

Article

[Dmitry Maslennikov](#) · Jul 25, 2022 8m read

## Introduction to Django part 2

In the first part, I've shown how to start a new project on Django, as well as define new models and add already existing models.

This time, I'll introduce an admin panel, available out of the box and how it can be useful.

Important note: do not expect that if you try to repeat actions from this post it will work for you, it does not. During the article, I had to do some fixes in the django-iris project, and even in DB-API driver made by InterSystems to fix some issues there as well, and I think this driver is still in development, and we will get more stable driver in future. Let's decide that this article only explains how it could be if we would have all done.

Let's return to our code and see what we have in urls.py, the main entrypoint for all web requests.

```
"""main URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/4.0/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path

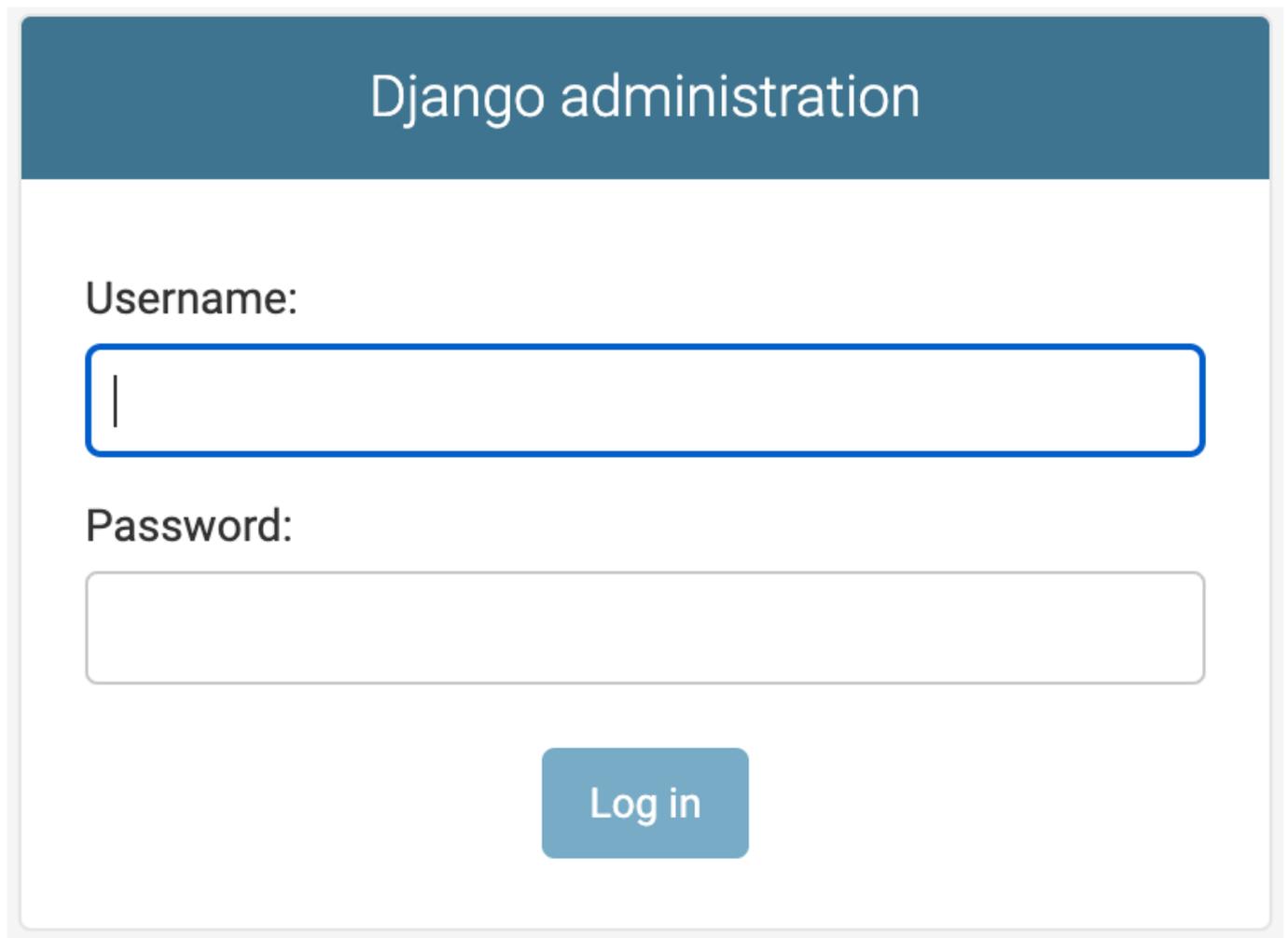
urlpatterns = [
    path('admin/', admin.site.urls),
]
```

