

Project Djora - Our Quora Clone

Its finally time to start with our mega project, we have all the basics that we need to start working on this project. However, do note that we just had a brief introduction to various features of Django, now its time to start diving in-depth.

I would advise to approach this project slowly and let all the concept sink in before moving to the next topic, we are going to build a lot of production grade functionality and new topics will be built upon the previous one.

Django also has amazing documentation, but it can be really intimidating for beginner, so I will be posting links from the official documentation for your reference.

Custom User

Previously we have used the default user model that ships with Django, however its **highly recommended** to create your own user model and is the officially recommended approach.

[Ref - Extending Default User Model](#)

User model can be extended using different approaches, we will be using '**AbstractUser**', its simple and effective. Incase you like to have more control over the User model, you can use '**AbstractBaseUser**', however covering it is beyond the scope of this tutorial.

[Ref - Default User Model](#)

[Ref - AbstractBaseUser](#)

[Ref - AbstractUser](#)

Project Setup

We would be starting with a fresh project -

```
$ mkdir djora
$ cd djora
$ pipenv install django
$ pipenv shell
$ django-admin startproject djora_project .
$ python manage.py startapp users
```

Register the app in project settings -

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

```
'django.contrib.sessions',  
'django.contrib.messages',  
'django.contrib.staticfiles',  
'users.apps.UsersConfig' # new  
)
```

Note -

DO NOT MIGRATE!

The reason for this is, we would be creating a custom model for our user and we need to migrate this custom model in order to signup all our users including superuser.

Extending User

We use a simple and robust approach for creating our custom user by extending '**AbstractUser**' model that already ships with Django.

We would be customizing abstract user and adding fields as desired by us.

There are two main changes that we will be making -

1. Earlier when signing up superuser we could skip the email and still sign up user, we want to change this and make **email** as **required**.
2. Change primary key from an integer to **UUID**.

Steps

1. Create custom user model extending 'AbstractUser'.
2. Register custom user in project settings.
3. Make user signup forms using the custom user model.
4. Adding custom user to admin.
5. Migrating our database

Note -

We are not yet adding views and templates, in the next chapter we will see how to create user profiles and then create all the views & templates.

However, we can create and check custom user using Django 'shell'.