python setup.py install
```

3. 利用 .whl 安装

wheel 文件是一个类似 zip 的文件包,其实用 pip 安装也是先安装 wheel 文件到本地,然后自动运行解压包的动作。首先下载 wheel 文件包,如 Django-1.10.3-py2.py3-none-any.whl 文件,其中 1.10.3 仍旧是 Django 的版本号,然后运行如下命令。

```
...>pip install Django-1.10.3-py2.py3-none-any.whl
```

4. 在 GitHub 上安装

可以利用类似于 Eclipse、Atom 到 GitHub 网站上安装 Django。https://github.com/django/django 是 Django 在 GitHub 上的地址。

1.3 HTTP 概述

超文本传输协议(HyperText Transfer Protocol, HTTP)是互联网上应用最为广泛的一种网络协议,所有的 3W 文件都必须遵守这个标准。设计 HTTP 最初的目的是为了提供一种发布和接收 HTML 页面的方法。1960 年,美国人 Ted Nelson 构思了一种通过计算机处理文本信息的方法,并称其为超文本(HyperText),这就是 HTTP 标准架构的发展根基,HTTP 的第一个版本 HTTP 0.9 是一种简单地用于网络间原始数据传输的协议。Ted Nelson 组织协调万维网协会(World Wide Web Consortium, WWW)和互联网工程工作小组(Internet Engineering Task Force, IETF)共同合作研究,最终发布了一系列的 RFC。HTTP 1.0 是在 RFC 1945 定义的,它在 HTTP 0.9 基础上做了改进,允许消息以类多用途因特网邮件扩展(Multipurpose Internet Mail Extensions, MIME)信息格式存在,包括请

求/响应范式中的已传输和修饰符等方面的内容。现在普遍使用的是 RFC 2616 定义的 HTTP 1.1,要求严格保证可服务性,增强了在 HTTP 1.0 中没有考虑分层代理服务器、高速缓存、持久连接需求以及虚拟主机方面的能力。

现在 HTTP 还推出了 HTTP 2.0 版本。这里简单地介绍一下 HTTP 2.0。百度百科中对于 HTTP 2.0 是这样定义的: HTTP 2.0 即超文本传输协议 2.0,是下一代 HTTP,是由互联网工程任务组(IETF)的 HyperText Transfer Protocol Bis (httpbis)工作小组开发的,是自 1999 年 HTTP 1.1 发布后的首个更新。HTTP 2.0 在 2013 年 8 月进行首次合作共事性测试。在开放互联网上,HTTP 2.0 将只用于“https://网址”,而“http://网址”将继续使用 HTTP 1.1,目的是在开放互联网上增加使用加密技术,以提供强有力的保护遏制主动攻击。DANE RFC6698 允许域名管理员不通过第三方 CA(认证授权)机构自行发行证书。

1.3.1 HTTP 的工作原理

