Báo cáo kết thúc dự án cá nhân – Blog Web API

Link repo: <https://github.com/toiQS/Blog-Web>

Mục lục

[**Giới thiệu** 1](#_Toc169397402)

[**Mục tiêu và Thách Thức của Dự Án** 1](#_Toc169397403)

[**Tổng Quan Giải Pháp** 2](#_Toc169397404)

[**Mô tả Use Case của hệ thống web API:** 3](#_Toc169397405)

[**Mô tả** 4](#_Toc169397406)

[**Yêu cầu** 4](#_Toc169397407)

[**Các bước thiết lập** 5](#_Toc169397408)

[**Thiết lập vùng nhớ tạm thời** 7](#_Toc169397409)

[**Chạy và Kiểm Thử:** 7](#_Toc169397410)

[**Sử dụng Swagger** 8](#_Toc169397411)

[**Sử dụng Post man** 9](#_Toc169397412)

[**Libraries and Frameworks** 13](#_Toc169397413)

# **Giới thiệu**

Blog-Web là một dự án mã nguồn mở nhằm cung cấp một giải pháp API mạnh mẽ để xây dựng ứng dụng blog. Sử dụng công nghệ như Entity Framework Core để quản lý cơ sở dữ liệu, Identity Framework để quản lý người dùng và xác thực, cùng với JWT (JSON Web Token) để xác thực và phân quyền an toàn, Blog-Web cung cấp một nền tảng toàn diện cho việc phát triển ứng dụng blog an toàn và đầy đủ tính năng.

# **Mục tiêu và Thách Thức của Dự Án**

Các thách thức chính được giải quyết trong việc phát triển Blog-Web bao gồm:

* **Quản lý Cơ sở Dữ liệu:** Lưu trữ và truy xuất dữ liệu một cách hiệu quả và bảo mật với Entity Framework Core, hỗ trợ nhiều hệ quản trị cơ sở dữ liệu khác nhau.
* **Quản lý Người Dùng:** Triển khai chức năng đăng ký, đăng nhập và quản lý người dùng sử dụng Identity Framework.
* **Xác thực và Phân Quyền:** Sử dụng JWT để xác thực người dùng an toàn và phân quyền dựa trên vai trò người dùng.
* **Quản lý Nội Dung:** Cung cấp các API cho các hoạt động CRUD (Create, Read, Update, Delete) trên bài đăng và bình luận, đảm bảo quản lý nội dung hiệu quả.

# **Tổng Quan Giải Pháp**

Blog-Web giải quyết các thách thức này thông qua các phương pháp sau:

* **Entity Framework Core:** Cung cấp phương pháp tiếp cận hướng đối tượng cho các tương tác cơ sở dữ liệu, đơn giản hóa thiết kế và bảo trì cơ sở dữ liệu.
* **Identity Framework:** Hỗ trợ quản lý người dùng với các tính năng như đăng ký, đăng nhập, quản lý vai trò và tích hợp với Entity Framework Core để lưu trữ và quản lý thông tin người dùng.
* **Xác thực JWT:** Tăng cường bảo mật bằng việc sử dụng JSON Web Tokens cho xác thực người dùng và quản lý phiên làm việc, giảm tải cho máy chủ và cải thiện quản lý phiên làm việc đồng thời.
* **API Quản lý Nội Dung:** Cho phép quản lý linh hoạt nội dung blog thông qua các API cho việc tạo, cập nhật và xóa bài đăng và bình luận.

**Sơ lược về cơ sở dữ liệu**

Dạng bảng:

A computer screen shot of a computer

Description automatically generated

# **Mô tả Use Case của hệ thống web API:**

1. **Quản lý người dùng:**
   * **Đăng ký người dùng:** Người dùng có thể đăng ký tài khoản mới trên hệ thống bằng cách cung cấp các thông tin như tên đăng nhập, địa chỉ email và mật khẩu.
   * **Đăng nhập:** Người dùng có thể đăng nhập vào hệ thống bằng cách cung cấp email và mật khẩu.
   * **Đăng xuất:** Người dùng có thể đăng xuất khỏi hệ thống để kết thúc phiên làm việc hiện tại.
2. **Quản lý chủ đề (Theme):**
   * **Lấy danh sách chủ đề:** Người dùng có thể xem danh sách các chủ đề hiện có trên hệ thống.
   * **Tìm kiếm chủ đề:** Người dùng có thể tìm kiếm chủ đề bằng cách nhập từ khóa vào ô tìm kiếm.
   * **Xem chi tiết chủ đề:** Người dùng có thể xem chi tiết của một chủ đề bằng cách xem thông tin chi tiết về chủ đề đó.
   * **Thêm chủ đề mới:** Người dùng có quyền admin có thể thêm một chủ đề mới vào hệ thống bằng cách cung cấp thông tin về chủ đề.
3. **Quản lý hình ảnh (Image):**
   * **Lấy danh sách hình ảnh:** Người dùng có thể xem danh sách tất cả hình ảnh có sẵn trên hệ thống.
   * **Tìm kiếm hình ảnh:** Người dùng có thể tìm kiếm hình ảnh bằng cách nhập từ khóa liên quan đến hình ảnh.
   * **Xem chi tiết hình ảnh:** Người dùng có thể xem chi tiết của một hình ảnh cụ thể bằng cách xem thông tin chi tiết về hình ảnh đó.
   * **Cập nhật hình ảnh:** Người dùng có quyền admin có thể cập nhật thông tin của một hình ảnh đã có sẵn trên hệ thống.
4. **Quản lý bình luận (Comment):**
   * **Lấy danh sách bình luận:** Người dùng có thể xem danh sách tất cả các bình luận được thêm vào các bài đăng trên hệ thống.
   * **Thêm bình luận:** Người dùng có thể thêm một bình luận mới cho một bài đăng cụ thể bằng cách cung cấp nội dung bình luận.
   * **Trả lời bình luận:** Người dùng có thể trả lời một bình luận đã có trên hệ thống bằng cách chỉ định bình luận gốc và cung cấp nội dung trả lời.
   * **Xóa bình luận:** Người dùng có quyền admin có thể xóa bỏ một bình luận khỏi hệ thống.
5. **Quản lý bài viết (Poster):**
   * **Lấy danh sách bài viết:** Người dùng có thể xem danh sách các bài viết hiện có trên hệ thống.
   * **Xem chi tiết bài viết:** Người dùng có thể xem chi tiết của một bài viết cụ thể bằng cách xem thông tin chi tiết về bài viết đó.
   * **Thêm bài viết mới:** Người dùng có quyền admin có thể thêm một bài viết mới vào hệ thống bằng cách cung cấp các thông tin về bài viết.
6. **Quản lý tệp tin và hình ảnh (File/Image Management):**
   * **Tải lên tệp tin:** Người dùng có thể tải lên các tệp tin và hình ảnh mới lên hệ thống để sử dụng trong các bài viết hoặc nội dung khác.
   * **Xóa tệp tin:** Người dùng có quyền admin có thể xóa bỏ các tệp tin và hình ảnh không cần thiết khỏi hệ thống để giải phóng không gian lưu trữ.

