

Tìm hiểu, so sánh grpc, restapi, websocket

Grpc	2
I. Khái niệm.....	2
II. Cấu trúc của Grpc.....	2
III. Ưu nhược điểm của Grpc	2
IV. Hoạt động của grpc	2
Restfull API	3
I. Khái niệm.....	3
II. Cấu trúc của RestfullAPI.....	3
III. Ưu nhược điểm của Restfull API.....	3
IV. Hoạt động của RestFull API	4
WebSocket	4
I. Khái niệm.....	4
II. Cấu trúc của websocket	4
III. Ưu nhược điểm của websocket.....	4
- Nhược điểm.....	5
IV. Hoạt động của websocket	5
So sánh giữa Grpc, RestFull API	5
I. Giống nhau.....	5
II. Khác nhau	5

Grpc

I. Khái niệm

- Grpc : Grpc là một công nghệ được phát triển bởi google sử dụng protocol buffers để xác định các dịch vụ và các thực thể được trao đổi giữa các ứng dụng.
- Hoạt động trên nhiều nền tảng khác nhau giúp phát triển các ứng dụng và có tính mở rộng, tốc độ truyền tải dữ liệu nhanh hơn so với công nghệ khác.

II. Cấu trúc của Grpc

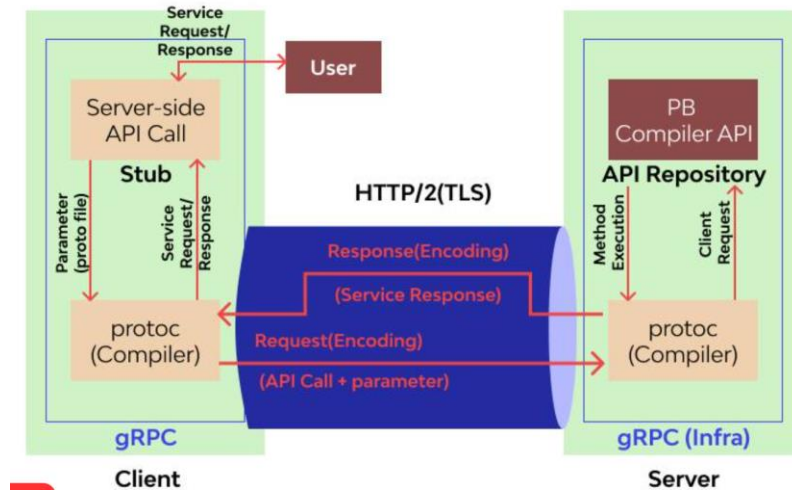
- Grpc sử dụng protobuf để xác định các giao thức và hình thức dữ liệu gửi và nhận của các kênh.
 - o Protobuf là một dạng dữ liệu nhị phân có cấu trúc được sử dụng để truyền tải và lưu trữ dữ liệu trong các ứng dụng phân tán cho phép định nghĩa các cấu trúc dữ liệu rõ ràng và hiệu suất mở rộng cao so với JSON XML
- Hỗ trợ hai loại kiến trúc: Unary RPCs và Streaming RPCs :
 - o Unary RPC là loại giao thức đơn giản nhất, trong đó client gửi một yêu cầu và server trả về một kết quả. Yêu cầu và kết quả đều là các message đơn.
 - o Streaming RPC, thay vì gửi một message đơn như Unary RPC, nó cho phép gửi và nhận nhiều message theo cơ chế liên tục và đồng bộ.

III. Ưu nhược điểm của Grpc

- Ưu điểm:
 - o Khả năng tạo ra mã dễ sử dụng và tự động hóa trong việc tạo client và server, giúp cho việc phát triển ứng dụng nhanh hơn và giảm thiểu việc phải viết lại mã lặp đi lặp lại.
 - o Hiệu năng cao, bởi vì GRPC sử dụng protobuf để mã hóa dữ liệu, giúp giảm thiểu dung lượng mạng truyền tải và tăng tốc độ truyền tải dữ liệu.
 - o Được thiết kế để hoạt động tốt với các ứng dụng phân tán và đa ngôn ngữ, vì vậy nó có thể được sử dụng cho các ứng dụng được phát triển trên nhiều nền tảng và ngôn ngữ.
- Hạn chế:
 - o Khó khăn trong việc triển khai vì yêu cầu sử dụng một số công nghệ phức tạp như Protobuf và HTTP/2.
 - o Không hỗ trợ tốt cho các ứng dụng có kích thước nhỏ, vì khi sử dụng GRPC, các ứng dụng phải tích hợp thêm một số thư viện khác để xử lý.

