

# 基于 Django 和 Python 的 Web 开发

王冉阳

(中国地质大学(北京), 北京 100083)

**摘要:** 随着 Web 技术的发展和软件工程的日益成熟, 人们逐渐提出敏捷开发的要求。但是传统的 Web 框架像 Struts, Spring, .NET 等由于其复杂性而很难满足人们的要求, 于是逐渐出现一些敏捷性 Web 框架。Django 是使用 Python 开发的优秀 Web 框架, 作为一款开源的敏捷开发框架, 受到越来越多的人的关注。本文主要介绍了 Web 框架 Django 和 Python 语言的特点, 并以一个简易的博客系统的创建, 介绍 Django 和 Python 在 Web 开发中的应用。

**关键词:** Django; Python; Web 开发; Blog; 开源

## Web Development based on Django and Python

WANG Ranyang

(China University of Geosciences(Beijing), Beijing 100083)

**Abstract:** With the development of Web technology and software engineering, people gradually put forward the requirement of agile development. Traditional web framework such as Struts, Spring, .NET, and so on, can not meet the requirement because of their complexity. Django is excellent Web framework developed with Python. As a open source agile framework, has attracted more and more attentions. This article introduced the features of Django and Python. Through the creation of a simple blog system, we introduce the the development of Web applications base Python and Django.

**Keywords:** Django; Python; Web development; Blog; Open Source

### 1 引言

因特网的发展使得 Web 大行其道, 当今几乎所有的企业信息化软件以及部分个人桌面都采用了 Web 界面。Web 界面已成为和普通 GUI 并行的一种选择, 这种变化使得对 Web 开发框架的选择变得极为重要。在众多的 Web 开发框架中, Django /Python——一款开源的 Web 开发框架以其优秀的性能和快捷的开发正越来越引起人们的重视。

### 2 Python

Python 是一种简单易学、功能强大的编程语言, 它有高效率的高层数据结构, 简单而有效地实现面向对象编程。Python 简洁的语法和对动态输入的支持, 再加上解释性语言的本质, 使得它在大多数平台上的许多领域都是一个理想的脚本语言, 特别适用于快速的应用程序开发。

#### 2.1 简单

Python 是一种代表简单主义思想的语言。阅读一个良好的 Python 程序就感觉像是在读英语一样, Python 的这种伪代码本质是它最大的优点之一, 它使你能够专注于解决问题而不是去搞明白语言本身。

#### 2.2 易学

Python 有极其简单的语法, 极其容易上手开源: Python 是 FLOSS (自由/开放源码软件) 之一。简单地说, 可以自由地发布这个软件的拷贝、阅读它的源代码、对它做改动、把它的一部分用于新的自由软件中。

#### 2.3 高层语言

当用 Python 语言编写程序的时候, 无需考虑诸如如何管理程序使用的内存一类的底层细节。

#### 2.4 可移植性

由于它的开源本质, Python 已经被移植在许多平台上(经过改动使它能够工作在不同平台上)。如果小心地避免使用依赖于系统的特性, 那么所有 Python 程序无需修改就可以在下述任何平台上面运行。

这些平台包括 Linux、Windows、FreeBSD、Macintosh、Solaris、OS/2、Amiga、AROS、AS/400、BeOS、OS/390、z/OS、Palm OS、QNX、VMS、Psion、Acom RISC OS、Vx-Works、PlayStation、Sharp Zaurus、Windows CE 甚至还有 PocketPC。

#### 2.5 解释性

而 Python 语言写的程序不需要编译成二进制代码。可以直接从源代码运行程序。在计算机内部, Python 解释器把源代码转换成称为字节码的中间形式, 然后再把它翻译成计算机使用的机器语言并运行。事实上, 由于不再需要担心如何编译程序, 如何确保连接转载正确的库等等, 所有这一切使得使用 Python 更加简单。只需要把 Python 程序拷贝到另外一台计算机上, 它就可以工作了, 这也使得 Python 程序更加易于移植。

#### 2.6 面向对象

Python 既支持面向过程的编程也支持面向对象的编程。在面向过程的语言中, 程序是由过程或仅仅是可重用代码的函数构建起来的。在面向对象的语言中, 程序是由数据和功