# **Mô tả**

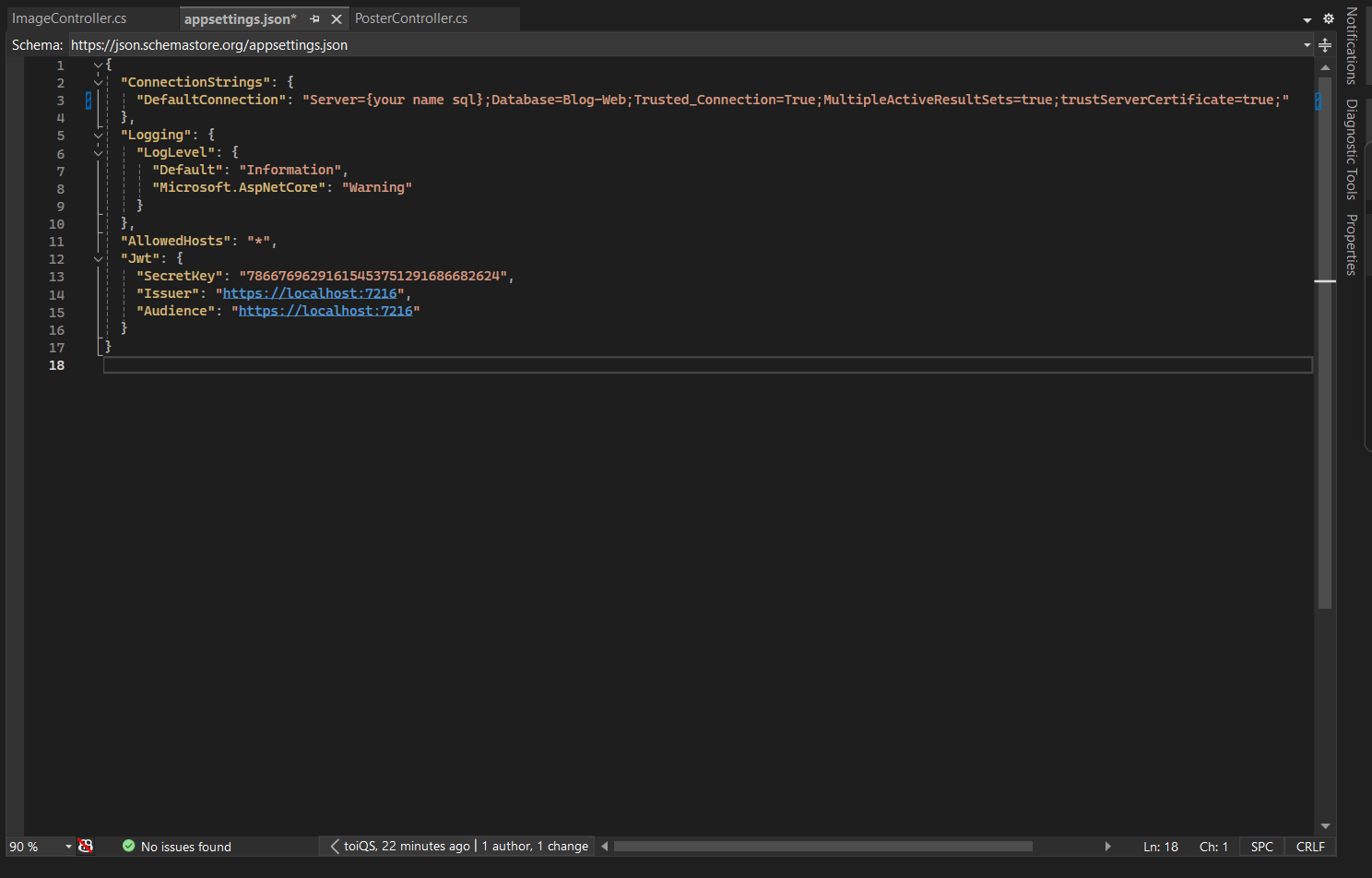
* **Người dùng (User):** Các người dùng của hệ thống có thể là người dùng thông thường hoặc quản trị viên (Admin). Người dùng có thể thực hiện các hoạt động quản lý tài khoản, quản lý bình luận, quản lý bài viết, quản lý hình ảnh, và quản lý chủ đề.
* **Hệ thống:** Là toàn bộ hệ thống Blog Web API cung cấp các dịch vụ và chức năng để quản lý các nội dung bài viết, hình ảnh, chủ đề và người dùng.

