

基于 Django 的 Blog 系统的开发与实现

Django-based Blog Development and Implementation

杨志庆

(同济大学 中德学院, 上海 200092)

摘要: Django 是使用 Python 开发的优秀开源 Web 框架, 在开发方面具有简洁、清晰、高效、安全的优点, 特别适合快速构建各类 Web 应用。通过分析 Django 框架结构及其特点并以一个简易的博客系统和一个登录验证界面的创建, 展示了 Django 在 Web 开发中的强大优势。实践证明, 使用 Django 可以降低 Web 应用开发的复杂性, 提高开发效率。

关键词: Django Python 网络应用 Blog 登录界面

DOI: 10.3969/j.issn.1007-080x.2013.09.013

Abstract: Django is an exceptional open-source Web frame developed by Python, characterized by conciseness, clarity, efficiency and security and suited at the fast structuring of various Web applications. Through analysing the Django framework and its characters and setting up a simplified blogging system and authentication system for log-in, it shows the good advantage of Django framework in the Web application development process. Practice proves that by using Django, the complexity of Web application development can be reduced so that the efficiency and quality of the development is improved.

Key words: Django Python internet application Blog log-in interface

0 引言

开发简单易学、可读性好的语言才能跟上 Web2.0 时代的潮流, 轻型、快速开发的框架才是生命力持久的框架。Python 是一个简单、解释型、交互式、可移植、面向对象的高级语言, 有一个交互式的开发环境, 因为它是解释运行的, 这大大节省了每次编译的时间。Python 为软件分析、设计、原型、编码、测试、调试、优化、文档、部署和维护等生命周期的每个阶段提供了非常高的效率。Django 是 Python 的一个开源 Web 开发框架, 相比于其他 Python Web 框架, 它安装简单且灵活多变, 符合 Python 开箱即用的特点。Django 框架遵循 MVC 开发模式, 内置多种 Web 开发

组件, 同时还集成了一个轻量级 Web Server, 使开发人员可以最少的代码、最快的速度, 方便高效地进行各种 Web 应用的开发和调试。经过长时间的测试, Python 3.0 发布, 其常被称为 Python 3000 或简称 Py3k。目前, 在世界范围内, 已有很多站点基于 Django 进行开发。本文根据 Django 的特点, 快速开发一个简单的带有登录验证功能的 Blog 应用。

1 关键技术介绍

1.1 Django 框架

Django 提供了一个高级的框架, 用它只需要很少的几行代码就能完成一个 Web 的应用。Django 框架的核心包括: 1 个面向对象的映射器, 用作数据模型

作者简介: 杨志庆 1990 年生, 硕士研究生。研究方向为信号与信息处理。

(以 Python 类的形式定义)和关联性数据库间的媒介;1 个基于正则表达式的 URL 分发器;1 个视图系统,用于处理请求;1 个模板系统^[6]。这种轻盈、强大、灵活的框架在方案的设计过程中发挥了极大的作用。

1.2 Django MVC 框架介绍

Django 是一种遵循 MVC 开发模式的框架。Models.py 用于创建数据库模型,处理如何存取,如何验证有效性、行为以及数据之间的关系等与数据相关的所有事务。Templates 文件夹中的各个模板文件代表视图(view),负责数据内容的显示;urls.py 定义了 Django 项目中的 URL 路由表,指定了 URL 与被调用类之间的对应关系,并和 views.py 中定义的各种处理函数(称为 Django 视图函数)代表控制器(controller)。其中的 urls.py 可以根据用户输入的 url 请求,调用 views.py 中相应的函数与数据模型和视图交互,响应用户的请求。基于 Django 的 Web 应用开发活动由于主要集中在 models.py、templates 文件夹中的各模板文件以及 views.py 之内,因此 Django 的开发模式通常也称为 MTV 开发模式。见图 1。

