

JWT

BẢO MẬT RESTFUL API VỚI JWT



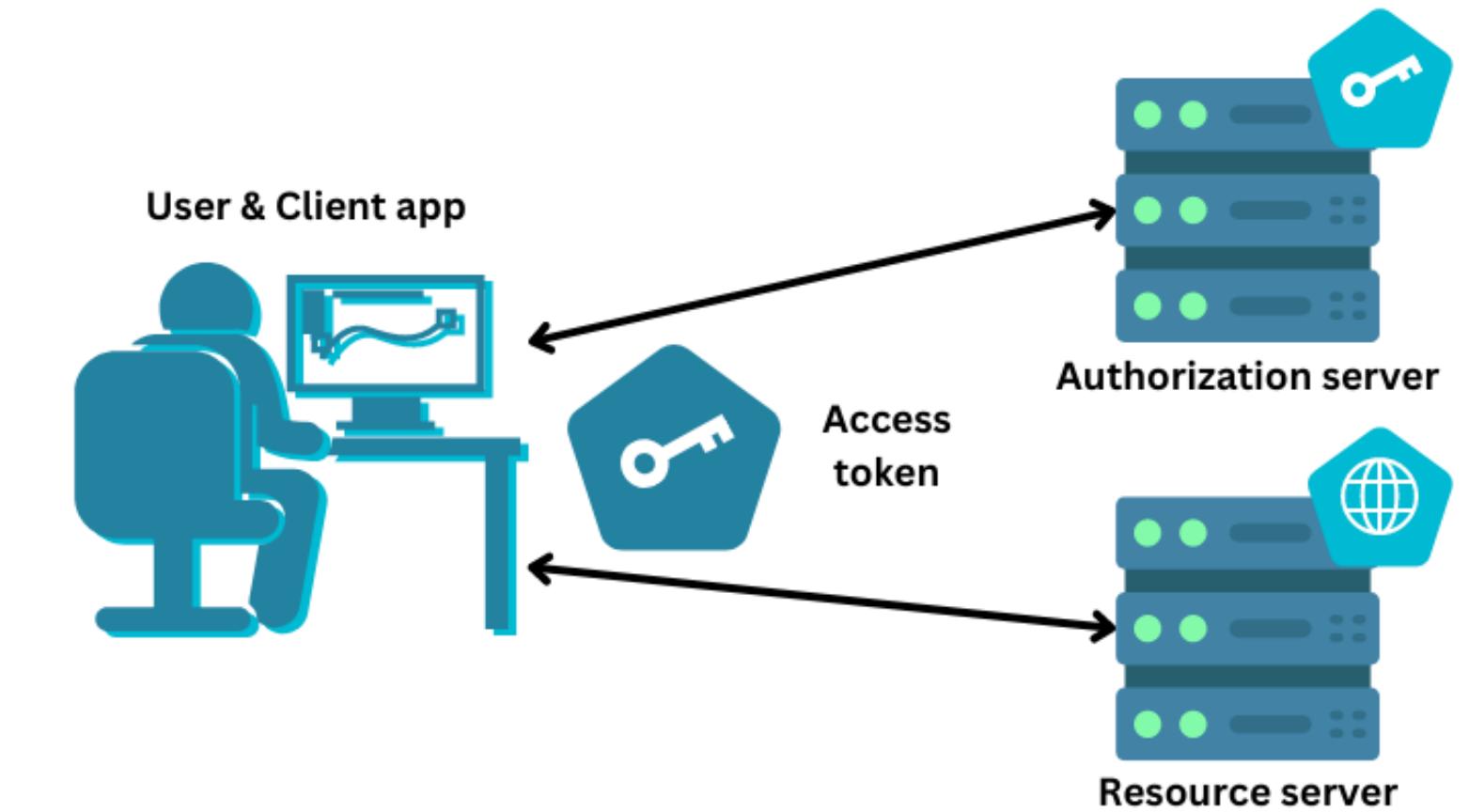
MỤC TIÊU CỦA BÀI THỰC HÀNH

- Hiểu cơ chế bảo mật của JWT
- Biết cách tạo, mã hóa và giải mã JWT
- Áp dụng JWT để bảo vệ API trong dự án PHP