# **Yêu cầu**

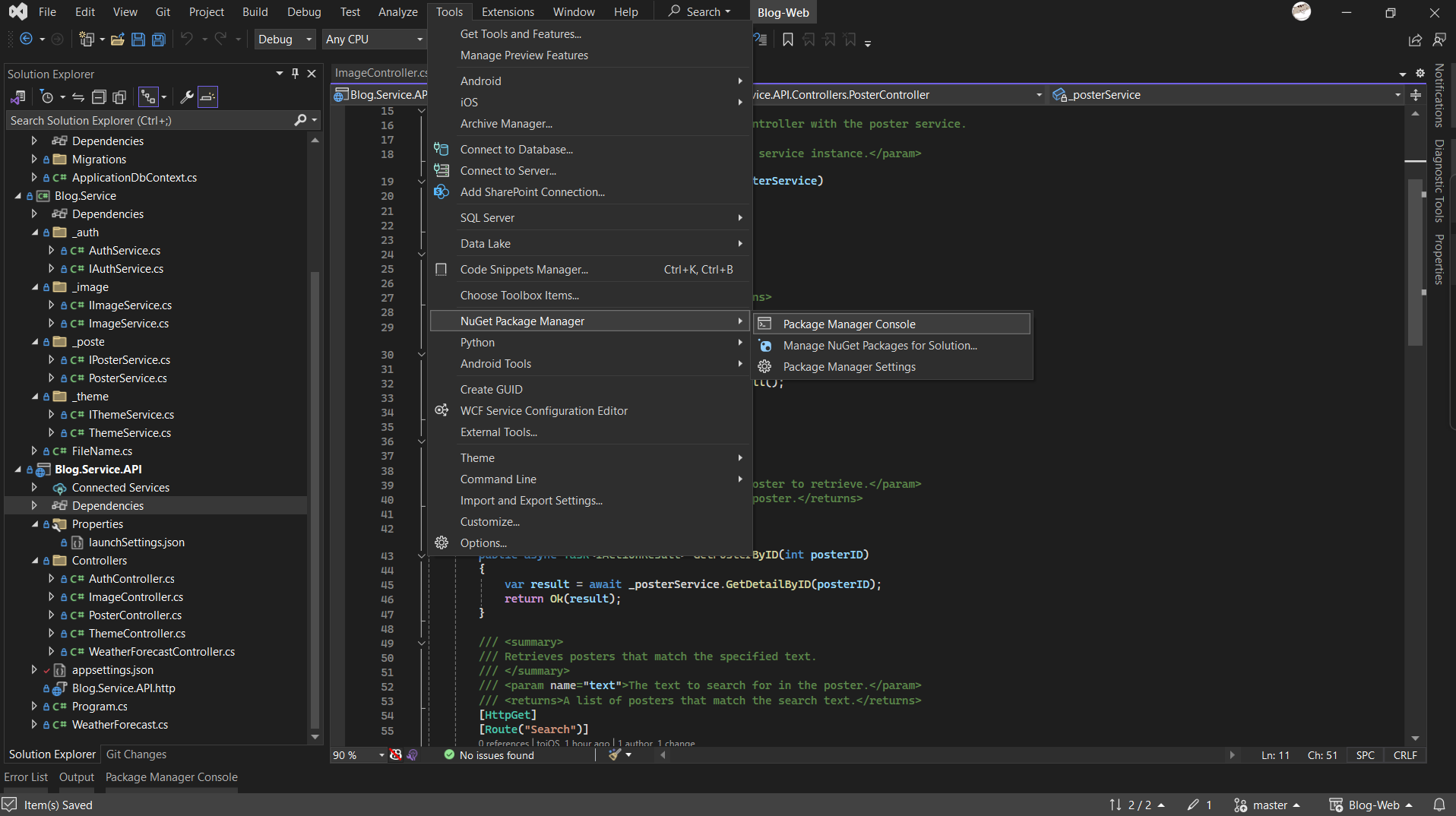
* Visual Studio 2017 trở lên
* SQL Server Management Studio (SSMS)
* Postman