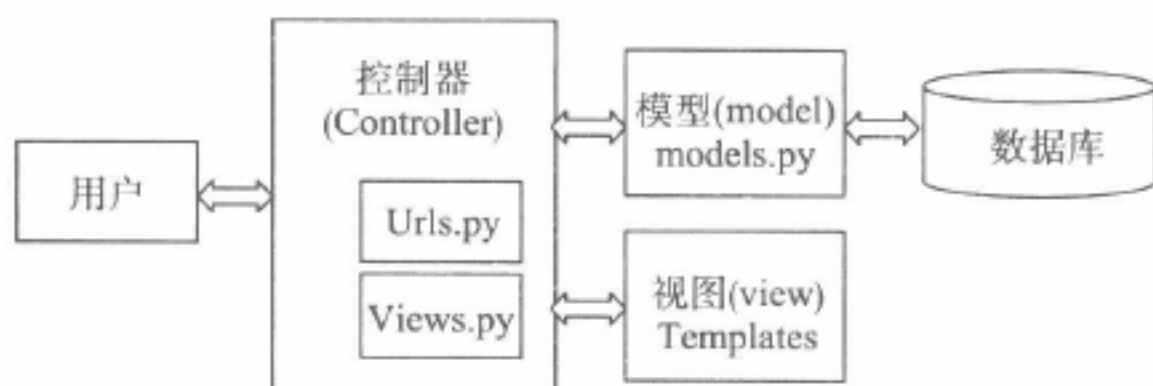


图 1 Django 框架的 MVC 开发模式

2 Blog 系统及其登录验证界面的开发与实现

2.1 创建项目

组织 Django 代码最简单的方式是使用 Django 的“项目”——1 个包含了组成单个网站的所有文件的目录。Django 提供了 1 个 django-admin.py 的命令来帮助创建这个项目的目录。在运行之前,需保证该命令所在路径必须在环境变量中。创建的项目包含以下文件:

——_init_.py 文件 会把这个项目目录变成 1 个 Python 的包——相关 Python 模块的 1 个集合;

——Manage.py 文件 是 1 个同这个 Django 项目一起工作的工具,可以从其目录列表中的权限里看到是可执行的;

——Setting.py 文件 包含该项目的默认设置,包括数据库信息、调试标志以及一些其他重要变量;

——Urls.py 文件 在 Django 里叫 URLconf,是 1 个将 URL 模式映射到应用程序上的配置文件。

2.2 Model.py 设计

在 model.py 文件中加入代码

```
class BlogPost(models.Model):
    title = models.CharField(max_length=150)
    body = models.TextField()
    timestamp = models.DateTimeField()
```

这是 1 个完整的 model,代表了有 3 个变量的“BlogPost”对象。这个新建的 BlogPost 类是 django.db.models.Model 的 1 个子类。这是 Django 为数据 model 准备的标准基类,是 Django 强大的对象映射(ORM)系统的核心。此外,每一个变量都和普通的类属性一样被定义为 1 个特定变量类(field class)的实例。这些变量类也在 django.db.models 里定义。

2.3 设置数据库

考虑到简单博客系统的规模,这里采用的数据库类型为 SQLite。它的速度很快,是 1 个轻量级的数据库,支持多种操作系统,占用资源低。SQLite 使用本地文件系统作为存储介质,并且用原生的文件系统权限来访问和控制主机、端口,用户或密码等信息一概不需要。我们只需为项目创建数据库的目录即可。在 setting.py 中加入:

```
DATABASE_ENGINE = 'sqlite3'
DATABASE_NAME = r'E:\py\django\db\blog.db'
```

现在就可以通过所提供的连接信息去连接数据库并且设置应用程序所需的表。

2.4 设置自动 admin 应用

自动化的后台应用程序 admin 称得上是 Django “皇冠上的明珠”。打开 setting.py 并在 INSTALLED_APPS 里的‘django.contrib.auth’下面添加:

```
‘django.contrib.auth’
‘django.contrib.admin’
```

每次往里添加后,重新运行 syncdb 来创建新的表:

```
Creating table django_admin_log
```



```
Installing index for admin. LogEntry model
```

```
Loding 'initial_data' fixtures...
```

```
No fixture found
```

设置完 App 后,我们需要为它制订 1 个 URL,这样才能访问。在自动生成的 urls.py 中有这样两行代码:

```
#Uncomment this for admin:
```

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

删除第二行开头的 # 号,可以保存文件。这就告诉 Django 去加载默认的 admin 站点,这是被用于 contrib admin 应用程序的 1 个特殊对象。

