

## Достоинство отладочного сервера:

При внесении изменений в проект он автоматически перезапускается с учетом сделанных изменений

### Смена порта

По умолчанию команда `runserver` запускает сервер разработки на внутреннем IP адресе с портом 8000.

Для смены порта передайте его аргументом в командной строке. Например, эта команда запускает сервер на порту 8080:

```
python manage.py runserver 8080
```



# Создание приложения Django в составе проекта **film**

## Проекты и приложения

В чем разница между проектом и приложением? Приложение - это веб-приложение, которое что-то делает, например, система Weblog, база данных публичных записей или небольшое приложение для опроса. Проект - это набор настроек и приложений для определенного сайта. Проект может содержать несколько приложений. Приложение может быть в нескольких проектах.

При создании приложения важно убедиться, что находитесь в том же каталоге, что и файл `manage.py`!



# Создание приложения в составе Django-проекта **film**

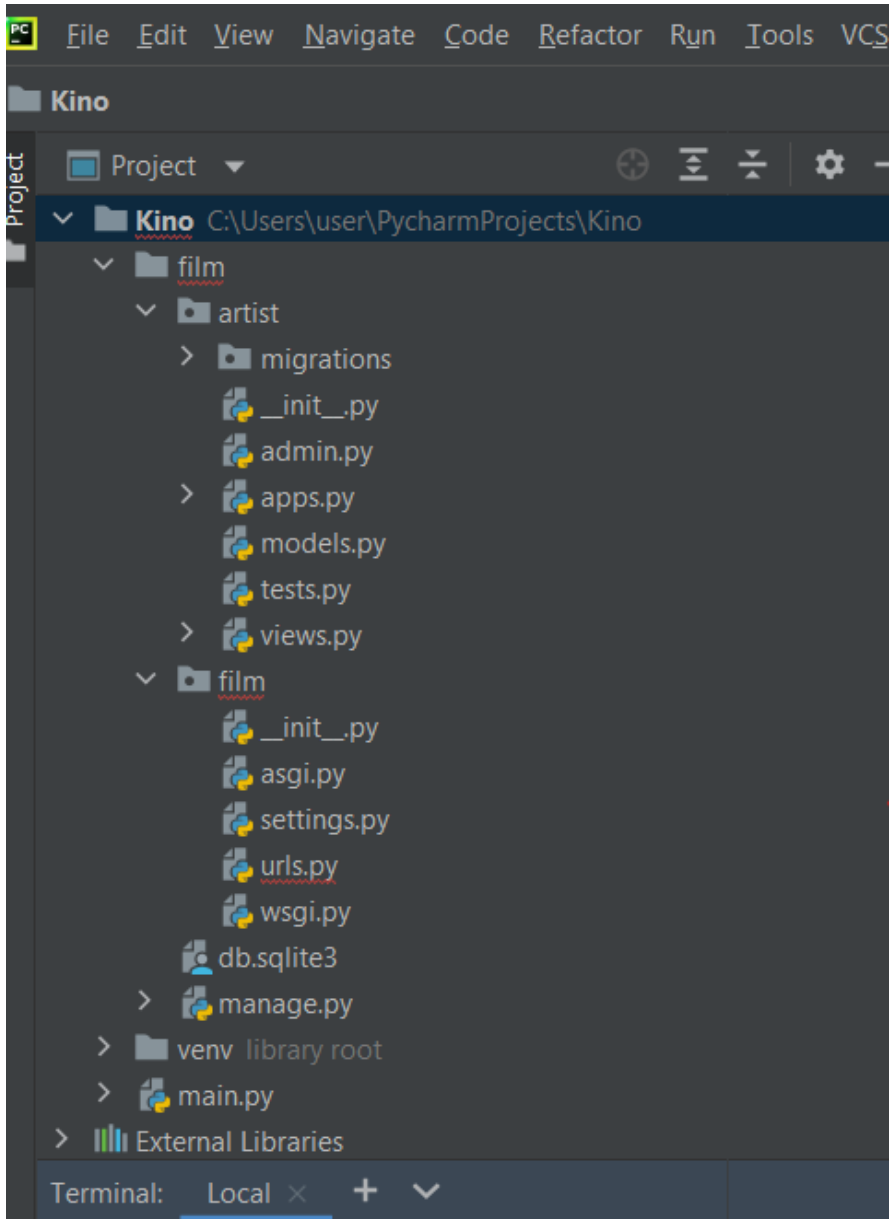
1. В окне терминала PyCharm перейдем в папку film (cd film) и выполним команду:

