

Nghiên cứu xây dựng Restful API sử dụng bộ công cụ Swagger

Trình bày: Nguyễn Thị Sim

Time: 2019 October

Mục lục

Tìm hiểu về tài liệu hướng dẫn sử dụng API.....	2
Tìm hiểu Swagger	3
Cài đặt và sử dụng Swagger Inspector Extension	5
Thêm Swagger vào web API project	7
Tổng kết.....	10

Tìm hiểu về tài liệu hướng dẫn sử dụng API

Sau khi đã xây dựng được ứng dụng RestfulAPI, một công việc tiếp theo chúng ta cần làm là cung cấp tài liệu hướng dẫn sử dụng API để bên thứ ba có thể sử dụng. Một công cụ rất mạnh mẽ để tạo một trang quản lý document cho API chúng ta sẽ cùng tìm hiểu về Swagger.

- Tài liệu hướng dẫn sử dụng API là một nội dung kỹ thuật, nó chứa tất cả các thông tin được yêu cầu để cs thể làm việc với API, với thông tin chi tiết về các tài nguyên, phương thức, các request và response, ... được hỗ trợ bởi các hướng dẫn và ví dụ.
- Tài liệu hướng dẫn sử dụng API thường được tạo và bảo trì bằng các trình soạn thảo văn bản thông thường. Các định dạng mô tả API giống như OpenAPI/Swagger Specification sẽ tự động hóa quá trình xử lý tài liệu, giúp các team dễ dàng hơn trong việc tạo và bảo trì chúng.
- Tài liệu hướng dẫn API rất quan trọng:
- Sản phẩm của chúng ta có thể là tốt nhất, nhưng sẽ không có ai sử dụng nó nếu họ không biết nó làm gì và sử dụng như thế nào.
- Cải thiện trải nghiệm người dùng: nếu bạn có tài liệu hướng dẫn tốt, nhiều người sẽ dễ dàng tìm thấy giá trị trong các dịch vụ của bạn và sử dụng chúng.
- Giúp nhiều người biết đến API của bạn: khi có nhiều người chấp nhận và sử dụng các API của chúng ta, họ sẽ giúp chúng ta giới thiệu về các sản phẩm tuyệt vời mà họ đã sử dụng.
- Tiết kiệm thời gian hỗ trợ và chi phí: tài liệu hướng dẫn tốt, cũng giảm lượng thời gian phải bỏ ra để hỗ trợ những người dùng mới, các thành viên mới của team hoặc đối tác. Tài liệu hướng dẫn tồi hoặc không có, nghĩa là sẽ có nhiều người dùng bực bội vì phải phụ thuộc vào team của bạn để hiểu cách làm việc với API. Ngược lại, khi chúng ta cung cấp cho người dùng khả năng thử nghiệm API trước khi triển khai nó, và cung cấp cho họ tài liệu chi tiết để bắt đầu, bạn sẽ tiết kiệm cho team của mình vô số thời gian trả lời email và các cuộc gọi hỗ trợ.
- Dễ bảo trì hơn: giúp team của chúng ta biết các chi tiết của tài nguyên, phương thức, các request và response, giúp cho việc bảo trì và cập nhật nhanh hơn.

OpenAPI Specification là một định dạng mô tả API dành cho REST APIs. Một file OpenAPI cho phép bạn mô tả toàn bộ API bao gồm:

- Cho phép những endpoints (/users) và cách thức hoạt động của mỗi endpoint (GET, POST, PUT, DELETE).
- Các tham số đầu vào & đầu ra của từng API.
- Phương thức xác thực.
- Thông tin liên hệ, chứng chỉ (HTTP/ HTTPS), điều khoản sử dụng và những thông tin khác.

API specifications có thể được viết bằng **YAML** hoặc **JSON**. Định dạng này dễ đọc, dễ hiểu cho cả người dùng lẫn ngôn ngữ máy tính.