# **Các bước thiết lập**

1. Cập nhật tệp appsettings.json để thay thế `{your name sql}` bằng tên của instance SSMS của bạn.

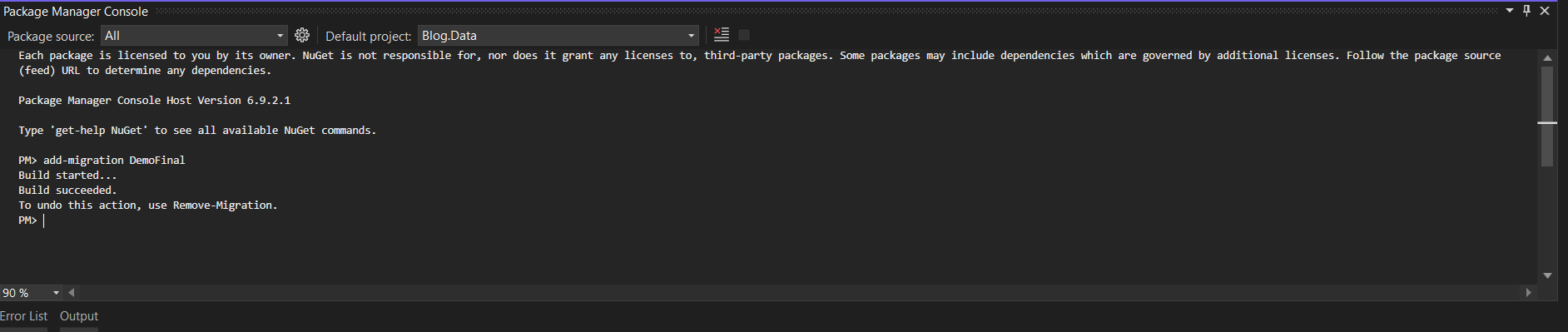


1. Mở Package Manager Console trong Visual Studio, sau đó thiết lập Default Project là Blog.Data.

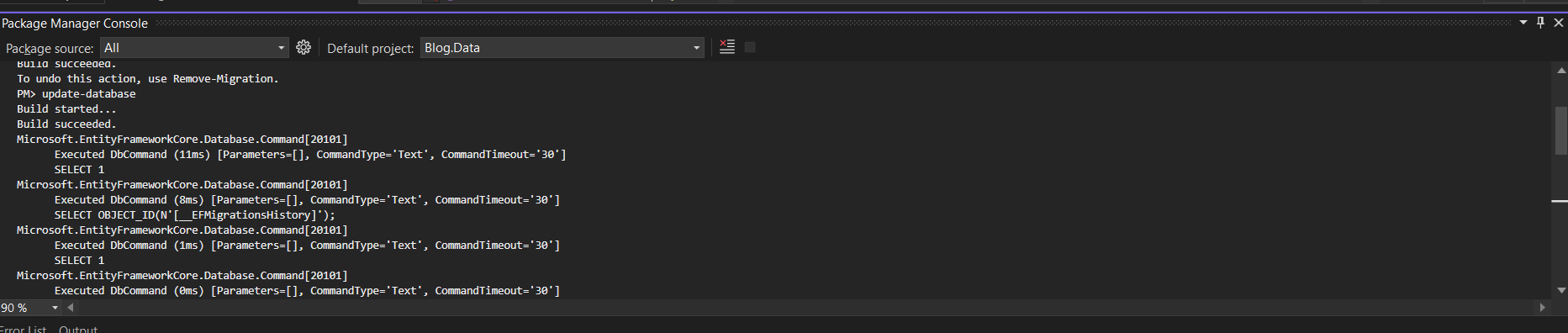


1. Nếu không có bất kỳ migration hiện tại nào trong Blog.Data:

* Sử dụng lệnh **add-migration {nameDb}** hoặc **entityframeworkcore\add-migration {nameDb}** để chuyển đổi các entities thành các bảng trong cơ sở dữ liệu.



* Chạy lệnh `update-database` để thêm các bảng vào trong SQL Server Management Studio (SSMS) của bạn.



1. Nếu đã có các migrations tồn tại hoặc không có thay đổi nào trong các entities, bạn chỉ cần chạy lệnh update-database. Lệnh này sẽ cập nhật cơ sở dữ liệu của bạn với các thay đổi mới nhất mà không cần tạo migrations mới.

Bạn có thể thực hiện lệnh update-database trong Package Manager Console của Visual Studio, sau khi đã chắc chắn rằng Default Project đã được thiết lập thành Blog.Data.

# **Thiết lập vùng nhớ tạm thời**

Thay đổi đường dẫn filePath thành đường dẫn cụ thể trỏ tới file memory.txt trong Blog.Service/\_auth/Memory.text.

Một số file cần thay đổi như CommentService, PosterService.

