**django用户注册、登录、注销和用户扩展**

原文地址：

<https://blog.csdn.net/clh604/article/details/9269449>

用户部分是一个网站的基本功能，django对这部分进行了很好的封装，我们只需要在django的基础上做些简单的修改就可以达到我们想要的效果

首先我假设你对django的session、cookie和数据库、admin部分都有一定的了解，不了解的可以参考这个教程：<http://djangobook.py3k.cn/2.0/>

1、用户登录：

首先假设有这样的登录界面：



处理登录的视图代码如下：

**[python]** [view plain](https://blog.csdn.net/clh604/article/details/9269449) [copy](https://blog.csdn.net/clh604/article/details/9269449)

1. def userLogin(request):
2. curtime=time.strftime("%Y-%m-%d %H:%M:%S",time.localtime());
4. if request.method=='POST':
5. print("POST")
6. username=request.POST.get('name','')
7. password=request.POST.get('password','')
8. user= auth.authenticate(username=username,password=password)＃a\*\*\*\*\*\*\*\*\*\*\*
9. if user and user.is\_active:
10. auth.login(request, user)＃b\*\*\*\*\*\*\*\*\*\*\*\*
11. return HttpResponseRedirect("/user")
13. return render\_to\_response("blog/userlogin.html",RequestContext(request,{'curtime':curtime}))

注：a、这里是用django自己的auth框架验证用户名和密码，有人会说，这样太不灵活了，我想用邮箱登录呢？后面我们会说直接用django.contrib.auth.models.User 模型来直接操作用户数据，这样就可以做自己想要的验证了。

b、用户信息被验证无误后需要把用户登录的信息写入session中

2、用户注销

注销比较简单，只需要在session中删除对应的user信息就ok了

**[python]** [view plain](https://blog.csdn.net/clh604/article/details/9269449) [copy](https://blog.csdn.net/clh604/article/details/9269449)

1. def userLogout(request):
2. auth.logout(request)
3. return HttpResponseRedirect('/user')

3、用户注册

注册的界面如下：



用户名、密码、邮箱是基本的注册信息，这是django自带的，下面的电话是扩展的用户信息，至于这么扩展用户信息，一会会讲，先透露下我采用的是profile的扩展方式（个人喜好吧，我觉得这种方式简单明了）

注册的视图view代码：

**[python]** [view plain](https://blog.csdn.net/clh604/article/details/9269449) [copy](https://blog.csdn.net/clh604/article/details/9269449)

1. def userRegister(request):
2. curtime=time.strftime("%Y-%m-%d %H:%M:%S",time.localtime());
4. if request.user.is\_authenticated():#a\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
5. return HttpResponseRedirect("/user")
6. try:
7. if request.method=='POST':
8. username=request.POST.get('name','')
9. password1=request.POST.get('password1','')
10. password2=request.POST.get('password2','')
11. email=request.POST.get('email','')
12. phone=request.POST.get('phone','')
13. errors=[]
15. registerForm=RegisterForm({'username':username,'password1':password1,'password2':password2,'email':email})#b\*\*\*\*\*\*\*\*
16. if not registerForm.is\_valid():
17. errors.extend(registerForm.errors.values())
18. return render\_to\_response("blog/userregister.html",RequestContext(request,{'curtime':curtime,'username':username,'email':email,'errors':errors}))
19. if password1!=password2:
20. errors.append("两次输入的密码不一致!")
21. return render\_to\_response("blog/userregister.html",RequestContext(request,{'curtime':curtime,'username':username,'email':email,'errors':errors}))
23. filterResult=User.objects.filter(username=username)#c\*\*\*\*\*\*\*\*\*\*\*\*
24. if len(filterResult)>0:
25. errors.append("用户名已存在")
26. return render\_to\_response("blog/userregister.html",RequestContext(request,{'curtime':curtime,'username':username,'email':email,'errors':errors}))
28. user=User()#d\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
29. user.username=username
30. user.set\_password(password1)
31. user.email=email
32. user.save()
33. #用户扩展信息 profile
34. profile=UserProfile()#e\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
35. profile.user\_id=user.id
36. profile.phone=phone
37. profile.save()
38. #登录前需要先验证
39. newUser=auth.authenticate(username=username,password=password1)#f\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
40. if newUser is not None:
41. auth.login(request, newUser)#g\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
42. return HttpResponseRedirect("/user")
43. except Exception,e:
44. errors.append(str(e))
45. return render\_to\_response("blog/userregister.html",RequestContext(request,{'curtime':curtime,'username':username,'email':email,'errors':errors}))
47. return render\_to\_response("blog/userregister.html",RequestContext(request,{'curtime':curtime}))