Tìm hiểu Swagger

Swagger là mã nguồn mở cung cấp các công cụ cho các lập trình viên và các bên khác (kể cả những bên phân tích product, khách hàng), một bộ công cụ mã nguồn mở để xây dựng OpenAPI specifications giúp chúng ta có thể thiết kế, xây dựng tài liệu và sử dụng REST APIs.

Swagger cung cấp một số công cụ cho các nhà phát triển:

- Swagger-Editor: dùng để thiết kế lên các APIs hoàn toàn mới hoặc edit lại các APIs có sẵn thông qua một file config.
- Swagger-UI: tự sinh ra tài liệu và giao diện cho API từ một file config.

Để viết document với Swagger có hai hướng tiếp cận chính như sau:

- Top-down approach: thiết kế các API trước khi code
- Bottom-up approach: từ các API có sẵn thiết kế file config để mô tả chúng.

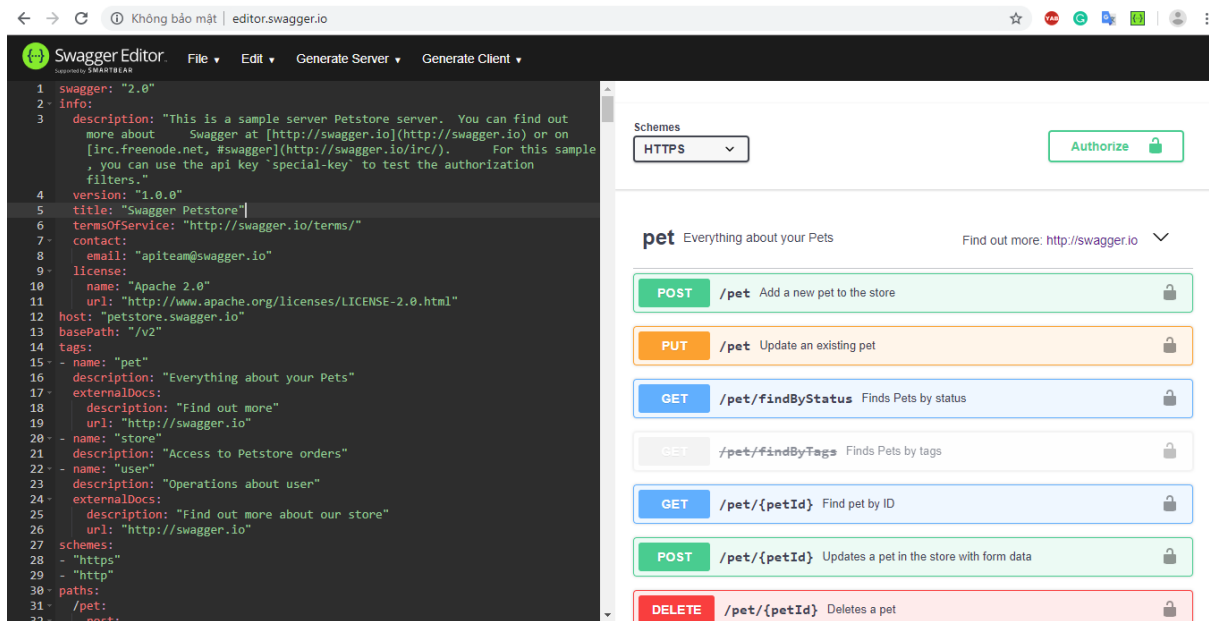
Swagger UI

Swagger UI là một mã nguồn mở giúp sinh ra giao diện cho tài liệu từ file config dưới chuẩn OpenAPI. Giao diện được hiện ra rõ ràng và tường minh. Dễ dàng đọc hiểu cho cả lập trình viên lẫn người dùng.

Cấu trúc cơ bản

Swagger có thể được viết bằng JSON hoặc YAML. Định dạng này dễ đọc, dễ hiểu cho cả người dùng lẫn ngôn ngữ máy tính.

