1. **Lý thuyết**
   1. **Kiến thức cần nhớ**

**Khái niệm Web API**

* Web API là công nghệ mới nhất của Microsoft để xây dựng các dịch vụ thành phần phân tán.
* Web API hỗ trợ mô hình MVC: Routing, Controller, ActionResult, Filter, Model Binder,…
* Web API hỗ trợ RESTfull đầy đủ phương thức: GET, POST, PUT, DELETE dữ liệu.
* REST (Representational State Tranfer) là kiến trúc được sử dụng trong việc giao tiếp giữa các máy khách chủ, trong việc quản lý các tài nguyên trên internet. REST được sử dụng rất nhiều trong việc phát triển các ứng dụng Web Services sử dụng giao thức HTTP trong giao tiếp thông qua mạng Internet. Các ứng dụng sử dụng kiến trúc REST này được gọi là ứng dụng phát triển theo RESTfull.

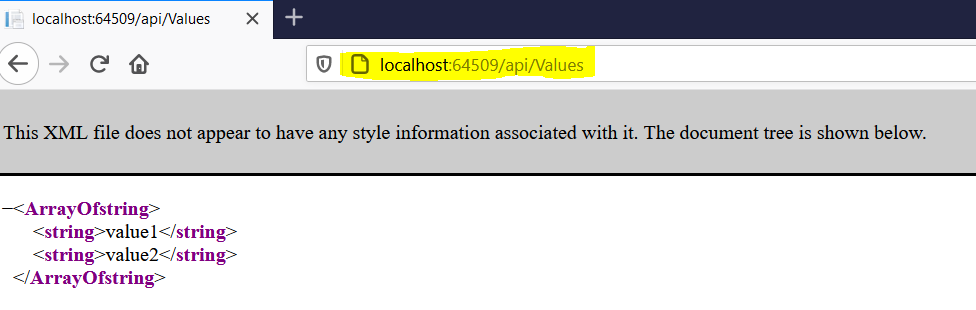
**Ưu điểm của Web API**

* Có độ hoàn thiện cao, có thể host trong ứng dụng hoặc IIS, là kiến trúc lý tưởng cho các thiết bị có băng thông giới hạn như smart phone, tablet. Web API Services sử dụng được ở hầu hết client như ứng dụng desktop, ứng dụng web, ứng dụng mobile.
* Web API trả về phía client dữ liệu có định dạng JSON, XML,…
* Xây dựng các HTTP Services đơn giản và nhanh chóng.
* Web API mã nguồn mở và có thể được sử dụng bởi bất kỳ một Client nào hỗ trợ JSON, XML.
* Hỗ trợ đầy đủ các thành phần HTTP: URI, request/ response header, caching, versioning, content format.

**Đặc điểm của ASP.NET Web API**

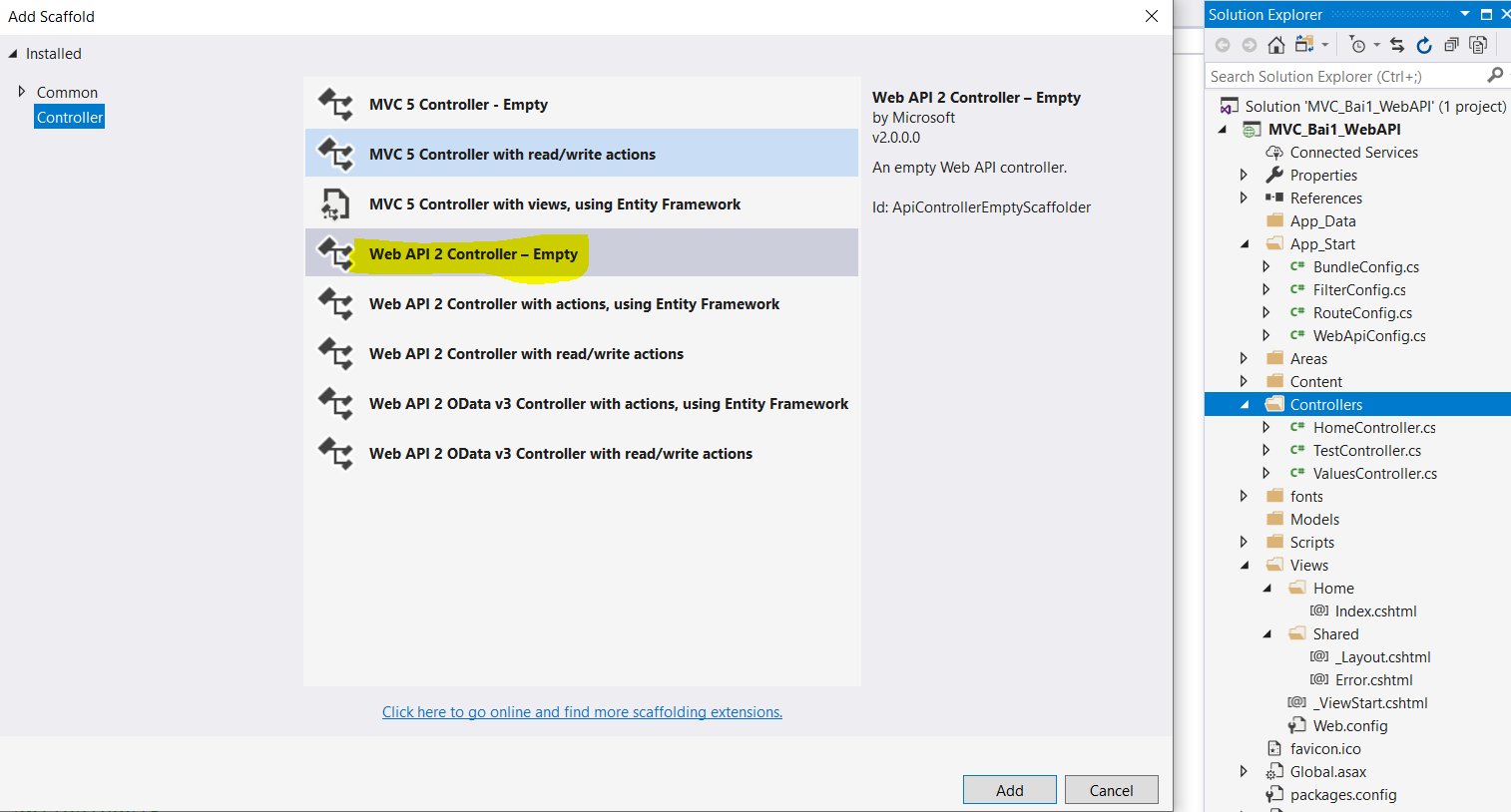
* ASP.NET Web API là nền tảng lý tưởng để xây dựng các dịch vụ RESTfull.
* ASP.NET Web API được xây dựng trên nền tảng ASP.net hỗ trợ request/ response
* ASP.NET Web API ánh xạ các động tử HTTP thành tên phương thức
* ASP.NET Web API hỗ trợ các định dạng khác nhau của dữ liệu phản hồi.
* ASP.NET Web API có thể được lưu trữ trong IIS, tự lưu trữ hoặc máy chủ web khác hỗ trợ . NET 4.0 +
* Khung ASP.NET Web API bao gồm HTTPClient mới để giao tiếp với máy chủ API Web. HTTPClient có thể được sử dụng trong phía máy chủ ASP.MVC, ứng dụng Windows Form, ứng dụng console hoặc các ứng dụng khác.
  1. **Giới thiệu bài tập mẫu**

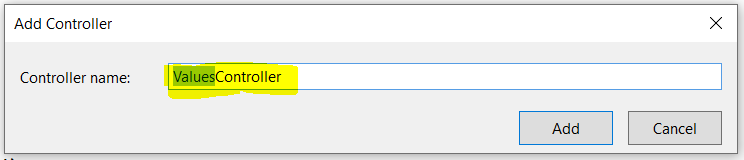
**Bài 1:** Tạo một Web Api Project MVC đặt tên là **MVC\_Bai1\_WebAPI** hiển thị danh sách 2 chuỗi “**value1**” và “**value2**” như sau