**python manage.py startapp artist** (artist - имя приложения. Имя может быть любым)

## Структура папок и файлов приложения **artist**

- Папка migrations – хранит информацию для сопоставления базы данных со структурой данных, описанных в модели.
- Файл `__init__.py` – указывает Python, что текущий каталог будет рассматриваться в качестве пакета.
- Файл `admin.py` – административные функции.
- Файл `apps.py` – содержит конфигурацию приложения.
- Файл `models.py` – содержит описание моделей данных, используемых приложением.
- Файл `tests.py` – тесты приложения.
- Файл `views.py` – функции, которые получают запросы пользователей, обрабатывают их и возвращают ответ.





## Структура папок и файлов приложения artist

- Папка migrations – хранит информацию для сопоставления базы данных со структурой данных, описанных в модели.
- Файл `__init__.py` – указывает Python, что текущий каталог будет рассматриваться в качестве пакета.
- Файл `admin.py` – административные функции.
- Файл `apps.py` – содержит конфигурацию приложения.
- Файл `models.py` – содержит описание моделей данных, используемых приложением.
- Файл `tests.py` – тесты приложения.
- Файл `views.py` – функции, которые получают запросы пользователей, обрабатывают их и возвращают ответ.



# Регистрация приложения **artist** в проекте Django

В файле **film/settings.py** в конец массива с приложениями проекта **INSTALLED\_APPS** добавим приложение **artist**:

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



```
settings x  
26 # SECURITY WARNING: don't run with debug turned on in production!  
27 DEBUG = True  
28  
29 ALLOWED_HOSTS = []  
30  
31  
32 # Application definition  
33  
34 INSTALLED_APPS = [  
35     'django.contrib.admin',  
36     'django.contrib.auth',  
37     'django.contrib.contenttypes',  
38     'django.contrib.sessions',  
39     'django.contrib.messages',  
40     'django.contrib.staticfiles',  
41     'artist',  
42  
43 ]
```



# Программирование отклика приложения **artist**

Корректировка файла представлений views.py

по умолчанию файл views.py выглядит так:

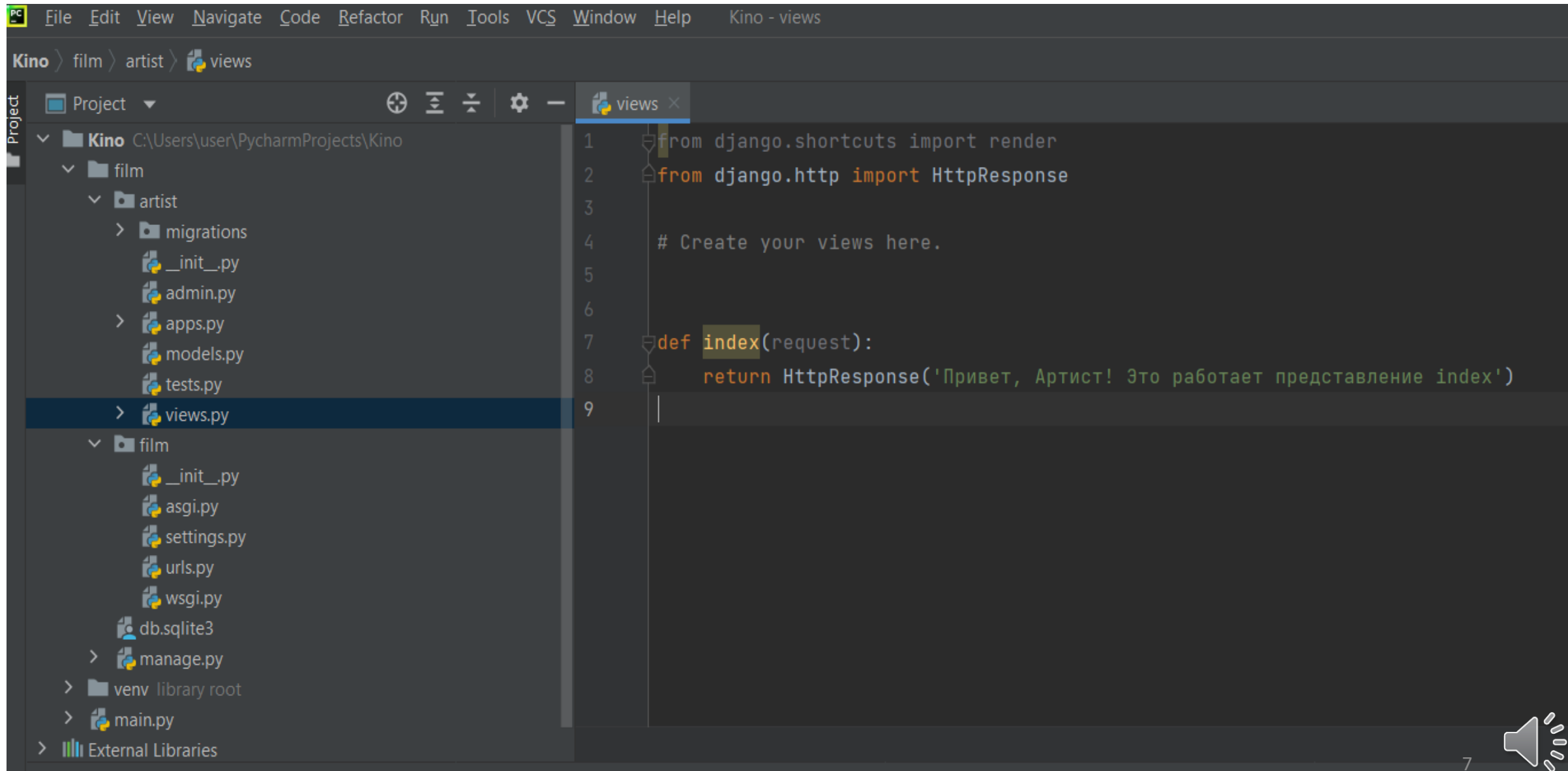
```
from django.shortcuts import render    # Ф-я render (предоставлять) для обработки входящих  
                                       запросов
```

```
#Create your views here
```