Ví dụ:



Nhìn vào giao diện, ta có thể thấy file config Swagger này có những thành phần cơ bản sau:

1. Metadata

Mỗi OpenAPI specifications sẽ bắt đầu với từ khóa `openapi` để khai báo phiên bản. Phiên bản này sẽ định nghĩa toàn bộ cấu trúc của API. Phần `info` sẽ chứa những thông tin của API như `title`, `description` (tùy chọn), `version`

- `title` là tên API của bạn
- `description` là thông tin mở rộng về API của bạn.
- `info` cũng hỗ trợ những từ khóa về thông tin liên lạc, chứng chỉ, điều khoản sử dụng và những thông tin khác.
- `Version` phiên bản API.

2. Servers

Đây là phần sẽ chỉ định đường dẫn của server để ta có thể test được API. Bạn có thể định nghĩa một hoặc nhiều server. Tất cả đường dẫn API sẽ là đường dẫn tương đối của URL mà bạn định nghĩa.

3. Paths

Đây là phần trọng tâm của API. Ở phần này bạn sẽ định nghĩa những paths trong API cũng như phương thức, tham số trong API.

- Phần này sẽ bắt đầu bằng từ khóa `paths`
- Path trong API (`/user/{userId}`)
- Phương thức của API (GET, POST, DELETE, PUT...)
- Summary là phần mô tả tóm tắt của API

- Parameters: sẽ là những tham số truyền vào API. Bạn có thể set tham số required hay không, mô tả nó hoặc validate. Đặc biệt trong phần này chúng ta có thể chỉ định 1 schema(Model) để có thể định nghĩa cho phần tham số thông qua schema và \$ref.
- Reponse là phần trả về của server. Chúng ta có thể định nghĩa những HTTP code: 200, 404, 500,.. với những mô tả cho từng trường hợp.

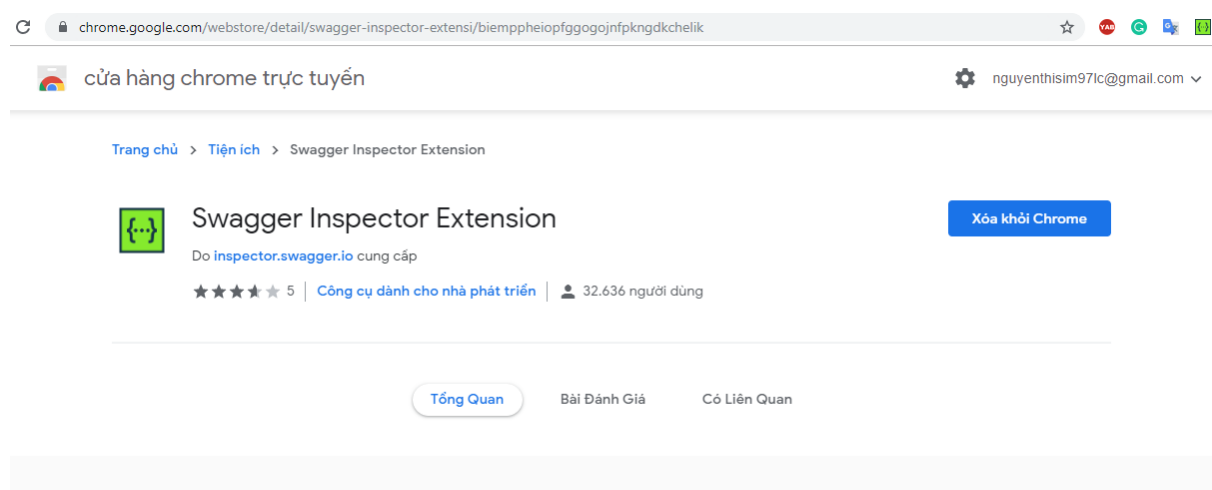
Những chú ý khi các parameters có khá nhiều khai báo sau từ khóa **in**:

- in:"body": tạo cho người dùng một input-text area mà ở đó người ta có thể nhập data body request vào(sử dụng cho methods PATH/PUT).
 - in:"path": tạo cho người dùng một input nhập vào giá trị khai báo trong routes, thường là id.
 - in:"formData": tạo cho người dùng những input đã định trước mà người ta sẽ nhập data request theo từng field đã định sẵn vào (sử dụng cho methods PATH/PUT).
 - in:"query": tạo cho người dùng một input nhập vào giá trị theo các filed định sẵn để gửi những query request(sử dụng trong methods GET).
 - in:"header": khai báo những giá trị trong header của request mà bạn cần quyền lên.
4. SecurityDefinitions : Authentication mà APIs sử dụng để cung cấp tài nguyên.
 5. Definitions: Định nghĩa các model sử dụng bởi APIs, bao gồm: tên của Model, kiểu(type) định dạng(object) và thuộc tính(properties) của model.

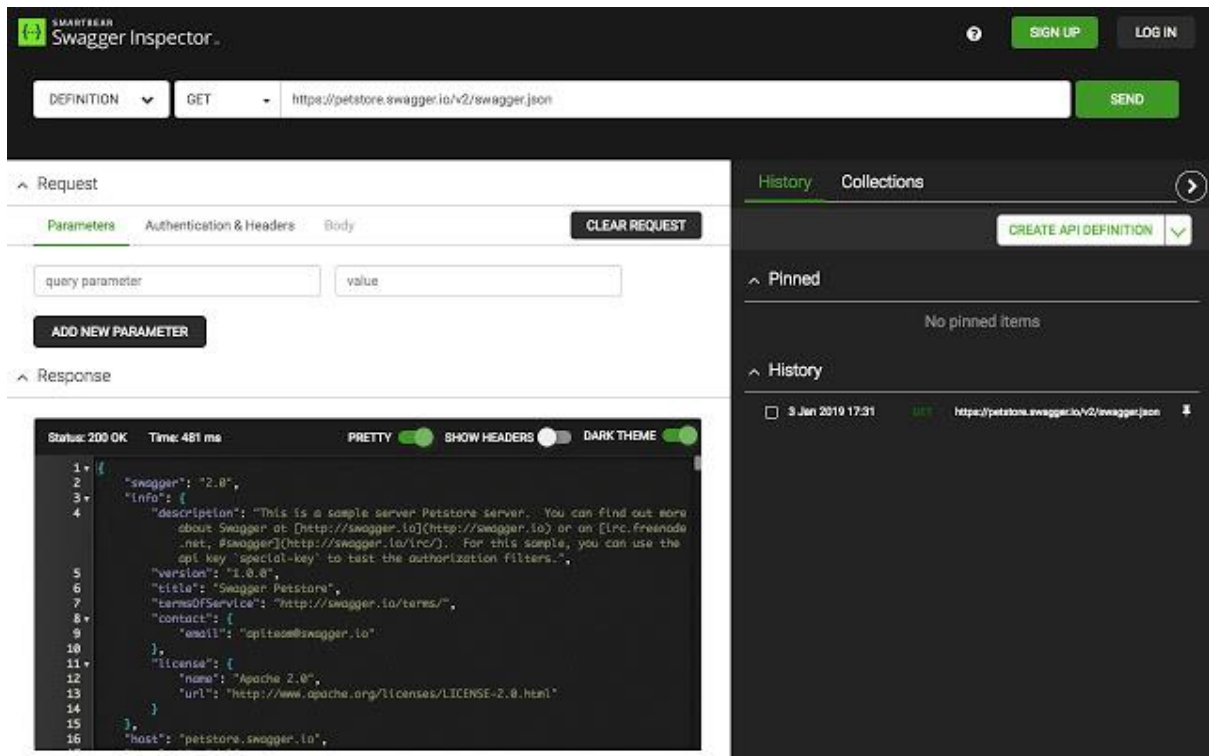
Cài đặt và sử dụng Swagger Inspector Extension

Vào chrome để thêm extension Swagger:

<https://chrome.google.com/webstore/detail/swagger-inspector-extensi/biemppeiofpggogojnfpkngdkchelik>