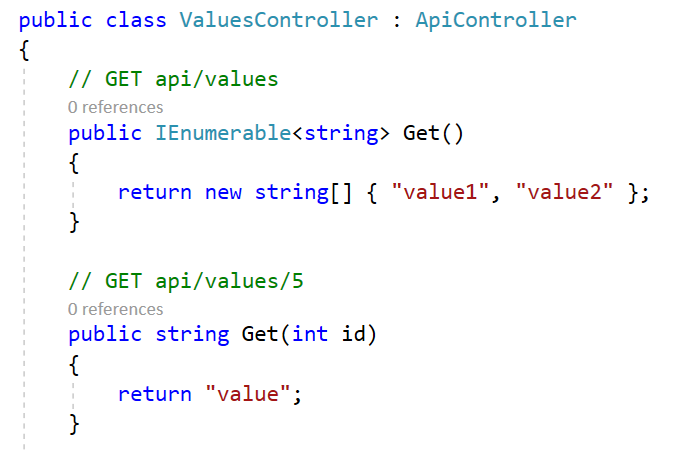
**Hướng dẫn:**

**B1: Rclick vào Folder Controller 🡪 Add Controller**

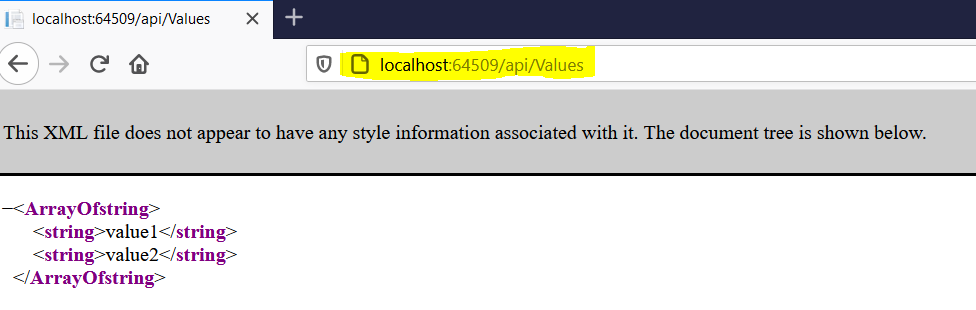


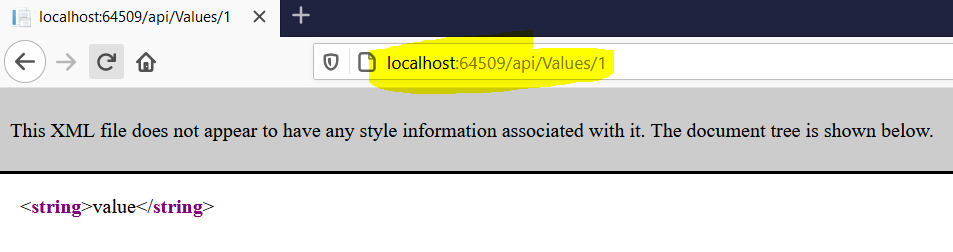


**B2: Xây dựng phương thức Get**

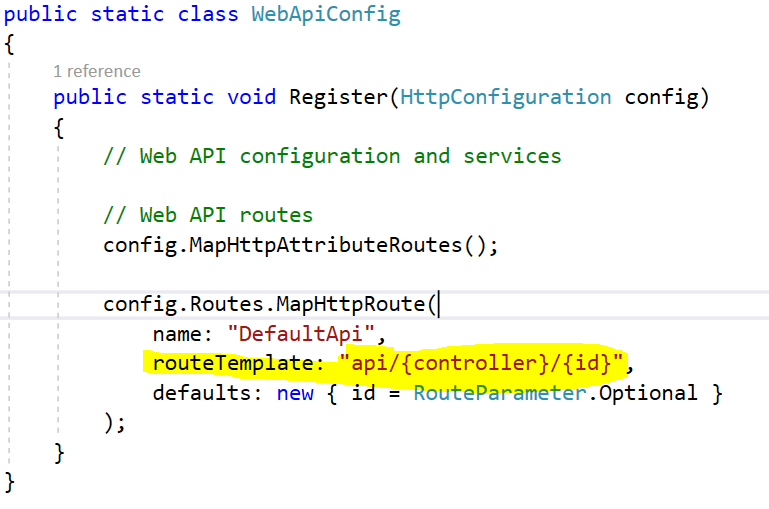


**B3: Gọi thực thi Web API**

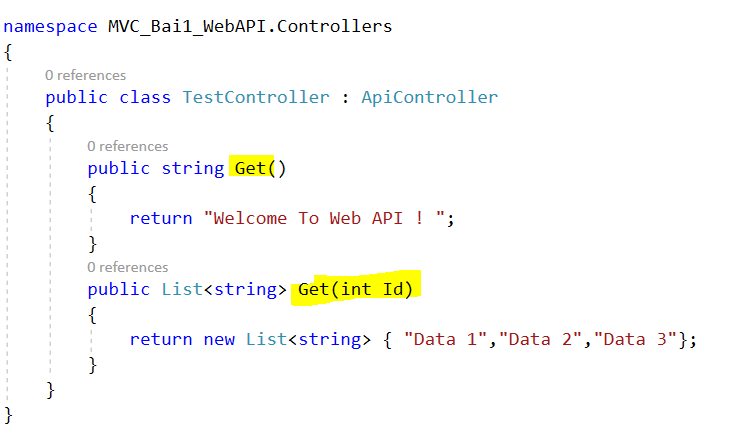




File **WebApiConfig.cs** được sinh ra trong thư mục App\_Start



**Bài 2:** Tạo một Web API 2 Controller Empty mới và đặt tên là **TestController.cs** trong đó xây dựng 2 Method như sau:

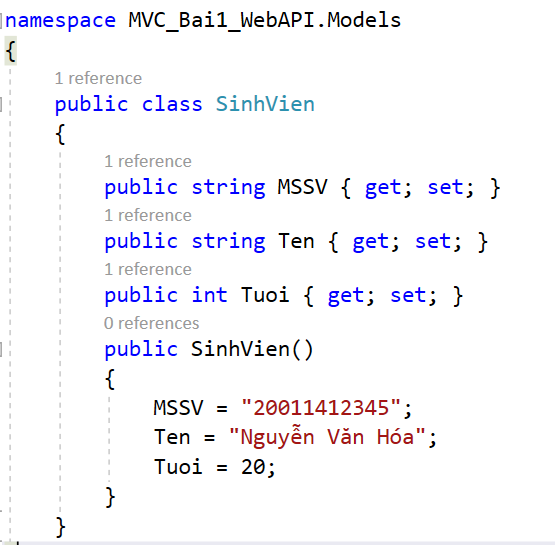


Build và gọi thực thi các phương thức vừa định nghĩa trong APIController **Test,**  nhận xét về dữ liệu trả về trên trang.

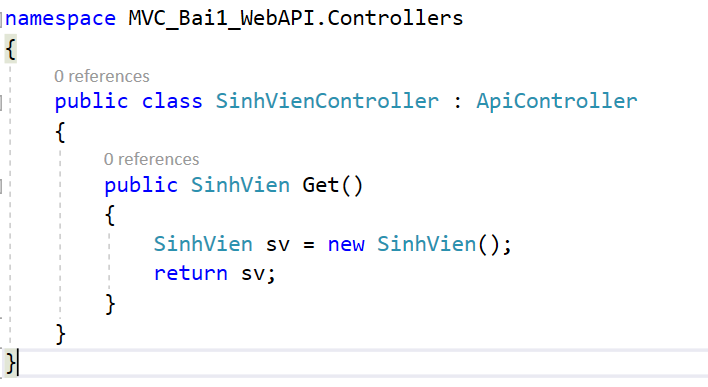
**Bài 3: Tạo API lấy dữ liệu từ một class**

