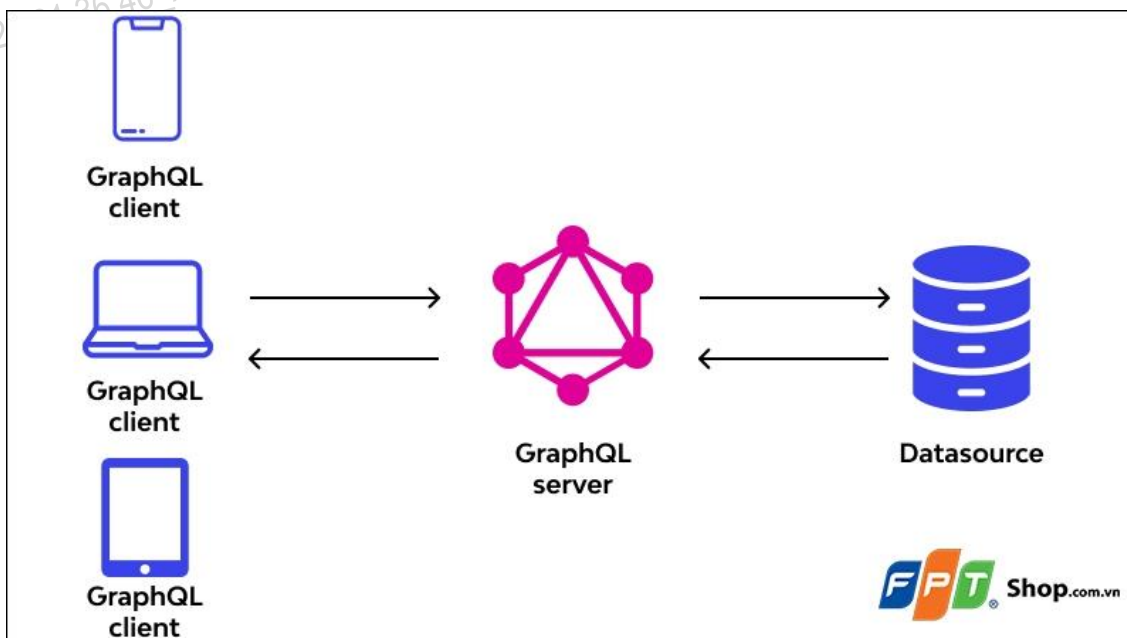
	VIETTEL AI RACE	TD176
	TỔNG QUAN VỀ GRAPHQL	Lần ban hành: 1

1. GraphQL là gì?


GraphQL là một cú pháp mạnh mẽ được thiết kế để mô tả cách yêu cầu lấy dữ liệu, thường được sử dụng để tải dữ liệu từ một server đặc biệt cho client. Đơn giản hóa, GraphQL có thể được coi là một ngôn ngữ truy vấn và thao tác dữ liệu nguồn mở cho các API. Quan trọng nhất, nó mang đến cho client khả năng đưa ra yêu cầu dữ liệu một cách chính xác và thuận tiện, đồng thời tối ưu hóa quá trình tương tác giữa client và server. GraphQL được xây dựng dựa trên ba đặc điểm chính:

- GraphQL giúp tổng hợp dữ liệu từ nhiều nguồn khác nhau một cách hiệu quả. Thay vì phải gửi nhiều yêu cầu đến nhiều API khác nhau, GraphQL cho phép lấy đồng thời tất cả các dữ liệu cần thiết chỉ trong một yêu cầu.
- GraphQL cung cấp sức mạnh cho các client khi có khả năng định dạng chính xác dữ liệu mà họ muốn nhận. Điều này loại bỏ tình trạng "over-fetching" hoặc "under-fetching" thông tin, giúp tối ưu hóa tải trọng dữ liệu.
- GraphQL sử dụng hệ thống kiểu dữ liệu (type system) để định nghĩa và khai báo cấu trúc dữ liệu. Điều này tạo ra một phương thức mạnh mẽ và dễ đọc để xác định kiểu dữ liệu, quy tắc kiểm soát và phục vụ mục đích tự mô tả của nó.