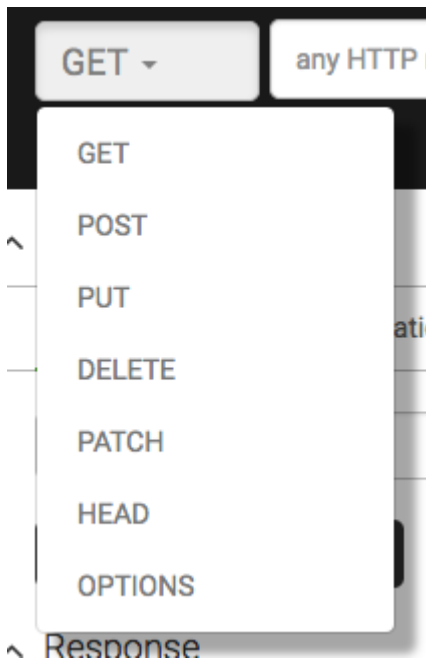


Sau khi thêm xong, ta mở giao diện và được kết quả:



Cách tạo một API call

Trên trang, có một dropdown các method chứa common HTTP như: GET,PUT and POST. Chọn method mà bạn muốn sử dụng.



Bên phải của method option là input URL mà bạn muốn gọi.

Bên dưới thẻ input URL có 3 tab Parameters, Authentication&Headers và Body.

- Parameters là các parameter query, như khi cần tìm kiếm ta có parameter và giá trị cần tìm kiếm. Chúng ta có thể thêm một query parameter trực tiếp vào URL hoặc thêm vào trong ô của tab.
- Authentication&Headers là nơi bạn muốn thêm vào headers, giống như content-type của request hay thêm authentication.
- Body là phần bạn muốn thêm body cho request của bạn.

Sử dụng Definition: Tính năng này cho phép bạn mock và tạo API call đến một API based trên API definition. Tính năng này support cho OpenAPI 2.0, OpenAPI 3.0 và file WSDL.

History & Right Panel: Lưu trữ các request đã được gọi, khoảng 1000 request và tự động xóa những request cũ nếu vượt quá 1000.

Thêm Swagger vào web API project

Để add Swagger vào một ASP.NET Web Api, chúng ta sẽ cài đặt một open source project là Swashbuckle thông qua nuget.

```
PM> Install-Package Swashbuckle -Version 5.2.1
```

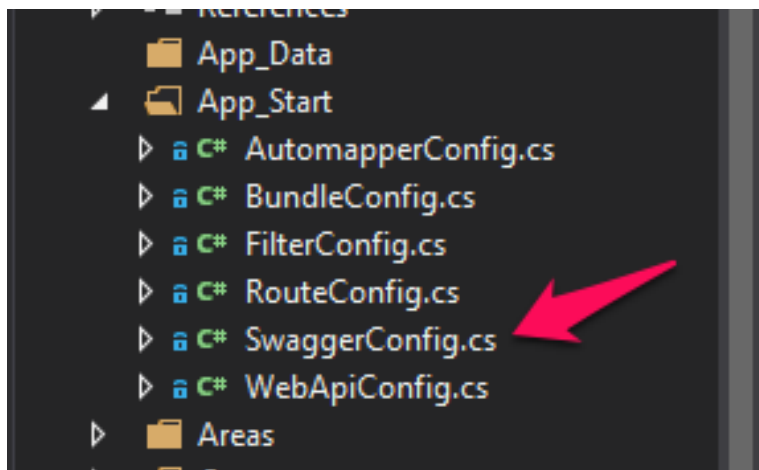
Trong Visual Studio:

- Chuột phải vào dự án của bạn trong Solution Explorer > Manage NuGet Packages
- Gõ Swashbuckle trong ô tìm kiếm
- Check "Include prerelease"
- Cài đặt Package source tới nuget.org
- Chọn Swashbuckle và install

Swashbuckle có 2 thành phần chính:

- Swashbuckle.SwaggerGen: cung cấp các chức năng để tạo ra tài liệu JSON Swagger mô tả các đối tượng, phương thức, kiểu dữ liệu trả về, tham số, ...
- Swashbuckle.SwaggerUI: tích hợp phiên bản Swagger UI, cái mà sử dụng chỉnh sửa mô tả, chức năng của Web API.