**Hướng dẫn:**

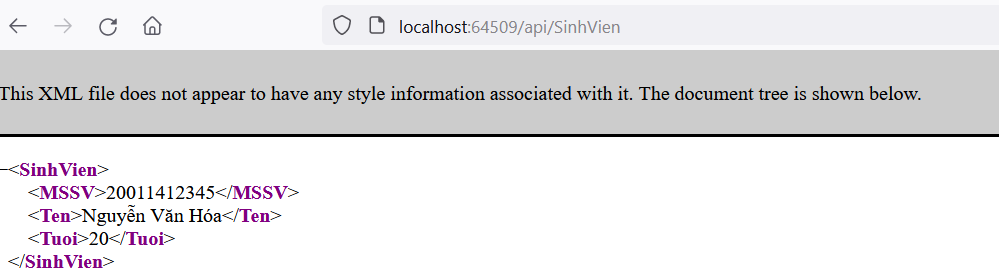
**B1: Xây dựng lớp**



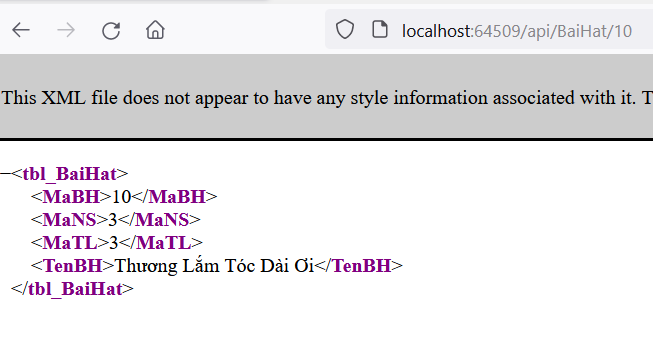
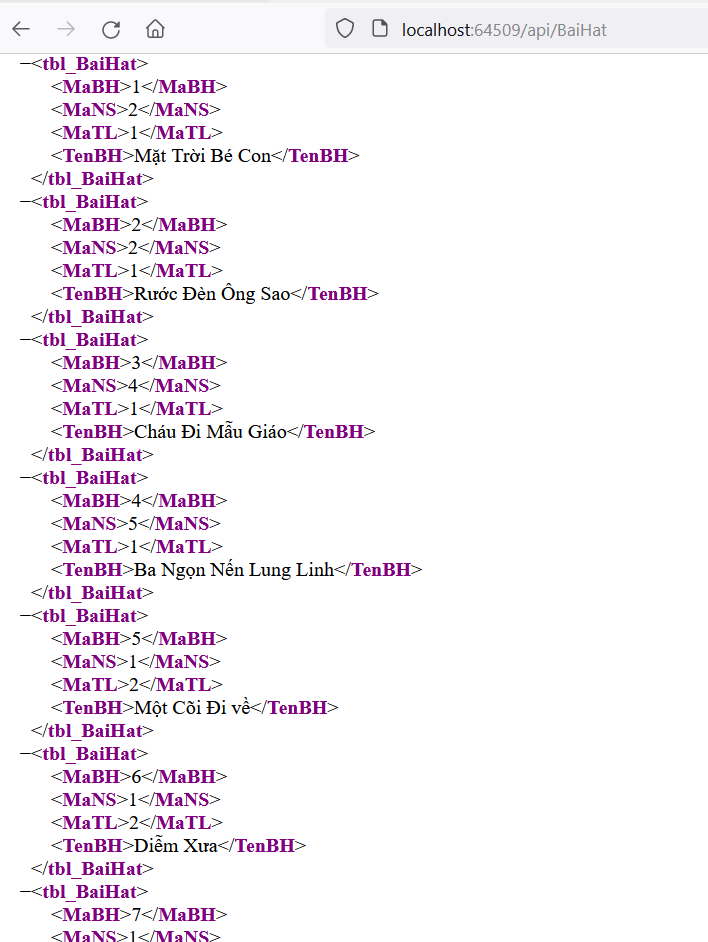
**B2: Tạo Controller kế thừa từ lớp API**



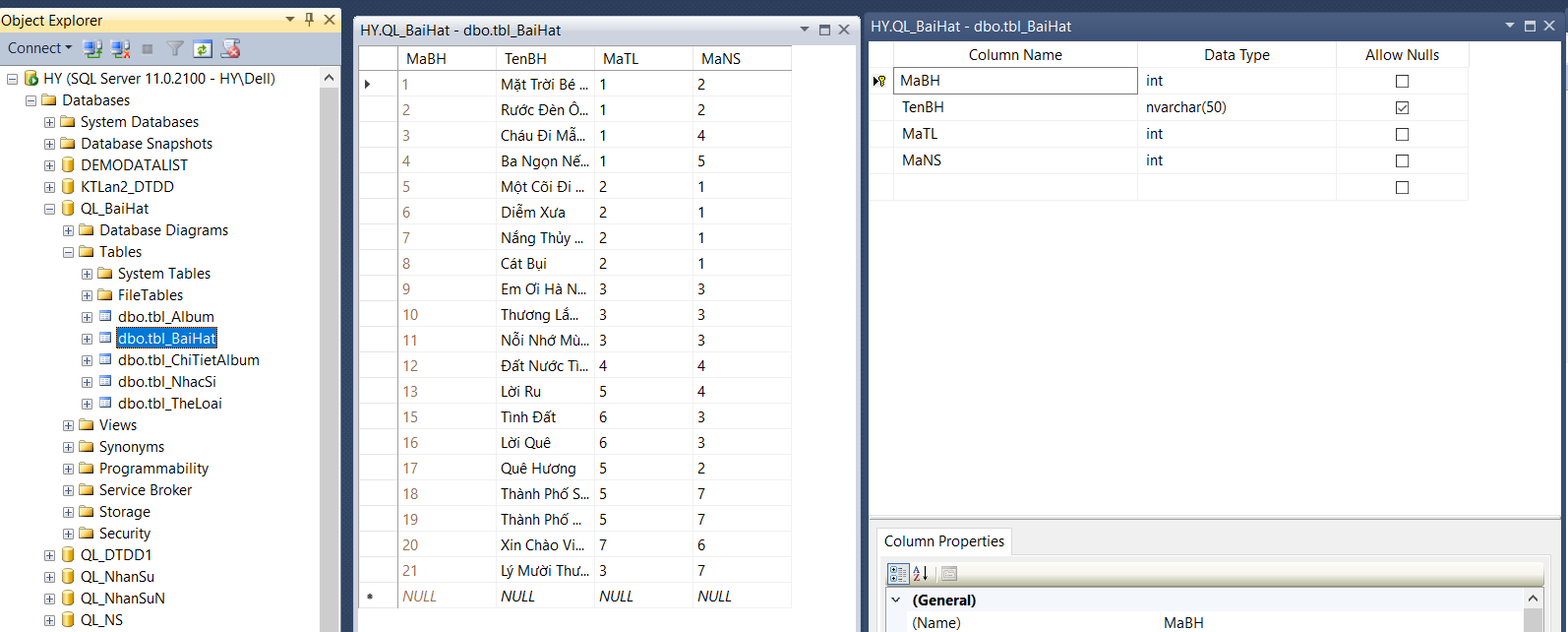
**B3: Thực thi gọi API SinhVien**



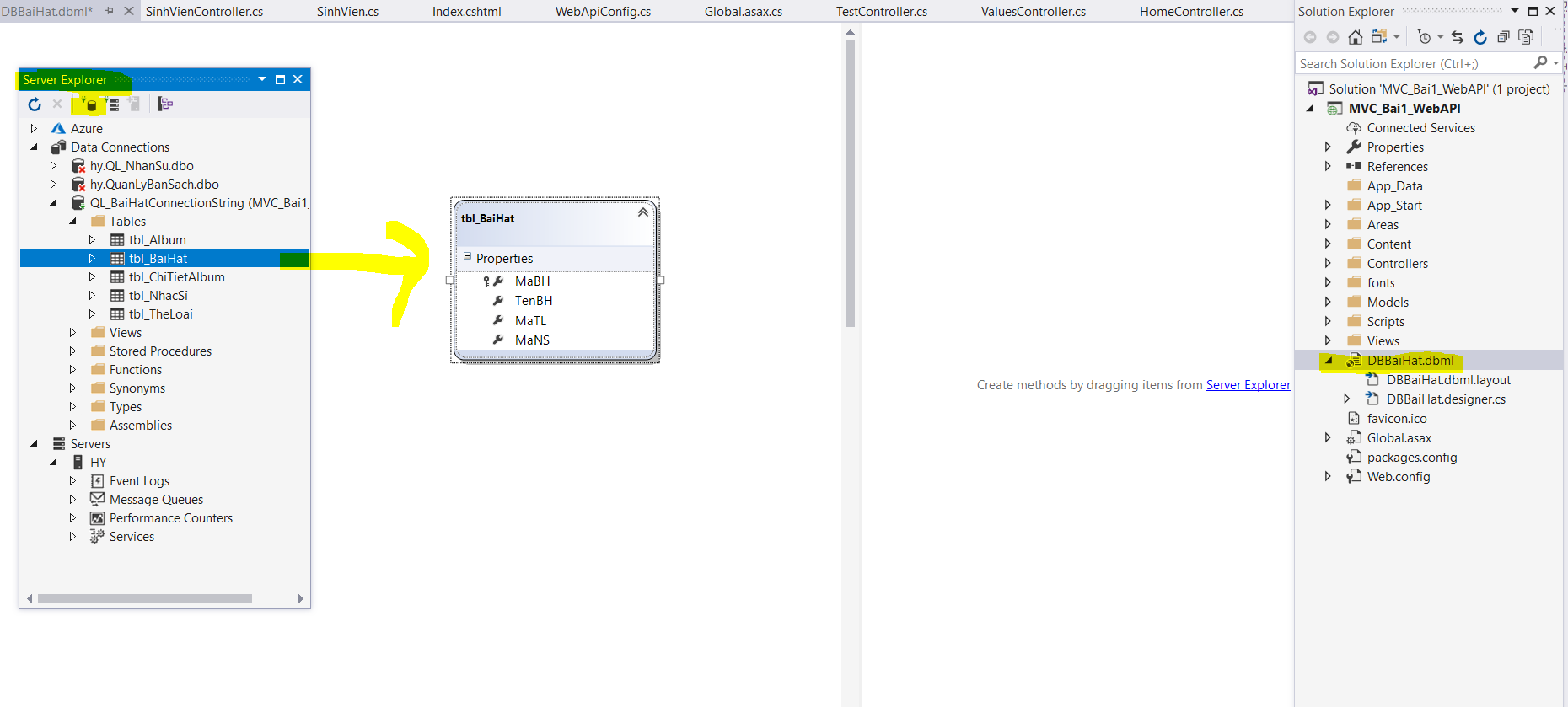
**Bài 4: Tạo 1 Web API lấy dữ liệu lưu trữ từ bảng BaiHat trong SQL Server 2012**



