**Практическая работа: Создание приложения**

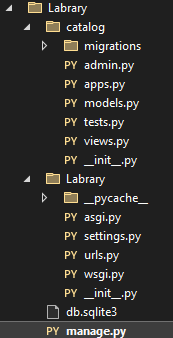
**Цель:**

Научится создавать приложения и выполнять подключения к БД в проекте Django.

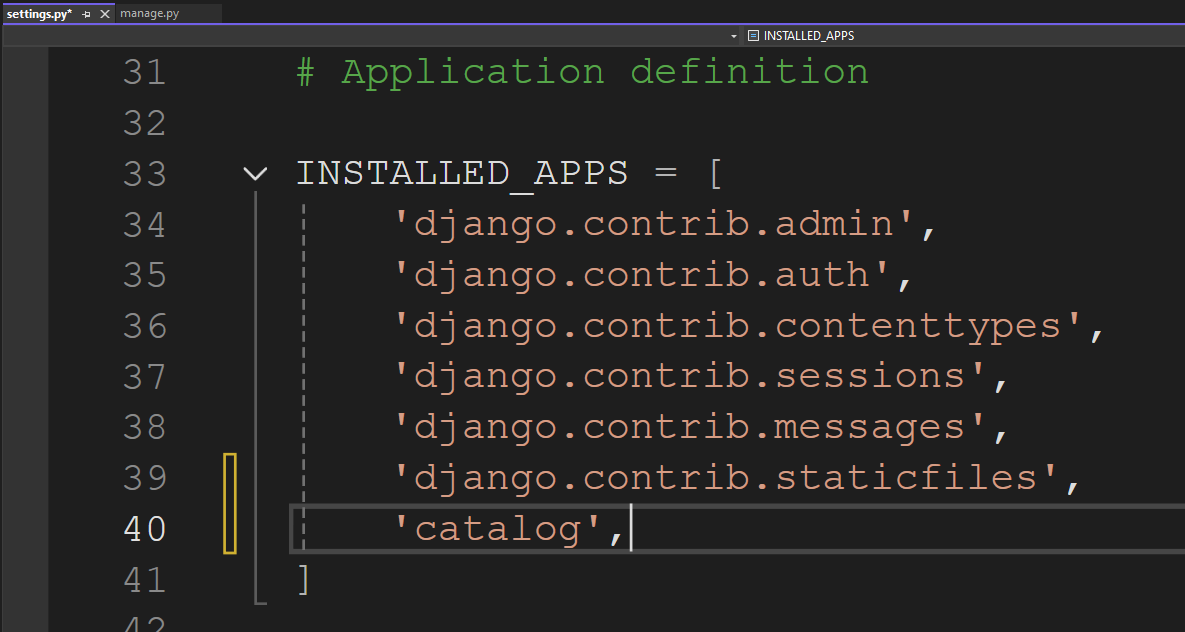
**Порядок выполнения:**

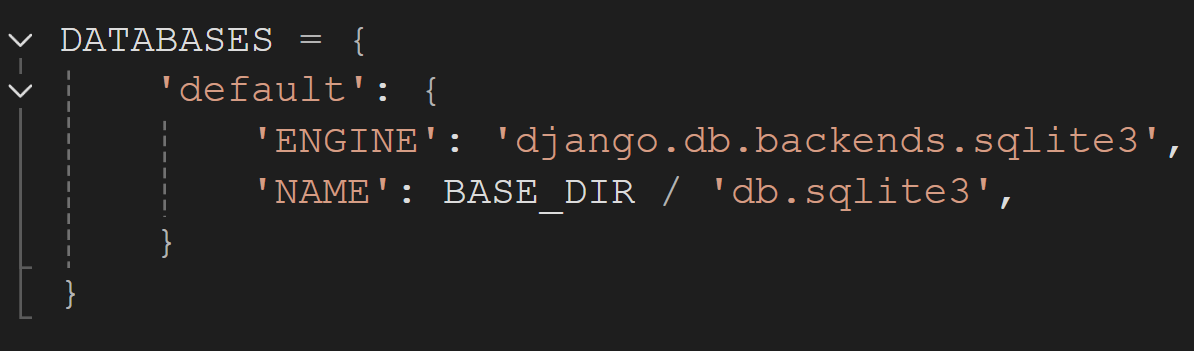
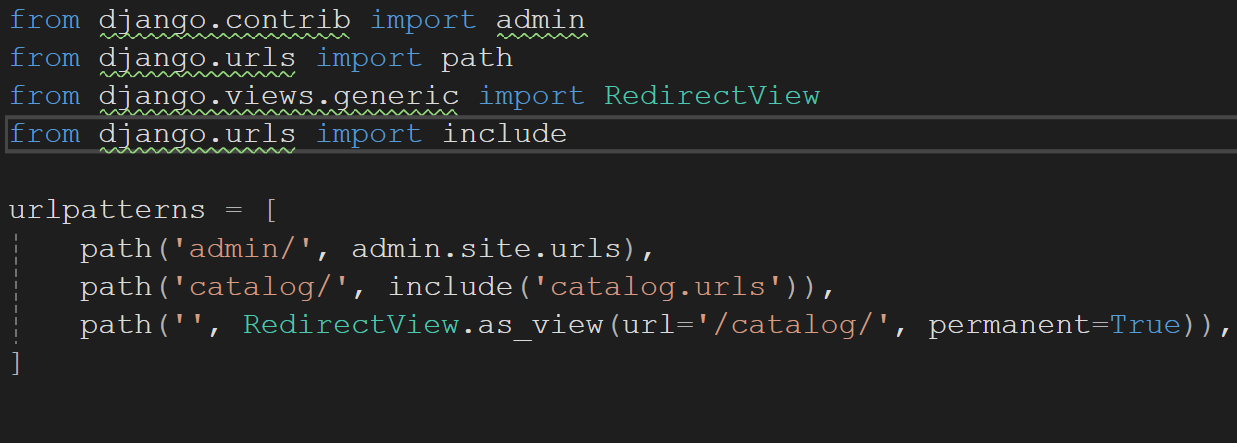
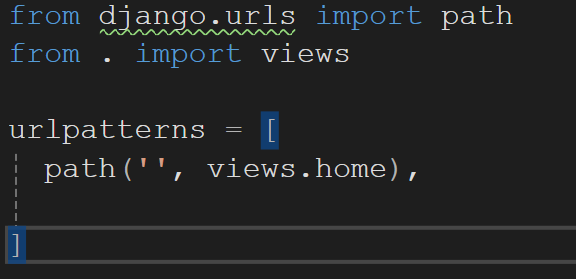
1. Откройте проект предыдущей практической работы и активируйте среду окружения.
2. Создайте приложение каталог выполнив следующую команду: python manage.py startapp catalog.

В результате в структуре проекта должна появится папка catalog



1. После создания приложения, нам нужно зарегистрировать его в проекте, чтобы различные утилиты затрагивали его своим действием. Приложения регистрируются добавлением их названий в список INSTALLED\_APPS в настройках проекта(который, как мы помним, называется **settings.py**). Откройте файл **library/locallibrary/settings.py** и найдите в нём список INSTALLED\_APPS . Затем добавьте новую строку в конец списка:



1. Настройка базы данных. На этом шаге обычно указывают базу данных для будущего проекта. Мы будем использовать базу данных SQLite для этого проекта, потому что не предполагаем большое количество одновременных запросов на неё. Вы можете видеть, что база данных уже настроена в **settings.py.  
   **
2. Измените в файле **settings.py** переменную TIME\_ZONE на одну из строк из таблицы, которая отвечает вашему часовому поясу <https://en.wikipedia.org/wiki/List_of_tz_database_time_zones>  
   Например, TIME\_ZONE = 'Europe/Moscow'
3. В файле присутствует две настройки, которые не нужно менять сейчас, но о назначении которых следует знать:
   1. SECRET\_KEY. Это секретный ключ, который используется Django для поддержки безопасности сайта. Если вы раскроете этот ключ в процессе разработки кому-либо, то необходимо будет его сменить (возможно считать его с какого-либо файла на сервере или переменной окружения) когда будете размещать проект на сервер.
   2. DEBUG. Включает подробные сообщения об ошибках, вместо стандартных HTTP статусов ответов. Должно быть изменено на False на сервере, так как эта информация очень много расскажет взломщикам.
4. Замените файл views.py в папке catalog на файл views.py в папке Library созданный на прошлом уроке.
5. Измените значение переменной urlpatterns в файле urls.py таким образом, чтобы он ссылался на файл views.py в папке catalog. Запустите сервер и проверьте работоспособность ссылки <http://127.0.0.1:8000>  
   Запустилась ли нужная страница?
6. Перенаправим корневой URL нашего сайта (например 127.0.0.1:8000) на URL 127.0.0.1:8000/catalog/  
   Чтобы это использовать, нам понадобится специальная функция (RedirectView), которая принимает первым параметром новый относительный URL на который следует перенаправлять (/catalog/) когда указанный в функции url() адрес соотносится с адресом запроса (корневой URL, в данном случае).  
   Импортируйте необходимую функцию и добавьте строку которая будет перенаправлять корневой URL и измените файл следующим образом:  
   
7. Создайте файл **urls.py** внутри папки **catalog**, и добавьте следующий код, чтобы определить urlpatterns. Сюда мы будем добавлять наши URL соотношения, по мере разработки сайта.  
   
8. Запустите сервер и протестируйте работу главной страницы.