GIỚI THIỆU TỔNG QUÁT VỀ JWT

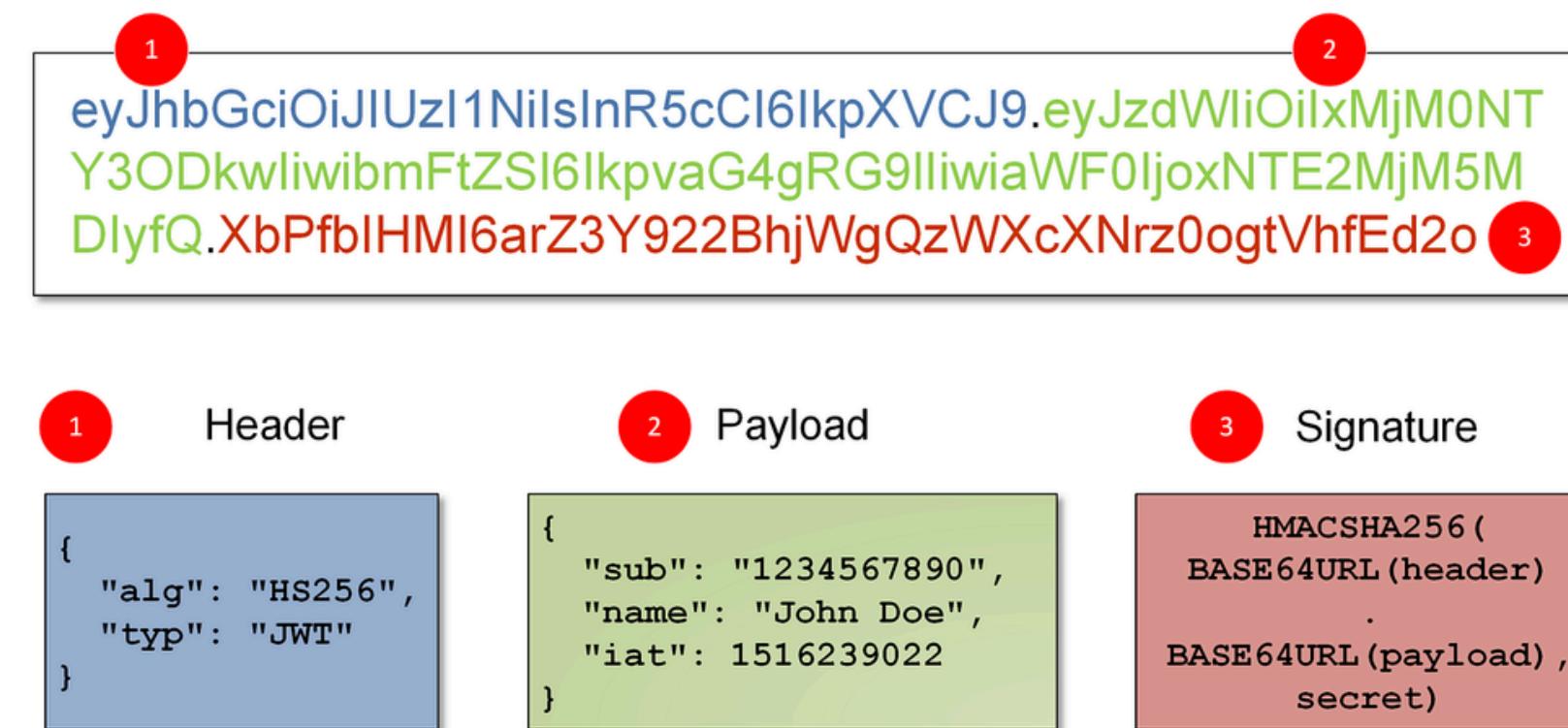
- JWT (JSON Web Token) là một tiêu chuẩn mở (RFC 7519).
- Dùng để truyền tải thông tin giữa các bên một cách an toàn và nhỏ gọn (compact & secure).
- Thường dùng để xác thực (authentication) và phân quyền truy cập (authorization) trong các ứng dụng web, mobile, API RESTful.



