

第123天: Web 开发 Django 管理工具

原创 極光 Python技术 2月10日

上次为大家介绍了 Django 的模型, 通过模型就可以操作数据库, 从而就可以改变页面的展示内容, 那问题来了, 我们只能通过手动编辑模型文件来配置模型吗? 当然不是, Django 为我们提供了强大的工具, 可以全自动地根据模型创建后台管理界面。管理界面不是为网站的访问者准备, 而是为站点管理者准备的。有了这个功能, 站点管理人员方便使用管理系统来对数据进行操作。

运行 Django Admin

首先我们上次我们已经创建了 `TestProject` 项目, 并且我们在这个项目中已经创建了 `polls` 应用, 并在该应用下我们创建了两个 models: `Question` 和 `Choice`。代码如下所示:

```
1 # polls/models.py
2
3 from django.db import models
4
5 class Question(models.Model):
6     question_text = models.CharField(max_length=200)
7     pub_date = models.DateTimeField('发布日期')
8
9
10 class Choice(models.Model):
11     question = models.ForeignKey(Question, on_delete=models.CASCADE)
12     choice_text = models.CharField(max_length=200)
13     votes = models.IntegerField(default=0)
```

接下来我们继续这个项目进行修改, 并运行 Django Admin 管理工具。

修改配置文件

在项目 `TestProject` 目录下找到 `settings.py` 文件, 打开编辑 `INSTALLED_APPS` 并增加 `django.contrib.admin` 等相关项, `django.contrib` 是一套庞大的功能集, 它是 Django 基本代码的组成部分, 而 Django 自动管理工具是 `django.contrib` 的一部分。编辑后结果如下:

```
1 # TestProject/settings.py
2
3 INSTALLED_APPS = [
4     'polls.apps.PollsConfig',
5     'django.contrib.admin',
6     'django.contrib.auth',
```

```
7     'django.contrib.contenttypes',
8     'django.contrib.sessions',
9     'django.contrib.messages',
10    'django.contrib.staticfiles',
11 ]
```

配置访问路径

在上面目录同级，有个 `urls.py` 文件，用来配置管理工具的访问路径。当然通常我们在生成项目时会在 `urls.py` 中自动设置好，我们只需去掉注释即可，请看如下代码：

```
1 # TestProject/settings.py
2
3 from django.contrib import admin
4 from django.urls import path, include
5
6 urlpatterns = [
7     path('polls/', include('polls.urls')),
8     path('admin/', admin.site.urls), # 配置 /admin 路径
9 ]
```

好了，一切都配置完成，下面我们就来运行下 Django 管理工具，和以前一样，先通过命令 `python manage.py runserver 127.0.0.1:8080` 启动本地 server，然后通过访问 URL `http://127.0.0.1:8080/admin` 就可以跳转到管理工具登陆页面，如下图所示：

登陆

已经看到登陆页面，接下来我们就需要输入用户名密码进行登陆。

初始化超级管理员

Django admin 管理工具的用户需要通过命令来初始化，回到项目根目录，执行 `python manage.py createsuperuser`，然后根据命令提示完成初始化，操作结果如下图：

初始化

用户名密码初始化完成，我们就可以在登陆页面输入，然后就能登陆到管理工具页面，如下图：

主页

管理应用

在上面登陆成功后，我们在管理页只看到用户和组相关的管理内容，这是管理工具自带的应用，并没有看到之前我们创建的 `polls` 应用，接下来我们来介绍下用 Django admin 管理工具如何管理我们的应用模型。首先修改 `polls` 应

用下的 `admin.py` 文件，增加如下代码：

```
1 # polls/admin.py
2
3 from django.contrib import admin
4 # 引入 polls 应用下的 models
5 from polls.models import Question, Choice
6
7 # 注册两个模型
8 admin.site.register(Question)
9 admin.site.register(Choice)
```

保存后退出，然后刷新主页面，就能看到我们的 `polls` 应用的模型管理界面了，如下图所示：

然后我们就可以对 `polls` 应用数据进行操作了，点击 `Questions` 模型对应的增加按钮，跳转到数据新增页面，并填写一个问题描述以及设置发布日期，如下图：

单击保存后退出到列表页面，并提示操作成功，然后用我们以前我们配置过的查询投票问题的URL `http://127.0.0.1:8080/polls/query` 进行查询，就会查到新增加的这条投票问题，查询结果如下图所示：

是不是很快捷方便？当然除了新增，还可以对数据进行修改和删除操作，你可以在列表中点击要操作的模型名下对应的记录，然后就跳转到修改和删除页面，如下图所示：

这些操作都比较简单，这里就不再详细介绍，不过需要注意的是：

1. 这个表单是从问题 `Question` 模型中自动生成的
2. 不同的字段类型（日期时间字段 `DateTimeField`、字符字段 `CharField`）会生成对应的 `HTML` 输入控件。每个类型的字段都知道它们该如何在管理页面里显示自己。
3. 每个日期时间字段 `DateTimeField` 都有 `JavaScript` 写的快捷按钮。日期有转到今天的快捷按钮和一个弹出式日历界面。时间有设为现在的快捷按钮和一个列出常用时间的方便的弹出式列表。

另外页面的底部提供了几个选项：

- 保存：保存改变，然后返回对象列表。
- 保存并继续编辑：保存改变，然后重新载入当前对象的修改界面。
- 保存并新增：保存改变，然后添加一个新的空对象并载入修改界面。
- 删除：显示一个确认删除页面。

总结

本文为大家介绍了 `Django Admin` 管理工具，可以通过应用的模型简单配置，生成出对应的后台数据管理页面，通过这个管理页面，我们可以方便的管理数据。当然除了简单模型，它还可以管理复杂模型，并可以自定义表单以及样式

等，感兴趣的朋友可以更深入的研究下。

参考

Django 中文官网: <https://docs.djangoproject.com/zh-hans/2.2>

示例代码: <https://github.com/JustDoPython/python-100-day>

系列文章

第122天: Flask 单元测试

第121天: 机器学习之决策树

从 0 学习 Python 0 - 120 大合集总结

PS: 公号内回复: Python, 即可进入Python 新手学习交流群, 一起**100天计划!**

-END-

Python 技术

关于 Python 都在这里