A computer screen with text

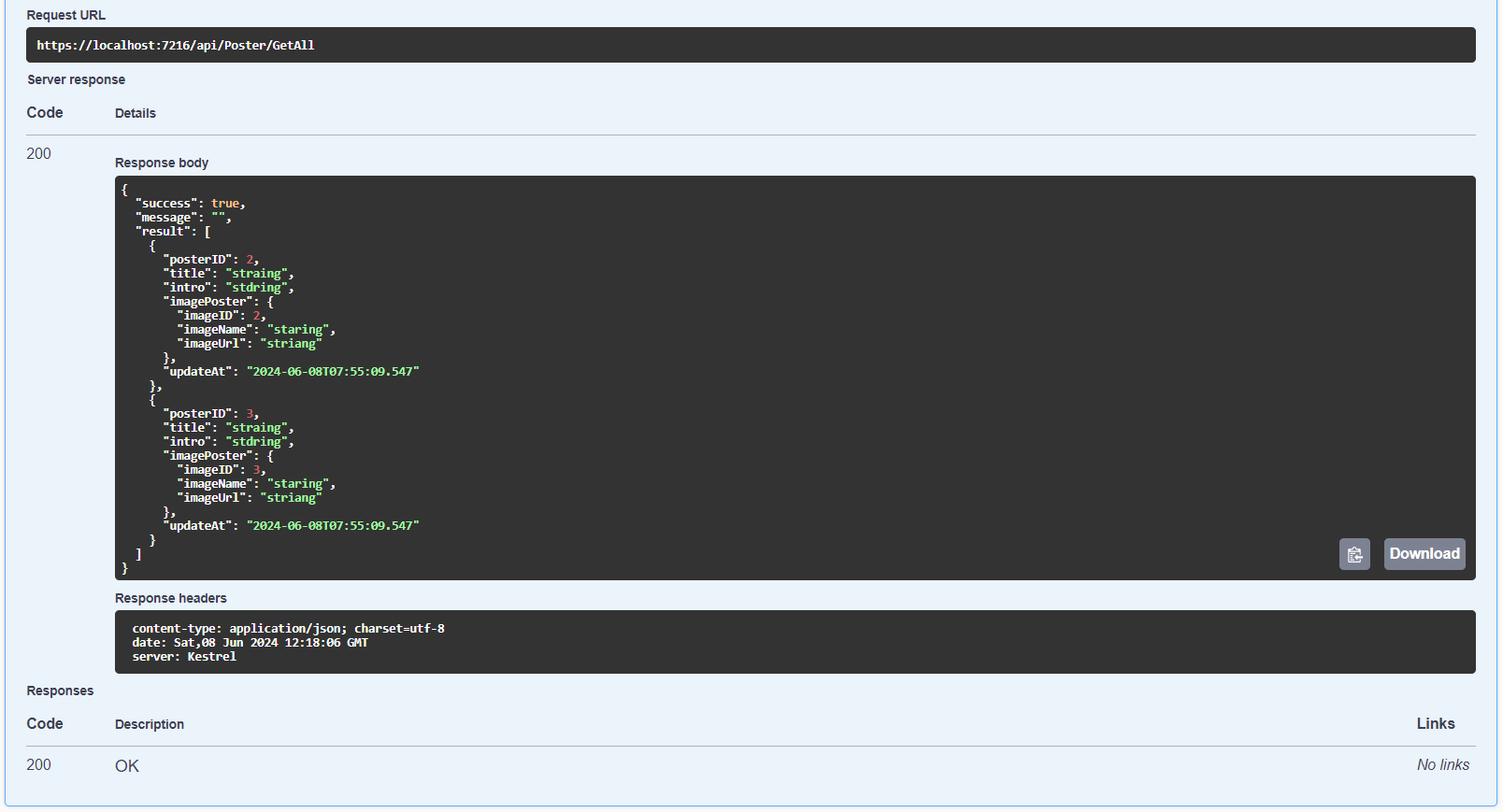
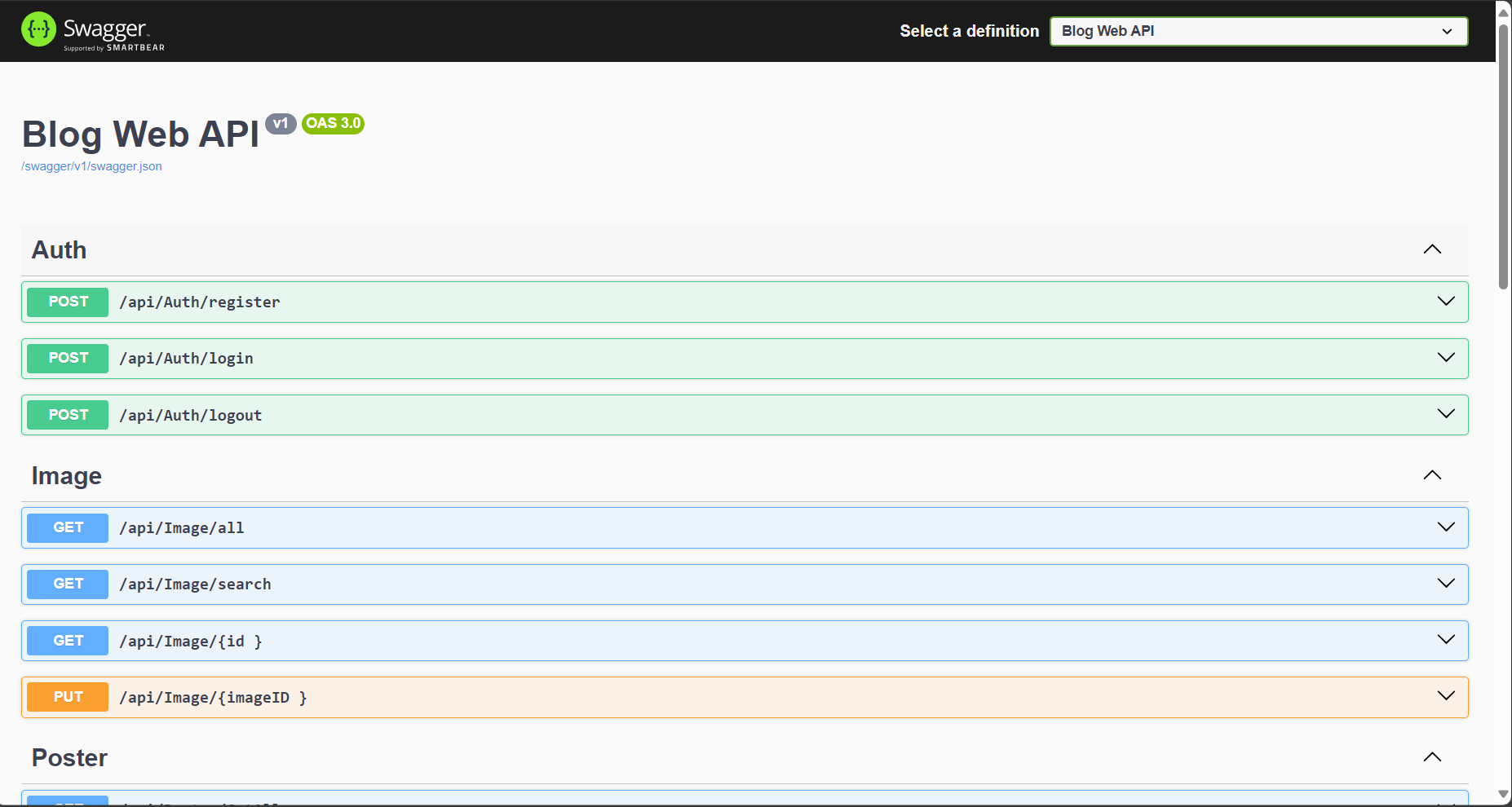
Description automatically generated

# **Chạy và Kiểm Thử**

* Khởi động ứng dụng để tương tác với các API sử dụng Swagger UI hoặc Postman.
* Các hoạt động không yêu cầu xác thực có thể được thử nghiệm trực tiếp thông qua Swagger UI.
* Đối với các hoạt động yêu cầu xác thực (ví dụ như CRUD), nhận mã JWT từ Swagger và sử dụng chúng trong Postman để kiểm thử API.

