

Graphql django api (часть 2)

№ урока: 8 **Курс:** Django Essential

Средства обучения: Pycharm Community, GIT, аккаунт в GitHub, Python 3.6 и выше, Django проект, Docker образ.

Обзор, цель и назначение урока

Научиться работать с Graphql в рамках Django-проекта. Научиться создавать схемы в Django проекте которые, поддерживают подход Graphql. Также рассмотреть на практике работу библиотеки graphene_django. Научиться создавать различные запросы и связи между ними.

Изучив материал данного занятия, учащийся сможет:

- Подключать к проекту Graphql.
- Создавать схему.
- Создавать запрос в Graphql.
- Создавать мутацию.
- Определять типы Graphql в Django-проекте.

Содержание урока

1. Установка библиотеки graphene_django
2. Создание моделей и загрузка данных
3. Создание схемы
4. Рассмотрение типов в graphene_django
5. Создание запроса (Query)
6. Создание мутаций
7. Подключение мутаций к схеме
8. Регистрация схемы в проекте
9. Запуск проекта
10. Тестирование Graphql на практике

Резюме

- Для работы с Graphql используем библиотеку graphene и graphene_django.
- Чтобы загрузить данные из файла используем команду loaddata. В качестве формата файла используем json.
- Для начала нужно создать файл schema.py.
- Для определения типов в GraphQL используем DjangoObjectType, ObjectType.
- С помощью DjangoObjectType создаем GraphQL тип с Django модели.
- Чтобы создать Query используем ObjectType. В Query необходимо добавить Resolvers. Это даст возможность GraphQL найти адрес, где искать запрашиваемую информацию.
- InputObjectType служит для определения того, какие поля можно использовать для изменения с помощью GraphQL.
- Для создания мутаций используем класс graphene.Mutation. При создании мутации нужно унаследоваться от этого класса.
- Для работы мутаций нужно внутри класса определить метод mutate и в нем прописывать логику для создания объекта в базе.

- С помощью мутаций можно как создавать, так и обновлять данные. Для этого рекомендуется использовать разные классы.
- Последний штрих в работе с мутациями — это создать тип мутаций (class Mutation).
- Чтобы подключить мутации к схеме, используем graphene.Schema.
- Для подключения схемы к проекту нужно обновить файл с настройками:
 - Добавить в INSTALLED_APPS graphene_django
 - Добавить GRAPHENE с параметрами: ключевое слово SCHEMA и линк на файл со схемой.
 - Добавить маршрут в urls.
- Для тестов используем внутренний сервер по адресу <http://127.0.0.1:8000/graphql/>.

Закрепление материала

- Какая библиотека используется в Django для реализации GraphQL?
- Какие есть типы в GraphQL?
- Назовите все основные составляющие GraphQL? Какова их роль?
- Как подключить мутацию к проекту?
- Как подключить схему к проекту?

Дополнительное задание

Задание

Написать код, который реализует подход GraphQL. Создать несколько новых схем и мутаций.

Самостоятельная деятельность учащегося

Задание 1

Выучить и понять какие есть тонкости использования GraphQL в Django проекте, и в каких случаях его использовать.

Задание 2

Создать модель с как можно большим количеством разных типов полей. Для модели создать схему и мутации. Протестировать какими будут ответы на запросы.

Задание 3

Создать мутацию для удаления из базы данных.

Рекомендуемые ресурсы

Документация:

- <https://graphql.org/>
- <https://docs.graphene-python.org/en/latest/quickstart/>

IDE:

- <https://github.com/graphql/graphiql>