**B1: Cho CSDL**



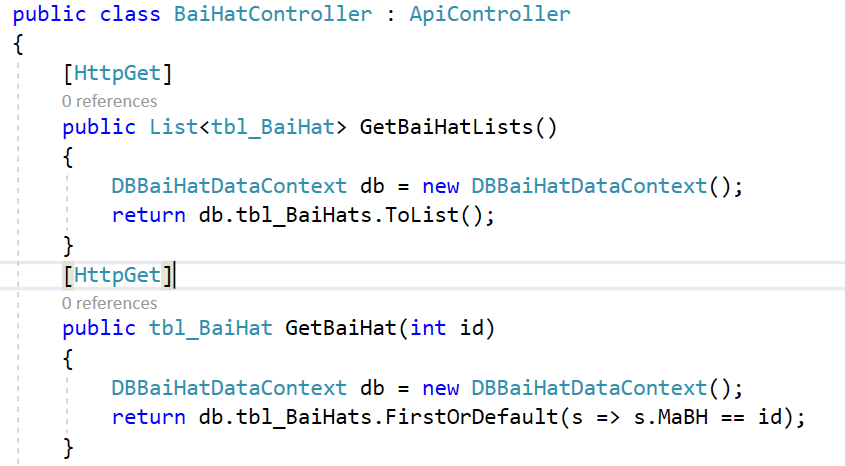
**B2: Tạo một kết nối đến CSDL QL\_BaiHat bằng LINQ to SQL Class**



**B3: Tạo Web API Controller BaiHat**

Viết các chức năng trong BaiHatController

**Phương thức truy vấn thông tin :** [HttpGet]



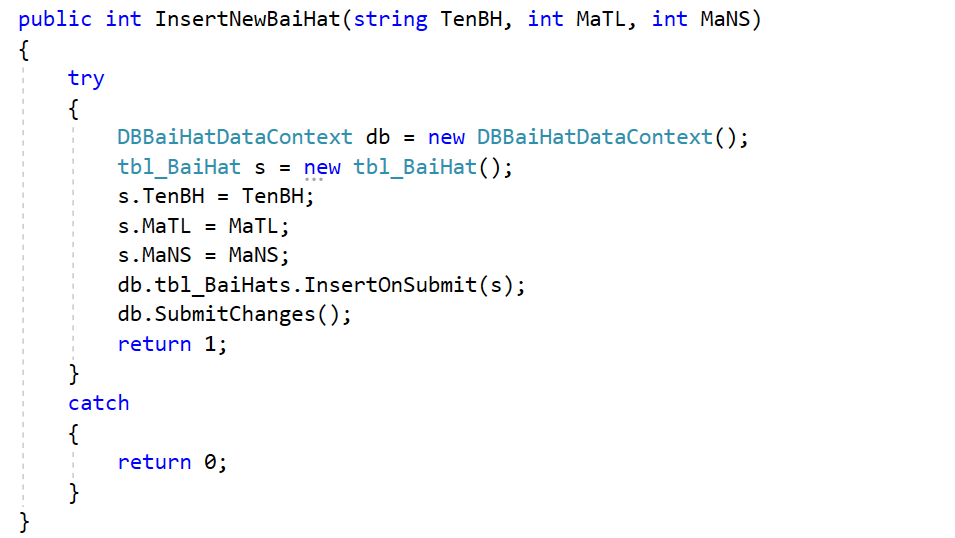
Post, Put và Delete về sau sẽ phải dùng 1 công cụ hỗ trợ để gửi request mới có thể xác định được dữ liệu thu được là gì, vì request gửi đi sẽ ở dạng gửi ngầm không thể nhìn thấy.

<https://reqbin.com/>

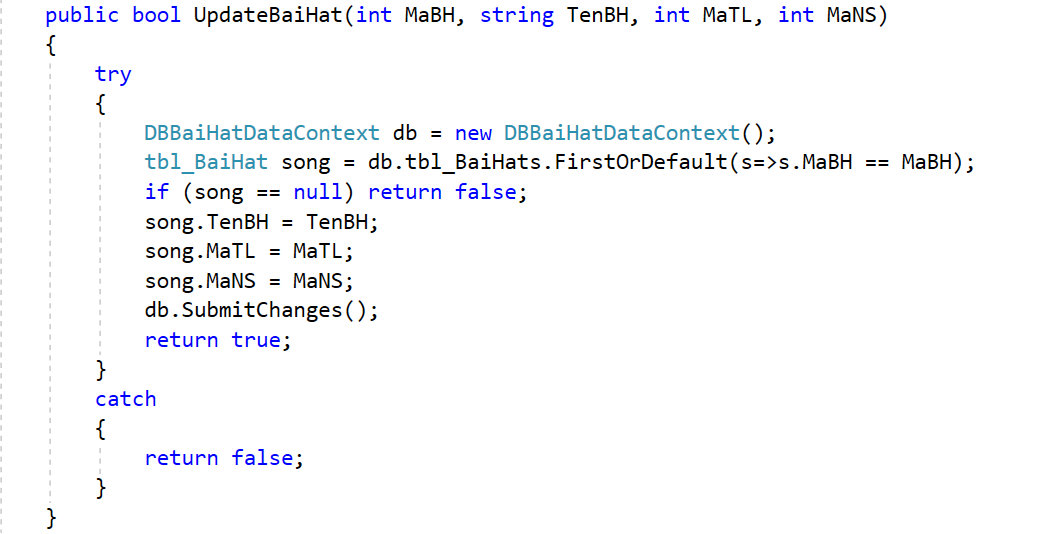
<https://www.postman.com/downloads/>

fiddler

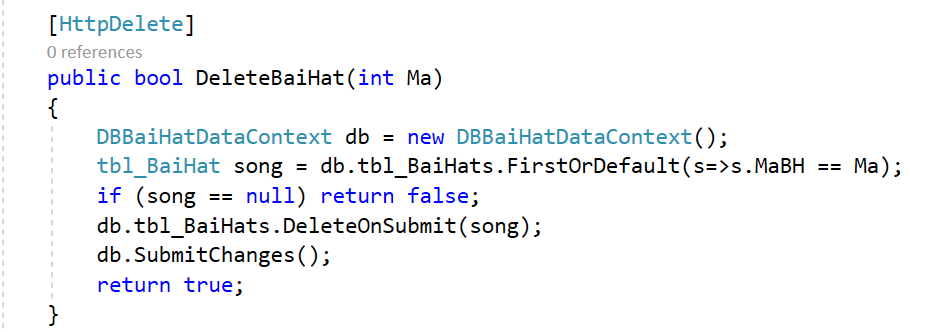
**Phương thức thêm mới thông tin:** [HttpPost]



**Phương thức thay đổi thông tin:** [HttpPut]

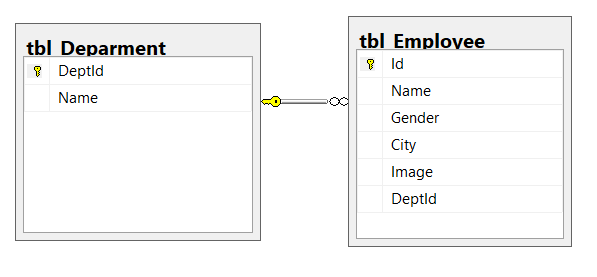


**Phương thức xóa thông tin:** [HttpDelete]

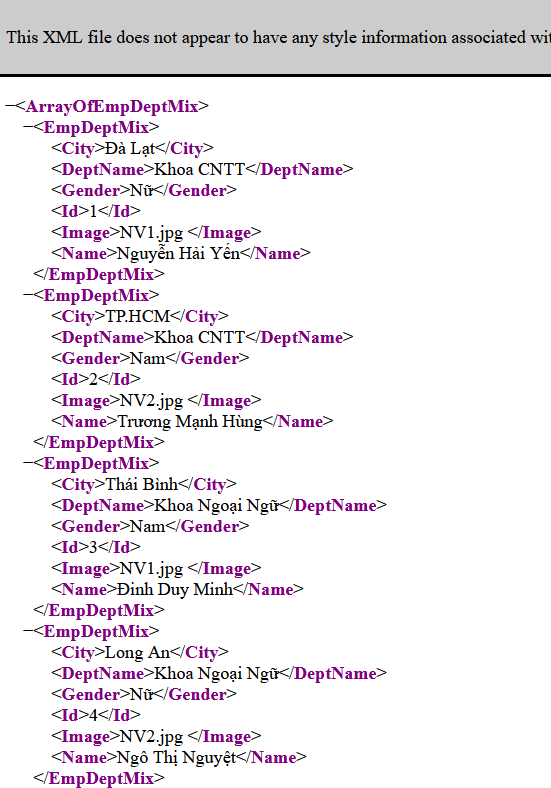
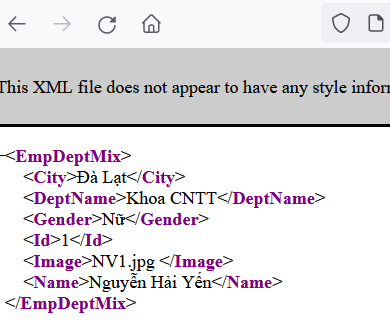


