**Практическая работа № 2: Модуль 1. Основные понятия Django. Вывод данных 1.**

Используя библиотеку requests сделать обращение к «фейковой api» - «jsonplaceholder» (https://jsonplaceholder.typicode.com/todos/) и записать эти данные в json-файл или переменную. 2. Полученные данные вывести в консоль Django. 3. Полученные данные вывести в веб-браузер через форматирование строки.

Шаг 1: Установка Django и Requests

Убедитесь, что у вас установлен Django и библиотека Requests. Вы можете установить их, выполнив следующую команду:

```

pip install django requests

```

Шаг 2: Создание Django проекта и приложения

Создайте Django проект, выполнив следующую команду:

```

django-admin startproject myproject

```

Затем создайте приложение внутри проекта:

```

cd myproject

python manage.py startapp myapp

```

Шаг 3: Настройка представления в Django

Откройте файл `myapp/views.py` и добавьте следующий код:

```

import requests

from django.http import HttpResponse

import json

def fetch\_data(request):

url = 'https://jsonplaceholder.typicode.com/todos/'

response = requests.get(url)

data = response.json()

# Сохранение данных в JSON-файл

with open('data.json', 'w') as file:

json.dump(data, file)

# Отправка данных в шаблон

return render(request, 'data.html', {'data': data})

```

Шаг 4: Создание шаблона в Django

Создайте файл шаблона `myapp/templates/data.html` и добавьте следующий код:

```html

<!DOCTYPE html>

<html>

<head>

<title>Data</title>

</head>

<body>

<h1>Data:</h1>

{% for item in data %}

<p>{{ item.title }}</p>

{% endfor %}

</body>

</html>

```

Шаг 5: Настройка URL маршрута в Django

Откройте файл `myproject/urls.py` и добавьте следующий код:

```python

from django.urls import path

from myapp.views import fetch\_data

urlpatterns = [

path('fetch\_data/', fetch\_data, name='fetch\_data'),

]

```

Шаг 6: Запуск сервера Django

Запустите сервер Django, выполнив следующую команду:

```

python manage.py runserver

```

Шаг 7: Просмотр результатов

Откройте веб-браузер и перейдите по адресу `http://127.0.0.1:8000/fetch\_data/`. Вы должны увидеть отформатированные данные, полученные из API, отображенные на странице.

Таким образом, вы сможете использовать библиотеку Requests для обращения к фейковому API, сохранить данные в JSON-файл и вывести их как в консоль Django, так и в веб-браузер через форматирование строки.