# **Sử dụng Swagger**

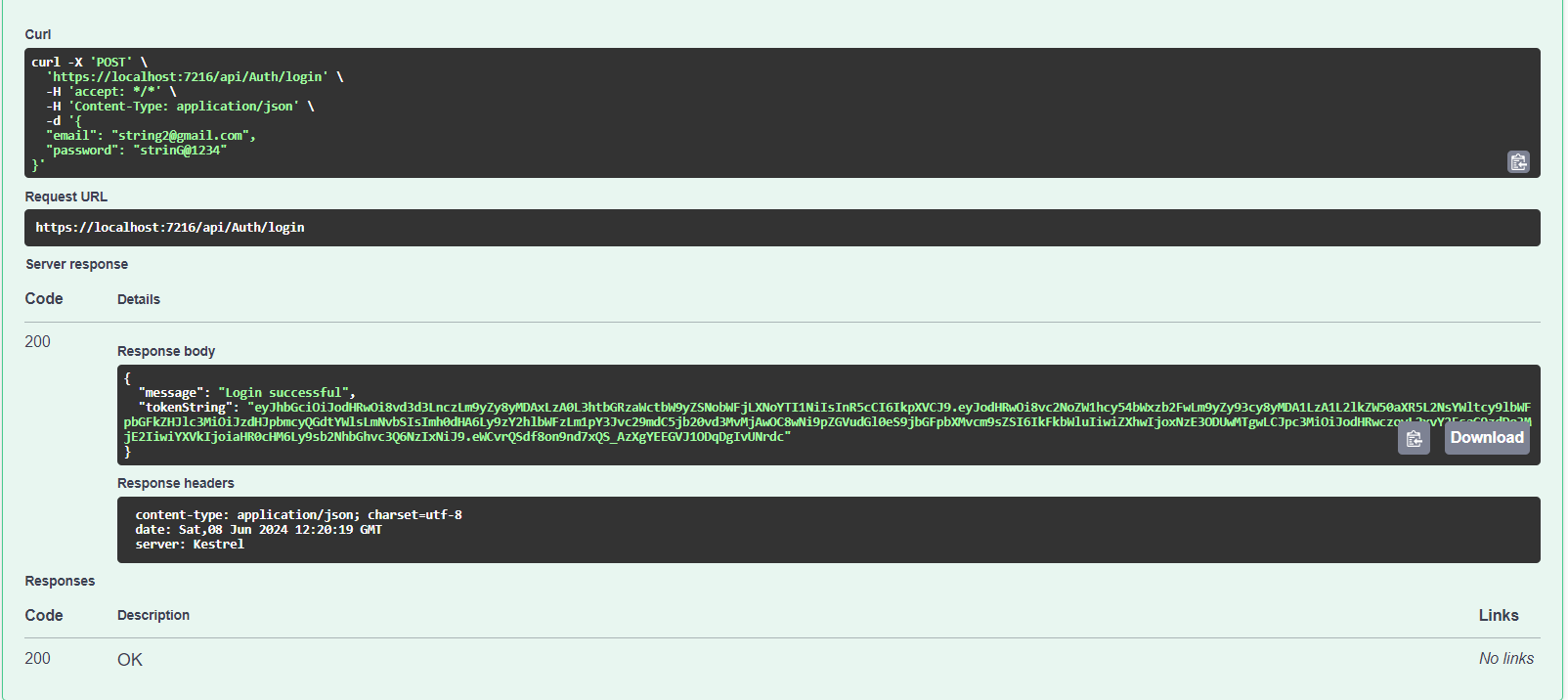
* Nhấn F5 để khởi động ứng dụng. Swagger UI sẽ mở trong trình duyệt.
* Tương tác với các tác vụ không yêu cầu xác thực, như GetAll.
* Đối với các tác vụ yêu cầu xác thực (ví dụ như các hoạt động CRUD), hãy đảm bảo đã có xác thực trước.



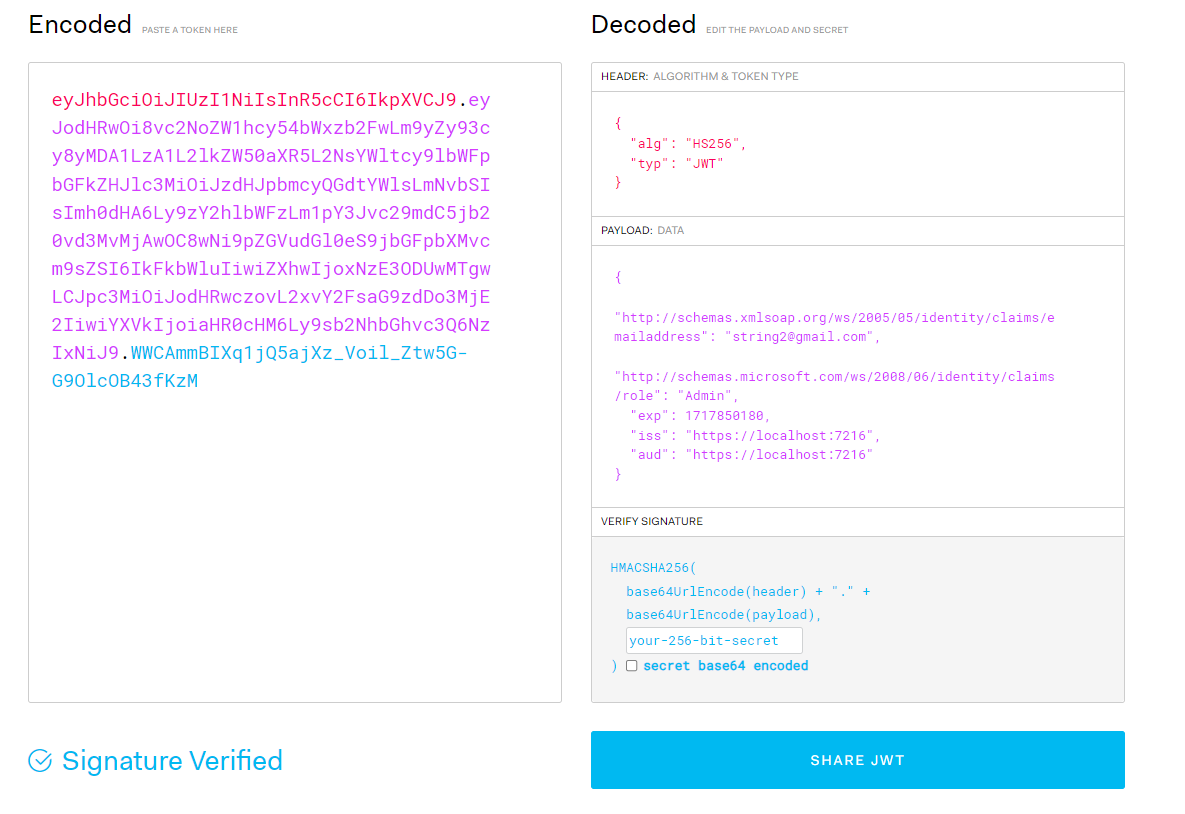
# **Sử dụng Post man**

* Sử dụng Postman để kiểm thử và tương tác với API. Đảm bảo xác thực đúng nơi cần thiết.
* Sau khi đăng nhập vào Swagger, lấy mã token có hiệu lực trong 16 phút.