能组合而成的对象构建起来的。与其他主要的语言如 C++ 和 Java 相比，Python 以一种非常强大又简单的方式实现面向对象编程。

2.7 可扩展性

如果需要一段关键代码运行得更快或者希望某些算法不公开，可以把部分程序用 C 或 C++ 编写，然后在 Python 程序中使用它们。

2.8 可嵌入性

可以把 Python 嵌入你的 C/C++ 程序，从而向程序用户提供脚本功能。

2.9 丰富的库

Python 标准库确实很庞大。它可以帮助处理各种工作，包括正则表达式、文档生成、单元测试、线程、数据库、网页浏览器、CGI、FTP、电子邮件、XML、XML-RPC、HTML、WAV 文件、密码系统、GUI（图形用户界面）、Tk 和其他与系统有关的操作。只要安装了 Python，所有这些功能都是可用的，这被称作 Python 的“功能齐全”理念。除了标准库以外，还有许多其他高质量的库，如 wxPython、Twisted 和 Python 图像库等。

3 Django

Django 是使用 Python 开发的 Web 框架。它的架构成为 MTV (Model-Template-View)，Model 用于数据库处理，Template 是模板系统，用于显示处理，View 是控制部分用于 Model 的处理、数据的加工和模板的调用等工作。由于 Django 是使用 Python 开发的，所以如果用 Django 开发站点，则用 Django 库写 Python 代码。

3.1 特征

3.1.1 可插入 (Plug-in) 管理的 Application

Django 提供了一个强大的命令行工具 `django-admin` 可以帮用户完成大量的初始化及有关数据库方面的工作。

3.1.2 对象相关的映射

完全在 Python 中定义数据模型。可以免费得到一个丰富的，动态访问数据库的 API——但如果需要仍然可以写 SQL 语句。

3.1.3 URL 分发

URL 的设计漂亮，`cruft-free` 没有框架的特定限定。

3.1.4 模版系统

使用 Django 强大而可扩展的模板语言来分隔设计、内容和 Python 代码。

3.1.5 Cache 系统

可以挂在内存缓冲或其他的框架实现超级缓冲——实现所需要的粒度。

3.1.6 自动化的管理界面

不需要花大量的工作来创建人员管理和更新内容的接界，Django 可以自动完成。

3.1.7 支持多种数据库

已经支持 Postgresql、MySQL、SQLite3、Oracle、`ado_`  
`msSQL`。

3.1.8 国际化

完全支持多国语言，允许针对应用程序提供特定国家语言的翻译字符

3.2 优势

3.2.1 高级 Web 框架

Django 免去了勾践动态 Web 站点的痛苦，它把常见的 Web 开发的问题抽象出来并提供了对频繁编程任务的捷径。

3.2.2 提供快速开发

Django 可以极大程度提高开发速度，由于采用 Python 语言，以为用几天、几星期、甚至几年的站点，用 Django 几个小时就可以做好。

3.2.3 干净实效的设计

Django 严谨的维护，干净的设计，它所有的代码让开发的 Web 程序遵循最佳实践，它的哲学是：Django 很轻松地做对事情。

4 应用

Django 是一种十分高效的敏捷 Web 框架，从 2005 年发布到现在已经出现了大量基于 Python 和 Django 开发的站点。以一个简易 Blog 开发为例，来讲述基于 Python 和 Django 的 Web 开发流程。该 Blog 的开发流程如图 1 所示。

