



GraphQL или REST: Сравнение функциональности, преимуществ и примеров использования

Описание

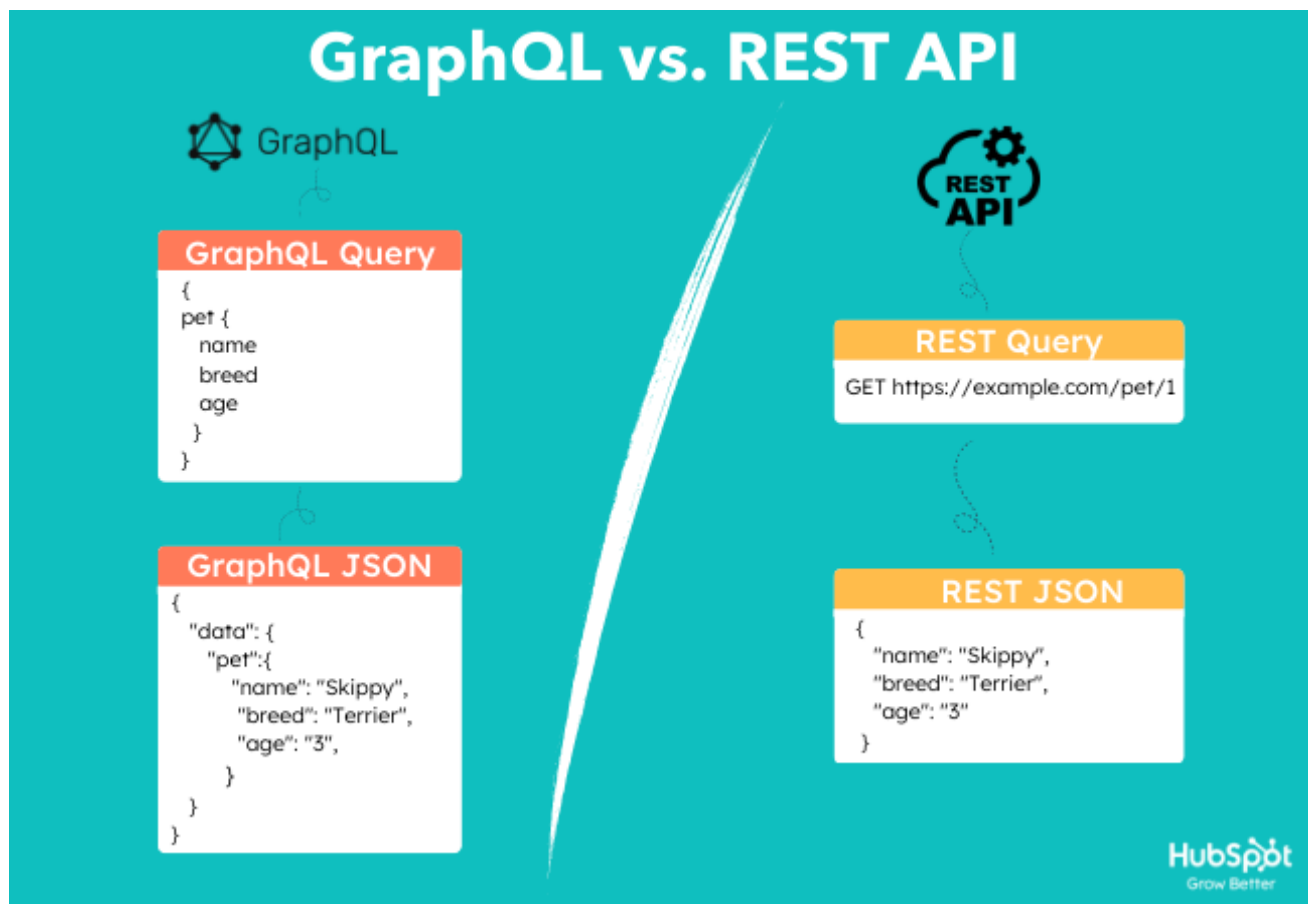
Выбирая между GraphQL и REST, думайте об этом как о настольной игре. Оба API предлагают различные возможности, которые помогают разработчикам создавать более совершенные приложения, но при этом действуют разные правила игры. Выбирая API для своего проекта, вы должны рассмотреть, какой сервис лучше всего подходит, исходя из его функциональности, преимуществ и потенциальных вариантов использования. В этой статье блога мы сначала рассмотрим сходства между GraphQL и REST, а затем сравним их преимущества, чтобы вы могли принять обоснованное решение, столкнувшись с этими вариантами.

Обзор GraphQL и REST

Подводя итог, можно сказать, что GraphQL и REST – это интерфейсы программирования веб-приложений, которые позволяют разработчикам получать доступ к данным. Они оба позволяют разработчикам создавать, читать, обновлять и удалять данные с сервера последовательным образом. Это делается с помощью набора функций “запросов” или “конечных точек”.