1. **Bài tập tại lớp**

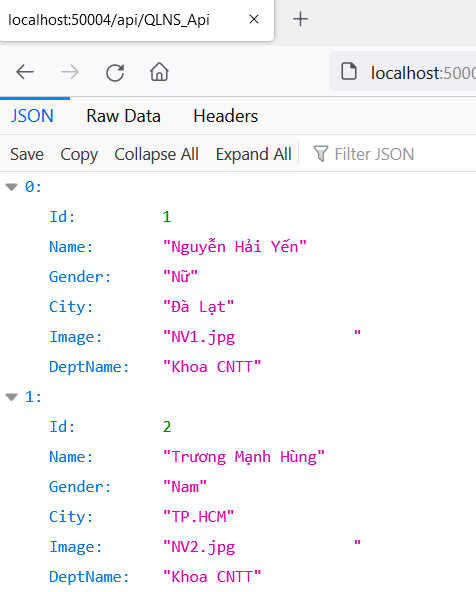
**Bài 5:** Cho CSDL :



Xây dựng Web API có nội dung như sau:

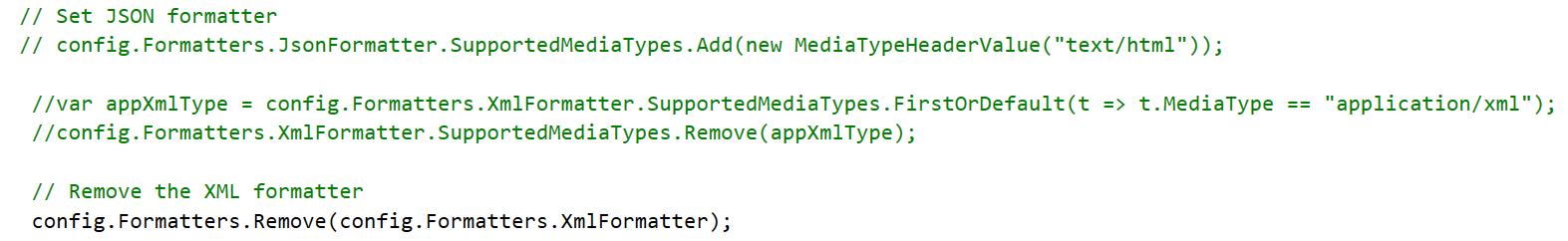
 

**Bài 6:** Hiển thị dữ liệu dạng XML và dạng JSON cho bài 5



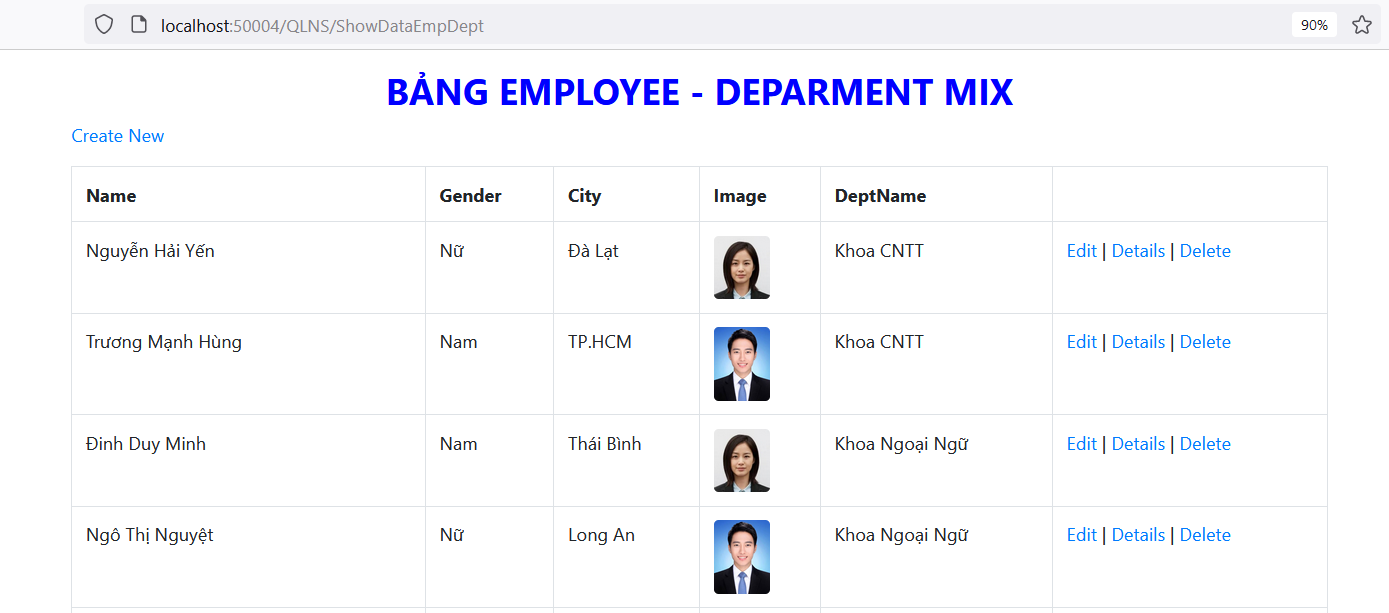
**Hướng dẫn:**

Vào App\_Start 🡪 WebApiConfig.cs



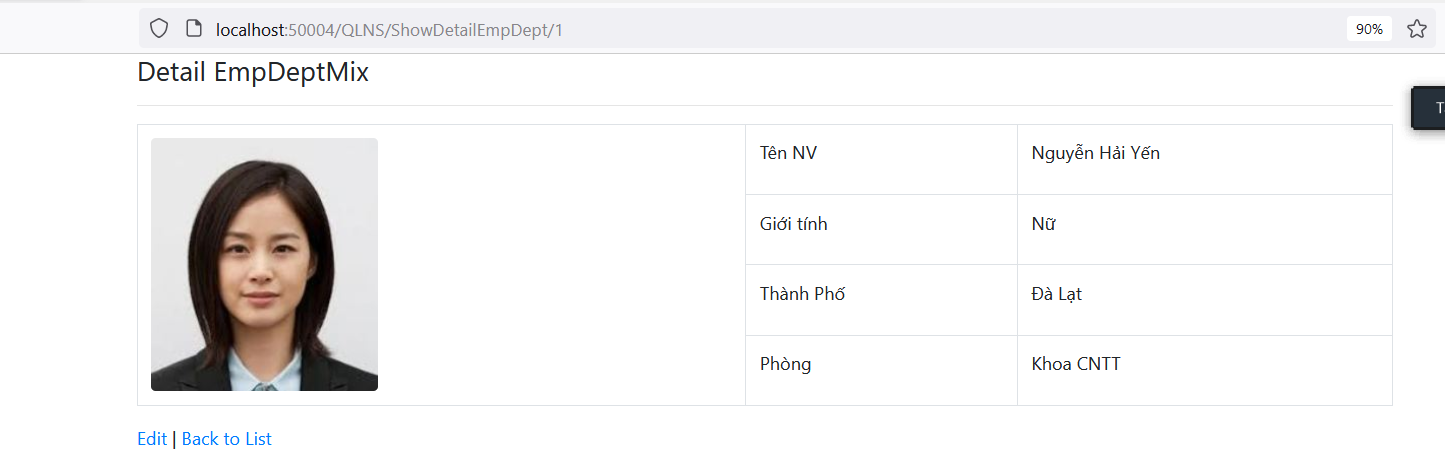
**Bài 7:** Hiển thị Web API lên trang

Cho Web API như bài 5, yêu cầu đọc và hiển thị lên trang như sau



**Câu 8:** Hiển thị Web API lên trang

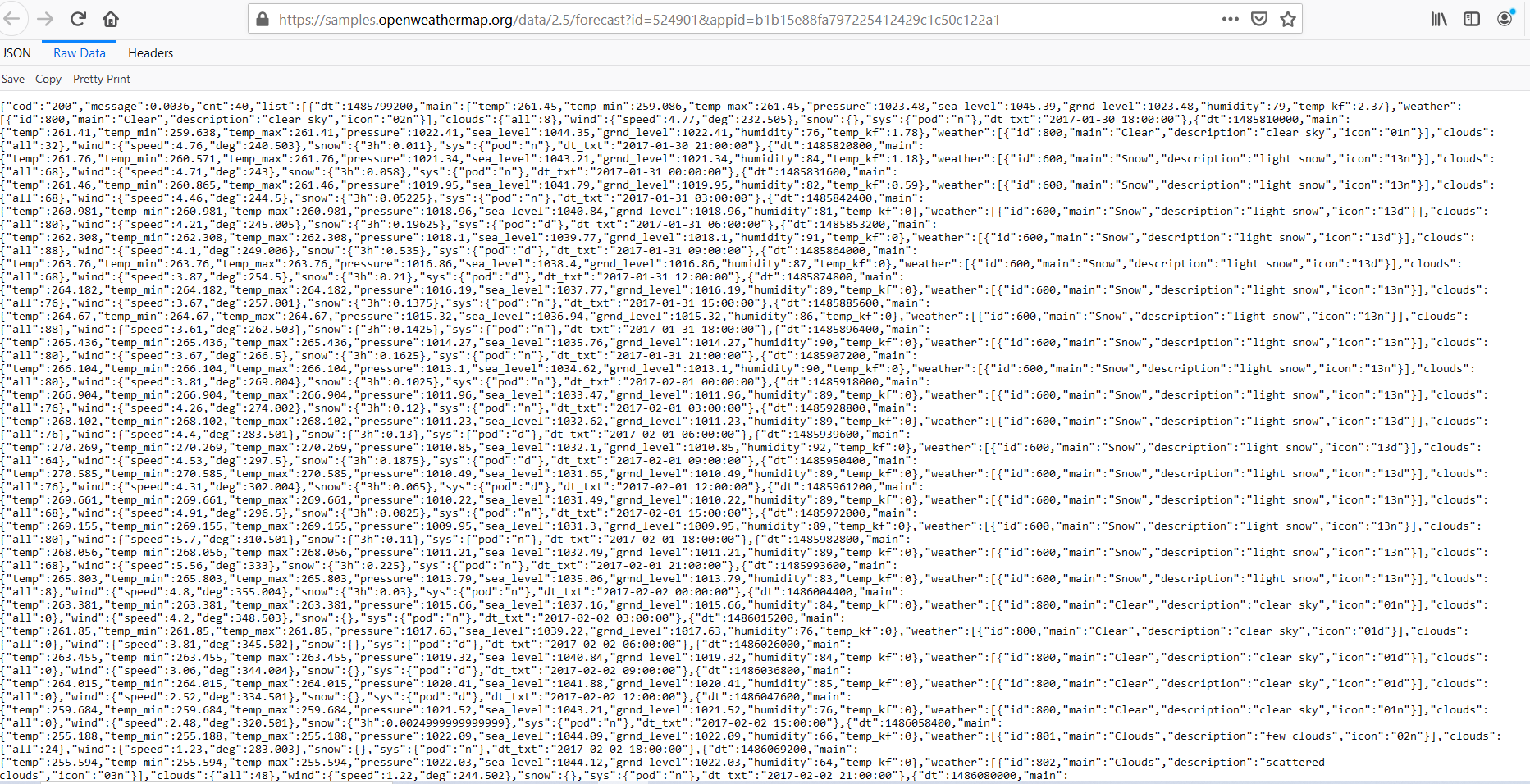
