1. **Xác định tài nguyên (Resources)**:

· Tài nguyên là các đối tượng hoặc dữ liệu mà API sẽ quản lý, được quản lí trong models. Mỗi tài nguyên được đại diện bằng một URI duy nhất.

1. **Xác định các phương thức HTTP**:

· Các phương thức HTTP như GET, POST, PUT, DELETE, PATCH được sử dụng để xác định các hành động khác nhau trên tài nguyên:

* + **GET**: Lấy dữ liệu của tài nguyên.
  + **POST**: Tạo mới một tài nguyên.
  + **PUT**: Cập nhật toàn bộ tài nguyên.
  + **PATCH**: Cập nhật một phần của tài nguyên.
  + **DELETE**: Xóa tài nguyên.

1. **Thiết kế URL**:

· URL phải tuân thủ các nguyên tắc REST và cần rõ ràng, dễ đọc. Chúng thường dựa trên danh từ và không chứa động từ, ví dụ: /users, /products/{id}, /orders/{orderId}.

1. **Định dạng dữ liệu**:

· Chọn định dạng dữ liệu mà API sẽ sử dụng, phổ biến nhất là JSON và XML. JSON thường được ưa chuộng hơn vì nhẹ và dễ sử dụng với JavaScript.

1. **Xử lý trạng thái**:

· RESTful API là stateless, nghĩa là mỗi yêu cầu từ client phải chứa đủ thông tin để server xử lý, không lưu trạng thái giữa các yêu cầu.

1. **Quản lý lỗi**:

· Thiết kế các phản hồi lỗi rõ ràng và nhất quán, sử dụng các mã trạng thái HTTP như 400 (Bad Request), 401 (Unauthorized), 404 (Not Found), 500 (Internal Server Error), v.v.

1. **Bảo mật**:

· Đảm bảo an toàn cho API bằng các phương pháp xác thực và phân quyền như OAuth, JWT (JSON Web Token), SSL/TLS.

1. **Tài liệu và mô tả API**:

· Cung cấp tài liệu chi tiết và rõ ràng cho API, giúp các nhà phát triển khác dễ dàng hiểu và tích hợp với hệ thống. Swagger và OpenAPI là những công cụ phổ biến hỗ trợ việc này.