CẤU TRÚC CỦA JWT

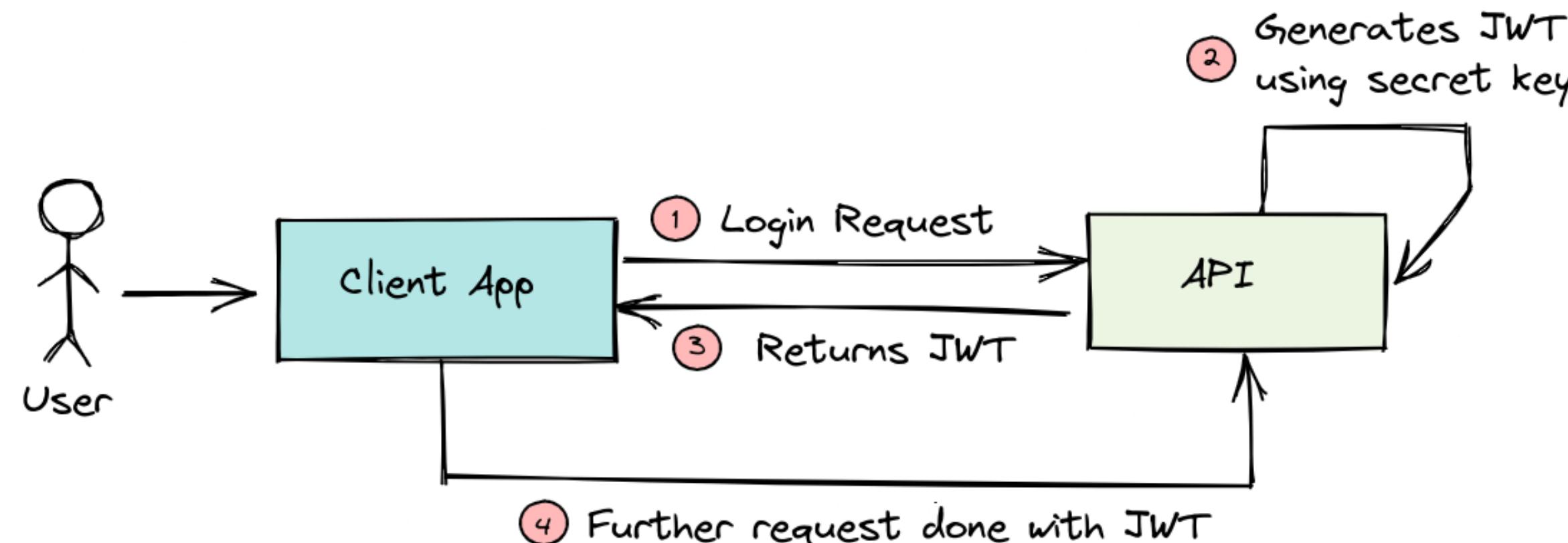
Cấu trúc của JWT gồm 3 phần chính:

1. **Header** – Thông tin về thuật toán mã hóa (ví dụ: HS256)
2. **Payload** – Dữ liệu muốn truyền đi (ví dụ: user_id, role, exp, ...)
3. **Signature** – Chữ ký số đảm bảo tính toàn vẹn



JWT ĐƯỢC DÙNG ĐỂ LÀM GÌ?