Cho Web API như bài , yêu cầu đọc và hiển thị chi tiết của một nhân viên có mã id = x ra màn hình trình duyệt



**Câu 9:**

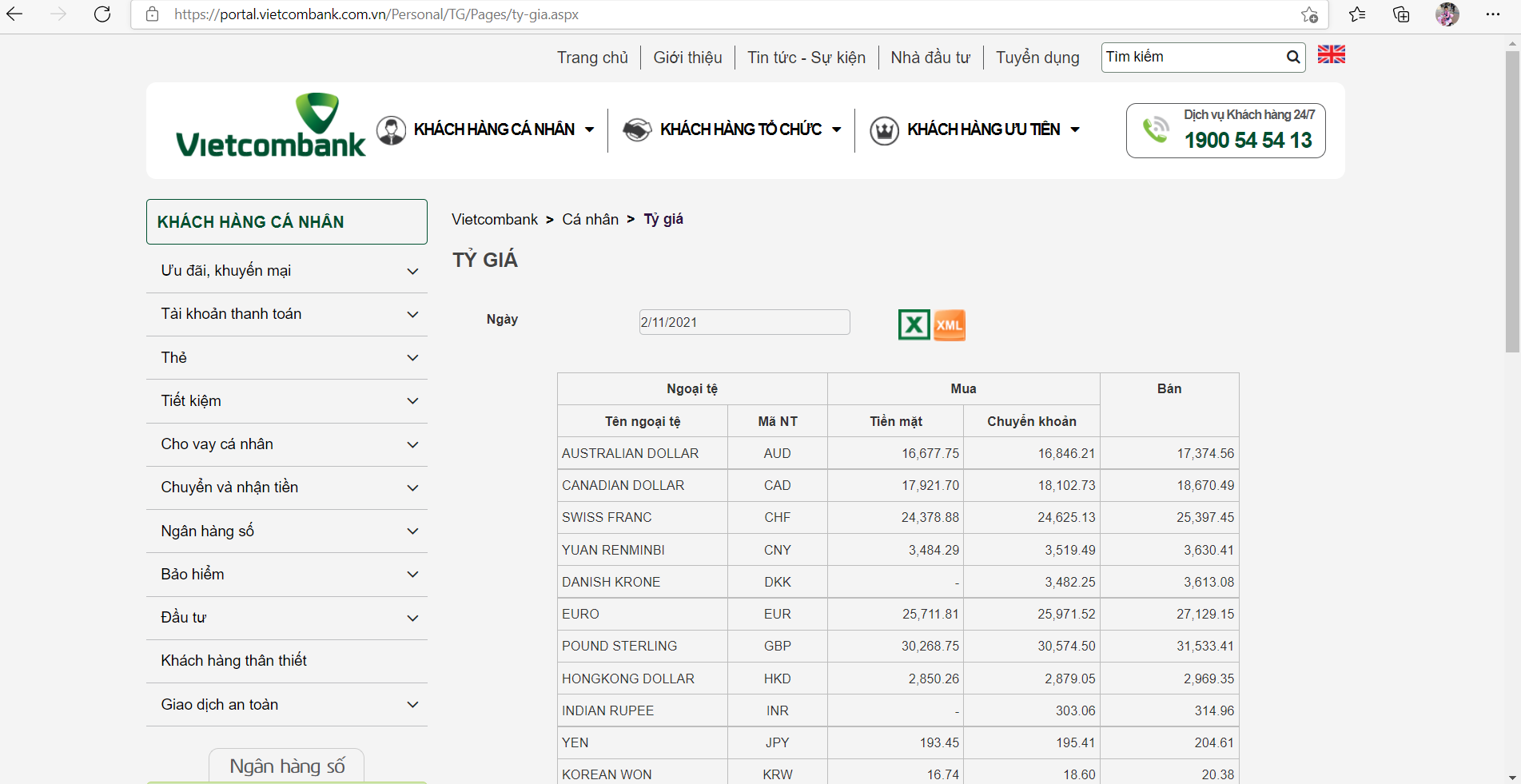
Tìm một số API: về thời tiết, chứng khoán. Sử dụng các Api để lấy dữ liệu

<https://samples.openwethermap.org/data/2.5/forecast?id=524901&appid=b1b15e88fa797225412429c1c50c122a1>



**Bài 10.**

**Nhúng Tỉ giá ngân hàng VietComBank vào ứng dụng Web**



Tỉ giá được mô tả bằng cấu trúc file XML như sau

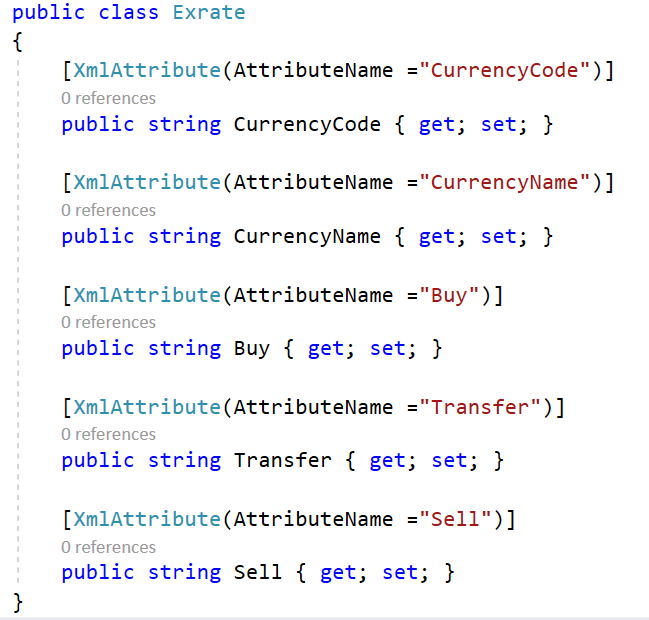


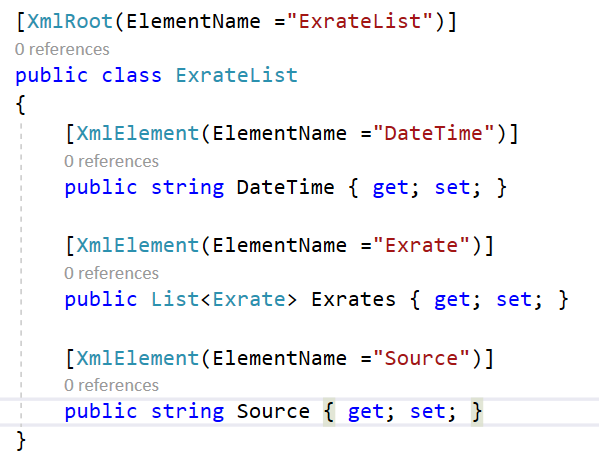
Tạo trang Web nhúng trực tiếp tỉ giá của ngân hàng VCB vào ứng dụng



