

Московский государственный технический университет
имени Н. Э. Баумана

В. И. Кузнецов

Web фреймворк Django

*Методические указания к выполнению
лабораторной работы №8 по дисциплине
«Языки интернет программирования»*

Редакция от 1.09.2025

**Факультет «Информатика и системы управления»
Кафедра «Компьютерные системы и сети»**

Кузнецов В.И.

Web фреймворк Django: методические указания к выполнению практикума № 8 и лабораторной работы №8 по дисциплине «Языки интернет-программирования» / В. И. Кузнецов.

В методическом пособии приведены основы фреймворка Django. Так же приведены инструкции по разворачиванию и запуску

Для студентов МГТУ им. Н.Э. Баумана, обучающихся по направлению «Информатика и вычислительная техника».

Теоретическая часть

Django — это высокоуровневый фреймворк для разработки веб-приложений на языке Python. Он создан для того, чтобы облегчить создание сложных, масштабируемых и безопасных сайтов с минимальными усилиями. Основная философия Django — принцип “Don’t repeat yourself” (DRY), то есть «не повторяйся».

Фреймворк появился в 2005 году и с тех пор стал одним из самых популярных инструментов для веб-разработки. Его активно используют такие компании, как Instagram, Spotify, Mozilla, Pinterest, Disqus и другие.

Django позволяет разработчику сосредоточиться на логике приложения, предоставляя готовые решения для: работы с базами данных, маршрутизации (URL), аутентификации пользователей, администрирования данных, шаблонизации страниц, API и REST-сервисов (через Django REST Framework).

Преимущества и недостатки Django

Преимущества:

- Быстрая разработка и масштабируемость
- Встроенная защита от XSS, CSRF и SQL-инъекций
- Админ-панель «из коробки»
- ORM и REST-интеграция

Недостатки:

- Избыточность для мелких проектов
- Крутая кривая обучения

Архитектура и структура проекта

Django основан на архитектурном шаблоне MVC (Model-View-Controller), но использует немного изменённую терминологию — MTV (Model-Template-View):

Model — описывает структуру и логику работы с данными (обычно таблицы в базе данных)

Template — отвечает за отображение информации пользователю, HTML-шаблон

View — обрабатывает запросы, управляет логикой и выбирает, какие данные и шаблон вернуть

Примеры синтаксиса

Модели (Models)

Модель определяет структуру данных, хранящихся в базе данных. Django использует ORM (Object-Relational Mapping) — это способ взаимодействия с базой через Python-классы вместо SQL-запросов.

Пример модели:

```
from django.db import models

class Post(models.Model):
    title = models.CharField(max_length=100)
    text = models.TextField()
    created_at = models.DateTimeField(auto_now_add=True)
```

Модели (Models)

Модель определяет структуру данных, хранящихся в базе данных. Django использует ORM (Object-Relational Mapping) — это способ взаимодействия с базой через Python-классы вместо SQL-запросов.

Пример модели:

```
from django.db import models

class Post(models.Model):
    title = models.CharField(max_length=100)
    text = models.TextField()
    created_at = models.DateTimeField(auto_now_add=True)
```

Панель администратора

Одно из сильных преимуществ Django — автоматически создаваемая админ-панель.

После добавления приложения в INSTALLED_APPS и регистрации модели:

```
from django.contrib import admin
from .models import Post
```

```
admin.site.register(Post)
```

можно перейти по адресу <http://127.0.0.1:8000/admin/> и управлять записями через веб-интерфейс.

Работа с базой данных

По умолчанию Django использует SQLite, но легко подключается к другим СУБД: PostgreSQL, MySQL, Oracle.

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'mydb',
        'USER': 'user',
        'PASSWORD': 'pass',
        'HOST': 'localhost',
        'PORT': '5432',
    }
}
```

URL-маршрутизация