- Thay thế session trong xác thực người dùng
- Truyền thông tin user giữa client ↔ server
- Không cần lưu trạng thái người dùng (stateless) trên server



GIỚI THIỆU VỀ COMPOSER

Composer là một trình quản lý thư viện (**dependency manager**) dành cho ngôn ngữ PHP. Cho phép bạn:

- Cài đặt thư viện bên ngoài dễ dàng.
- Quản lý phiên bản thư viện.
- Tự động tải (autoload) class.
- Cấu trúc dự án chuyên nghiệp hơn.

Composer hoạt động như thế nào?

- Dựa vào tệp composer.json để xác định các thư viện cần thiết.
- Tự động tải thư viện và các phụ thuộc vào thư mục vendor.
- Tạo file composer.lock để đảm bảo phiên bản ổn định.

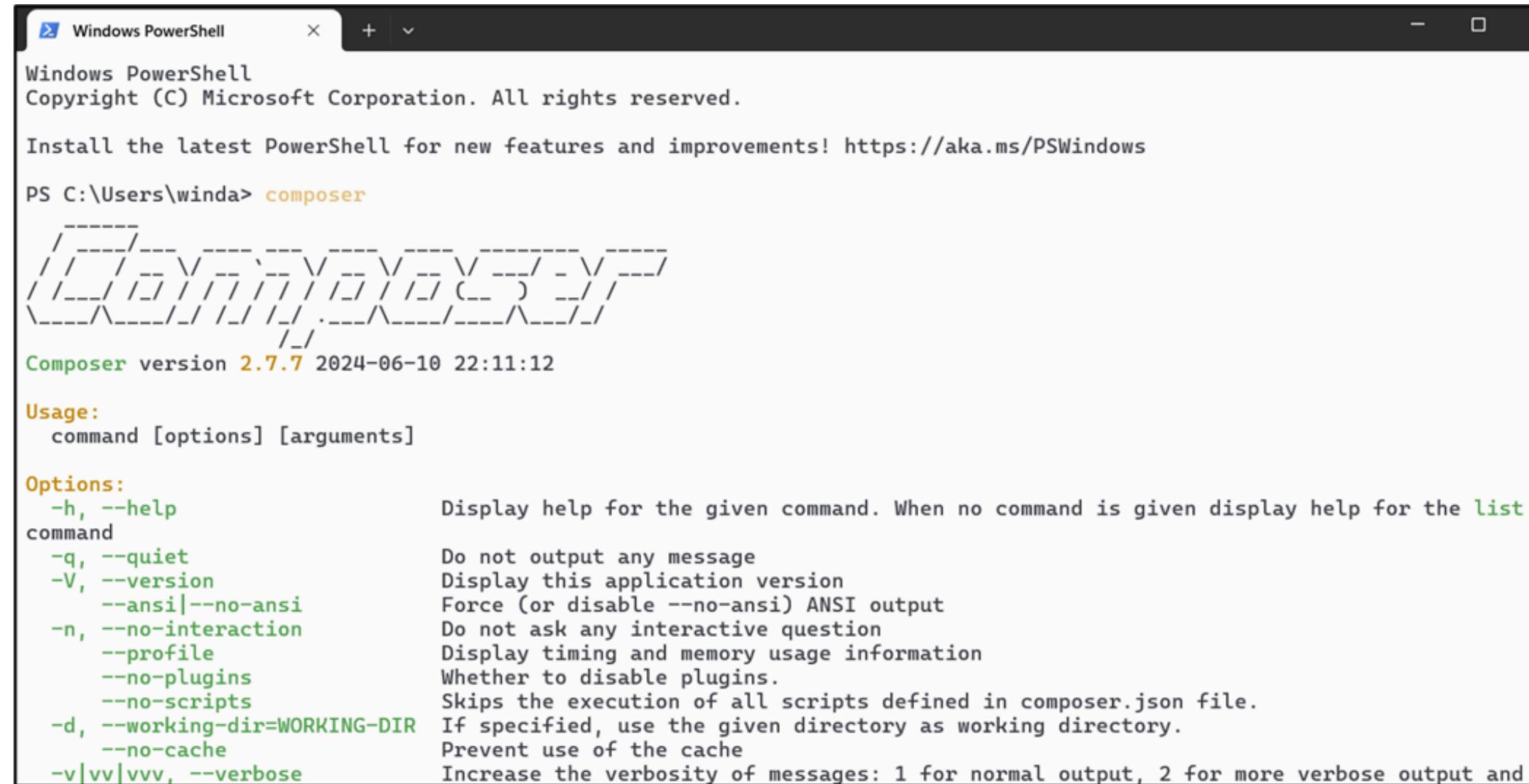
🔗 Truy cập trang chính thức: <https://getcomposer.org>



