Báo cáo Rest API

* Rest API ( REpresentational State Transfer ) là phong cách kiến trúc cho các hệ thống phân tán hybermedia

1. **Nguyên tắc của REST**

* Clent – server: bằng cách tách các mối quan tâm về giao diện người dùng và mối quan tâm về lưu trữ dữ liệu, nó giúp cải thiện tính linh hoạt của giao diện trên nhiều nền tảng và cải thiện khả năng mở rộng bằng cách đơn giản hóa các thành phần của máy chủ.
* Stateless: Mỗi request từ client đến server phải chứa tất cả thông tin cần thiết để hiểu yêu cầu và không thể tận dụng bất kì thứ gì trên server. Trạng thái session do đó được giữ hoàn toàn trên client
* Cacheable: cache hạn chế yêu cầu dữ liệu trong 1 response. Nếu 1 response là cachable, sau đó bộ nhớ cache của client được cho quyền tái sử dụng.
* Uniform interface: bằng cách áp dụng nguyên tắc kĩ thuật phần mềm tổng quát của component interface, kiến trúc hệ thống tổng thể được đơn giản hóa và khả năng hiển thị của các tương tác được cải thiện. Để có được một giao diện đồng nhất (uniform interface ), nhiều ràng buộc kiến trúc là cần thiết để hướng dẫn hành vi của các component. Rest api được xác định thông qua 4 yêu tố: Nhận dạng các tài nguyên, thao tác tài nguyên thông qua đại diện, thông điệp tự mô tả, hypermedia như năng lượng của trạng thái ứng dụng.
* Layered system: Kiểu hệ thống lớp cho phép 1 kiến trúc được bao gồm các lớp phân cấp bằng các hạn chế hành vi thành phần và mỗi thành phần không thể tương tác vượt ra ngoài lớp ngay lập tức mà nó đang tương tác.

1. **Cách làm việc**

* 1 Rest api tạo ra một loạt các module nhỏ. Mỗi module xử lí 1 phần cơ bản cụ thể của transaction. Các module này cung cấp rất nhiều tính linh hoạt
* Một Rest api sử dụng các HTTP method. Sử dụng GET để lấy tài nguyên. PUT để thay đổi trạng thái hoặc cập nhật tài nguyên. POST để tạo tài nguyên mới. DELETE để loại bỏ tài nguyên.

1. **Sử dụng**

* Bởi vì các yêu cầu là stateless, Rest api là rất hữu dụng trong các ứng dụng đám mây. Các thành phần stateless có thể được tái triển khai một cách tự do và chúng có thể thay đổi quy mô để phù hợp với tải trọng. Điều này là do yêu cầu bất kì có thể được dẫn đến bất kì trường hợp nào của một thành phần, không có gì phải nhớ trong giao dich tiếp theo. Điều đó làm cho Rest api thích hợp cho việc sử dụng web và trong các dịch vụ đám mây vì ràng buộc với một dịch vụ thông qua api là vấn đề kiểm soát URL.

1. **Xác thực**

* Các resquest api có thể được xác thực để đảm bảo người dùng đã đăng nhập và có quyền thích hợp. Điều này tránh cho việc bất kì ai cũng có thể lấy dữ liệu, thay đổi dữ liệu
* Có 4 cách xác thực cơ bản như:
  + HTTP basic authentication: Một header Authentication có chứa tên người dùng và mật khẩu được mã hóa base64 thông qua resquest header.
  + API keys: Một ứng dụng của bên thứ 3 được cấp quyền sử dụng API bằng cách sử dụng một API key có thể có các quyền cụ thể hoặc bị giới hạn cụ thể. Key được thông qua trong 1 HTTP request header hoặc thông qua trên chuỗi truy vấn.
  + Oauth: 1 token thu được trước khi bất cứ request có thể thực hiện bằng các gửi 1 client ID và có thể cần 1 client secret tới server. Oauth sau đó được gửi với mỗi yêu cầu API cho đến khi hết hạn
  + JWT: token xác thực điện tử được truyền một cách an toàn trong header của request hoặc response.
* Xác thực API sẽ khác nhau tùy thuộc vào ngữ cảnh sử dụng.

1. **Bảo mật**

* Một số lưu ý về bảo mật:
  + Sử dụng https
  + Sử dụng một phương pháp xác thực mạnh
  + Sử dụng cors để hạn chế các request phía client đến server
  + Cung cấp những chức năng tối thiếu, không tạo tùy chọn DELETE mà không yêu cầu xác thực
  + Tránh show API token trong JS phía clent
  + Yêu cầu sử dụng cùng một API token hoặc địa chỉ API giới hạn N phút