HTTP 是基于 TCP 的,同时也可以承载 TLS 或 SSL 协议层,这里把承载 TLS 或 SSL 协议称作为 HTTPS。一般情况下,HTTP 为 80 端口,而 HTTPS 为 443 端口。图 1-5 是 HTTP 协议栈。图 1-6 是 HTTPS 协议栈。

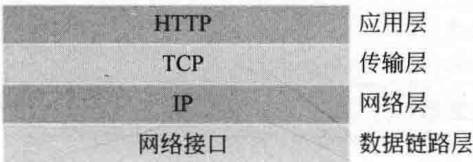


图 1-5 HTTP 协议栈

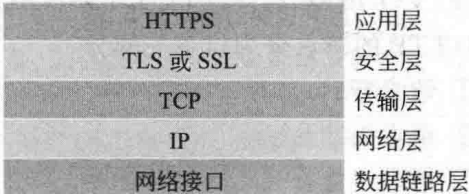


图 1-6 HTTPS 协议栈

通过图 1-7 可以更好地了解 HTTP 在整个网络中的位置。

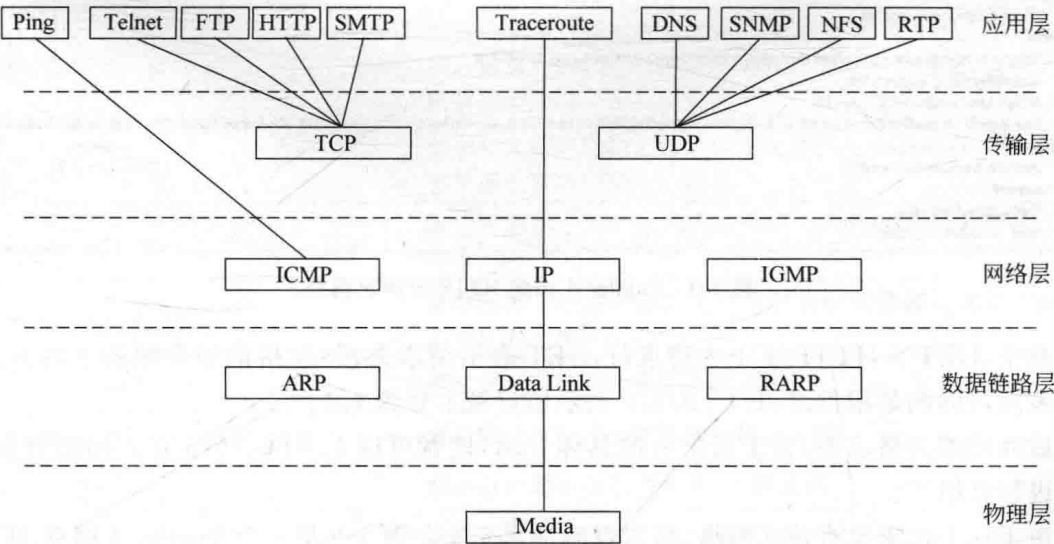


图 1-7 HTTP 在其他协议中的位置

1.3.2 HTTP 的请求

HTTP 的请求方式共分为 OPTIONS、GET、HEAD、POST、PUT、DELETE、TRACE 和 CONNECT 8 种(注意: 这些方法均为大写), 其中比较常用的为 GET 和 POST。

(1) OPTIONS: 返回服务器针对特定资源所支持的 HTTP 请求方法, 也可以利用向 Web 服务器发送 * 的请求来测试服务器的功能性。

(2) HEAD: 向服务器所要与 GET 请求相一致的响应, 只不过 HEAD 的请求响应体不会被返回(也就是说, GET 请求的响应为 HEAD 请求的响应+响应体)。这种方法可以在不必传输整个响应内容的情况下, 就可以获取包含在响应小消息头中的元信息。

(3) GET: 向特定的资源发出请求。注意: GET 方法不应当被用于产生“副作用”的操作中。例如, 在 Web Application 中, 其中一个原因是 GET 可能会被网络蜘蛛等随意访问。

(4) POST: 向指定资源提交数据处理请求(如提交表单或者上传文件)。数据被包含在请求体中。POST 请求可能会导致新资源的建立和(或)已有资源的修改。

(5) PUT: 向指定资源位置上传其最新内容。

(6) DELETE: 请求服务器删除 Request-URL 所标识的资源。

(7) TRACE: 回显服务器收到的请求, 主要用于测试或诊断。

(8) CONNECT: HTTP 1.1 协议中预留给能够将连接改为管道方式的代理服务器。

HTTP 的请求分为以下三部分。

① 请求行。

② 请求头。

③ 请求正文。

图 1-8 是一个用 Fiddler 4 捕捉到的访问 <http://www.3testing.com> 网站的请求内容。

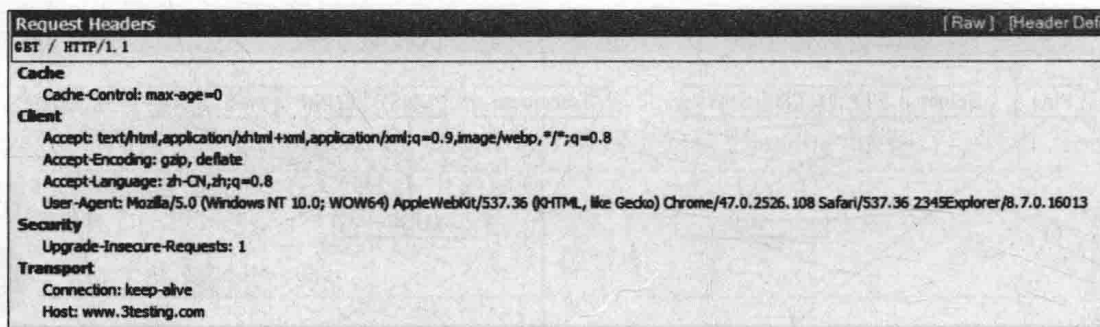


图 1-8 Fiddler 4 捕捉 HTTP 的请求信息

其中, GET / HTTP/1.1 为请求行, GET 表示请求方法, 包括前面介绍的 8 种方法之一; / 表示访问的是根目录; HTTP/1.1 表示协议版本号为 1.1。

后面的都为请求头, 关于请求头的具体介绍, 读者可以上 RFC 2616 官方网站查询, 这里不进行介绍了。

由于这个请求没有请求数据, 所以没有请求正文。图 1-9 是一个 Fiddler 4 捕捉 HTTP 的请求正文例子。

