


Trường ĐH CNTP TP.HCM Khoa: CNTT Bộ môn: Khoa học máy tính CÔNG NGHỆ WEB	BÀI 13,14 WEB API	
---	------------------------------------	---

A. MỤC TIÊU:

- Xây dựng được Web API lấy dữ liệu từ một đối tượng
- Xây dựng Web API lấy dữ liệu từ 1 danh sách đối tượng
- Xây dựng Web API lấy dữ liệu từ danh sách đối tượng ánh xạ từ EF

B. NỘI DUNG THỰC HÀNH

1. Cơ sở lý thuyết

1.1. 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 Transfer) 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.

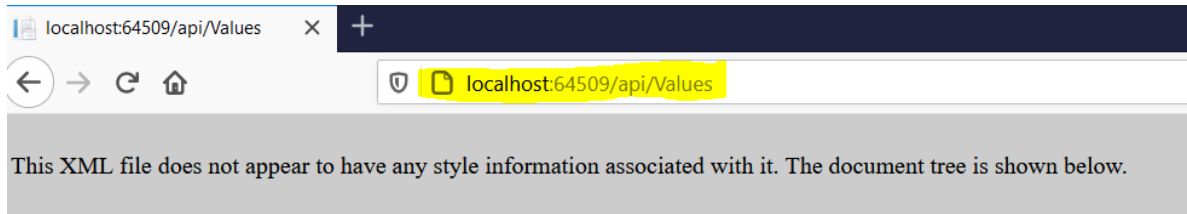
Đặc điểm của ASP.NET Web API

- 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.

1.2. Giới thiệu bài tập mẫu

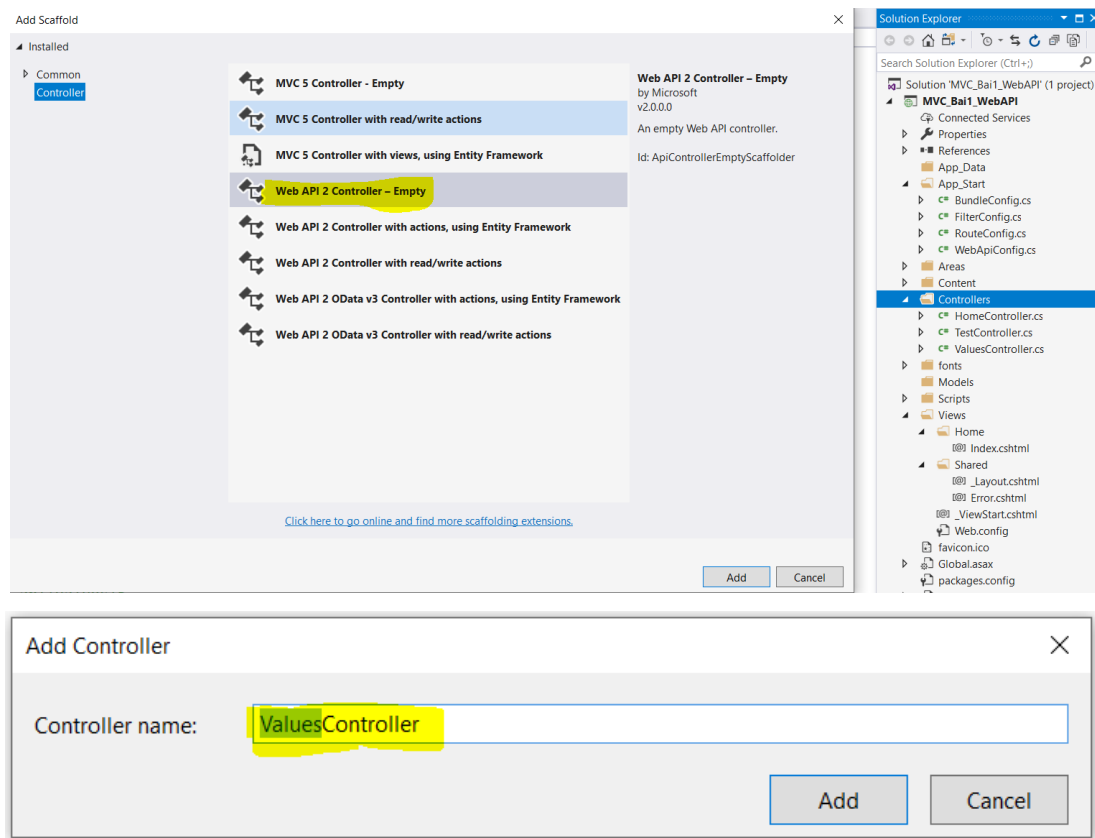
Bài 1: Tạo một Web Api Project MVC đặt tên là **MVC_Bai1_WebAPI** hiển thị danh sách 2 chuỗi “**value1**” và “**value2**” như sau



```
<ArrayOfstring>
  <string>value1</string>
  <string>value2</string>
</ArrayOfstring>
```

