

《Django 教程》

- 讲师: 魏明择
- 时间: 2019

目录

- Django中的用户认证 (使用Django认证系统)
 - [auth基本模型操作:](#)
 - [自定义User模型](#)

Django中的用户认证 (使用Django认证系统)

- Django带有一个用户认证系统。它处理用户账号、组、权限以及基于cookie的用户会话。
- 作用:
 1. 添加普通用户和超级用户
 2. 修改密码
 3. 登陆和退出管理
 4. 查看已登陆用户
- 文档参见
 - <https://docs.djangoproject.com/en/1.11/topics/auth/>
- User模型类
 - 位置: `from django.contrib.auth.models import User`
- 默认user的基本属性有:

属性名	类型	是否必须存在
username	用户名	是
password	密码	是
email	邮箱	可选
first_name	名	
last_name	姓	
is_superuser	是否是管理员帐号(/admin)	
is_staff	是否可以访问admin管理界面	
is_active	是否是活跃用户,默认True。一般不删除用户,而是将用户的is_active设为False。	
last_login	上一次的登录时间	

属性名	类型	是否必须存在
date_joined	用户创建的时间	

- 数据库表现形式

```
mysql> use myauth;
mysql> desc auth_user;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
password	varchar(128)	NO		NULL	
last_login	datetime(6)	YES		NULL	
is_superuser	tinyint(1)	NO		NULL	
username	varchar(150)	NO	UNI	NULL	
first_name	varchar(30)	NO		NULL	
last_name	varchar(30)	NO		NULL	
email	varchar(254)	NO		NULL	
is_staff	tinyint(1)	NO		NULL	
is_active	tinyint(1)	NO		NULL	
date_joined	datetime(6)	NO		NULL	

```
11 rows in set (0.00 sec)
```

auth基本模型操作:

- 创建用户
 - 创建普通用户create_user

```
from django.contrib.auth import models
user = models.User.objects.create_user(username='用户名',
password='密码', email='邮箱',...)
...
user.save()
```

- 创建超级用户create_superuser

```
from django.contrib.auth import models
user = models.User.objects.create_superuser(username='用户名',
password='密码', email='邮箱',...)
...
user.save()
```

- 删除用户

```

from django.contrib.auth import models
try:
    user = models.User.objects.get(username='用户名')
    user.is_active = False # 记当前用户无效
    user.save()
    print("删除普通用户成功! ")
except:
    print("删除普通用户失败")
return HttpResponseRedirect('/user/info')

```

- 修改密码set_password

```

from django.contrib.auth import models
try:
    user = models.User.objects.get(username='laowei')
    user.set_password('654321')
    user.save()
    return HttpResponse("修改密码成功! ")
except:
    return HttpResponse("修改密码失败! ")

```

- 检查密码是否正确check_password

```

from django.contrib.auth import models
try:
    user = models.User.objects.get(username='laowei')
    if user.check_password('654321'): # 成功返回True,失败返回False
        return HttpResponse("密码正确")
    else:
        return HttpResponse("密码错误")
except:
    return HttpResponse("没有此用户! ")

```

自定义User模型

- 当django自带的 django.contrib.auth.models.User 的属性不能满足我们现有的需求时，可以自定义User模型
- 如，向用户中加入性别和家庭住址字段等
- 自定义User 类需要继承自 django.contrib.auth.models.AbstractUser, 如:

```

from django.contrib.auth.models import AbstractUser

class User(AbstractUser):
    address = models.CharField("住址", max_length=100)

```

```
mobile = models.CharField("手机号", max_length=15)
...
```

- 自定义User类后需要再次做迁移操作
- 文档参见<https://docs.djangoproject.com/en/1.11/topics/auth/customizing/>