注：

a、验证用户是否登录了，已经登录就没必要注册了（当然这只是练习使用，实际生产情况可能不一样）

b、注册表单传过来的数据需要一些基本的验证，怎么验证表单数据可以参考这个教程：<http://djangobook.py3k.cn/2.0/chapter07/>

c、用User模型查找要注册的用户名是否存在，如果用户已经存在就需要提示注册的客户更换用户名

d、直接利用User模型把通过验证的用户数据存入数据库，需要注意的是，保存密码信息时需要使用set\_password方法（因为这里有个加密的过程）

e、存储用户的扩展信息（这里是用户的电话号码），这里用到自定义的用户扩展模型UserProfile，具体怎么扩展用户后面会讲

f、用户登录前需要先进行验证，要不然会出错

g、用户登录

4、用户扩展

网上关于django的用户扩展方式有好几种，个人比较倾向于Profile的方式，主要是这种方式简单清楚，扩展步骤如下：

A、在你App的models中新建一个UserProfile模型

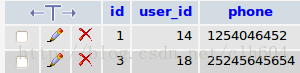
**[python]** [view plain](https://blog.csdn.net/clh604/article/details/9269449) [copy](https://blog.csdn.net/clh604/article/details/9269449)

1. from django.contrib.auth.models import User
3. class UserProfile(models.Model):
4. user=models.OneToOneField(User,unique=True,verbose\_name=('用户'))#a\*\*\*\*\*\*
5. phone=models.CharField(max\_length=20)#b\*\*\*\*\*\*

注:a、UserProfile其实就是一个普通的model，然后通过这一句与django的User模型建立联系

    b、扩展的用户信息

B、python manage.py syncdb 在数据库内创建userprofile的表



C、如何调用user的扩展信息呢?很简单,先得到user，然后通过user提供的get\_profile()来得到profile对象,比如

user.get\_profile().phone

D、如何更新和存储user的profile信息呢，其实在之前的用户注册部分我们已经使用了这样的功能，userprofile其实也是一个model，我们只要通过user模型得到user的id,就可以通过UserProfile模型来操作对应的profile信息：

**[python]** [view plain](https://blog.csdn.net/clh604/article/details/9269449) [copy](https://blog.csdn.net/clh604/article/details/9269449)

1. user=User()
2. user.username=username
3. user.set\_password(password1)
4. user.email=email
5. user.save()
6. #用户扩展信息 profile
7. profile=UserProfile()
8. profile.user\_id=user.id
9. profile.phone=phone
10. profile.save()

E、我们能在程序中操作用户扩展信息了，那我想在admin后台中编辑扩展信息要怎么做呢：

很简单，只要在你的APP的admin.py中添加下面的语句就行了

**[python]** [view plain](https://blog.csdn.net/clh604/article/details/9269449) [copy](https://blog.csdn.net/clh604/article/details/9269449)

1. class UserProfileInline(admin.StackedInline):
2. model=UserProfile
3. fk\_name='user'
4. max\_num=1
6. class UserProfileAdmin(UserAdmin):
7. inlines = [UserProfileInline, ]
9. admin.site.unregister(User)
10. admin.site.register(User,UserProfileAdmin)