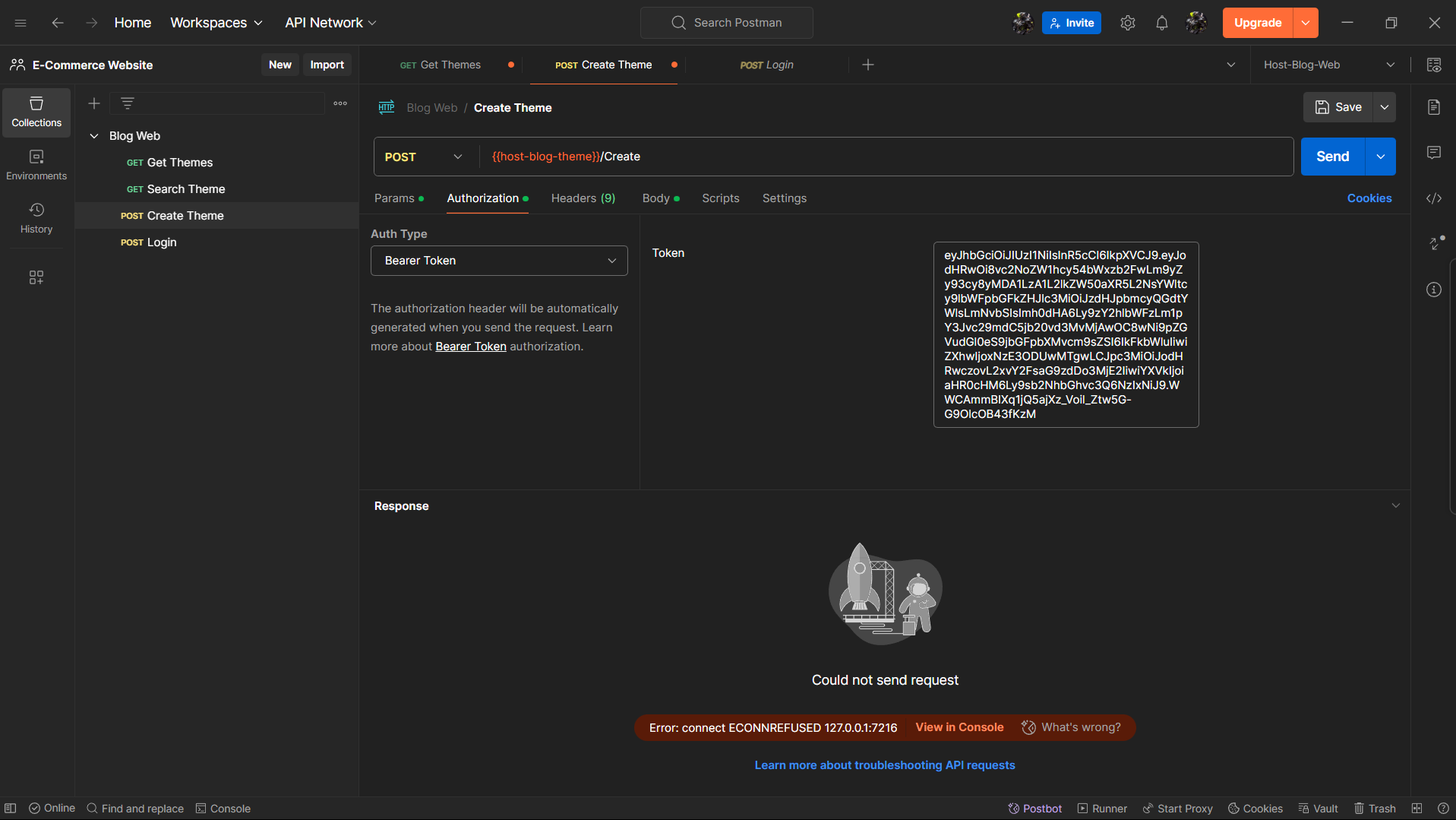
1. Để sử dụng mã token trong Postman: Sao chép mã token từ Swagger.



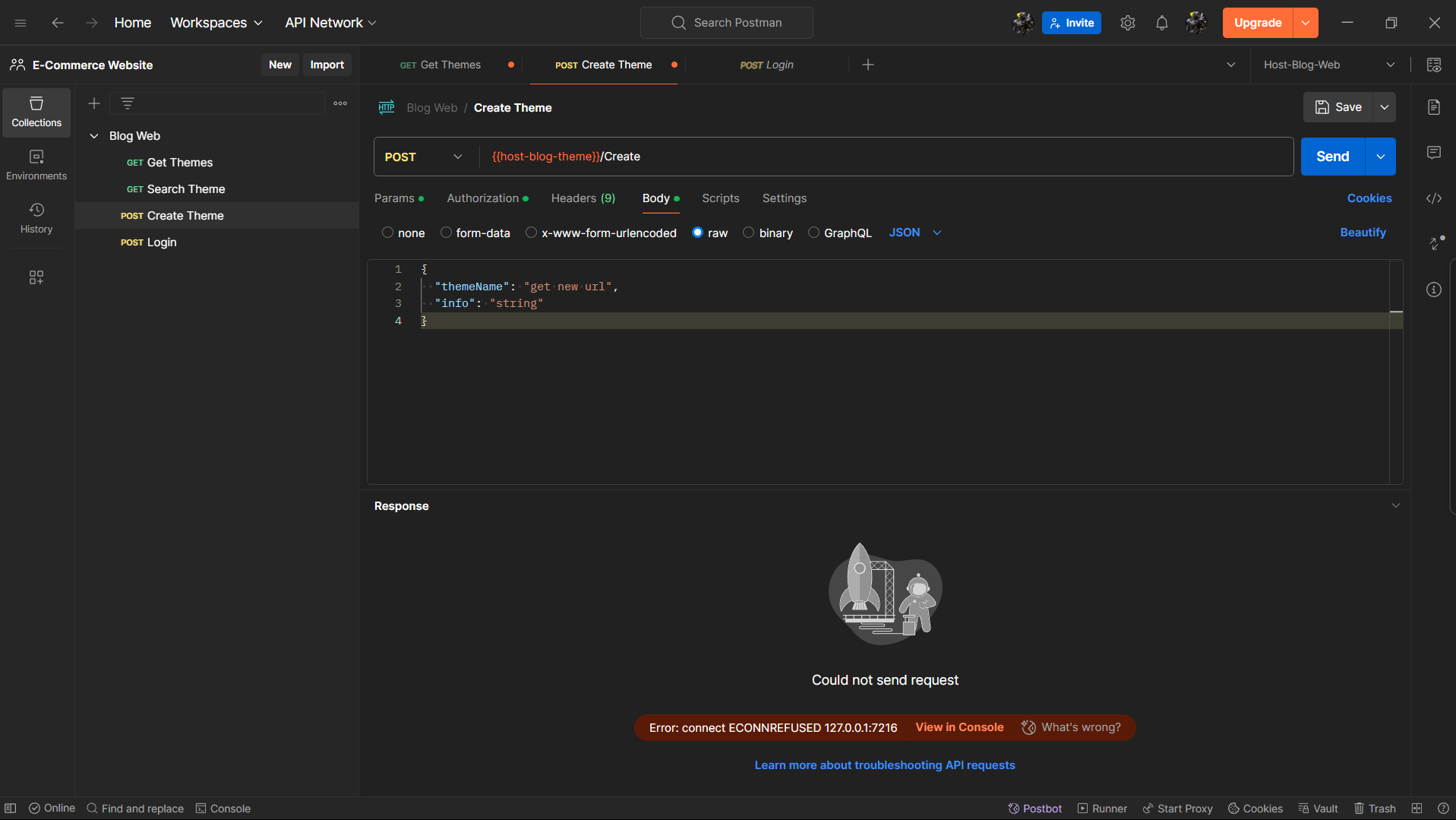
* Kiểm thử token lại trên Jwt.io.