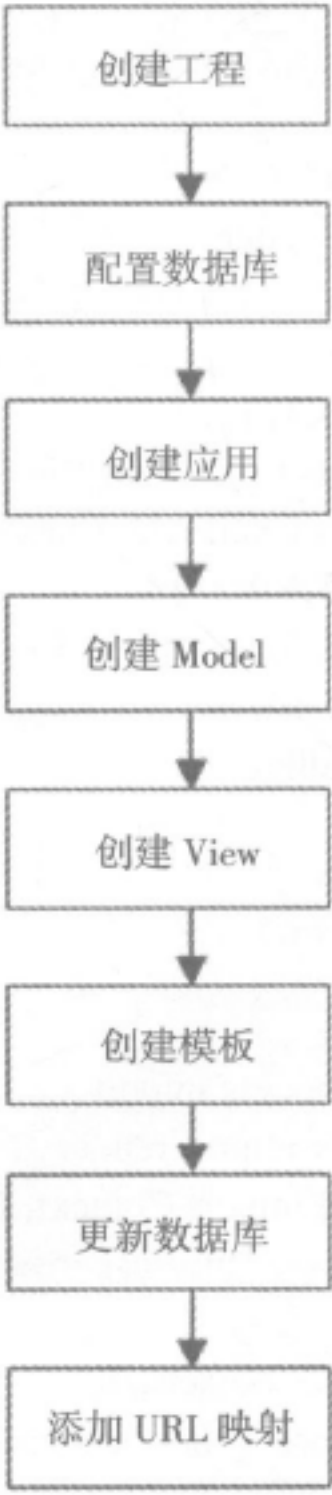


图 1 Blog 开发流程

具体的开发流程如下：

(1) 创建工程

使用下面的命令创建一个 project：



django-admin startproject webblog

## (2) 配置数据库

进入 webblog 目录, 修改 settings.py:

DATABASE\_ENGINE='sqlite3'

DATABASE\_NAME='webblog.db'

## (3) 创建应用

使用下面的命令创建一个 app:

manage.py startapp blog

修改 setting.py, 把新创建的 app 和模板加进去:

TEMPLATE\_DIRS = (

" d:/webblog/" ,

)

INSTALLED\_APPS = (

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.sites',

'webblog.blog',

'django.contrib.admin',

)

## (4) 创建 Model

进入 blog 目录, 修改 models.py:

class Reporter (models.Model):

full\_name = models.CharField (maxlength=70)

def \_\_str\_\_ (self):

return self.full\_name

class Admin:

pass

class Article (models.Model):

pub\_date = models.DateTimeField ()

headline = models.CharField (maxlength=200)

article = models.TextField ()

reporter = models.ForeignKey (Reporter)

def \_\_str\_\_ (self):

return self.headline

class Admin:

pass

## (5) 创建 View

修改 view.py:

from webblog.blog.models import \*

from django.shortcuts import render\_to\_response

from django.template import Context, loader

from django.http import HttpResponse

def index (request):

article\_list = Article.objects.all ()

return render\_to\_response ('htmls/index.html', {'article\_list': article\_list})

## (6) 创建模板

在 webblog 目录下创建 htmls 目录, 进入该目录, 创建 index.html:

{% extends " base.html" %}

{% block title %} Articles {% endblock %}

{% block content %}

<h1>Articles</h1>

{% for article in article\_list %}

<p>Headline : {{ article.headline }} </p>

<p>By {{ article.reporter.full\_name }} </p>

<p>Published {{ article.pub\_date|date:" F j, Y" }} </p>

{% endfor %}

{% endblock %}

创建 base.html:

<html>

<head>

<title> {% block title %} {% endblock %} </title>

</head>

<body>

{% block content %} {% endblock %}

</body>

</html>

## (7) 更新数据库

在 weblog 目录下, 执行:

manage.py syncdb

## (8) 添加 URL 映射

修改 url.py, 将访问地址加入进去:

from django.conf.urls.defaults import \*

urlpatterns = patterns ('',

(r'^\$', 'blog.views.index'),

(r'^admin/', include ('django.contrib.admin.urls')),

)

这样一个简单的 Blog 系统就搭建成功了, 整个系统的开发用了不到 30 分钟, 基于 Django 和 Python 的 Web 开发, 能够很好地满足快速开发的要求。

## 5 结语

Django 是一个优秀的平台。它提供了 Web 开发所必需的组件和工具, 极大地方便用户的开发, 并辅之功能强大的 Web 开发语言 Python。Django/Python 构成了一个优良的 Web 开发架构和平台, 足以用来实现功能强大和易维护的 Web 应用。

## 参考文献

- [1] 李迎辉. Python 开发 Rails 框架——Django 框架介绍. 程序员, 2006, (11) .
- [2] 王亚卿, 方龙. 浅析 python 在地图处理中的运用. 华东森林经理. 2008, (01) .
- [3] 郭力子. 基于开源的 Web 开发架构——ZOE/Python. 计算机时代. 2008, (07) .

## 作者简介

王冉阳, 男 (1983-), 学士, 图形图象, Web 开发。