最后,应用程序需要告诉 Django 要在 admin 窗口中显示哪一个 model 以供编辑,因此只要在之前提到的默认 admin 站点,并向其注册 BlogPost model 的代码即可。完成应用的数据库部分和 admin 部分以后,我们建立 Blog 的公共部分——1 个页面具有 3 个典型的组件:

——1 个模板(template) 负责将传递进来的信息显示出来;

——1 个视图(view)函数 负责获取要显示的信息,通常都从数据库里取得;

——1 个 URL 模式 用来把收到的请求和视图函数匹配,有时也会向视图传递一些参数。

这样,一个简单的博客系统差不多就搭建完毕了,整个系统的开发相比其他开发工具,开发周期短,工作量也小很多(见图 2)。

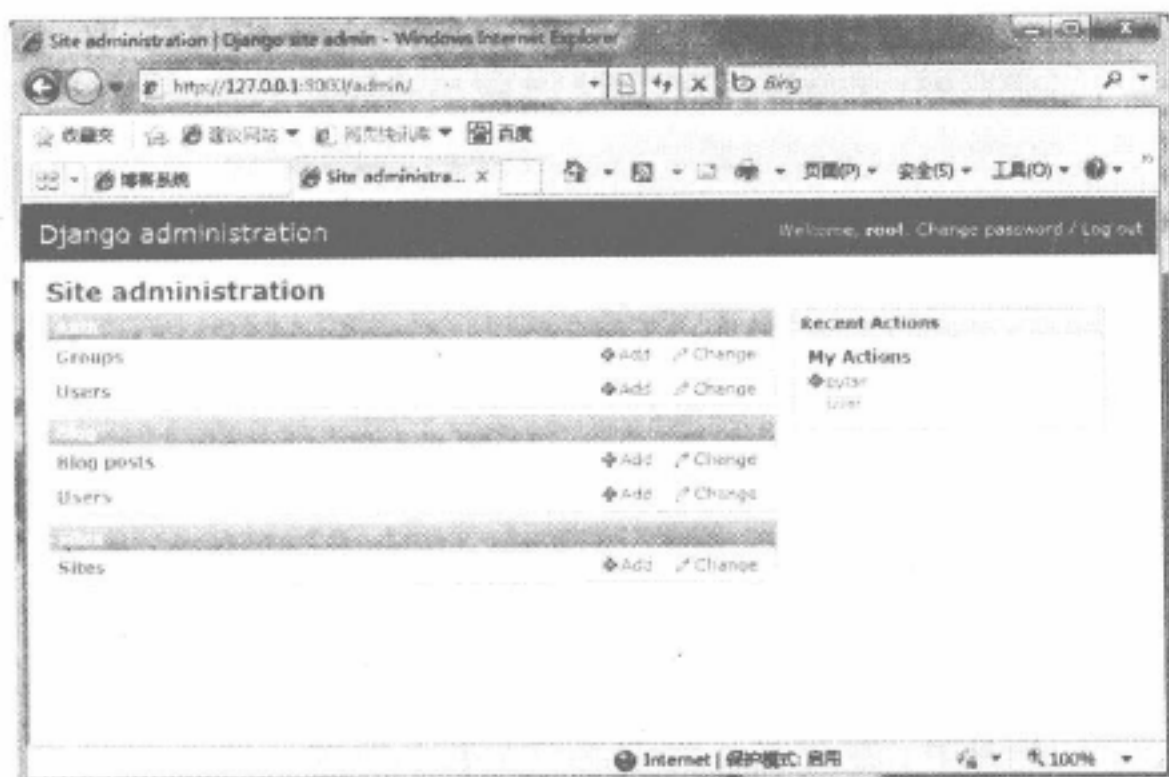


图 2 Blog 系统 admin 主页

2.5 登录验证界面的开发与实现

通过随机验证功能的添加,我们可以升级博客用

户账户的安全等级。考虑到用户对色彩敏感度的差异性,在设计随机验证码的时候设置了不同的颜色。与此同时,为了博客登录系统便捷性考虑,验证码的设计上采用了不区分大小写的方式。下面是 1 段产生随机验证码产生函数:

```
< script language = " javascript " type = " text/
javascript " >
var code ; //在全局定义验证码
function createCode() {
code = "";
var codeLength = 4; //验证码的长度
var checkCode = document.getElementById ( "ch-
eckCode" );
checkCode.value = "";
var selectChar = new Array('2','3','4','5','6','7','8','9',
'A','B','C','D','E','F','G','H','J','K','L','M','N',
'P','Q','R','S','T','U','V','W','X','Y','Z'); //从这些字符
中随机产生验证码
var colors = new Array ( " Red ", " Green ", " Gray ",
" Blue ", " Maroon ", " Aqua ", " Fuchsia ", " Lime ",
" Olive ", " Silver " ); //随机产生 1 种颜色
for( var i = 0 ; i < codeLength ; i ++ ) {
var charIndex = Math.floor( Math.random()
* 32 );
code + = selectChar[ charIndex ];
if( code.length ! = codeLength ) { createCode(); }
checkCode.innerHTML = code;
checkCode.style.color = colors[ Math.floor( Math.
random() * 10 ) ]; }
< /script >。
```

以上函数通过 4 次 (codeLength) 循环产生 4 个 0~31 的随机数作为数组 selectChar 的索引值,得到 4 个字符;同样地,函数在数组 colors 中会得到 1 个随机的颜色。然后,系统将得到的 4 位随机字符串赋值给 checkCode.innerHTML 并渲染上颜色,这个颜色也是随机产生的(见图 3)。