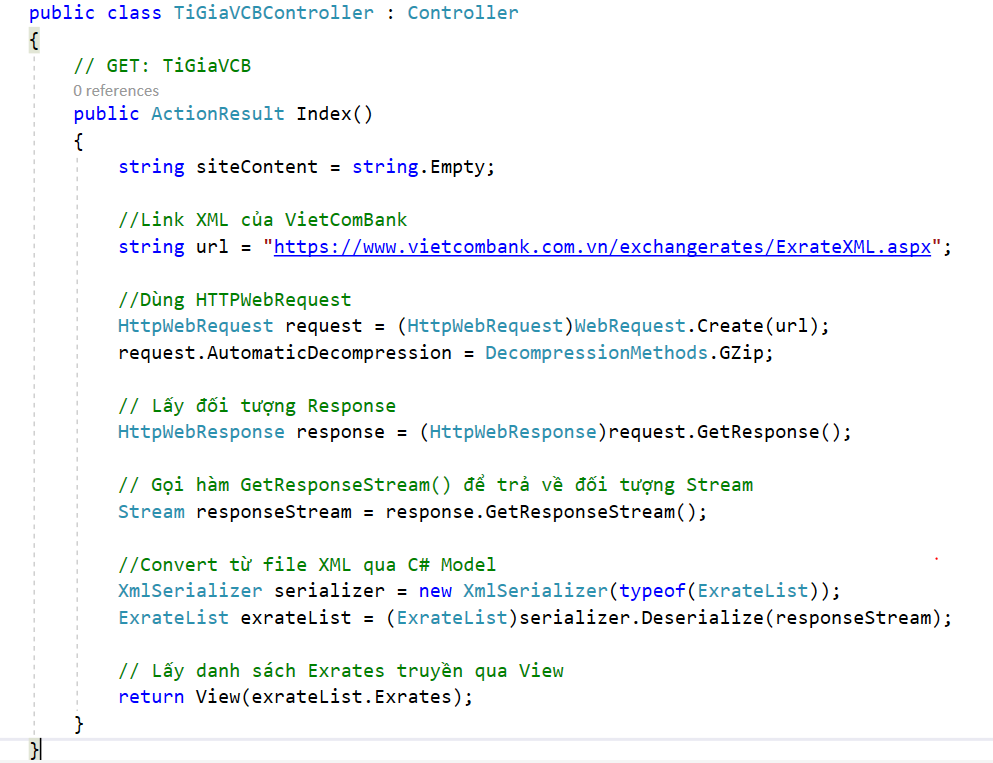
**Hướng dẫn:**

**B1: Tạo Class lưu trữ thông tin Exrate và ExrateList**

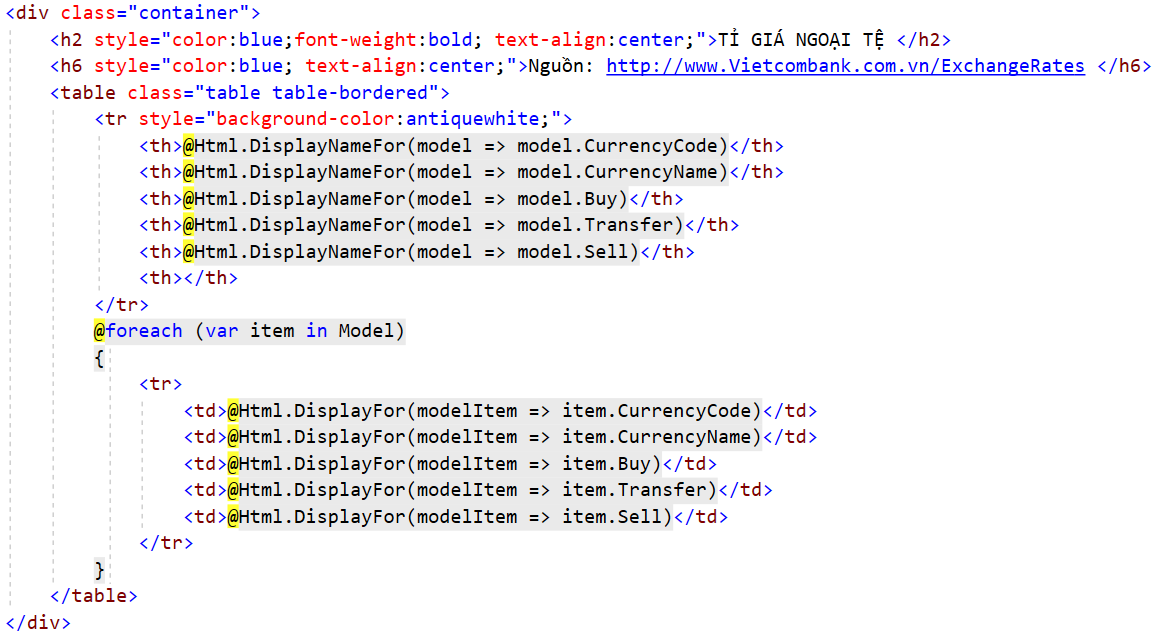




**B2: Tạo Controller đọc dữ liệu từ file Xml vào class ExrateList**



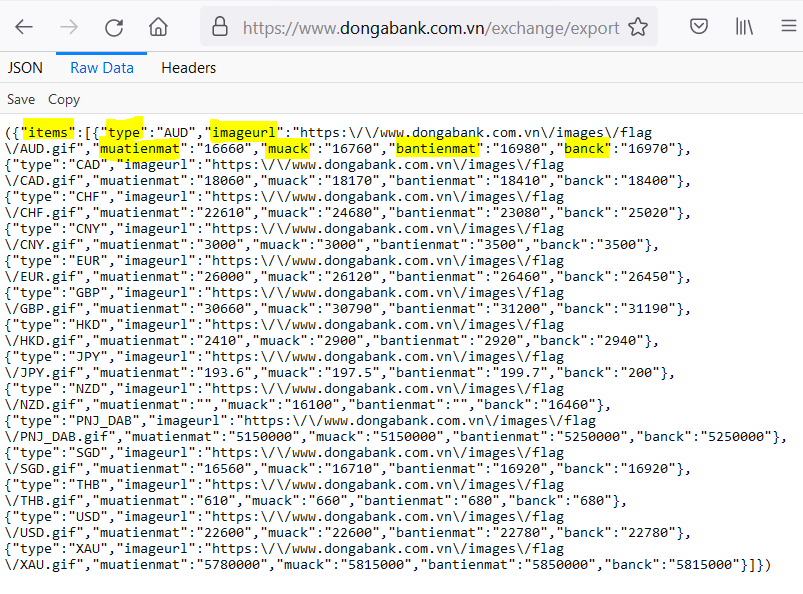
**B3: Tạo View Hiển Thị Dữ liệu ExrateList**