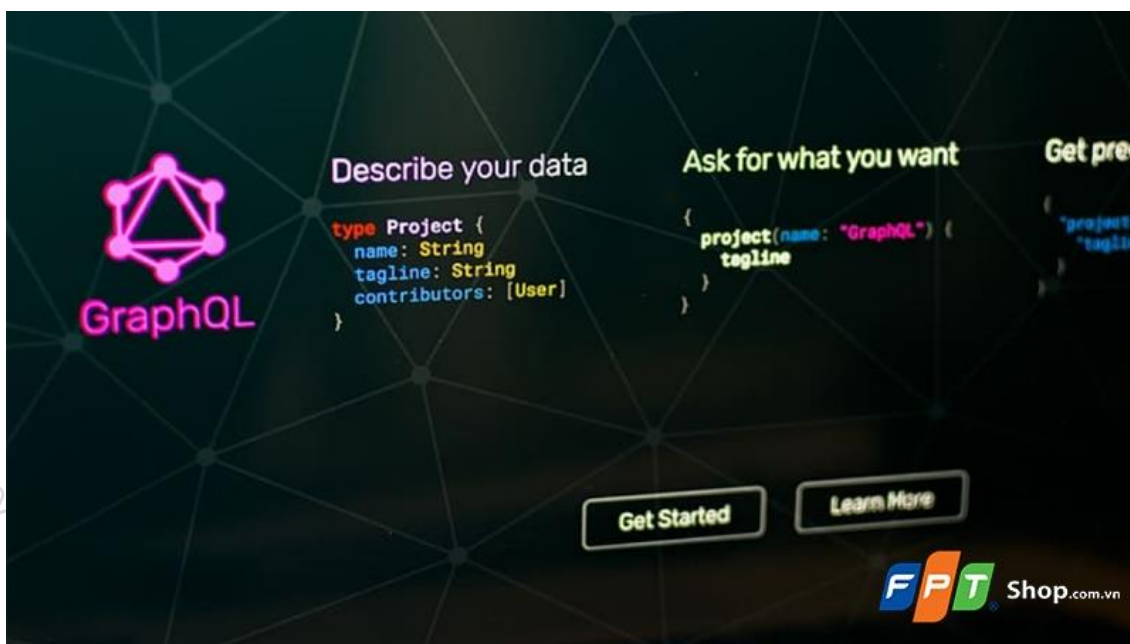
2. Một số tính năng của GraphQL

Khi khám phá về GraphQL là gì, hãy chắc chắn rằng bạn không bỏ lỡ những đặc điểm nổi bật dưới đây:

	VIETTEL AI RACE	TD176
	TỔNG QUAN VỀ GRAPHQL	Lần ban hành: 1

Thay thế cho REST: Do REST đôi khi gặp khó khăn khi trả về dữ liệu quá nhiều hoặc quá ít, gây ảnh hưởng đáng kể đến hiệu suất ứng dụng. Điều này làm cho việc GraphQL trở thành sự lựa chọn tự nhiên, bởi nó cho phép khai báo thông tin và dữ liệu tại các điểm mà client có thể xác định chính xác những gì họ cần từ API.

Mutation: GraphQL cho phép gửi các truy vấn được gọi là mutation, một tính năng có cú pháp tương tự như khi truy vấn dữ liệu (Fetching Data), nhưng bắt đầu bằng một từ khóa cụ thể. Tính năng này được coi là linh hoạt và hữu ích, giúp hệ thống đạt hiệu quả trong việc phân tích và xử lý dữ liệu hoạt động.



Defining Schema và Type System: GraphQL có một hệ thống đặc biệt để xác định schema của bất kỳ API nào. Hiện tại, toàn bộ type system được liệt kê trong một API cụ thể sử dụng GraphQL Schema Definition Language để thực hiện các thao tác và hoạt động cần thiết cho ứng dụng.

Realtime updates và Subscription: Để đáp ứng nhu cầu kết nối với máy chủ và thực hiện các chức năng realtime, GraphQL mang đến một loại khái niệm và thông tin liên quan được gọi là Subscription.

Fetching data_Query: GraphQL không phụ thuộc vào client để xác định thông tin cần thiết, mà thay vào đó, nó nắm giữ khả năng nạp dữ liệu và thông tin một cách linh hoạt. Điều này đặt GraphQL vào một vị trí thuận lợi để phát triển và giới thiệu những cải tiến trong quá trình truy xuất dữ liệu. GraphQL có khả năng khắc phục các hạn chế của các phương thức truy xuất truyền thống và tận dụng ưu điểm để xây dựng và mở rộng hệ thống một cách toàn diện.

	VIETTEL AI RACE	TD176
	TỔNG QUAN VỀ GRAPHQL	Lần ban hành: 1



3. Các ưu điểm và nhược điểm của GraphQL

Để triển khai GraphQL một cách hiệu quả, bạn cần hiểu rõ cả ưu điểm và nhược điểm của ngôn ngữ này, từ đó tận dụng các ưu thế và đối mặt với nhược điểm để xây dựng hệ thống một cách hiệu quả.