```
-----WebKitFormBoundarybWudMLpxmT9YA752
Content-Disposition: form-data; name="csrfmiddlewaretoken"

ljRgbiYPTVkyYjxeYC9bWMNW3RljmNDPp740LYHp2X0OWOJgEe8WbCqdAwOKGa1
-----WebKitFormBoundarybWudMLpxmT9YA752
Content-Disposition: form-data; name="username"

cindy
-----WebKitFormBoundarybWudMLpxmT9YA752
Content-Disposition: form-data; name="password"

123456
```

图 1-9 Fiddler 4 捕捉 HTTP 的请求正文例子

1.3.3 HTTP 的应答

HTTP 的应答返回码包含服务器响应情况,见表 1-1。

表 1-1 HTTP 的应答返回码

消 息	描 述
100 Continue	客户应该和自己的请求继续。中间的应答被用于告知客户请求的初始部分已经收到,并且还没有被服务器拒绝
101 Switching Protocols	服务器转换协议: 服务器将遵从客户的请求转换到另外一种协议
200 OK	请求成功(其后是对 GET 和 POST 请求的应答文档)
201 Created	请求被创建完成,同时新的资源被创建
202 Accepted	提供处理的请求已被接受,但是处理未完成
203 Non-authoritative Information	文档已经正常返回,但一些应答头可能不正确,因为使用的是文档的副本
204 No Content	没有新文档。浏览器应该继续显示原来的文档。如果用户定期刷新页面,而 Servlet 可以确定用户文档足够新,这个状态代码就是有用的
205 Reset Content	没有新文档。但浏览器应该重置它所显示的内容,用来强制浏览器清除表单中输入的内容
206 Partial Content	客户发送了一个带有 Range 头的 GET 请求,服务器响应了该请求
300 Multiple Choices	多重选择。链接列表。用户可以选择某链接到达目的地。最多允许五个地址
301 Moved Permanently	请求的页面已经转移至新的 URL
302 Found	请求的页面已经临时转移至新的 URL
303 See Other	请求的页面可在别的 URL 下被找到
304 Not Modified	未按预期修改文档。客户端有缓冲的文档并发出了一个条件性的请求(一般是提供 If-Modified-Since 头表示客户只想比指定日期更新的文档)。服务器告诉客户,原来缓冲的文档还可以继续使用