You may see, that there is already URL admin defined there

Let's start development server, by command

```
python manage.py runserver
```

If you go by URL <http://localhost:8000/admin>, you will the login form to Django administration



Django administration

Username:

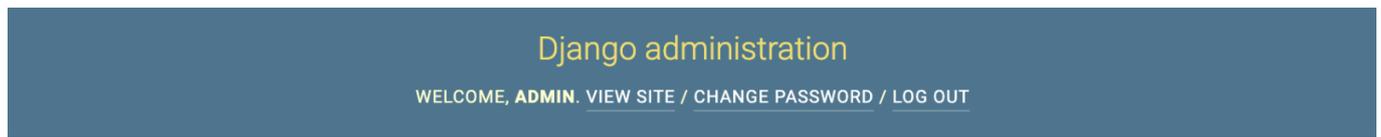
Password:

Log in

To enter here, we need some user, and we can create it with this command

```
$ python manage.py createsuperuser
Username (leave blank to use 'daimor'): admin
Email address: admin@example.com
Password:
Password (again):
The password is too similar to the username.
This password is too short. It must contain at least 8 characters.
This password is too common.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
```

We can use this login and password now. It's quite empty at the moment, but it already gives access to Groups and Users.



## Site administration

AUTHENTICATION AND AUTHORIZATION		
<b>Groups</b>	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
<b>Users</b>	<a href="#">+ Add</a>	<a href="#">✎ Change</a>

### Recent actions

#### My actions

None available

## More data

Previously I've already installed post-and-tags package with zpm, you can do it too

```
zpm "install posts-and-tags"
```

Now we can get the models for all tables (community.post, community.comment, community.tag) installed by this package

```
$ python manage.py inspectdb community.post community.comment community.tag > main/models.py
```

This will produce a bit long file, so, I put it in a spoiler

```
main/models.py
```

The administration dashboard in Django, can be extended by developer. And it's possible to add some more tables. To do so, we need to add a new file named main/admin.py. I've added a few comments right in the code, to explain some lines.

```
from django.contrib import admin

# import our community models for our tables in IRIS
from .models import (CommunityPost, CommunityComment, CommunityTag)

# register class which overrides default behaviour for model CommunityPost
@admin.register(CommunityPost)
class CommunityPostAdmin(admin.ModelAdmin):
    # list of properties to show in table view
    list_display = ('posttype', 'name', 'publisheddate')
    # list of properties to show filter for on the right side of the tablee
    list_filter = ('posttype', 'lang', 'published')
    # default ordering, means from the latest date of PublishedDate
    ordering = ['-publisheddate', ]

@admin.register(CommunityComment)
class CommunityCommentAdmin(admin.ModelAdmin):
```

```
# only this two fields show, (post is numeric by id in table post)
list_display = ('post', 'created')
# order by date of creation
ordering = ['-created', ]
```

```
@admin.register(CommunityTag)
class CommunityTagAdmin(admin.ModelAdmin):
    # not so much to show
    list_display = ('name', )
```

## Extending portal

Let's return back to our Django administration page, and see, some new items added there

The screenshot shows the Django administration interface for 'Community posts'. The main content area displays a table of posts with the following data:

POSTTYPE	NAME	PUBLISHEDDATE
Question	How to programmatically decide whether a session have unresolved errors in Interoperability?	2022-06-10 11:03:07
Question	How to Copy NTE and append segments HL7 ensemble	2022-06-10 11:03:07
Question	Not able to connect to IRIS data platform in AWS using JDBC	2022-06-10 11:03:07
Announcement	See Schedule of Morning Presentations	2022-06-10 11:03:07
Question	ADO.NET component InterSystems.Data.IRISClient is not closing the connections even after calling Close() method.	2022-06-10 11:03:07
Question	Unable to logon to server ENSEMBLE.	2022-06-09 20:03:34
Announcement	Coursera Subscription - new reward on Global Masters	2022-06-09 20:03:34
Article	Debugging Trick with SQL	2022-06-09 20:03:34
Question	Error #6301 and #5090 - Compiling Pipeline	2022-06-09 20:03:34
Question	Restrict ODBC Access Based On Application	2022-06-09 20:03:33
Question	PowerBI and IRIS generating "The column '...' of the table wasn't found."	2022-06-09 20:03:33
Question	Search bar interoperability prining through invalid action	2022-06-09 20:03:33
Digest	Unanswered Questions on InterSystems Developers for June 08, 2022.	2022-06-09 20:03:33

The filter panel on the right shows the following options:

- By posttype: All, ANNOUNCEMENT, ARTICLE, DIGEST, DISCUSSION, QUESTION, -
- By lang: All, EN
- By published: All, Yes, No, Unknown

On the right side, you will see the filter panel, and what's important is, that it's got all the possible values in particular fields and showed there.

Unfortunately, InterSystems SQL does not support LIMIT, OFFSET feature, expected on or another way by Django. And Django does not support TOP. So, pagination here will be shown but does not work. And no way to make it work at the moment, and I don't think will ever work, unfortunately.

You can even dive into the object, and Django will show form, with correct field types. (note: This dataset does not contain data in the Text field)

Django administration WELCOME, ADMIN VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Main > Community posts > CommunityPost object (519921)

Start typing to filter...

- AUTHENTICATION AND AUTHORIZATION
  - Groups + Add
  - Users + Add
- MAIN
  - Community comments + Add
  - Community posts + Add
  - Community tags + Add

### Change community post

CommunityPost object (519921) HISTORY

Acceptedanswers:  Date:  Today | 📅  
Time:  Now | 🕒  
Note: You are 4 hours ahead of server time.

Author:  📄

Avgvote:

Commentsamount:

Created:  Date: 2022-06-09 Today | 📅  
Time: 16:29:06 Now | 🕒  
Note: You are 4 hours ahead of server time.

Deleted:  ▾

Favscount:

Hascorrectanswer:  ▾

Hash:

Lang:

Name:

Nid:

Start typing to filter...

- AUTHENTICATION AND AUTHORIZATION
  - Groups + Add
  - Users + Add
- MAIN
  - Community comments + Add
  - Community posts + Add
  - Community tags + Add

Name:

Nid:

Posttype:

Published:  ▾

Publisheddate:  Date: 2022-06-10 Today | 📅  
Time: 11:03:07 Now | 🕒  
Note: You are 4 hours ahead of server time.

Subscount:

Tags:

Text:

Translated:  ▾

Type:

Views:

Votesamount:

## Comments object

Start typing to filter...

- AUTHENTICATION AND AUTHORIZATION
  - Groups + Add
  - Users + Add
- MAIN
  - Community comments + Add
  - Community posts + Add
  - Community tags + Add

### Change community comment

CommunityComment object (comment|192386) HISTORY

Id1:

Acceptedanswers: **Date:**  **Time:**  Note: You are 4 hours ahead of server time.

Author:

Avgvote:

Correct:

Created: **Date:**  **Time:**  Note: You are 4 hours ahead of server time.

Hash:

Id:

Post:

Text: 

```
<p>Michael</p>
<p>Thanks for the interest, and I will update you if there are details we can share! At this point, the work is still exploratory and a couple approaches are in the works</p>
<p>Kind Regards,<br>Thomas</p>
```

Texthash:

Type:

Issues with licenses to expect on Community Edition

[#Python](#) [#InterSystems](#) [IRIS](#)

Source URL: <https://community.intersystems.com/post/introduction-django-part-2>