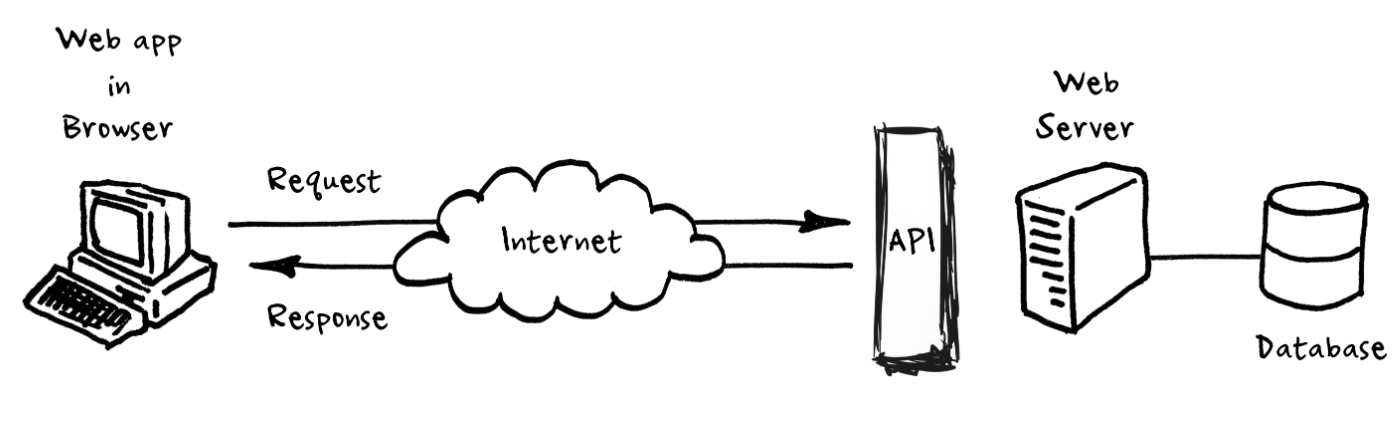
**API TESTING**

1. Giới thiệu API
2. API là gì?

**API (Application programming Interface) là cầu nối giữa 2 đối tượng**  

Nói trên phương diện mô hình client – server: API là cầu nối giữa client và server. Client có thể là máy tính, điện thoại với hệ điều hành khác nhau với ngôn ngữ khác nhau.Server backend cũng được viết bằng ngôn ngữ khác nhau. Để server và client giao tiếp được với nhau, cần dùng đến API.

1. Vì sao phải test API?

* Đảm bảo tính đúng đắn của API khi server và client giao tiếp với nhau.
* Đảm bảo logic giữa BE và FE
* Tìm ra nguyên nhân gốc rễ của lỗi là do server hay client
* Kiểm tra tính bảo mật : dữ liệu truyền qua API có được mã hoá, xác thực và uỷ quyền khi sử dụng yêu cầu truy cập các API,…

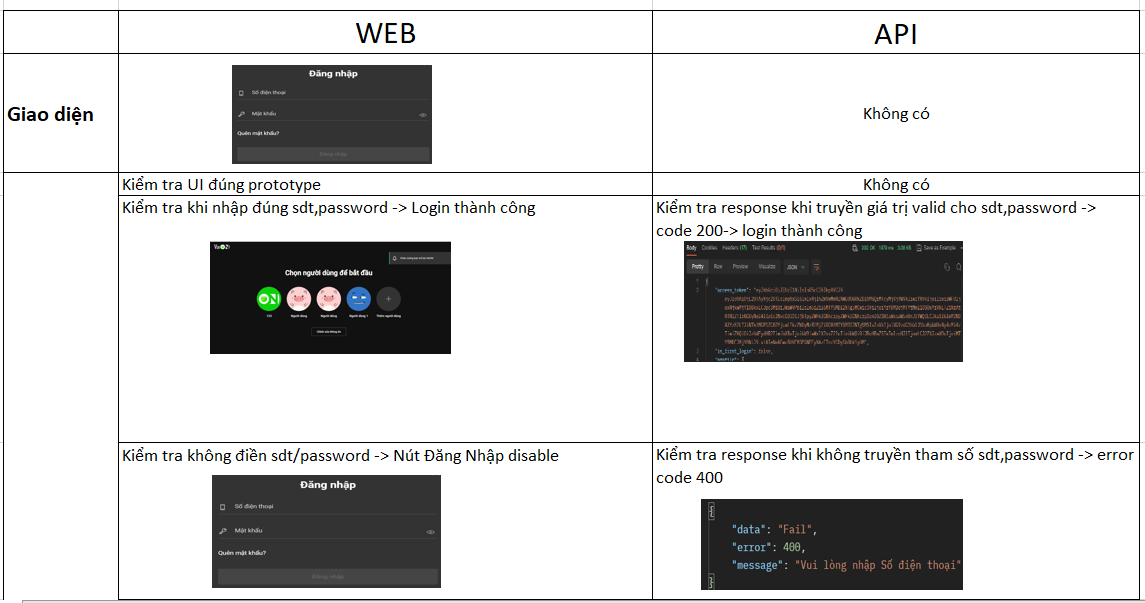
1. API Testing
2. Giới thiệu API Testing