**Bài 11. Nhúng tỉ giá Ngân Hàng Đông Á vào ứng dụng Web**

**https://www.dongabank.com.vn/exchange/export**

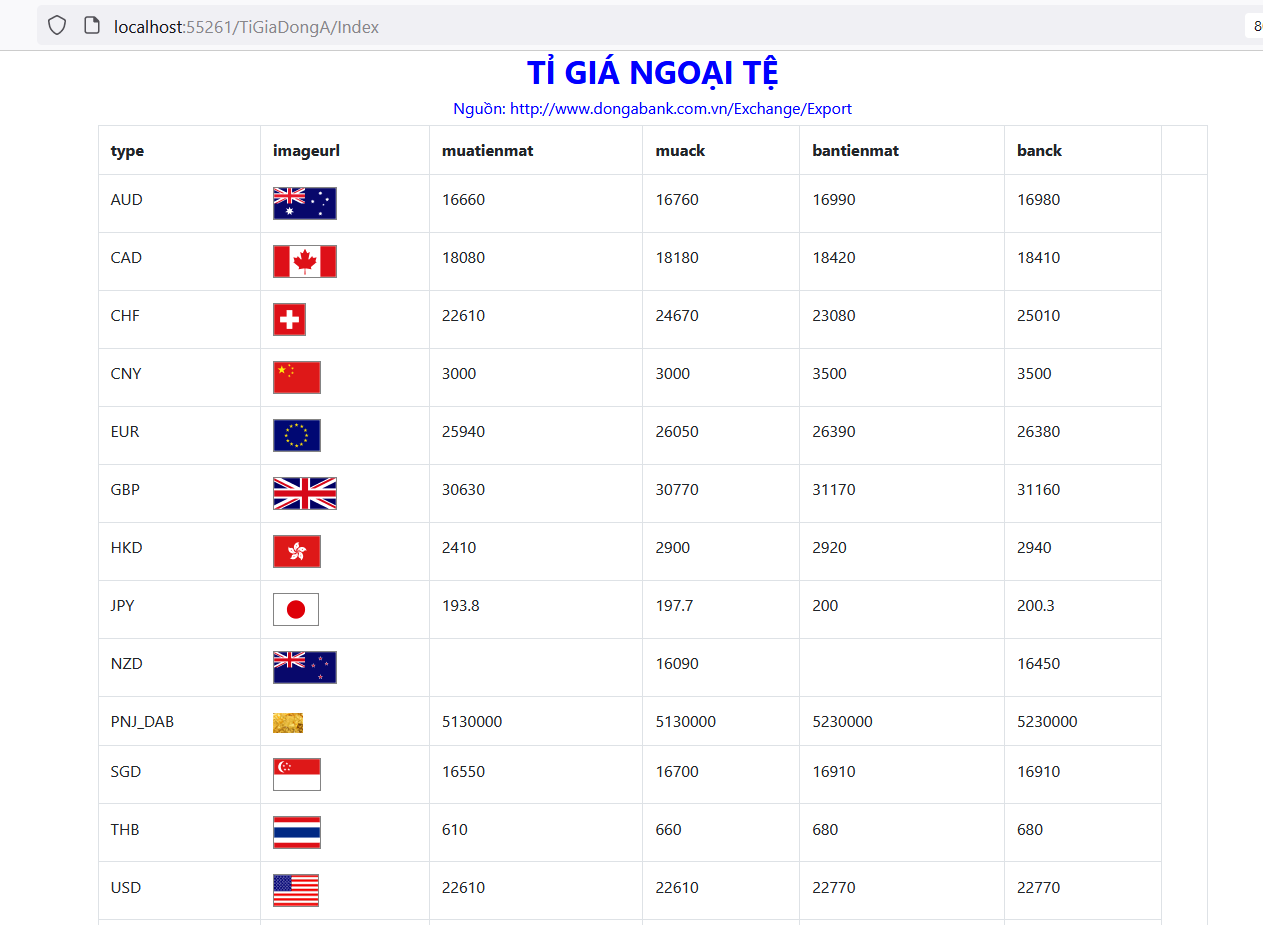
Tỉ giá được mô tả bằng cấu trúc file JSON như sau



Cấu trúc File JSON của Ngân hàng Đông Á Có một số đặc điểm:

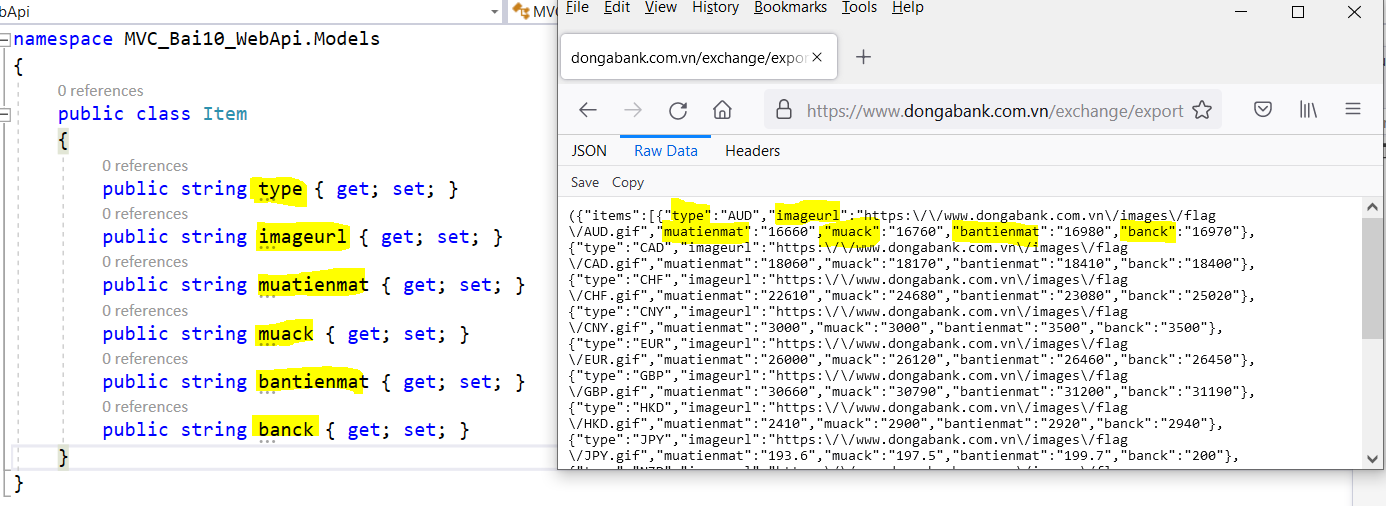
* + NH đã sửa cấu trúc file JSON bằng cách thêm ngoặc tròn bao lấy hai đầu của dữ liệu 🡪 ta cần xóa bỏ để dữ liệu trở về đúng cấu trúc của JSON.
  + Cấu trúc bên ngoài là một JSON, bên trong lại là 1 mảng các tỉ giá được lưu vào biến items. Mỗi một phần tử trong items là một dòng tỉ giá, nó có các thuộc tính : type, imageurl, muatienmat, muack, bantienmat, banck.
  + API nàu của Đông Á Bank đã cấm cách truy xuất thông thường, ta phải bổ sung các Header: “User-Agent” = “Mozilla/5.0 (compatible)” và “Accept” = “\*/\*”

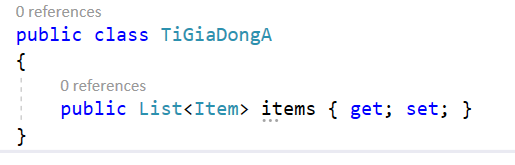
Kết quả thực thi



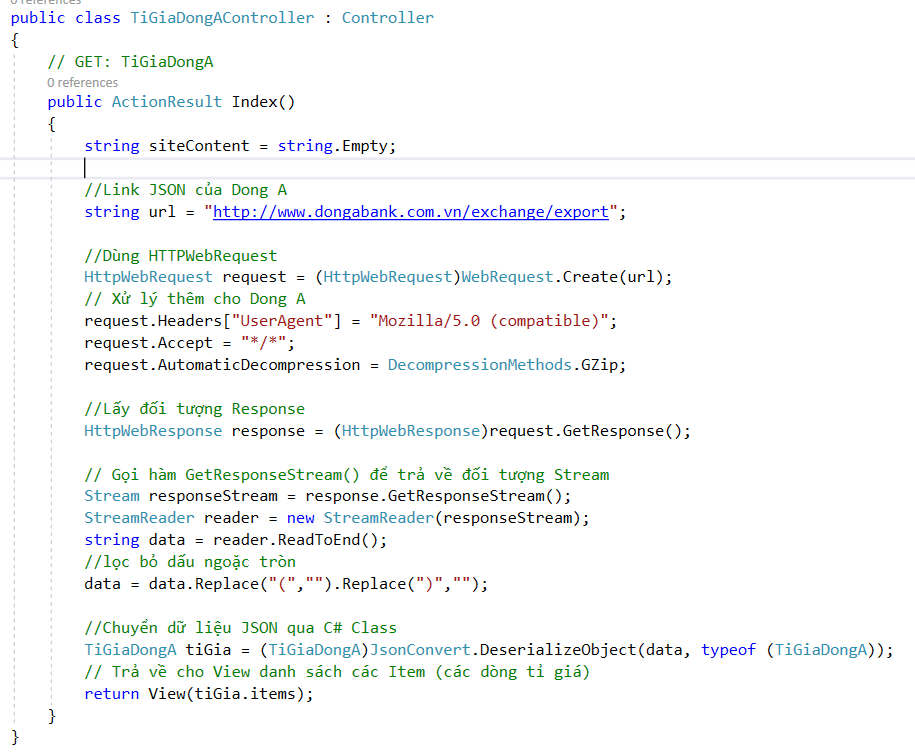
**Hướng dẫn:**

**B1:** Tạo Class **Item.cs** chứa các thông tin của một tỉ giá và lớp **TiGiaDongA.cs**





**B2:** Tạo Controller để đọc dữ liệu kiểu JSON



**B3:** Tạo View hiển thị Tỉ giá

1. **Bài tập về nhà**

**Bài 1.**

Tìm một API về thời tiết, Corona Virus,…. Đọc và hiển thị dữ liệu lên trang