Sau khi cài đặt xong, vào App_start trong Solution Explorer. File cấu hình Swagger là SwaggerConfig.cs.



Bởi vì mặc định Swagger không hiển thị XML Comments, vậy để project chạy tất cả các comment XML được lưu trong file. Đầu tiên, cần đến properties của project và chọn Build section check XML documention file, và có được path. Sau đó lưu nó bằng Ctrl+S.

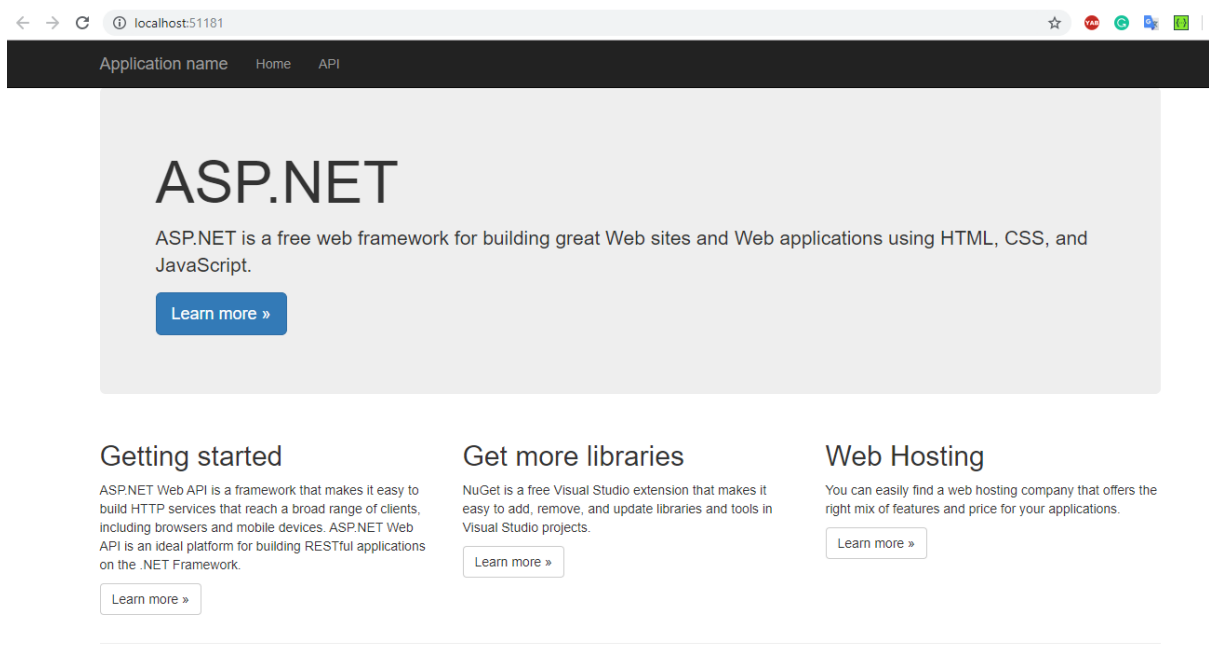
Giờ đi đến SwaggerConfig file và uncomment

```
c.IncludeXmlComments(GetXmlCommentsPath());
```

Và viết phương thức sử dụng intellisense hoặc bạn viết một phương thức ở cuối file config.