3 结束语

该博客系统的创建体现了 Django 在 Web 开发中的强大优势,减少了工作量,同时也缩短了开发周期。

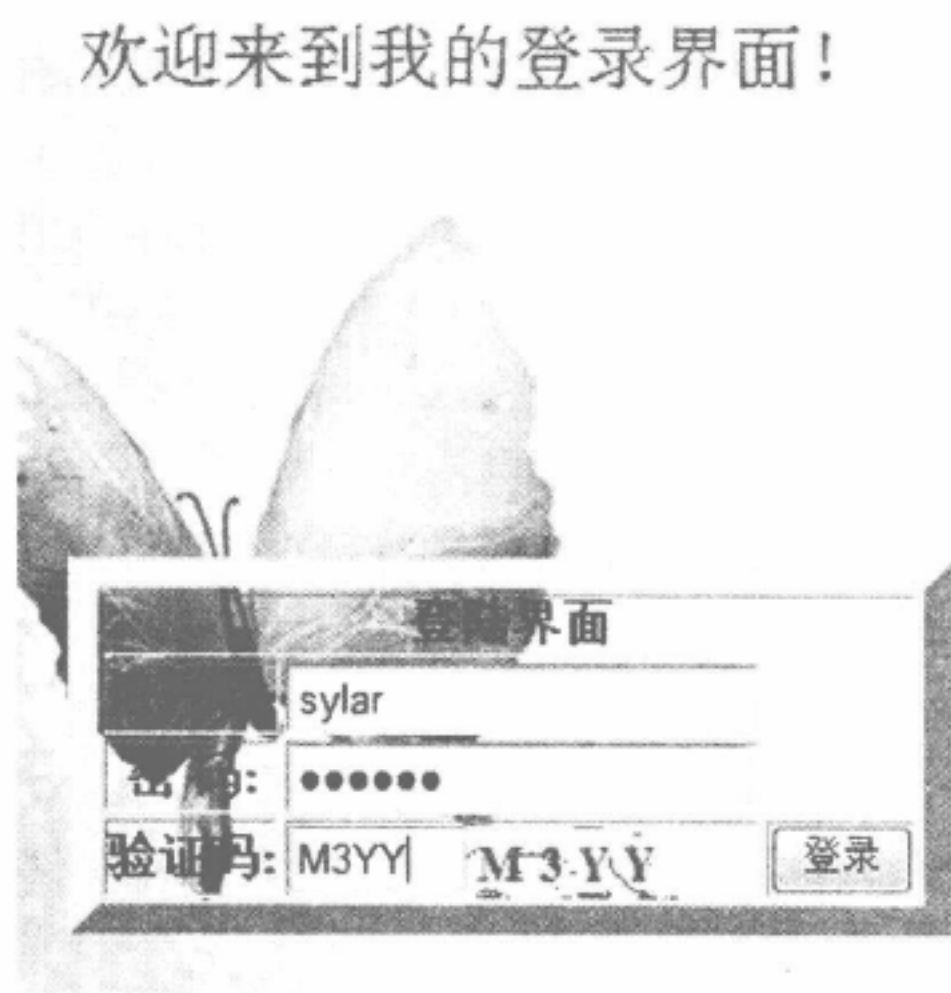


图3 带有随机验证码的登录界面

虽然该博客系统的结构简单,能实现的功能相对单一,但是具有了创建用户账户、书写博文等基本功能。此外,本系统的结构简单,组织清晰,可以在此基础上继续开发,使其功能更多、呈现画面更美。Django 的优势在于免去了构建动态 Web 站点的痛苦,把常见的 Web 开发的问题抽象出来并提供了针对频繁编程任务的捷径,极大程度提高开发速度。

(上接第 32 页)

- [3] 全国汽车标准化技术委员会. GB 11555-2009 汽车车窗玻璃除霜和除雾系统的性能和试验方法[S]. 北京:中国标准出版社,2009.
- [4] 全国汽车标准化技术委员会. GB 11562-1994 汽车驾驶员前方视野要求及测量方法[S]. 北京:中国标准出版社,2004.

(上接第 46 页)

- [2] 陈勇. 大功率机车柴油机组式活塞裙部型面研究[D]. 上海:同济大学,2002.
- [3] 马正颖. 柴油机活塞裙部润滑摩擦分析[D]. 天津:天津大学,2010.
- [4] 戴旭东,袁小阳,谢友柏. 缸套-活塞系统润滑行为与动力学行为耦合分析[J]. 摩擦学学报,2003,23(6):519-523.
- [5] 孟祥慧,谢友柏. 内燃机活塞裙部二次型线的弹性流

- [1] 董焰民,王保保. Python 与 Matlab 混合语言编程[J]. 现代电子技术,2007(14):109-110.
- [2] 孙广磊. 征服 PYTHON——语言基础与典型应用思想[M]. 北京:人民邮电出版社,2007.
- [3] 谭翼,王开,周兰江. 使用 Python 快速开发 Web 应用[J]. 电脑知识与科技,2006(35):50-51.
- [4] MARK LUTZ. Programmingpython[M]. 4th Revised edition. Sebastopol:O'Reilly and Associates,2011.