IV. Hoạt động của grpc

- Grpc sử dụng HTTP/2 để truyền tải dữ liệu giữa các ứng dụng được gọi là client và server để giao tiếp chuyển đổi các dữ liệu. Có một định dạng dữ liệu chung giữa client và server để có thể hiểu được các message trong quá trình trao đổi dữ liệu.
- Hiểu theo cách đơn giản là client sẽ gọi các service của server thông qua http/2 và được encode/decode với các tệp proto chung có trong client và server.



Ảnh 1. Ví dụ về cơ chế hoạt động của grpc

Restfull API

I. Khái niệm

- RESTful API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động...), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.
- Rest là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.

II. Cấu trúc của RestfullAPI

- Restfull api sử dụng các phương thức http để định nghĩa các hoạt động cần thực hiện. Các phương thức http phổ biến bao gồm POST, PUT, GET, DELETE.
- Mỗi resource có thể có nhiều biểu diễn (representation) khác nhau. Biểu diễn đó có thể là định dạng JSON, XML hoặc text/plain, phụ thuộc vào yêu cầu của client.
- RESTful API sử dụng URI để xác định các resource và cho phép client thao tác với chúng. URI phải được thiết kế có cấu trúc rõ ràng để giúp client dễ dàng tìm và truy cập các resource.
- RESTful API sử dụng hypermedia để cung cấp các liên kết giữa các resource và giúp client điều hướng các resource. Các liên kết này được gắn vào các response của server và cho phép client thực hiện các yêu cầu khác.

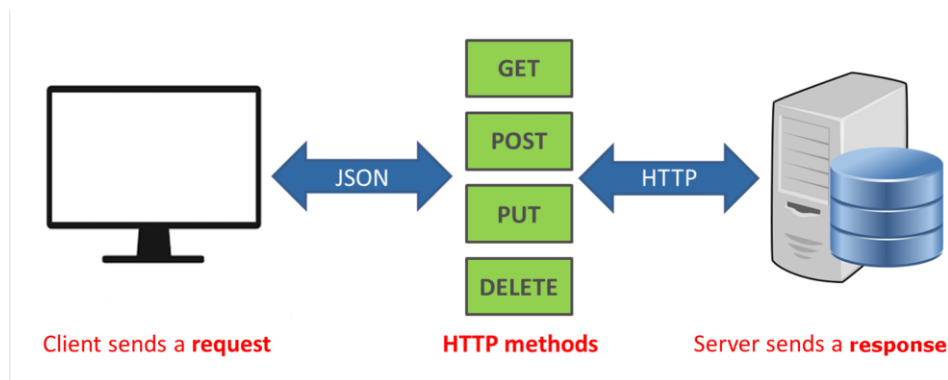
III. Ưu nhược điểm của Restfull API

- Ưu điểm:
 - o Độc lập với ngôn ngữ lập trình, tương thích với nhiều loại ứng dụng và thiết bị khác nhau.
 - o Sử dụng HTTP, một giao thức phổ biến và hỗ trợ rộng rãi trên nhiều nền tảng.
 - o Sử dụng các phương pháp HTTP tiêu chuẩn như GET, POST, PUT, DELETE giúp dễ dàng quản lý các hoạt động.
 - o Tính mở rộng cao, có thể mở rộng hệ thống để đáp ứng với nhu cầu tăng cao của số lượng người dùng.
- Nhược điểm:
 - o Không có quy chuẩn về cách định nghĩa API nên dễ dẫn đến nhầm lẫn và khó khắc phục khi phát triển ứng dụng quy mô lớn.

- Có thể bị giới hạn về tính tương thích khi API thay đổi hoặc được thêm mới, đặc biệt là với các ứng dụng đang sử dụng phiên bản cũ của API.
- Các API RESTful không thể đảm bảo tính toàn vẹn và bảo mật cho dữ liệu truyền tải mà phải dựa vào các giải pháp khác như mã hóa và chữ ký số.

IV. Hoạt động của RestFull API

- RestFull Api theo cơ chế client server. Dựa trên các method mà http cung cấp phía server sẽ cung cấp các dịch vụ theo các đường dẫn URI để thao tác với tài nguyên trên server. Client sẽ dùng các method này để thao tác với tài nguyên thông qua các URI mà server cung cấp. Sử dụng dữ liệu dạng chuẩn như JSON, XML để chuyển đổi giữa client và server.
- Thao tác thông qua URI giúp giảm bớt sự phụ thuộc giữa client và server, giúp cho việc đa nền tảng dễ dàng hơn.



Ảnh 2. Cách thức hoạt động của Restfull API

WebSocket

I. Khái niệm

- WebSocket là một giao thức truyền tải hai chiều (full duplex) cho phép các ứng dụng giao tiếp với nhau một cách thời gian thực. WebSocket cho phép thiết lập một kết nối liên tục giữa trình duyệt và máy chủ, giúp các ứng dụng có thể gửi và nhận dữ liệu một cách nhanh chóng và hiệu quả hơn.
- Trong giao thức HTTP truyền thống, trình duyệt phải gửi một yêu cầu mới tới máy chủ để nhận phản hồi. Với WebSocket, kết nối mạng được duy trì liên tục giữa trình duyệt và máy chủ, cho phép các ứng dụng gửi và nhận dữ liệu một cách thời gian thực, đồng thời giảm thiểu độ trễ và tăng tốc độ xử lý dữ liệu.

