Простое руководство по пониманию GraphQL

Описание

Назвать GraphQL сменой парадигмы в области запросов к данным – это преуменьшение. Это больше похоже на разницу между ходьбой и полетом – нет никакого сравнения. Чтобы понять, почему, давайте рассмотрим, чем традиционные REST API отличаются от GraphQL. Традиционные REST-запросы требуют множества запросов для получения всей необходимой информации, и по мере расширения ваше приложение будет перегружено управлением этими запросами. С GraphQL вы можете запросить именно то, что вам нужно, с помощью одного запроса и легко выразить сложные требования к данным – не говоря уже о потенциальной поддержке подписок в реальном времени. Так что если повышение производительности, упрощение интеграции между клиентами, ускорение разработки и снижение затрат на обслуживание звучат привлекательно, присоединяйтесь к нам, чтобы узнать все о GraphQL.

Что такое GraphQL?

GraphQL – это язык запросов, созданный Facebook для построения и подключения API. Он позволяет разработчикам определять, какие данные нужны клиенту в одном запросе. Вместо того чтобы иметь конечную точку для каждого типа данных, которые могут понадобиться клиенту, GraphQL позволяет вам сделать один вызов и получить то, что вам нужно, быстро и эффективно. Его преимущества перед традиционными REST API заключаются в следующем:

- **Ускоренное время разработки:** Синтаксис GraphQL легче изучить и

внедрить, поэтому разработчики тратят меньше времени на написание кода.

- **Улучшенная производительность:** Позволяя клиентам запрашивать все необходимые им данные одним запросом, вы улучшаете время отклика сервера.
- **Снижение затрат на обслуживание:** Единая конечная точка GraphQL упрощает обслуживание сервера и снижает необходимость мониторинга нескольких конечных точек.

Как работает GraphQL?

По своей сути GraphQL используется для создания API. Он использует так называемую систему типов для определения данных и способов доступа к ним. С помощью GraphQL пользователи могут определить, какие данные им нужны, в одном запросе и получить их обратно в виде ответа в формате JSON.

GraphQL состоит из трех компонентов:

- **Язык запросов (Query Language)** – с его помощью разработчики определяют, какие данные нужны клиенту от API.
- **Язык определения схемы (SDL)** – с его помощью разработчики определяют структуру данных, которые можно запросить у API.
- **Резольвер** – сервер использует его для обработки запроса GraphQL и возврата запрошенных данных.

GraphQL также очень гибок, позволяя разработчикам добавлять или удалять поля из запроса без изменения основного кода. Это уменьшает необходимость в нескольких конечных точках и упрощает изменение структуры данных.

Пример GraphQL

Давайте рассмотрим пример работы GraphQL. Допустим, у вас есть сервер с двумя типами книг: художественная и нехудожественная литература. При использовании традиционного REST API вам пришлось бы делать два отдельных запроса для каждого типа книг. С помощью GraphQL вы можете определить, какие данные нужны клиенту, в одном запросе. Чтобы сделать запрос на GraphQL, сначала необходимо определить нужный тип данных. Это делается с помощью так называемой схемы. Схема определяет структуру имеющихся данных и способы их комбинирования. Как только система типов определена, клиенты могут получить доступ к тому, что им нужно, с помощью запроса.

Давайте посмотрим, как выглядит запрос GraphQL в действии. Ниже приведен пример запроса GraphQL:

```
query {
    FictionBooks {
        name
        author
        yearPublished
    }
    NonFictionBooks {
        name
        author
        yearPublished
    }
}
```

Этот запрос возвращает то, что запросил клиент, в одном ответе. Затем сервер с помощью резольвера обрабатывает запрос и возвращает то, что было запрошено. GraphQL делает запрос данных быстрее и проще, позволяя разработчикам создавать мощные API, которые быстро предоставляют клиентам то, что им нужно.

Почему GraphQL особенный?

Мы рассмотрели GraphQL, что он делает и как работает. Но что делает GraphQL таким особенным? Прежде всего, GraphQL упрощает запрос данных, позволяя клиентам делать один запрос на то, что им нужно. Это экономит время и ресурсы, поскольку серверу не нужно делать несколько запросов, чтобы получить то, что нужно клиенту. GraphQL также обеспечивает поддержку подписки в реальном времени, позволяя клиентам получать обновления, когда на сервере выполняется операция. Это означает, что приложения могут быть более реактивными, предоставляя пользователям данные практически в реальном времени по мере их изменения. Наконец, GraphQL разработан как язык-агностик, что означает, что он

может быть использован с любым языком программирования. Это облегчает разработчикам использование GraphQL в существующих приложениях и позволяет быстро создавать мощные API.

Дата Создания
17.04.2023