CÀI ĐẶT COMPOSER (WINDOWS/LINUX/MACOS)

🔗 Truy cập trang chính thức: <https://getcomposer.org>

Cài đặt và kiểm tra bằng lệnh: **composer**



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\winda> composer
-----
Composer version 2.7.7 2024-06-10 22:11:12

Usage:
  command [options] [arguments]

Options:
  -h, --help                                Display help for the given command. When no command is given display help for the list
  command
  -q, --quiet                               Do not output any message
  -V, --version                             Display this application version
  --ansi|--no-ansi                          Force (or disable --no-ansi) ANSI output
  -n, --no-interaction                      Do not ask any interactive question
  --profile                                 Display timing and memory usage information
  --no-plugins                            Whether to disable plugins.
  --no-scripts                            Skips the execution of all scripts defined in composer.json file.
  -d, --working-dir=WORKING-DIR           If specified, use the given directory as working directory.
  --no-cache                                Prevent use of the cache
  -v|vv|vvv, --verbose                     Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and
```

CÀI ĐẶT THƯ VIỆN JWT TRONG PHP

Cài đặt thư viện JWT với Composer

- ● ●

```
1 // Sử dụng Composer để cài đặt:  
2 composer require firebase/php-jwt  
3  
4 // Nếu chưa có composer.json, bạn có thể khởi tạo bằng:  
5 composer init  
6
```

Cài đặt thư viện JWT với Composer

- ● ●

```
1 project/  
2   └── vendor/  
3     └── firebase/  
4       └── php-jwt/  
5         └── composer.json  
6         └── index.php
```

TẠO JWT SAU KHI ĐĂNG NHẬP

Khi nào tạo JWT?

- Sau khi người dùng đăng nhập thành công (kiểm tra tài khoản + mật khẩu đúng).
- Thay vì tạo session, ta sẽ tạo JWT và gửi về client.
- Client lưu token trong localStorage hoặc Authorization header.

Ghi chú quan trọng:

- exp: Hạn dùng của token (tính bằng giây).
- HS256: Thuật toán mã hóa **HMAC SHA-256**
- JWT nên được lưu ở client và gửi với mỗi request cần xác thực.

