

Django auth应用模块

在设计开发任何一个站点的时候都需要有“用户”的概念，从用户的注册、登陆、退出或者注销，到用户的浏览、订阅等，这些都需要用户管理系统来实现。如果没有时间积累和相关开发工作经验，设计一个优秀的用户系统是很困难的。那么 Django 又是如何解决这个问题的呢？

1. Django的auth模块

Django 作为功能完善的 Web 框架充分考虑到这一点，它提供的 auth 模块能够快速的实现用户模块的基本功能。

新建项目后，Django 就把 auth 模块的所有功能提供给了开发者使用，开发者可以调用相应的接口，实现不同的功能需求。auth 模块定义了一张名叫 auth_user 的数据表，该表是 auth 模块的内建用户表，开发者调用 auth 模块的相应接口生成此表，auth_user 表的字段以及字段类型，如下所示。

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(150)	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 (0.02 sec)#auth_user表

现在新建一张 auth_user 用户表，并为此表添加一个新用户 user。首先用如下方式引入 auth模块的 User 方法：

```
from django.contrib.auth.models import User
```

然后通过下面方法创建新用户 user，如下所示：

```
user=User.objects.create_user(username='c语言中文网',password='123456',email='664104694@qq.com')
save()#调用该方法保存数据
```

同时也可以使用如下方法修改密码：

```
user.set_password(password='12345abc')#会对原密码进行修改
```

根据具体的业务需求，还可以对表的字段进行增加、删除、更改。

当涉及到用户概念的时候也会产生用户权限问题，比如，如何划分普通用户和超级管理员用户？针对权限问题，Django 也提供了解决问题的方案，auth 模块提供了标准的权限管理系统，它配合 Admin 后台可以快速建立网站管理系统。

auth 模块提供了认证用户功能，可以用下面方式引入后使用：

```
from django.contrib.auth import authenticate
```

然后使用关键字传参的方法来传递用户凭证，从而达到用户认证的目的：

```
user = authenticate(username='c语言中文网', password='12345abc')
```

2. auth应用模块的其他作用

auth 模块还实现一些其它的功能，比如：

- 用户的登录（login）、退出（logout）功能，封装在 django.contrib.auth 里；
- 用户权限系统封装在 django.contrib.auth.models.Permission 中，可以对用户的权限进行增加、修改、删除；
- 用户组可以通过 from django.contrib.auth.models.Group 导入后来创建组或者删除组。

列举一些简单应用方法：

```
user.user_permissions.add(permission)#给某个用户权限的添加权限
```

```
group = Group.objects.create(name=group_name)#添加新的用户组
```

```
group.save() #保存新建好的用户组
```

```
group.delete()#删除用户组
```

3. auth模块总结

auth 模块提供的主要功能总结如下：

- 实现并维护了用户与用户组的增加、删除、更改功能；

- 实现了用户权限与用户组权限的增加、删除、更改；
- 实现了可以自定义用户权限与用户组权限功能。

除了以上功能外，Django auth 模块还提供了权限验证等功能以及一些常用的方法。

auth 模块帮助开发人员提高了工作效率，因为每个 Web 站点的权限管理模块或者用户管理模块基本都是相同的逻辑，Django 的 auth 模块使开发者们不必在为一一些重复性的逻辑进行构建。在实际开发工作中，用户模块与权限管理模块需要与实际相结合，开发者可以根据业务需求自定义或者重写相应方法，以达到和实际业务相契合的目的。