**Ставим Django:** pip install Django  
  
**Создаём новый проект:** python -m django startproject myproject  
  
**Создаём новое приложение внутри проекта:** python manage.py startapp orders

**Добавляем приложение в INSTALLED\_APPS в myproject/settings.py**  
  
**Создание модели данных, открываем orders/models.py и вносим данный код:**  
from django.db import models

class Customer(models.Model):

name = models.CharField(max\_length=255)

email = models.EmailField(unique=True)

phone = models.CharField(max\_length=20)

def \_\_str\_\_(self):

return self.name

class Product(models.Model):

name = models.CharField(max\_length=255)

description = models.TextField()

price = models.DecimalField(max\_digits=10, decimal\_places=2)

def \_\_str\_\_(self):

return self.name

class Order(models.Model):

customer = models.ForeignKey(Customer, on\_delete=models.CASCADE)

product = models.ForeignKey(Product, on\_delete=models.CASCADE)

quantity = models.IntegerField()

order\_date = models.DateTimeField(auto\_now\_add=True)

total\_price = models.DecimalField(max\_digits=10, decimal\_places=2)

def save(self, \*args, \*\*kwargs):

self.total\_price = self.quantity \* self.product.price

super().save(\*args, \*\*kwargs)

def \_\_str\_\_(self):

return f"Order #{self.id} for {self.customer.name}"

**Делаем регистрацию модели в админке:** открываем orders/admin.py и добавляем регистрацию моделей в админку Django:  
  
from django.contrib   
import admin from .models import Customer, Product, Order   
admin.site.register(Customer)   
admin.site.register(Product)

admin.site.register(Order)  
  
**После создаём миграции в базе данных:**   
python manage.py makemigrations orders  
  
**Применяем миграции:** python manage.py migrate  
  
Теперь в бд будут таблицы Customer, Product и Order  
  
**Создаём кастомную команду, все 3 команды вводим в Powershell:**  
New-Item -Path . -Name "orders" -ItemType "Directory"

New-Item -Path "orders" -Name "management" -ItemType "Directory"

New-Item -Path "orders\management" -Name "commands" -ItemType "Directory"  
  
**Теперь создаём пустой файл \_\_init\_\_.py:**   
New-Item -Path "orders\management\commands" -Name "\_\_init\_\_.py" -ItemType "File"  
  
**Создаём файл для кастомной команды generate\_test\_data.py:**  
New-Item -Path "orders\management\commands" -Name "generate\_test\_data.py" -ItemType "File"  
  
**Добавляем в generate\_test\_data.py следующий код:**  
from django.core.management.base import BaseCommand

from orders.models import Customer, Product, Order

import random

from faker import Faker

fake = Faker()

class Command(BaseCommand):

help = 'Генерация тестовых данных для заказов'

def handle(self, \*args, \*\*kwargs):

for \_ in range(10):

customer = Customer.objects.create(

name=fake.name(),

email=fake.email(),

phone=fake.phone\_number()

)

for \_ in range(5):

product = Product.objects.create(

name=fake.word(),

description=fake.text(),

price=random.randint(100, 1000)

)

customers = Customer.objects.all()

products = Product.objects.all()

for \_ in range(20):

order = Order.objects.create(

customer=random.choice(customers),

product=random.choice(products),

quantity=random.randint(1, 5)

)

self.stdout.write(self.style.SUCCESS('Тестовые данные успешно сгенерированы'))  
  
**Ставим библиотеку faker – фейк данные:** pip install faker  
  
**Запускаем кастомную команду для генерации тестовых данных:**  
  
python manage.py generate\_test\_data – это создаст случайных заказчиков, продукты и заказы в бдшке.  
  
**Открываем orders/views.py и вносим следующий код:**  
  
from django.http import JsonResponse

from orders.models import Order

def order\_list(request):

orders = Order.objects.select\_related('customer', 'product').all()

data = [

{

'order\_id': order.id,

'customer': order.customer.name,

'product': order.product.name,

'quantity': order.quantity,

'total\_price': order.total\_price,

'order\_date': order.order\_date.strftime('%Y-%m-%d %H:%M:%S')

}

for order in orders

]

return JsonResponse(data, safe=False)  
  
Открываем или создаём orders/urls.py и вносим следующий код:  
  
from django.urls import path

from .views import order\_list

urlpatterns = [

path('orders/', order\_list, name='order\_list'),

]  
  
Добавляем URL для этого представления в orders/urls.py:  
from django.urls import path

from .views import order\_list

urlpatterns = [

path('orders/', order\_list, name='order\_list'),

]  
  