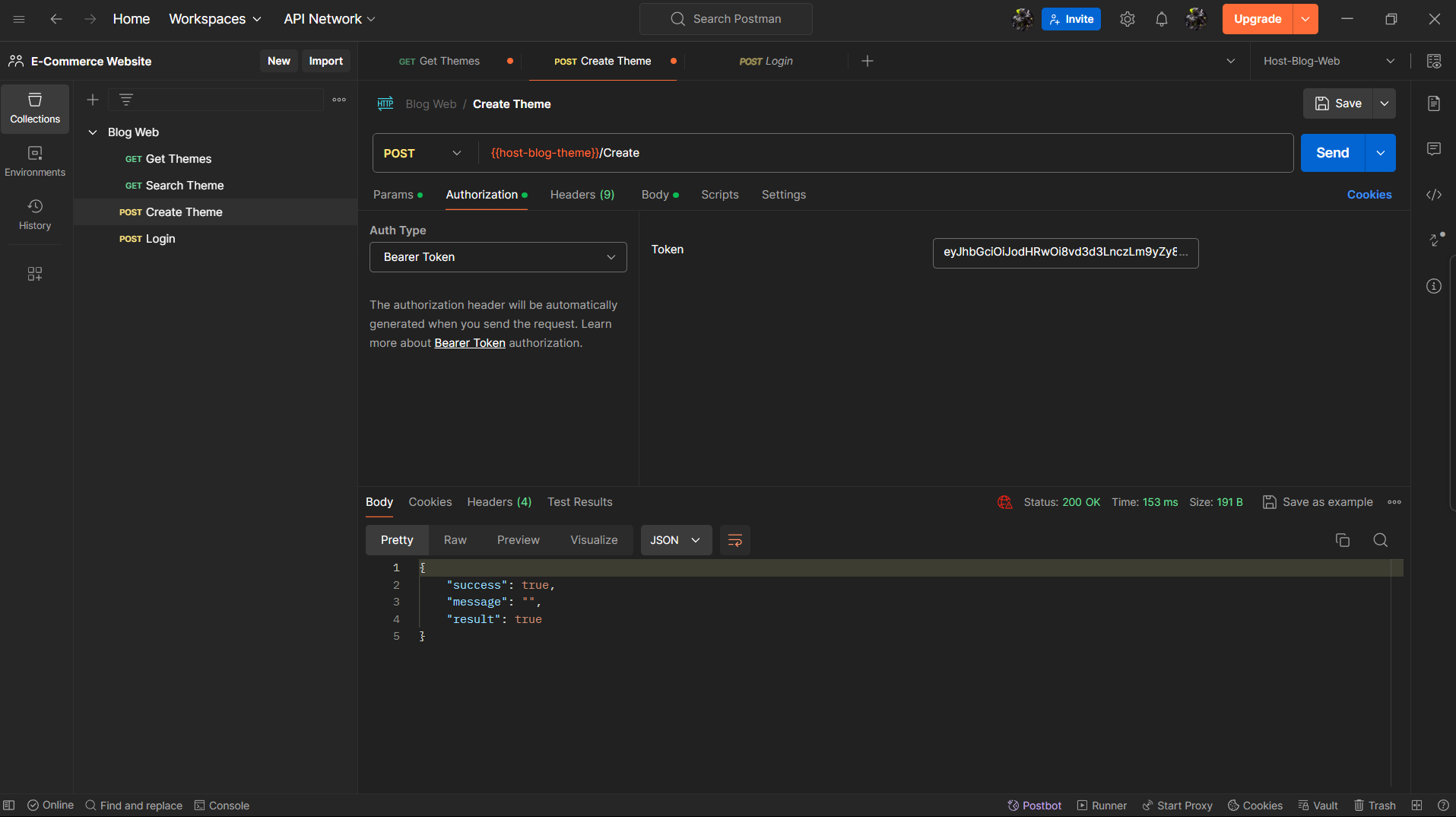
1. Trong Postman, tạo một yêu cầu mới và nhập URL điểm cuối API.
2. Chuyển đến tab Authorization, chọn Bearer Token trong danh sách Type dropdown, và dán mã token vào trường Token.



1. Chuyển đến tab Body, và chọn loại body phù hợp (ví dụ như raw với JSON) cho các hoạt động CRUD, nếu cần thiết.



1. Nhấn vào nút Send để thực hiện yêu cầu.



# **Libraries and Frameworks**

* Entity Framework Core
* Identity Framework
* Jwt Bearer