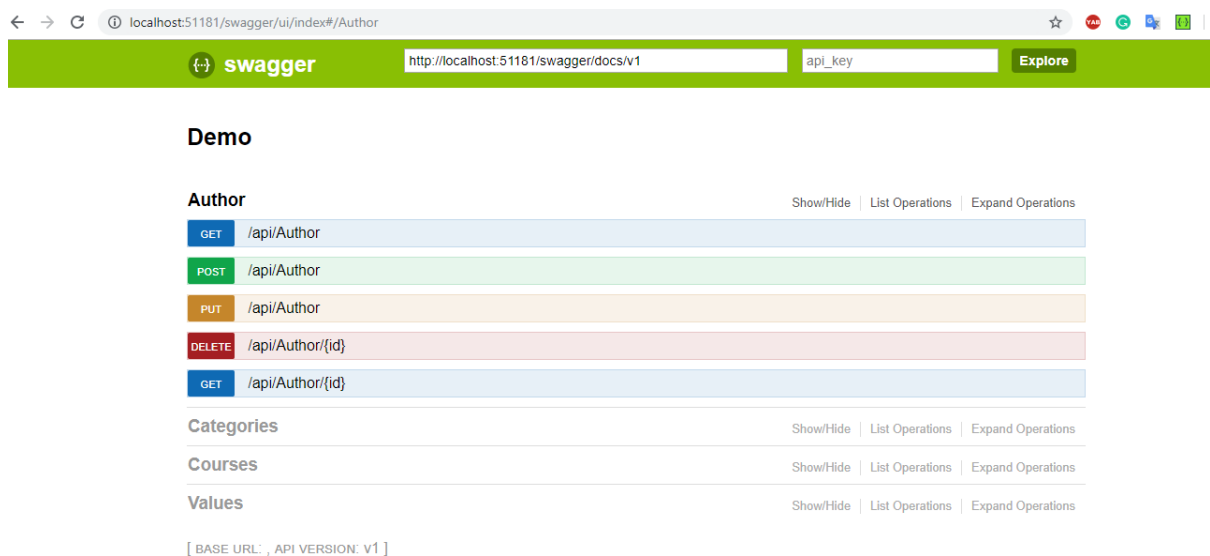
```
private static string GetXmlCommentsPath()
{
    return System.AppDomain.CurrentDomain.BaseDirectory +
@"\bin\DemoWebAPIWithSwagger.XML";
}
```

Giờ chúng ta chạy project.