* Là kiểm thử **business logic, security, performance** phía server
* Test API là kiểm tra: dữ liệu trả về có chính xác không, kiểm tra HTTP status code, thời gian phản hồi, error code khi API trả ra lỗi,kiểm tra authorization, performance testing, security testing,…
* Ví dụ: Login VieOn

1. Vào trang web VieOn, chọn chức năng Đăng Nhập
2. Điền thông tin sdt, password
3. Click button Đăng nhập
4. Hệ thống kiểm tra thông tin
5. Xử lí thông tin
   * Sdt, password đúng -> Login thành công
   * Sdt sai hoặc Password sai -> Login không thành công
6. Server trả về status code
7. Hiển thị thông báo trên giao diện cho người dùng biết.

So sánh login của Web và API Testing

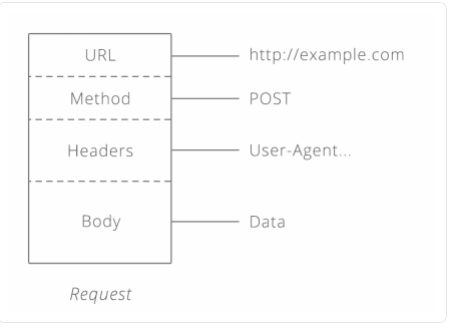
Khi nào cần test web, api?







1. API Testing với Postman
2. Thành phần API
   1. Request



URL: là địa chỉ định vị một tài nguyên dữ liệu duy nhất

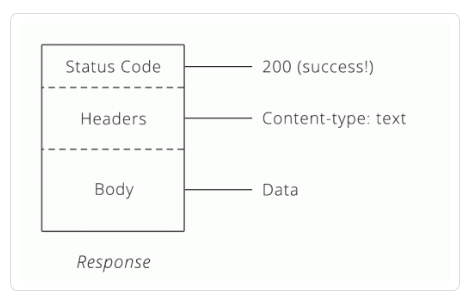
Methods: là hành động mà client muốn tác động lên “resource”. Có 4 loại method hay dùng:

* GET: yêu cầu serve trả dữ liệu về để hiển thị
* POST: đẩy dữ liệu lên để lưu vào database
* PUT: cập nhật dữ liệu
* DELETE: xoá 1 resource

Headers: nơi chứa thông tin của 1 request:loại thiết bị đang sử dụng, loại định dạng response mà client có thể đọc được,…

Body: nơi chứa thông tin client sẽ điền

* 1. Response
* Sau khi nhận được request từ phía client, serve sẽ xử lí và gửi lại một response



Status code: mã code gồm 3 số và có ý nghĩa.

+ 1xx: Mã thể hiện trạng thái thông tin

+ 2xx: Thể hiện sự thành công

+ 3xx: Thể hiện sự điều hướng

+ 4xx: Thể hiện lỗi từ client

+ 5xx: Thể hiện lỗi từ server

Headers: nơi chứa thông tin của 1 reponses:thời gian gửi request, loại định dạng response mà client có thể đọc được, tên server,…

Body: nơi chứa thông tin client sẽ điền

1. Collection

* **Là nơi lưu trữ các request**.
* Import: dạng link hoặc json
* Export: dạng json
* Run collection : gọi tuần tự các request bên trong collection

1. Environment

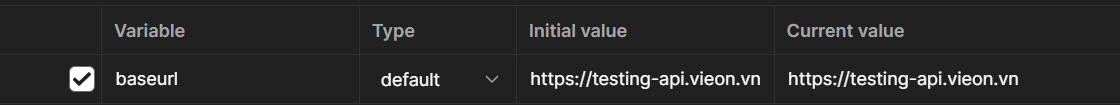
* Có 2 loại environment:

+ **Local**: chỉ định rõ sẽ được dùng cho request hay collection.

+**Globals**: tất cả key-value được dùng chung cho tất cả request.

* Import: dạng link hoặc file json
* Export: xuất dạng json
* Set key-value:

Ví dụ:



1. Cách test API

* Syntax testing: tập trung vào method check điều kiện: valid data đúng hay reject data sai hay không.

Ví dụ :

* Bỏ trống trường bắt buộc
* Bỏ trống trường không bắt buộc
* Nhập sai format,sai thông tin,…
* Functional testing: kiểm tra xử lí dữ liệu và thực hiện 1 chức năng có đúng hay không.

Ví dụ:

* Thực hiện API Update Phone -> Kiểm tra lại API Get Phone
* Test scenarios: ghép các API lại và test theo từng testcase.

Ví dụ:

* Thực hiện API register user -> API nhập OTP -> API login -> API Menu -> API logout
* Thực hiện API login -> API Menu-> API Get Profile -> API logout

1. Check valid json schema

Mục đích: dùng để kiểm tra kiểu dữ liệu của các trường dữ liệu

Ví dụ : Copy response dạng json sau đó dùng tool để generate nó thành schema sẽ cho kết quả là: Trường id có type là String, Trường status có type là integer,…