- [5] DANA MOORE, RAYMOND BUDD, WILLIAM WRIGHT. Web 2.0 Programming with Django and TurboGears Wright 2007[M]. Indianapolis: Wiley Publishing,2007.
- [6] PYTHON SOFTWARE FOUNDATION. Python 3.0.1. [EB/OL]. (2009-02-13) [2013-05-09]. <http://www.python.org/download/releases/3.0.1/>.
- [7] 刘班. 基于 Django 的快速开发 Web 应用[J]. 电脑知识与技术,2009(7):1616-1618.
- [8] JEFF FORCIER, PAUL BISSEX, CHUN WESLEY. Django Web 开发指南[M]. 徐旭铭,译. 北京:机械工业出版社,2009.
- [9] 马建乐,杨军,李琼. 基于 Django 的通用精品课程内容管理系统的快速开发[J]. 中国职工教育,2012(12):95-97.
- [10] ANDRE LESSA. 深入学习:Python 程序开发[M]. 张晓晖,张晓昕,王艳斌,等译. 北京:电子工业出版社,2001.

- [5] 莫志娇,杨易,邓峰. 车用空调风口布置 CFD 数值仿真研究[J]. 建筑热能通风空调,2008,27(1):63-67.
- [6] 朱娟娟,苏秀平,陈江平. 汽车空调除霜风道结构优化研究[J]. 汽车工程,2004,26(6):747-749.
- [7] 宋思洪,杨晨,苟小龙. 空调车室气流流场和温度场的数值模拟[J]. 计算机仿真,2004,21(9):167-169.

- 体润滑分析[J]. 润滑与密封,2008,33(11):1-4.
- [6] 成中清. 内燃机活塞环摩擦学特性对活塞动力学行为的影响研究[D]. 南宁:广西大学,2010.
- [7] 姜恩沪,夏建新,陈立志. 内燃机活塞二阶运动的模拟计算[J]. 内燃机学报,1989,7(3):265-273.
- [8] 吉林大学汽车工程系. 汽车构造[M]. 北京:人民交通出版社,2008.