II. Cấu trúc của websocket

- WebSocket server là một ứng dụng máy chủ chạy trên máy chủ web, cung cấp các API để tạo kết nối WebSocket và xử lý các thông điệp được gửi và nhận từ WebSocket client.
- WebSocket client là trình duyệt web hoặc ứng dụng khác sử dụng WebSocket API để thiết lập một kết nối WebSocket với WebSocket server.

III. Ưu nhược điểm của websocket

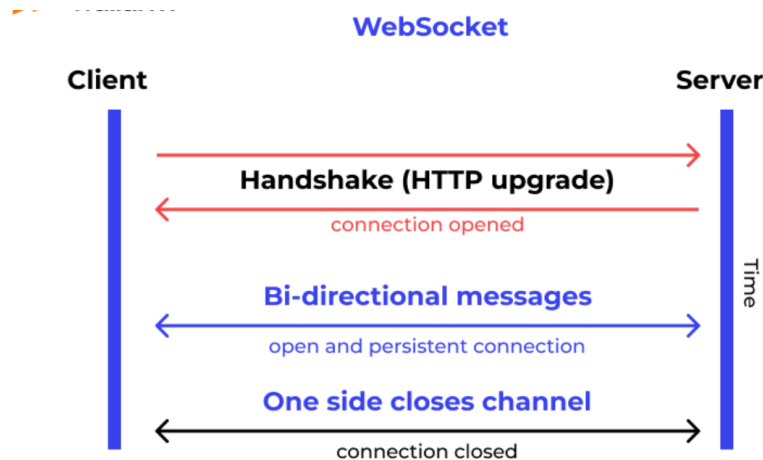
- Ưu điểm:
 - Cơ chế real time cho phép giao tiếp 2 chiều giữa client và server một cách hiệu quả và nhanh chóng.
 - Kích thước dữ liệu nhỏ hơn: WebSocket giảm thiểu kích thước gói tin và số lần giao tiếp yêu cầu/phản hồi so với HTTP. Do đó, kích thước dữ liệu được truyền tải sẽ nhỏ hơn, làm giảm lưu lượng mạng.
 - Không cần phải gửi yêu cầu mới để nhận dữ liệu mới: Khi sử dụng HTTP, trình duyệt phải gửi yêu cầu mới để nhận dữ liệu mới từ máy chủ. Tuy nhiên, với

WebSocket, máy chủ có thể gửi dữ liệu mới cho trình duyệt mà không cần phải yêu cầu mới.

- Nhược điểm
 - o Tăng khối lượng dữ liệu gửi đi: Khi sử dụng WebSocket, các thông tin định danh và các trạng thái phiên truy cập cần phải được gửi đi cùng với các gói tin, tăng kích thước của các gói tin truyền tải.
 - o Phức tạp hơn trong việc triển khai: WebSocket yêu cầu thiết lập một server riêng biệt để quản lý các kết nối, điều này có thể làm tăng độ phức tạp của hệ thống.

IV. Hoạt động của websocket

- Client sẽ gửi một HTTP request tới server với header để yêu cầu thiết lập một kết nối websocket. Server sẽ phản hồi request xác nhận client và thực hiện trao đổi dữ liệu.
- Truyền tải dữ liệu liên tục giữa client server mà không cần thiết lập lại kết nối cho đến khi client hay server đóng kết nối.



Ảnh 3. Cơ chế hoạt động của websocket

So sánh giữa Grpc, RestFull API

I. Giống nhau

- Cả gRPC và RESTful API đều sử dụng HTTP làm giao thức truyền tải.
- Cả hai đều được sử dụng để xây dựng các dịch vụ web và được sử dụng để giao tiếp giữa các ứng dụng.

II. Khác nhau

So sánh	grpc	rest	websocket
Giao thức sử dụng	Sử dụng HTTP/2	Sử dụng HTTP/1.1	
Định dạng dữ liệu	Dựa và protobuf để trao đổi	Các định dạng như JSON, XML	
Cơ chế giao tiếp	Sử dụng cơ chế message	Sử dụng cơ chế request/response	
Lợi thế	Thiên về sức mạnh, bảo mật và hiệu suất	Linh động hơn cho phép các ứng dụng khác nhau nhận dữ liệu và cơ chế khác nhau.	

Cấu trúc	Sử dụng các file proto để định nghĩa service và message	Không quy chuẩn rõ ràng	
----------	---	-------------------------	--