```
urlpatterns = [
    path('', views.index, name='home'),
    path('post/<int:id>', views.detail, name='detail'),
]
```

Django REST Framework (DRF)

DRF упрощает разработку RESTful API. Она предоставляет сериализацию, аутентификацию и пагинацию данных.

Пример сериализатора:

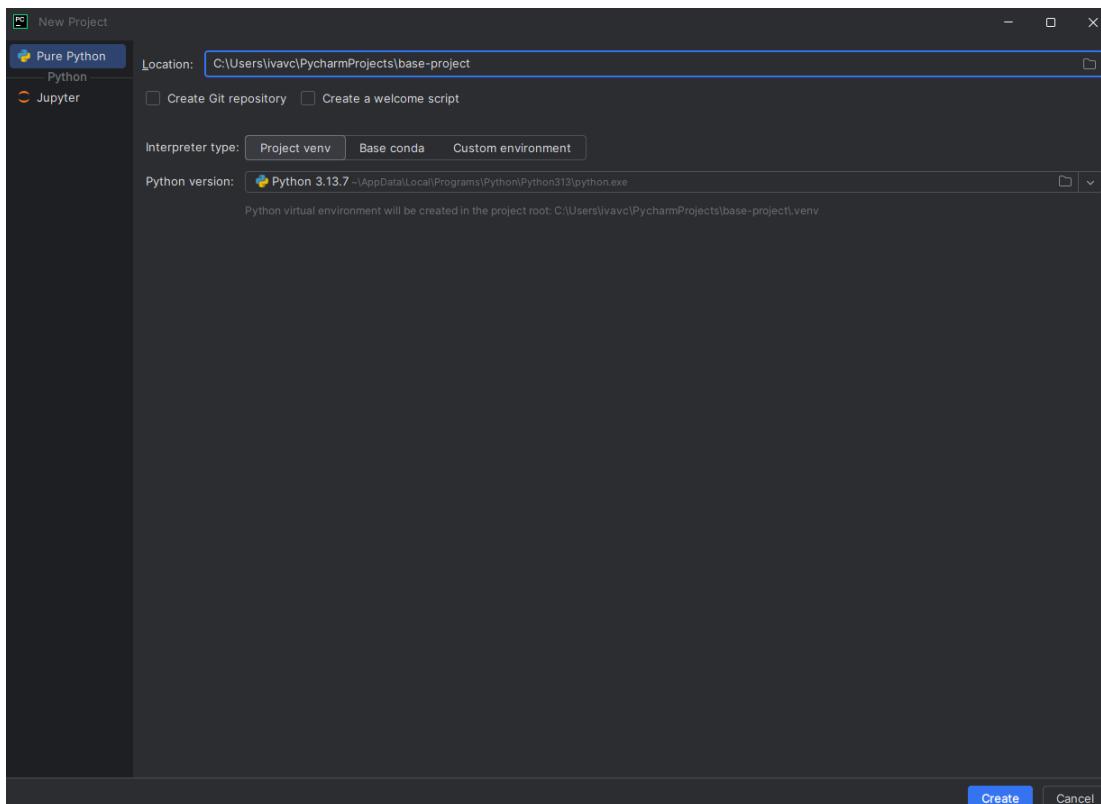
```
from rest_framework import serializers
from .models import Post

class PostSerializer(serializers.ModelSerializer):
    class Meta:
        model = Post
        fields = '__all__'
```

Практическая часть

Для создания приложения необходимо иметь установленную версию интерпретатора Python старше 3 версии. Для удобства разработки рекомендуется среда разработки JetBrains PyCharm. Для

Для начала необходимо создать пустой проект в среде разработки.



Далее открыв терминал windows через среду, разработки необходимо выполнить команду `pip install django`. Данная команда выполниться в виртуальном окружении python и будет хранится изолированно для данного проекта с данным виртуальным окружением.

The screenshot shows the PyCharm IDE interface. In the top navigation bar, it says "base-project" and "Version control". The main area is a terminal window titled "Terminal Local (2)". The terminal output shows the following command and its execution:

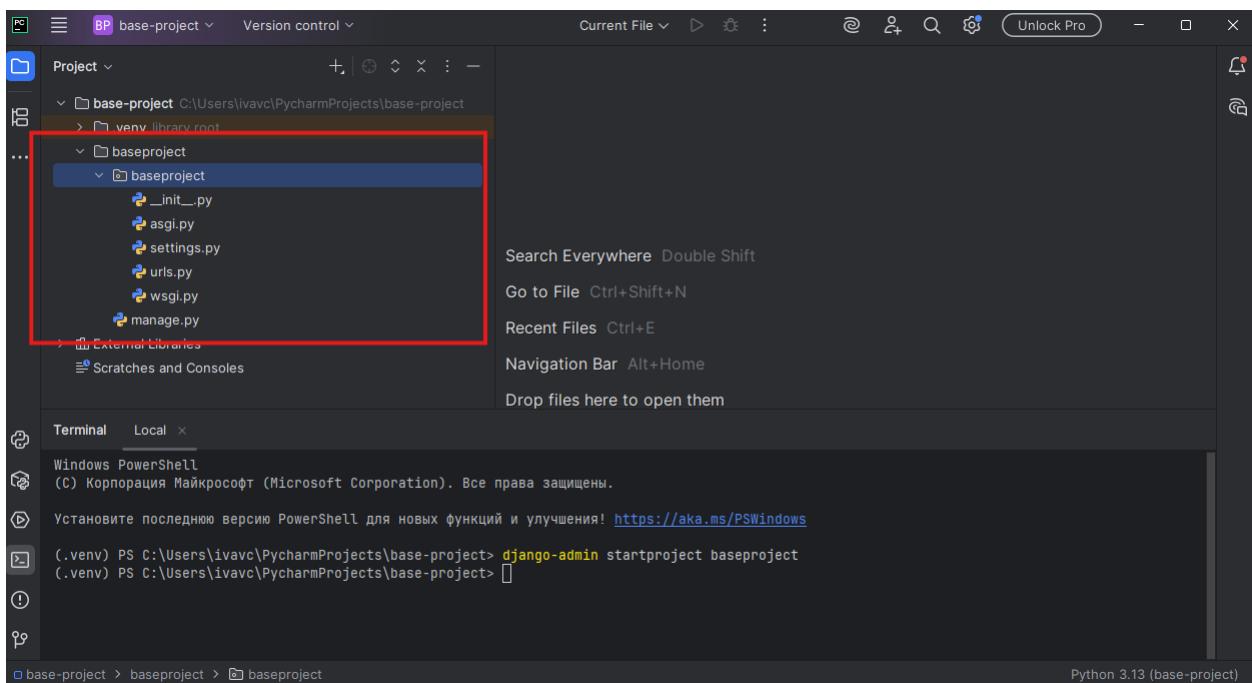
```
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Установите последнюю версию PowerShell для новых функций и улучшения! https://aka.ms/PSWindows

