**1. Tìm hiểu lý thuyết**

**API là gì?**

API (Application Programming Interface) là cầu nối giao tiếp giữa các phần mềm, cho phép chúng truy cập vào dữ liệu hoặc chức năng của nhau một cách có kiểm soát. API hoạt động như một giao diện, giúp nhà phát triển sử dụng các tính năng của phần mềm khác mà không cần hiểu rõ cách nó hoạt động bên trong.

Ví dụ: Khi bạn sử dụng một ứng dụng thời tiết để kiểm tra nhiệt độ hôm nay, ứng dụng này gọi API từ dịch vụ thời tiết để lấy dữ liệu.

**REST API là gì?**

REST (Representational State Transfer) là một phong cách thiết kế API phổ biến, được dùng nhiều trong các ứng dụng web và di động. Nó dựa trên giao thức HTTP và tuân theo một số quy tắc:

* **Stateless**: Mỗi yêu cầu từ client tới server đều độc lập, không lưu trạng thái phiên làm việc trước đó.
* **Tài nguyên**: REST API truy cập dữ liệu dưới dạng tài nguyên (ví dụ: người dùng, bài viết, sản phẩm).
* **Các phương thức HTTP**:
  + **GET**: Lấy dữ liệu.
  + **POST**: Thêm dữ liệu mới.
  + **PUT/PATCH**: Cập nhật dữ liệu.
  + **DELETE**: Xóa dữ liệu.

**JSONPlaceholder** là một REST API miễn phí dùng để thử nghiệm. Ví dụ: https://jsonplaceholder.typicode.com/users trả về danh sách người dùng.

**Gọi API trong React Native như thế nào?**

Trong React Native, việc gọi API được thực hiện qua các thư viện hoặc phương pháp như:

1. fetch:
   * Là API tích hợp sẵn trong JavaScript.
   * Dùng để gửi yêu cầu HTTP (GET, POST, PUT, DELETE).
   * Cách sử dụng đơn giản nhưng cần xử lý thủ công các vấn đề như lỗi hoặc timeout.

**Ví dụ:**

javascript

fetch('https://jsonplaceholder.typicode.com/users')

.then(response => response.json())

.then(data => console.log(data))

.catch(error => console.error('Error:', error));

1. axios:
   * Là thư viện HTTP mạnh mẽ, cung cấp nhiều tính năng tiện lợi hơn fetch.
   * Hỗ trợ xử lý lỗi tự động, cấu hình request dễ dàng, và timeout.
   * Cần cài đặt qua npm/yarn: npm install axios.

**Ví dụ:**

javascript

import axios from 'axios';

axios.get('https://jsonplaceholder.typicode.com/users')

.then(response => console.log(response.data))

.catch(error => console.error('Error:', error));

**Sự khác biệt giữa fetch và axios**

|  |  |  |
| --- | --- | --- |
| **Tiêu chí** | **fetch** | **axios** |
| Tích hợp sẵn | Có | Không, cần cài đặt thư viện. |
| Hỗ trợ cấu hình timeout | Không | Có. |
| Xử lý lỗi | Phải xử lý thủ công | Tự động xử lý lỗi. |
| Cấu hình yêu cầu HTTP nâng cao | Hạn chế | Linh hoạt và dễ dàng. |
| Tự động chuyển đổi JSON | Không (cần dùng response.json) | Có. |