```
1 <?php
2
3 use Firebase\JWT\JWT;
4
5 $secret_key = "my_secret_key";
6 $payload = [
7     "iss" => "http://localhost",
8     "aud" => "http://localhost",
9     "iat" => time(),
10    "exp" => time() + 3600, // hết hạn sau 1 giờ
11    "data" => [
12        "id" => $user["id"],
13        "username" => $user["username"]
14    ]
15 ];
16
17 $jwt = JWT::encode($payload, $secret_key, 'HS256');
18
```

GỬI TOKEN TRONG HEADER

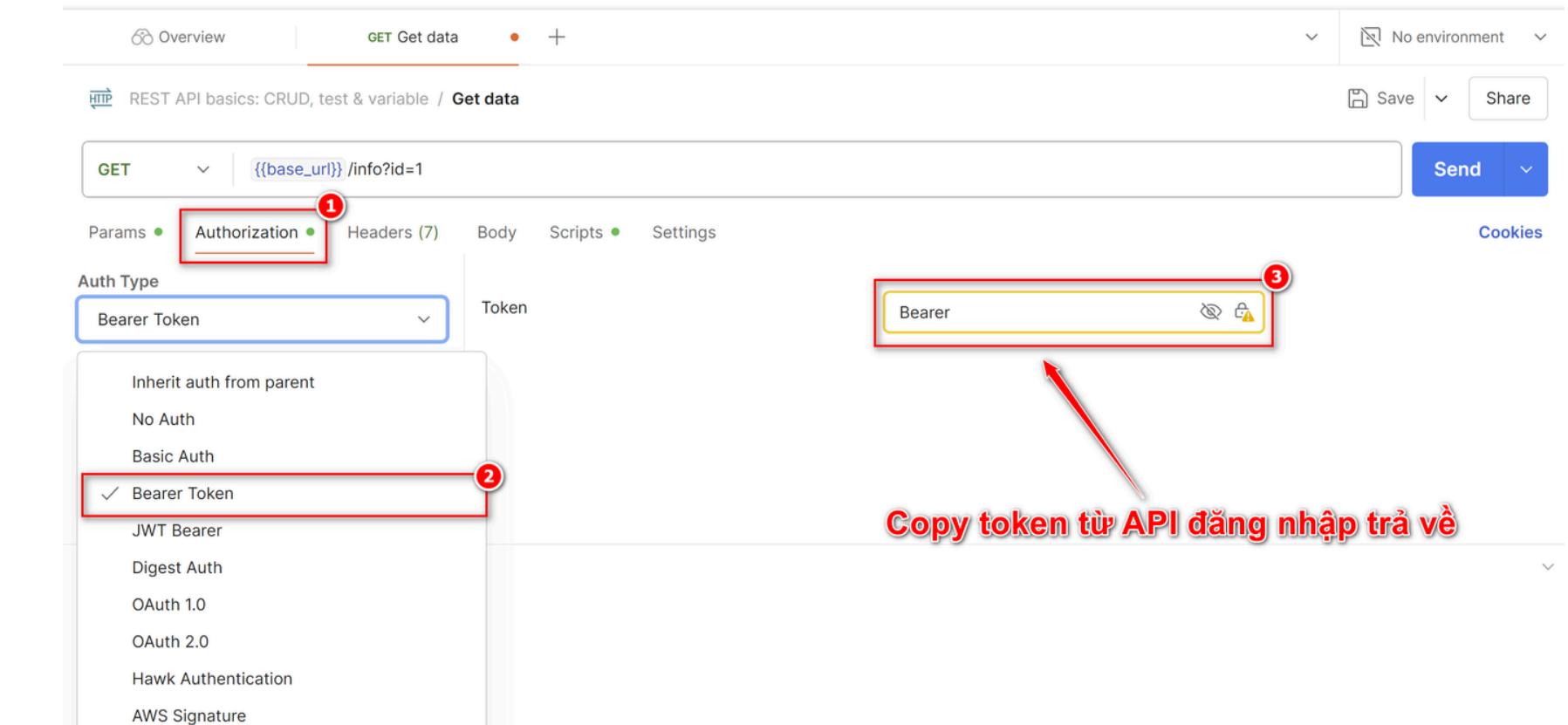
Vì sao gửi token qua Header?

- Bảo mật hơn so với gửi qua query string hoặc body.
- Tuân thủ chuẩn HTTP Authorization Header.
- Dễ kiểm soát trong middleware và gateway API.

Gửi JWT bằng Postman

Mã kiểm tra Header trong PHP