3.1. Ưu điểm

Ưu điểm của GraphQL bao gồm:

- **Tự động tạo nguồn thông tin:** GraphQL schema có khả năng tự động tạo ra một nguồn thông tin đáng tin cậy, giúp quản lý dữ liệu một cách hiệu quả.
- **Request hiệu quả:** Chỉ với một yêu cầu, client có thể lấy được lượng thông tin cần thiết để giảm thiểu số lượng request, tối ưu hóa giao tiếp giữa client và server.
- **Kiểm soát và xử lý kiểu dữ liệu:** GraphQL hỗ trợ người dùng trong việc kiểm soát và xử lý kiểu dữ liệu, giảm sai lệch trong giao tiếp giữa các thành phần.
- **Phát triển API mở rộng:** GraphQL giúp phát triển API mà không ảnh hưởng đến các truy vấn hiện có, tạo sự linh hoạt trong quá trình phát triển ứng dụng.
- **Tương thích với rest API:** GraphQL có thể hoạt động như một rest API và làm việc tốt với các công cụ API hiện có, không đưa ra yêu cầu đặc biệt về kiến trúc ứng dụng.

	VIETTEL AI RACE	TD176
	TỔNG QUAN VỀ GRAPHQL	Lần ban hành: 1

- **Tài liệu chi tiết:** Có sẵn tài liệu dành cho GraphQL, giúp người dùng dễ dàng học tập và tiếp thu kiến thức.



3.2. Nhược điểm

GraphQL cũng đối mặt với một số thách thức:

- **Tương thích với Rest API:** Mặc dù có nhiều extension mã nguồn mở cho GraphQL, nhưng không phải tất cả đều hoạt động tốt và tương thích với rest API, điều này có thể tạo ra những thách thức khi tích hợp vào các hệ thống hiện có.
- **Tăng công việc cho server:** Việc chuyển nhiều truy vấn lên server có thể làm tăng khối lượng công việc, đặc biệt là khi server phải xử lý nhiều yêu cầu phức tạp cùng một lúc, điều này có thể ảnh hưởng đến hiệu suất của server.
- **Quản lý bộ nhớ đệm phức tạp:** So với rest, việc quản lý bộ nhớ đệm trong GraphQL có thể trở nên phức tạp hơn nhiều, đặc biệt là khi cần xử lý một lượng lớn dữ liệu động.
- **Bảo trì GraphQL Schema:** Người viết API phải đảm bảo bảo trì đúng GraphQL Schema, điều này đòi hỏi họ phải chịu trách nhiệm và liên tục cập nhật để đảm bảo tính thống nhất của hệ thống.
- **Chiến lược triển khai đa dạng:** GraphQL có thể phụ thuộc vào cách triển khai để đáp ứng các yêu cầu khác nhau về chiến lược quản lý API, điều này có thể tạo ra sự phức tạp và đa dạng trong việc triển khai.

	VIETTEL AI RACE	TD176
	TỔNG QUAN VỀ GRAPHQL	Lần ban hành: 1



4. Yếu tố quan trọng của GraphQL

GraphQL phát triển dựa trên các thành phần quan trọng như sau:

Query: GraphQL sử dụng khái niệm query (truy vấn) để mô tả các yêu cầu dữ liệu từ client. Khi query được định nghĩa, client có thể sử dụng các từ khóa để đặt tên cho các trường dữ liệu. Việc này giúp hỗ trợ cho việc truy xuất các trường lồng nhau cho tất cả các hệ thống ứng dụng một cách linh hoạt. GraphQL đảm bảo rằng client không cần phải lo lắng về nguồn dữ liệu, vì nó quản lý và thực hiện mọi thứ.

Schema: GraphQL sử dụng hệ thống kiểu dữ liệu để định nghĩa API một cách chính xác. Tất cả các kiểu dữ liệu trong API đều được xác định bằng schema thông qua GraphQL Schema Definition Language (SDL). Điều này làm cho schema trở thành một công cụ quan trọng đến việc quy ước cách client và server giao tiếp, cùng lúc đó xác định cách client truy cập dữ liệu.

	VIETTEL AI RACE	TD176
	TỔNG QUAN VỀ GRAPHQL	Lần ban hành: 1



Resolvers: Để xác định nơi và cách lấy dữ liệu cho các trường của truy vấn, GraphQL sử dụng resolvers. Resolvers giúp GraphQL hiểu rõ nơi dữ liệu cần được truy xuất và làm thế nào nó có thể được thu thập. Điều quan trọng là resolvers không bị hạn chế hoặc ảnh hưởng bởi dữ liệu và thông tin được thu thập, tạo ra sự linh hoạt trong quá trình xử lý dữ liệu.

Tóm lại, GraphQL là một ngôn ngữ truy vấn mạnh mẽ, linh hoạt và hiệu quả, được thiết kế để giải quyết những thách thức gặp phải trong việc truy vấn dữ liệu. Với khả năng tương tác động, khả năng tùy chỉnh linh hoạt, và sự hỗ trợ cho ứng dụng di động và web, GraphQL đã trở thành một công nghệ ngày càng phổ biến trong cộng đồng phát triển phần mềm. Việc hiểu rõ về cách hoạt động của GraphQL là gì, ưu điểm và nhược điểm của nó sẽ giúp nhà phát triển tận dụng toàn bộ tiềm năng của ngôn ngữ này để xây dựng các ứng dụng mạnh mẽ và linh hoạt.