

Водич за Django

Вовед во Django

Django е високо ниво Python веб фрејмворк кој го олеснува развојот на веб апликации со брзина и безбедност. Django следи "Don't Repeat Yourself" (DRY) принцип и Model-View-Template (MVT) архитектура.

Инсталација

Предуслови:

- Python (верзија 3.8+)
- pip (Python пакет менаџер)
- Virtualenv (препорачливо за изолирана средина)

Инсталирање на Django

1. Креирање на виртуелна средина:
2. `python -m venv myenv`
3. `source myenv/bin/activate` # За Linux/macOS
4. `myenv\Scripts\activate` # За Windows
5. Инсталација на Django:
6. `pip install django`
7. Проверка на инсталацијата:
8. `django-admin --version`

Креирање на Django проект

1. Креирање на нов проект:
2. `django-admin startproject myproject`
3. Навигација до проектната папка:
4. `cd myproject`
5. Стартување на развоен сервер:
6. `python manage.py runserver`

Структура на проектот

- `manage.py` – главен скрипт за управување со проектот.
- `settings.py` – конфигурациски поставки.

- `urls.py` – дефинирање на патеки (routes).
- `models.py` – дефиниција на модели (бази на податоци).
- `views.py` – дефиниција на логика на апликацијата.
- `templates/` – HTML темплејти.

Работа со апликации

Креирање на апликација

```
python manage.py startapp myapp
```

Регистрација на апликацијата

Додајте ја апликацијата во `INSTALLED_APPS` во `settings.py`:

```
INSTALLED_APPS = [
    ...,
    'myapp',
]
```

Модели и база на податоци

Дефинирање на модел во `models.py`

```
from django.db import models
```

```
class Product(models.Model):
    name = models.CharField(max_length=255)
    price = models.DecimalField(max_digits=10, decimal_places=2)
    created_at = models.DateTimeField(auto_now_add=True)
```

Миграции

1. Генерирање на миграции:
2. `python manage.py makemigrations`
3. Аплицирање на миграции:
4. `python manage.py migrate`

Django Admin

Креирање на супер корисник

```
python manage.py createsuperuser
```

Додавање на модел во Admin Panel (admin.py)

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

```
admin.site.register(Product)
```

Views и Templates

Дефинирање на view во views.py

```
from django.shortcuts import render
```

```
def home(request):  
    return render(request, 'home.html')
```

Дефинирање на патека во urls.py

```
from django.urls import path  
from .views import home
```

```
urlpatterns = [  
    path("", home, name='home'),  
]
```

Креирање на темплејт во templates/home.html

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
    <title>Welcome</title>  
</head>  
  
<body>  
    <h1>Welcome to Django</h1>  
</body>  
</html>
```

Forms и User Authentication

Дефинирање на форма (forms.py)

```
from django import forms  
from .models import Product
```

```
class ProductForm(forms.ModelForm):  
    class Meta:  
        model = Product  
        fields = ['name', 'price']
```

User Authentication (Login/Logout/Register)

Django вклучува систем за автентикација:

```
from django.contrib.auth import login, logout, authenticate
```

REST API со Django Rest Framework (DRF)

Инсталирање на DRF

```
pip install djangorestframework
```

Конфигурација во settings.py

```
INSTALLED_APPS = [  
    ..., 'rest_framework',  
]
```

Дефинирање на API View (views.py)

```
from rest_framework.decorators import api_view  
from rest_framework.response import Response
```

```
@api_view(['GET'])  
def api_home(request):  
    return Response({"message": "Hello, API!"})
```

Додавање на API route (urls.py)

```
from django.urls import path  
from .views import api_home
```

```
urlpatterns = [  
    path('api/', api_home, name='api_home'),  
]
```

Deployment

Подготовка за продукција

1. Конфигурирајте ALLOWED_HOSTS во settings.py
2. Користете PostgreSQL наместо SQLite
3. Конфигурирајте DEBUG = False

Хостирање на сервер (Heroku, AWS, DigitalOcean)

```
pip install gunicorn
```

Заклучок

Django е моќен веб фрејмворк кој овозможува брз развој и лесно скалирање на апликации. Со овој водич имате основа за изградба на модерни веб апликации.

1. Креирање на Django проект

► Основни поими:

- **Проект:** Целата веб-апликација (на пример, блог, продавница, портал).
- **Ап (апликација):** Поделба на функционалности во проектот (на пример, blog, accounts, products).

► Чекори:

1. Креирање на проект:

```
bash
```

```
CopyEdit
```

```
django-admin startproject myproject
```

- Ќе се креира папка myproject со конфигурациски фајлови.

2. Влегување во проектот:

```
bash
```

```
CopyEdit
```

```
cd myproject
```

3. Креирање на апликација:

bash

CopyEdit

```
python manage.py startapp myapp
```

⚠ Не заборавај да ја додадеш апликацијата во INSTALLED_APPS во settings.py:

python

CopyEdit

```
INSTALLED_APPS = [
```

```
    ...,
```

```
    'myapp',
```

```
]
```

2. Django Модели (Models)

► Што е модел?

- Моделите се **класи кои го дефинираат обликот на податоците**.
- Секој модел е поврзан со табела во базата на податоци.

► Пример:

python

CopyEdit

```
from django.db import models
```

```
class Book(models.Model):
```

```
    title = models.CharField(max_length=200)
```

```
    author = models.CharField(max_length=100)
```

```
    published = models.DateField()
```

Ова ќе креира табела со колони: title, author, published.

► Најчести типови на полиња:

Поле	Опис
CharField	Текст (ограничена должина)
TextField	Долг текст
IntegerField	Цел број
FloatField	Децимален број
BooleanField	True / False
DateTimeField	Датум
DateTimeField	Датум + Време
EmailField	Текст што мора да е валиден емаил
ForeignKey	Врска еден-на-многу
ManyToManyField	Врска многу-на-многу
OneToOneField	Еден-на-еден врска

► Миграции

1. Креирање на миграции:

bash

CopyEdit

```
python manage.py makemigrations
```

2. Примена на миграции:

bash

CopyEdit

```
python manage.py migrate
```

3. Django Admin

► Што е Django Admin?

- Django автоматски нуди **веб интерфејс** каде што можеш да:
 - додадеш, уредуваш и бришеш податоци

- управуваш со модели без да пишуваш HTML
-

► Активирање:

1. Во settings.py:

python

CopyEdit

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

2. Во urls.py:

python

CopyEdit

```
from django.contrib import admin  
from django.urls import path
```

```
urlpatterns = [  
    path('admin/', admin.site.urls),  
]
```

3. Креирај суперкорисник:

bash

CopyEdit

```
python manage.py createsuperuser
```

► Регистрирање на модели:

Едноставно:

```
python
CopyEdit
from django.contrib import admin
from .models import Book
```

```
admin.site.register(Book)
```

Напредно (со ModelAdmin):

```
python
CopyEdit
@admin.register(Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ('title', 'author', 'published')
    search_fields = ('title', 'author')
    list_filter = ('published',)
    ordering = ('-published',)
```

► Корисни опции на ModelAdmin

Опција	Опис
list_display	Кои колони да се прикажат во табелата
search_fields	По кои полиња да се пребарува
list_filter	Додава филтри од страна
date_hierarchy	Навигација по датуми
ordering	Подразбирачки редослед
readonly_fields	Полиња што не може да се менуваат