```
1 <?php
2
3 $headers = getallheaders();
4 if (isset($headers['Authorization'])) {
5     $authHeader = $headers['Authorization'];
6     $token = str_replace('Bearer ', '', $authHeader);
7     // Tiếp tục kiểm tra token...
8 }
```



XÁC THỰC TOKEN Ở SERVER

Mục tiêu xác thực

- Đảm bảo request đến từ người dùng đã đăng nhập
- Token hợp lệ: chưa hết hạn, được ký bằng secret key đúng

Các bước xác thực JWT

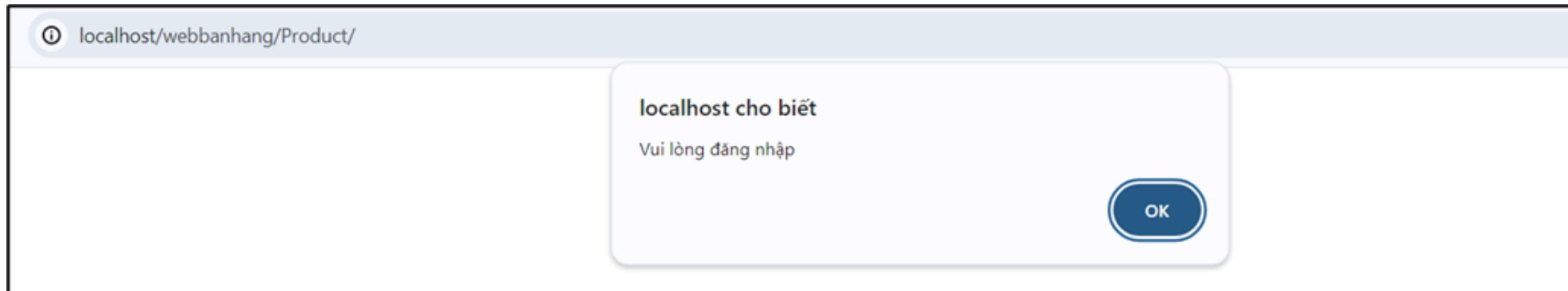
- Lấy token từ header
- Giải mã (decode) token
- Kiểm tra chữ ký (signature) và hạn (exp)
- Thực hiện chức năng sau:
 - Nếu hợp lệ → cho phép tiếp tục
 - Nếu không hợp lệ → trả lỗi 401 Unauthorized

