

# Authentication

## User Object

مودل المستخدم . هذا المودل يتم استخدامه لتمكين الجانقو من التعرف على مستخدمي الموقع ، التعامل مع تسجيل الدخول ، عرض البيانات ، الحد من صلاحيات المستخدم و التصاريح .

مميزات موديل المستخدم الاساسية في الجانقو

- اسم المستخدم username
- كلمة المرور password
- البريد الإلكتروني email
- الإسم الأول first\_name
- الإسم الأخير last\_name

## انشاء مستخدم جديد

اسرع طريقة لإنشاء مستخدم جديد هو استيراد الموديل من auth و من ثم انشاءه باستخدام الدالة create\_user

```
#import the user model
from django.contrib.auth.models import User
user = User.objects.create_user('john', 'lennon@thebeatles.com', 'johnpassword')
user.save()
```

## التعديل على كلمة المرور

الجانقو لا يحفظ كلمة المرور بشكل نصي مباشر في قاعدة البيانات ، و انما يتم تشفيرها . لتغيير كلمة المرور لأحد المستخدمين ، نستخدم الدالة set\_password من موديل الـ User

```
u = User.objects.get(username='john')
u.set_password('new password')
u.save()
```

يمكننا ايضا التعديل على كلمة المرور من خلال الترمينال باستخدام `manage.py`

```
python manage.py changepassword *username*
```

و ايضا يمكنك تعديلها مباشرة من خلال موقع الإدارة الـ `admin`

## المصادقة على بيانات مستخدم

للمصادقة على بيانات مستخدم (التأكد من صحة هوية و معلوماته) نستخدم الدالة `authenticate` ، هذه الدالة تأخذ اسم المستخدم وكلمة المرور و تتأكد من صحتها و مطابقتها لما هو موجود على قاعدة البيانات . اذا كانت صحيحة ترجع موديل `User` أو `None`

```
#first we import authenticate
from django.contrib.auth import authenticate

user = authenticate(request, username='john', password='secret')

if user is not None:
    # A backend authenticated the credentials
else:
    # No backend authenticated the credentials
```

## تسجيل دخول المستخدم

اذا تأكدت من صحة بيانات المستخدم و أردت ان تسجل دخوله كـ جلسة `session` . يمكنك عمل ذلك من خلال الدالة `login` .

```

from django.contrib.auth import authenticate, login

def my_view(request):
    username = request.POST['username']
    password = request.POST['password']
    user = authenticate(request, username=username, password=password)

    if user is not None:
        login(request, user)
        # Redirect to a success page.
        ...
    else:
        # Return an 'invalid login' error message.

```

## تسجيل خروج المستخدم

لتسجيل خروج مستخدم من الجلسة session على الموقع . نستخدم الدالة logout

```

from django.contrib.auth import logout

def logout_view(request):
    logout(request)
    # Redirect to a success page.

```

لاحظ أن الـ **logout** ستقوم بمسح جميع بيانات المستخدم من الجلسة الحالية و اي بيانات اخرى محفوظة في الجلسة (حتى تلك الغير متعلقة بالمستخدم الحالي) .

## تقييد الدخول على الصفحة للمستخدمين المسجلين فقط

عندما تعمل `login` لمستخدم ، فإن بياناته يتم حفظها في الجلسة ، و بالتالي يتم توفير موديل المستخدم للـ `request` . و من هذا يمكننا التأكد من ان المستخدم سجل دخول و تم التأكد من صحة بيانات عبر دالة `is_authenticated` . كالتالي:

```
def my_view(request):
    if not request.user.is_authenticated:
        return render(request, 'myapp/login_error.html')
```

ملاحظة : يتم تمرير المستخدم الحالي داخل `user` في اي قالب `template` .

استخدام **decorator** لتسهيل عملية التأكد من تسجيل دخول المستخدم

```
from django.contrib.auth.decorators import login_required

@login_required
def my_view(request):
```

اذا كان المستخدم قد سجل الدخول سيتم تنفيذ الصفحة كالمعتاد، لكن اذا لم يتم بتسجيل الدخول سيتم اعادة توجيه المستخدم الى `settings.LOGIN_URL` . و تمرير رابط الصفحة التالية كـ `query` داخل المفتاح `next` . على سبيل المثال :  
`/accounts/login/?next=/polls/3/`

يمكنك ايضا ان تزود رابط لإعادة التوجيه داخل الـ `login_requierd` كالتالي :

```
@login_required(login_url="/path/to/login/")
def my_view(request):
```