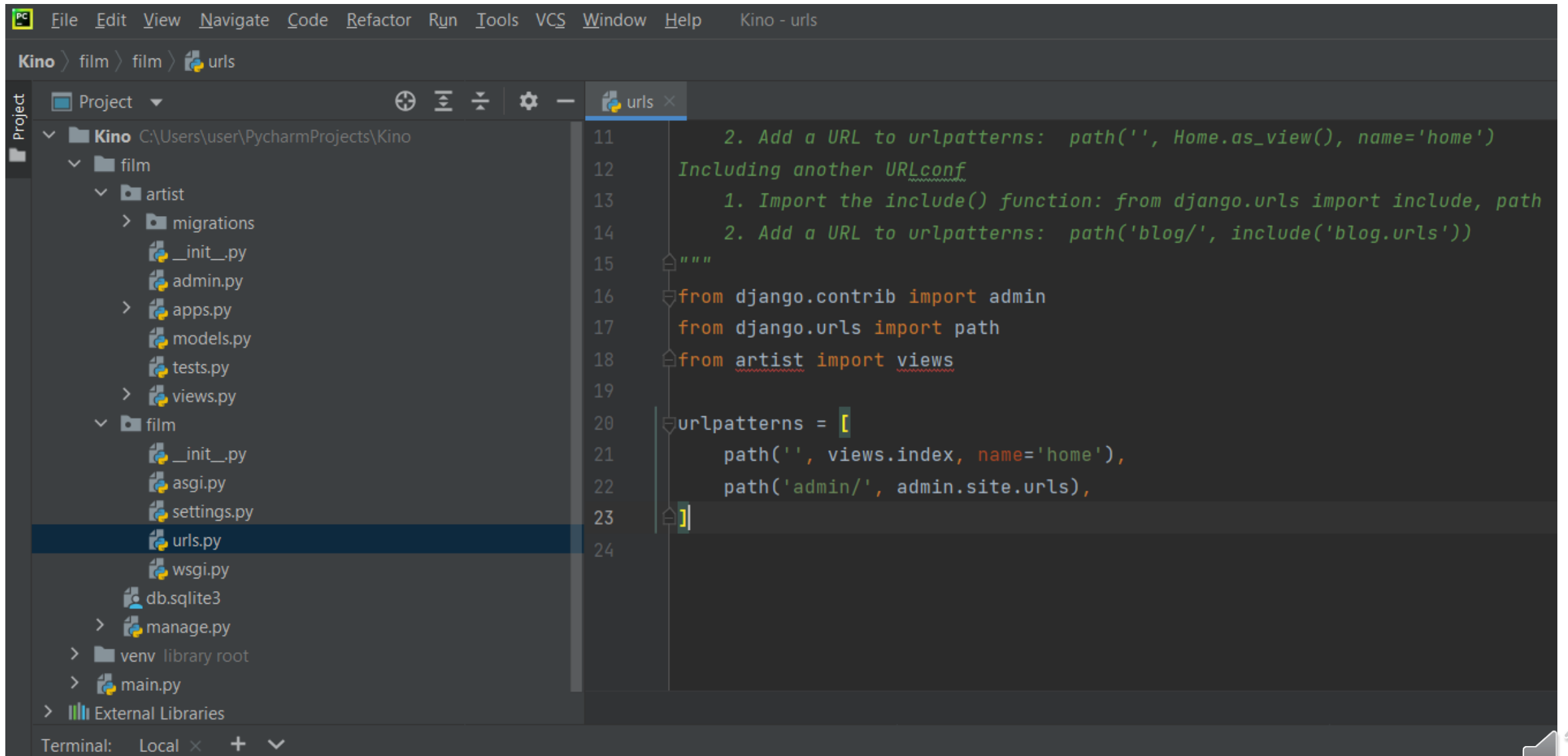
Дополним файл и создадим новые функции:



В файле views.py создадим функцию index():



В файле urls.py пропишем маршрут для функции index():



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help Kino - urls
Kino > film > film > urls
Project
  Kino C:\Users\user\PycharmProjects\Kino
  film
    artist
      migrations
      __init__.py
      admin.py
      apps.py
      models.py
      tests.py
      views.py
    film
      __init__.py
      asgi.py
      settings.py
      urls.py
      wsgi.py
      db.sqlite3
      manage.py
    venv library root
    main.py
  External Libraries
11      2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
12      Including another URLconf
13      1. Import the include() function: from django.urls import include, path
14      2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15      """
16      from django.contrib import admin
17      from django.urls import path
18      from artist import views
19
20      urlpatterns = [
21          path('', views.index, name='home'),
22          path('admin/', admin.site.urls),
23      ]
24
```



В терминальном режиме снова запустим локальный (отладочный) веб-сервер:

```
python manage.py runserver
```

В окне Терминала появится ссылка для перехода на локальный сервер:

<http://127.0.0.1:8000>



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help Kino - urls
Kino > film > film > urls

Project
Kino C:\Users\user\PycharmProjects\Kino
film
  artist
    migrations
      __init__.py
      admin.py
    apps.py
    models.py
    tests.py
    views.py
  film
    __init__.py
    asgi.py
    settings.py
    urls.py
    wsgi.py
  db.sqlite3
  manage.py
  venv library root
  main.py
  External Libraries

Terminal: Local x + v
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, con
Run 'python manage.py migrate' to apply them.
November 21, 2021 - 16:40:39
Django version 3.2.9, using settings 'film.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

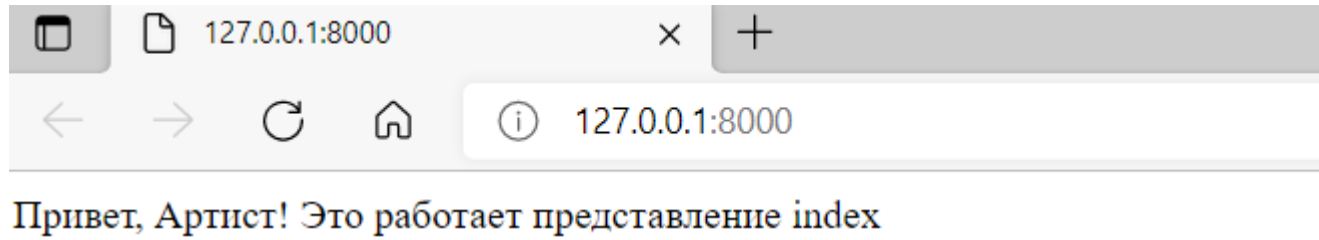
```
11 2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
12 Including another URLconf
13 1. Import the include() function: from django.urls import include, path
14 2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15 """
16 from django.contrib import admin
17 from django.urls import path
18 from artist import views
19
20 urlpatterns = [
21     path('', views.index, name='home'),
22     path('admin/', admin.site.urls),
23 ]
24
```

В окне Терминала  
появится ссылка  
для перехода на  
локальный сервер:  
<http://127.0.0.1:8000>

Перейдем по этой  
ссылке



В браузере отобразится страница:



# Документация Django

[Документация Django 3.1 на русском языке | Документация Django 3.1 | Все о фреймворке Джанго и его библиотеках](#)

[Документация Django — Документация Django 3.0 \(djbook.ru\)](#)



**Спасибо за внимание!**