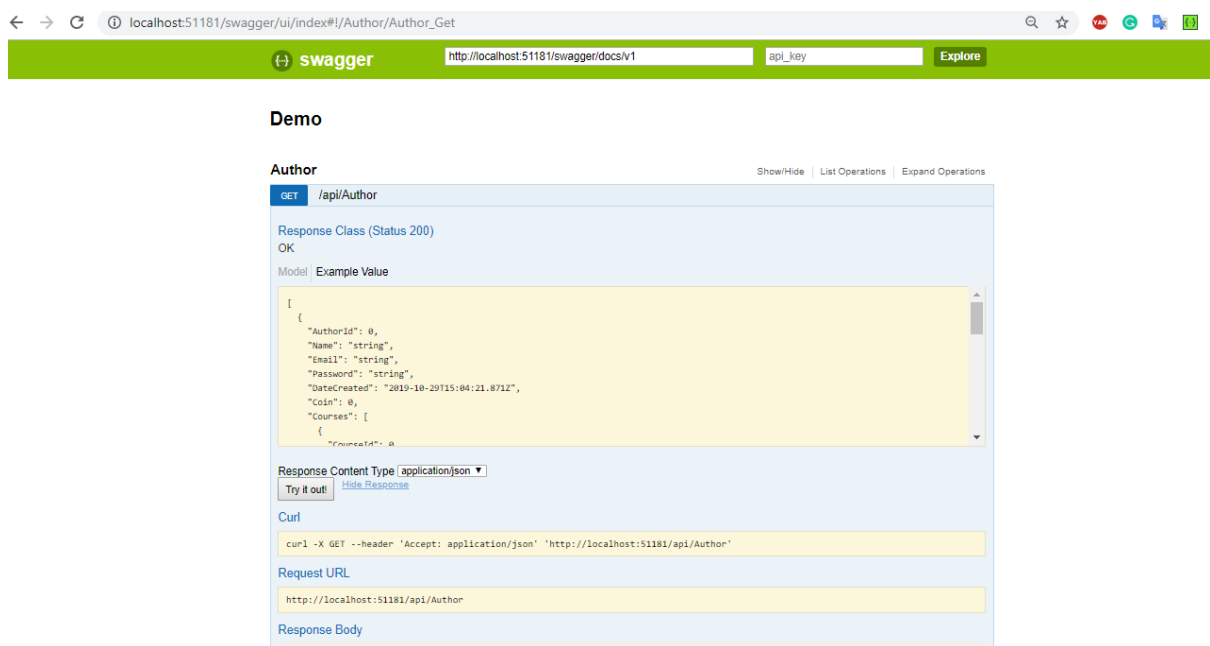


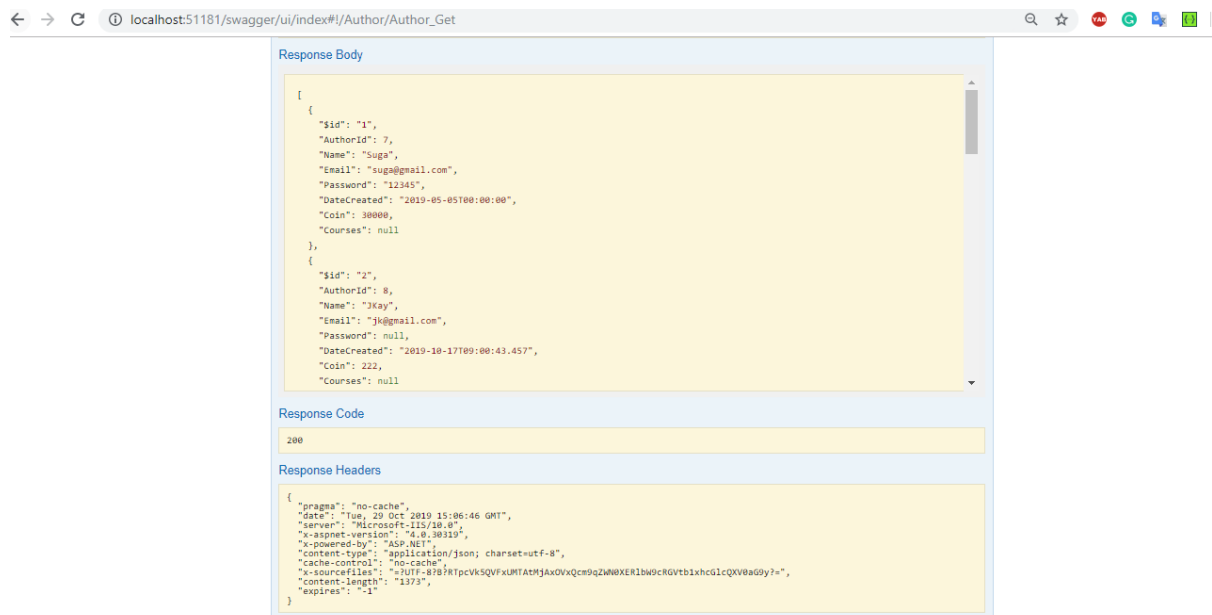
Add Swagger in url => <http://localhost:51181/swagger>

Giao diện Swagger UI của chúng ta sẽ ở đây,



Kiểm tra,





Trên là phần implement và customizing Swagger UI in ASP.NET Web APIs sử dụng Swashbuckle.

Sau khi có được file config này, ta có thể sử dụng trong Swagger Editor (<https://editor.swagger.io>) để Generate ra Server hoặc Client cho quá trình sử dụng.

Tổng kết

Trong bài nghiên cứu này, chúng ta đã nghiên cứu Swagger và OpenAPI là những công cụ mạnh mẽ để giải quyết vấn đề gắn kết các bên, theo dõi thông tin, tiến độ và kiểm tra API một cách dễ dàng. Vậy là chúng ta đã hoàn thành xong việc viết một trang document cho RESTful API bằng việc sử dụng Swagger, đồng thời cài đặt extension Swagger để kiểm tra data.