

# Django Admin

وفر Django واجهة خاصة ب Admin يساعد في إنشاء وتعديل البيانات الخاصة ب Model عن طريق إنشاء Superuser قبل البدء بإنشاء Superuser، قم بالدخول على ملف settings.py، لاحظ أن أحد التطبيقات لدينا هو تطبيق Admin

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',
```

وأيضا عند فتح ملف urls.py نلاحظ وجود Admin path

```
from django.contrib import admin  
from django.urls import path  
  
urlpatterns = [  
    path('admin/', admin.site.urls),  
]
```

نقوم بفتح ملف admin.py الخاص بالتطبيق الذي نريد اضافته في الـ Admin ونقوم بكتابة الأوامر التالية:

```
1 from django.contrib import admin
2 from .models import Movies_Info
3
4 admin.site.register(Movies_Info)
5
```

في الأسطر السابقة قمنا بالتالي:

- عمل import لـ Movies\_Info, Publisher, Contributor, MovieContributor, Review
- استخدام register method وهي تساعد بجعل models متاحة لتطبيق Admin.

نقوم بتنفيذ الأوامر:

```
python manage.py createsuperuser
```

ثم نقوم بإدخال username و email و password الذي نريد، ثم كتابة الأمر.

```
python manage.py runserver
```

نقوم بالدخول على الرابط التالي (<http://127.0.0.1:8000/admin>)

الآن بعد الدخول على صفحة admin سوف نلاحظ ظهور جميع Models التي أضفنا ويمكن إضافة أو تعديل البيانات.

## Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	<a href="#">+ Add</a>	<a href="#">Change</a>
Users	<a href="#">+ Add</a>	<a href="#">Change</a>
MOVIESLISTAPP		
Contributors	<a href="#">+ Add</a>	<a href="#">Change</a>
Movie contributors	<a href="#">+ Add</a>	<a href="#">Change</a>
Movies_ infos	<a href="#">+ Add</a>	<a href="#">Change</a>
Publishers	<a href="#">+ Add</a>	<a href="#">Change</a>
Reviews	<a href="#">+ Add</a>	<a href="#">Change</a>

## تخصيص واجهات Admin

### تخصيص واجهات Admin باستخدام list\_display

حتى نستطيع عمل تخصيص (customization) لموقع Admin بحيث نستطيع عرض معلومات Publisher كجدول يحتوي معلوماته يمكن تنفيذ ذلك عن طريق list\_display attribute، بحيث نقوم بالدخول على ملف admin.py ثم تعديل التالي:

```
1 class PublisherAdmin(admin.ModelAdmin):
2     list_display = ('name', 'website', 'email')
3
4 admin.site.register(Publisher, PublisherAdmin)
```

- كتابة list\_display مع إضافة fields التي نريد ظهورها في الجدول.
- تعديل register بإضافة PublisherAdmin.
- قم بالدخول على الصفحة سوف تلاحظ أن الواجهة تغيرت كالتالي:

Select publisher to change ADD PUBLISHER +

Action:  Go 0 of 2 selected

<input type="checkbox"/>	NAME	WEBSITE	EMAIL
<input type="checkbox"/>	Publisher_B	https://www.publisherb.com	Publisher_B@example.com
<input type="checkbox"/>	Publisher_A	https://www.Publisher_A.com	Publisher_A2@gmail.com

2 publishers

## تخصيص واجهات Admin باستخدام list\_filter

إحدى الطرق أيضا لعمل تخصيص على ModelAdmin عن طريق list\_filter وهي:

- تسهل علينا الوصول للمعلومات عندما يكون لدينا معلومات كثيرة.
- تشبه filter الظاهر في الجانب ( by staff status, by superuser status, and active )

بداخل admin.py نقوم بكتابة التالي:

```

1 class MoviesAdmin(admin.ModelAdmin):
2     list_filter = ('publisher',)
3
4 admin.site.register(Movies_Info, MoviesAdmin)
```

نقوم بتشغيل الموقع و نلاحظ أن الواجهة تغيرت وتم إضافة filter بناء على publisher كما في الشكل التالي:

WELCOME, **ADMIN**. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Select movies\_ info to change

ADD MOVIES\_ INFO +

Action: 

-----

Go

 0 of 1 selected

☐ MOVIES\_ INFO

☐ **Movie1**

1 movies\_ info

FILTER

By publisher

All

Publisher\_A

Publisher\_B

الإسم الظاهر لنا في Filter هو اسم publisher وهو يمثل مآكتبناه عن طريق ميثود () \_\_str\_\_ الموجودة في Publisher model