GraphQL – это язык запросов с открытым исходным кодом, который позволяет разработчикам запрашивать данные из нескольких источников с помощью одного вызова API. Он популярен благодаря своей гибкости в перемещении данных. GraphQL также имеет более быстрое время отклика, чем REST, который требует больше вызовов. С другой стороны, REST (representational state transfer)

представляет собой архитектурный стиль для сетевого взаимодействия между системами и чаще используется для веб-приложений. REST известен своей масштабируемостью и имеет более простой синтаксис, чем GraphQL.



Когда следует использовать GraphQL?

Есть несколько факторов, которые необходимо учитывать при принятии решения о том, является ли GraphQL правильным выбором для вашего проекта. Ниже приведены несколько ключевых вопросов, которые следует иметь в виду:

- Являются ли ваши данные графоподобными (отношения “многие-ко-многим”)? Если да, то GraphQL может быть правильным выбором.
- Нужно ли вам перемещать большое количество данных? Гибкость GraphQL в этой области является преимуществом.
- Вам важна производительность? Более быстрое время отклика GraphQL может стать преимуществом.

Теперь давайте рассмотрим некоторые ключевые преимущества GraphQL.

Использование различных источников данных

GraphQL отлично подходит для разработчиков, которым необходимо получить доступ к нескольким источникам данных с помощью одного запроса. Способность GraphQL предлагать более точные и отзывчивые запросы может принести пользу проектам с большим количеством данных или частыми запросами.

Более быстрое время отклика

GraphQL позволяет разработчикам получать данные быстрее, чем REST, который может завершить запрос всего одним вызовом API. GraphQL возвращает данные на основе точного запроса и позволяет разработчикам делать несколько запросов одновременно.

Уменьшение проблем с пропускной способностью

Ограничение пропускной способности может стать досадной проблемой для небольших устройств, таких как мобильные телефоны и смарт-часы. К счастью, GraphQL является отличным инструментом для минимизации проблемы ограниченной пропускной способности. С помощью этой технологии клиенты могут точно указать, какие данные они требуют от сервера; лишняя или ненужная информация не засоряет производительность приложения при ограничении пропускной способности.

Когда следует использовать REST?

REST – отличный выбор для приложений, в которых масштабируемость является ключевым фактором. Ниже приведены некоторые моменты, которые следует учитывать при принятии решения о том, является ли REST правильным выбором:

- Нужен ли вам предсказуемый рабочий процесс? REST часто используется для веб-приложений и может быть более последовательным при работе с данными.
- Вам нужна масштабируемость? REST – отличный вариант, поскольку он легко и быстро масштабируется.
- Вам нужно управлять небольшим объемом данных? Более простой синтаксис

REST может быть правильным выбором для вашего проекта.

Теперь давайте рассмотрим некоторые ключевые преимущества REST.

Более простой синтаксис

REST гораздо проще понять, чем GraphQL, поскольку он основан на существующих протоколах HTTP. REST также имеет много существующей документации, что облегчает работу начинающим разработчикам. По сравнению с GraphQL, синтаксис REST проще и легче для понимания.

Масштабируемость

REST известен своей масштабируемостью и гибкостью при работе с многочисленными запросами от различных устройств. Веб-приложения часто используют его из-за его способности обрабатывать множество клиентов и запросов.

Нестационарность

API REST спроектированы таким образом, что они не имеют статических данных, то есть запросы к серверу содержат все данные, необходимые серверу для выполнения поставленной задачи. Это позволяет легко создавать, читать, обновлять и удалять данные без необходимости отслеживать какие-либо дополнительные данные.

Сравнение функциональности и сценариев использования

Решая, какой вариант подходит для вашего проекта, важно учитывать функциональность GraphQL и REST и то, для каких случаев использования они предназначены. GraphQL отлично подходит для графоподобных данных и приложений, которые должны выполнять несколько запросов одновременно. Это также отличный выбор, когда важна производительность, поскольку он может быстро возвращать данные. Кроме того, его способность использовать различные источники данных делает его идеальным для проектов с большим количеством данных или частыми запросами.

REST, с другой стороны, лучше всего подходит для масштабируемых приложений, которым необходим предсказуемый рабочий процесс. Он также отлично подходит для управления небольшими объемами данных или в тех случаях, когда важны масштабируемость и гибкость. Простой синтаксис REST делает его более понятным, чем GraphQL, а имеющаяся документация облегчает начинающим разработчикам быстрый ввод в эксплуатацию.

Независимо от вашего выбора, GraphQL и REST предлагают отличные решения для разработчиков. Важно учитывать функциональность каждого варианта и то, для какого случая использования он лучше всего подходит, когда вы решаете, какой из них подходит для вашего проекта. В конечном итоге, GraphQL и REST сводятся к пониманию ваших требований и выбору правильного инструмента для работы.

Дата Создания

17.04.2023