```
● ○ ●
1 <?php
2
3 use Firebase\JWT\JWT;
4 use Firebase\JWT\Key;
5
6 $jwt = $token_from_header;
7 $secret_key = "my_secret_key";
8
9 try {
10     $decoded = JWT::decode($jwt, new Key($secret_key, 'HS256'));
11     // Truy cập payload: $decoded->data
12 } catch (Exception $e) {
13     http_response_code(401);
14     echo json_encode(["message" => "Token không hợp lệ"]);
15     exit;
16 }
17
```

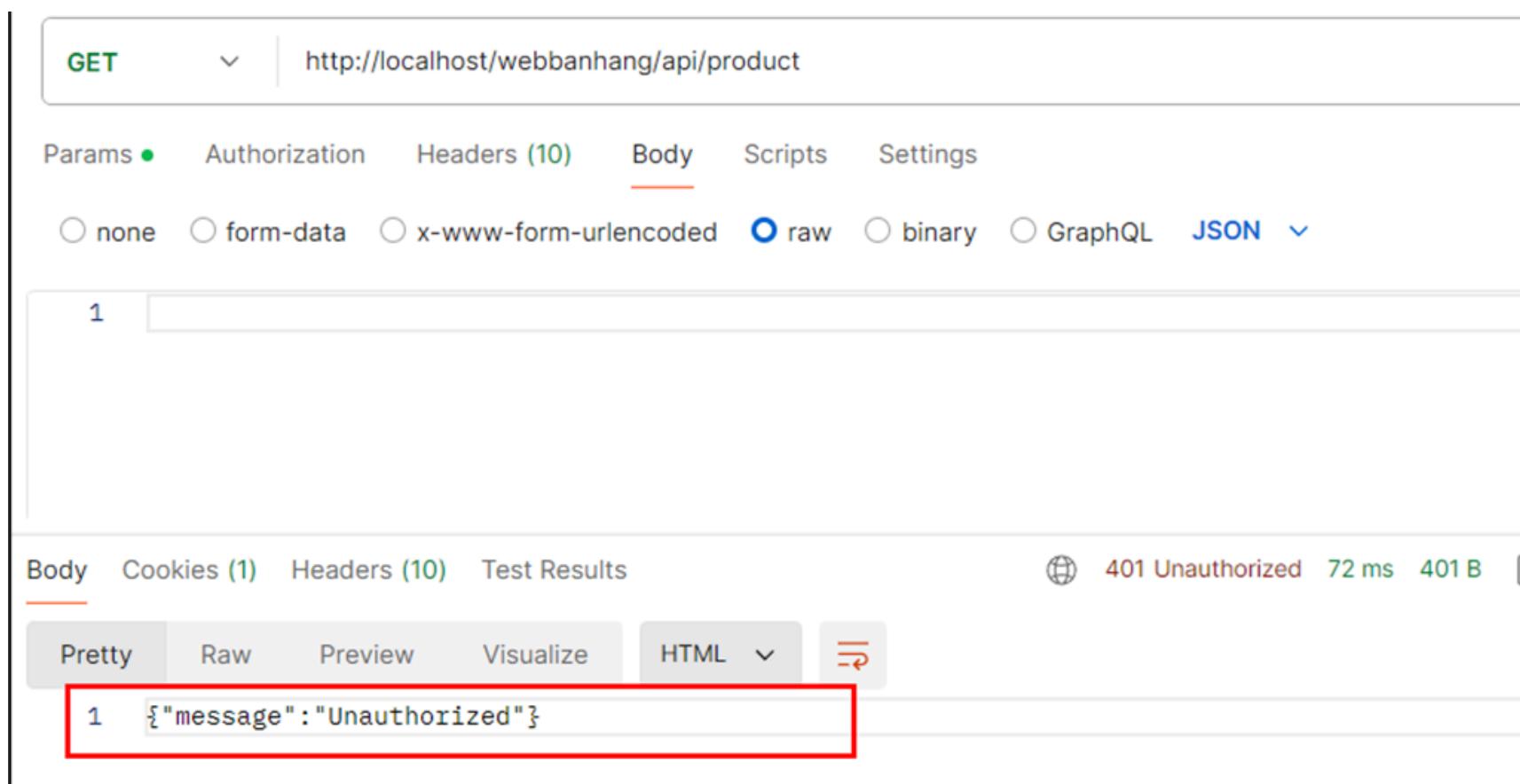
LỖI THƯỜNG GẶP KHI XỬ LÝ JWT

1. Token không tồn tại do không gửi token trong Authorization Header hoặc sai định dạng Bearer.
=> **Cách khắc phục:** *Tăng exp* nếu cần lâu hơn, Thông báo người dùng đăng nhập lại.
2. Token hết hạn (exp).
=> **Cách khắc phục:** Đảm bảo giải mã *dùng đúng key* và thuật toán như khi tạo token.
3. Sai secret key hoặc thuật toán dùng sai secret_key so với lúc mã hóa token.
=> **Cách khắc phục:** Không giải mã thủ công, luôn *dùng thư viện (như Firebase|JWT)*
4. Token bị sửa hoặc không hợp lệ signature không còn đúng.
=> **Cách khắc phục:** Cần luôn bọc JWT::decode() trong *try { ... } catch (Exception \$e) { ... }*
5. Không xử lý ngoại lệ (Exception) gây lỗi trống hoặc crash server.

KẾT QUẢ KHỞI CHẠY DỰ ÁN



Kiểm thử trên trình duyệt



GET http://localhost/webbanhang/api/product

Params • Authorization Headers (10) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

1

Body Cookies (1) Headers (10) Test Results

Pretty Raw Preview Visualize HTML

1 {"message": "Unauthorized"}

Kiểm thử trong Postman