(.venv) PS C:\Users\livavc\PycharmProjects\base-project> pip install django
Collecting django
  Using cached django-5.2.8-py3-none-any.whl.metadata (4.1 kB)
  Collecting asgiref>=3.8.1 (from django)
    Downloading asgiref-3.11.0-py3-none-any.whl.metadata (9.3 kB)
  Collecting sqlparse==0.3.1 (from django)
    Using cached sqlparse-0.5.3-py3-none-any.whl.metadata (3.9 kB)
  Collecting tzdata (from django)
    Using cached tzdata-2025.2-py2.py3-none-any.whl.metadata (1.4 kB)
  Using cached django-5.2.8-py3-none-any.whl (8.3 MB)
  Downloading asgiref-3.11.0-py3-none-any.whl (24 kB)
  Using cached sqlparse-0.5.3-py3-none-any.whl (44 kB)
  Using cached tzdata-2025.2-py2.py3-none-any.whl (347 kB)
Installing collected packages: tzdata, sqlparse, asgiref, django
Successfully installed asgiref-3.11.0 django-5.2.8 sqlparse-0.5.3 tzdata-2025.2

[notice] A new release of pip is available: 25.1.1 -> 25.3
[notice] To update, run: python.exe -m pip install --upgrade pip
(.venv) PS C:\Users\livavc\PycharmProjects\base-project>
```

Далее необходимо создать само приложение Django в нашем пустом проекте. Для этого необходимо выполнить команду `django-admin startproject baseproject`



Базовое приложение создано и его уже можно запустить. Для этого необходимо перейти в каталог созданного проекта и выполнить команду запуска приведенную на скриншоте.

```

urls.py      manage.py      __init__.py      asgi.py      settings.py
1  #!/usr/bin/env python
2  """Django's command-line utility for administrative tasks."""
3
4  import ...
5
6
7  def main():
8      """Run administrative tasks."""
9      os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'baseproject.settings')
10     try:
11         from django.core.management import execute_from_command_line
12     except ImportError as exc:
13         raise ImportError(
14             "Couldn't import Django. Are you sure it's installed and "
15             "available on your PYTHONPATH environment variable? Did you "

```

Terminal Local +

Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Установите последнюю версию PowerShell для новых функций и улучшения! <https://aka.ms/PSWindows>

(.venv) PS C:\Users\ivavc\PycharmProjects\base-project> cd ..\baseproject<
(.venv) PS C:\Users\ivavc\PycharmProjects\base-project\baseproject> python manage.py runserver

Watching for file changes with StatReloader

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, contenttypes, sessions.

Run 'python manage.py migrate' to apply them.

November 21, 2025 - 22:56:03

Django version 5.2.8, using settings 'baseproject.settings'

Starting development server at <http://127.0.0.1:8000/>

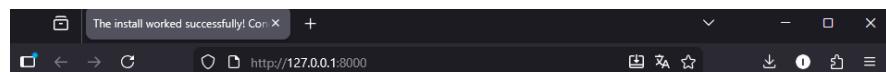
Quit the server with CTRL-BREAK.

WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead.

For more information on production servers see: <https://docs.djangoproject.com/en/5.2/howto/deployment/>

1:1 CRLF UTF-8 4 spaces Python 3.13 (base-project)

Запущенное приложение выведет количество невыполненных миграций баз данных и выведет адрес и порт по которому доступно приложение и при открытии данного адреса в браузере мы получим стартовую страницу пустого приложения.



The install worked successfully! Congratulations!

[View release notes](#) for Django 5.2

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.

django

Для завершения работы сервера достаточно нажать комбинацию клавиш **Cntrl+C**.

Задания

Задание 1

Создать базовое пустое приложение Django

Задание 2

Создать метод GET возвращающий по пути /student/ текст с ФИО студента и номером группы

Задание 3

Создать метод POST принимающий сообщение {"lab8": "pass"} и отправляет пустой успешный ответ

Требования к отчету

1. Приложить скриншоты каждой выполняемой консольной команды
2. Приложить скриншоты логов запущенного приложения
3. В отчете предоставить проведенные запросы к серверу через среды тестирования POSTMAN/INSOMNIA

Контрольные вопросы

1. Что такое Django и для чего он используется?
2. Что такое «модель» (Model) в архитектуре Django и зачем она нужна?
3. Для чего используется файл settings.py в Django-проекте?
4. Что делает команда python manage.py runserver?
5. Что такое URLconf (urls.py) и какую роль он играет в Django-приложении?