**Подключаем orders.urls в myproject/urls.py:**  
from django.contrib import admin

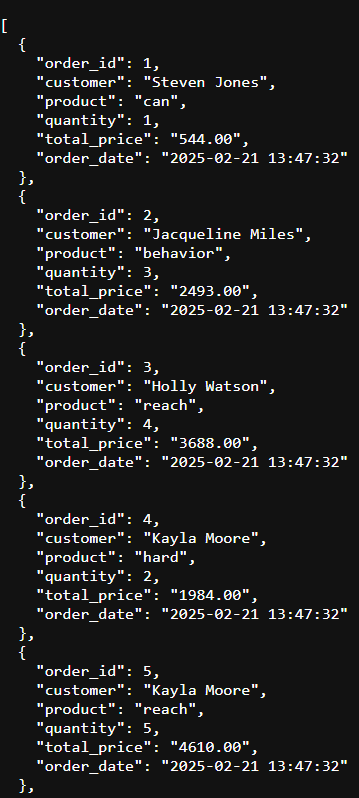
from django.urls import path, include

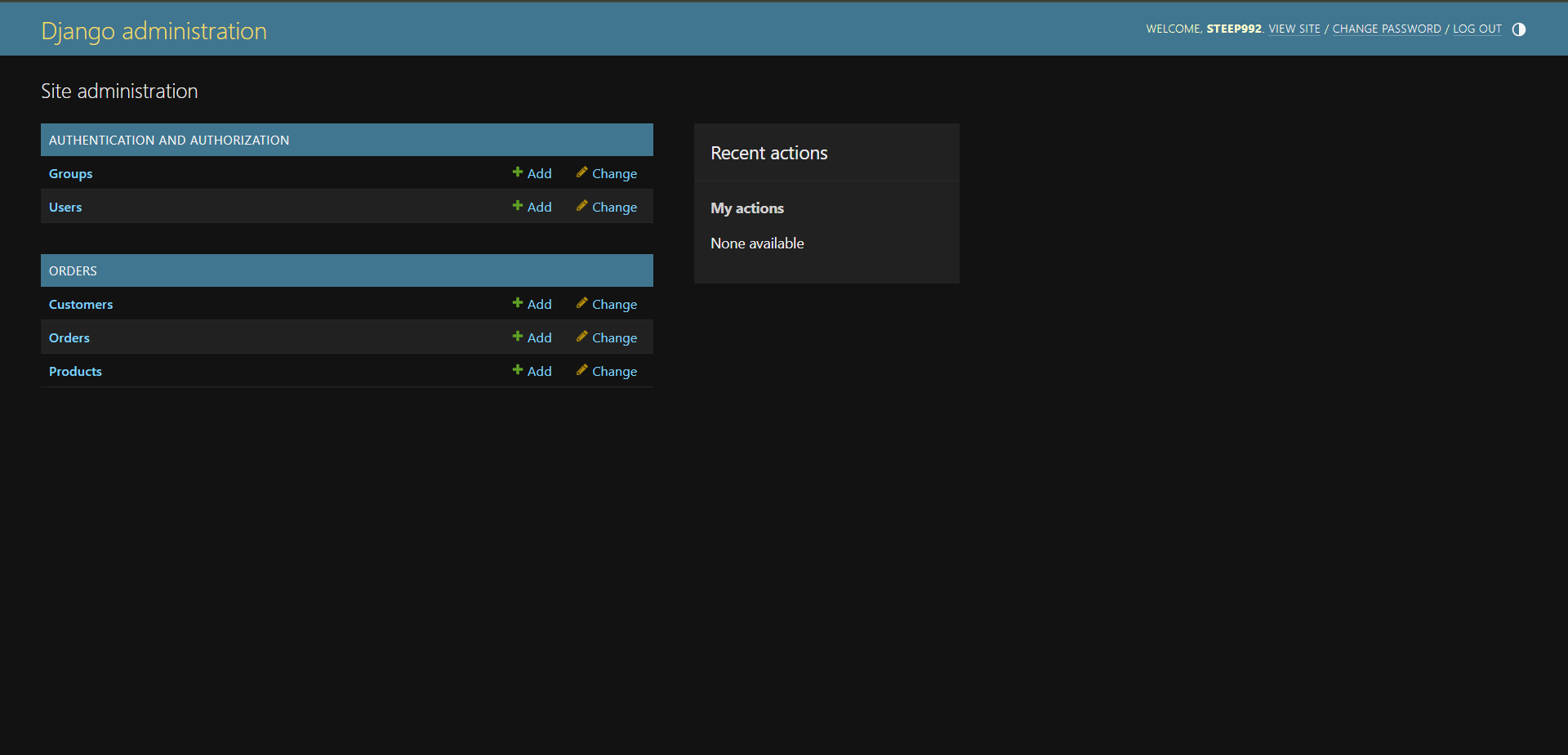
urlpatterns = [

path('admin/', admin.site.urls),

path('api/', include('orders.urls')), # Добавляем путь к API

]

**Теперь если мы запустим сервер и перейдём по** [**http://127.0.0.1:8000/api/orders/**](http://127.0.0.1:8000/api/orders/) **то получим все заказы в формате JSON:**  


**Создание суперпользователя в админку Django:** python manage.py createsuperuser, логин и пароль запоминаем, после переходим в админку: <http://127.0.0.1:8000/admin/>, авторизуемся и увидим то что сделали: ORDERS(Customers, Orders, Products)  
Все инструкции взяты из открытых источников!