Hướng dẫn:

B1: Click vào Folder Controller → Add Controller



B2: Xây dựng phương thức Get

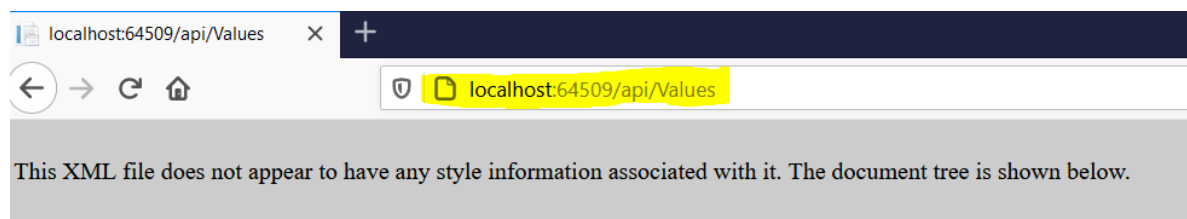
```

public class ValuesController : ApiController
{
    // GET api/values
    0 references
    public IEnumerable<string> Get()
    {
        return new string[] { "value1", "value2" };
    }

    // GET api/values/5
    0 references
    public string Get(int id)
    {
        return "value";
    }
}

```

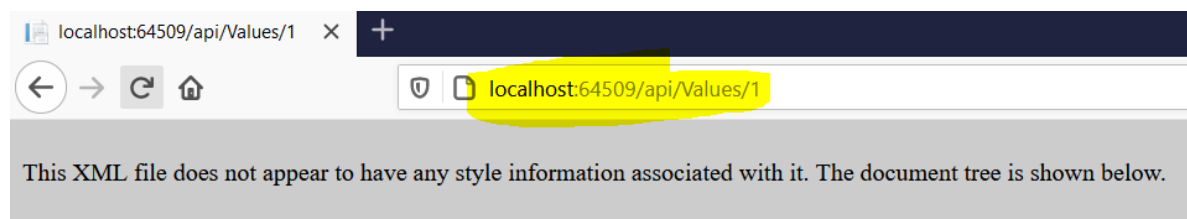
B3: Gọi thực thi Web API



```

<ArrayOfstring>
  <string>value1</string>
  <string>value2</string>
</ArrayOfstring>

```



```

<string>value</string>

```

File **WebApiConfig.cs** được sinh ra trong thư mục App_Start

```

public static class WebApiConfig
{
    1 reference
    public static void Register(HttpConfiguration config)
    {
        // Web API configuration and services

        // Web API routes
        config.MapHttpAttributeRoutes();

        config.Routes.MapHttpRoute(
            name: "DefaultApi",
            routeTemplate: "api/{controller}/{id}",
            defaults: new { id = RouteParameter.Optional }
        );
    }
}

```

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

```
namespace MVC_Bai1_WebAPI.Controllers
{
    0 references
    public class TestController : ApiController
    {
        0 references
        public string Get()
        {
            return "Welcome To Web API ! ";
        }
        0 references
        public List<string> Get(int Id)
        {
            return new List<string> { "Data 1","Data 2","Data 3"};
        }
    }
}
```

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:

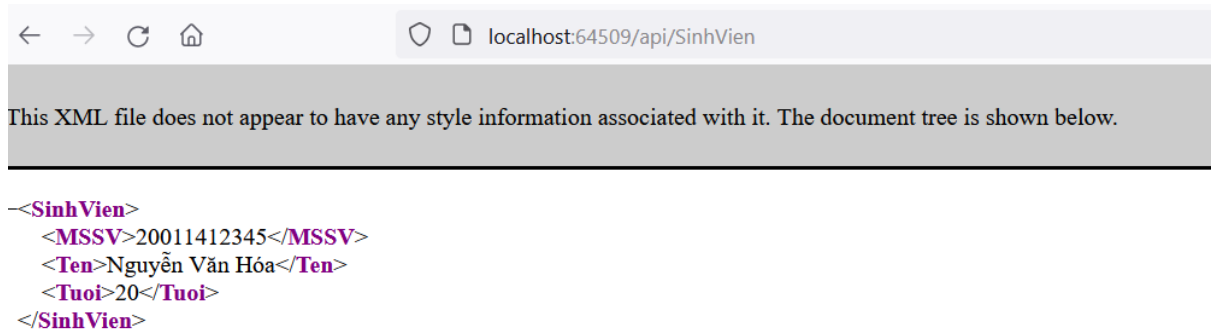
B1: Xây dựng lớp

```
namespace MVC_Bai1_WebAPI.Models
{
    1 reference
    public class SinhVien
    {
        1 reference
        public string MSSV { get; set; }
        1 reference
        public string Ten { get; set; }
        1 reference
        public int Tuoi { get; set; }
        0 references
        public SinhVien()
        {
            MSSV = "20011412345";
            Ten = "Nguyễn Văn Hóa";
            Tuoi = 20;
        }
    }
}
```

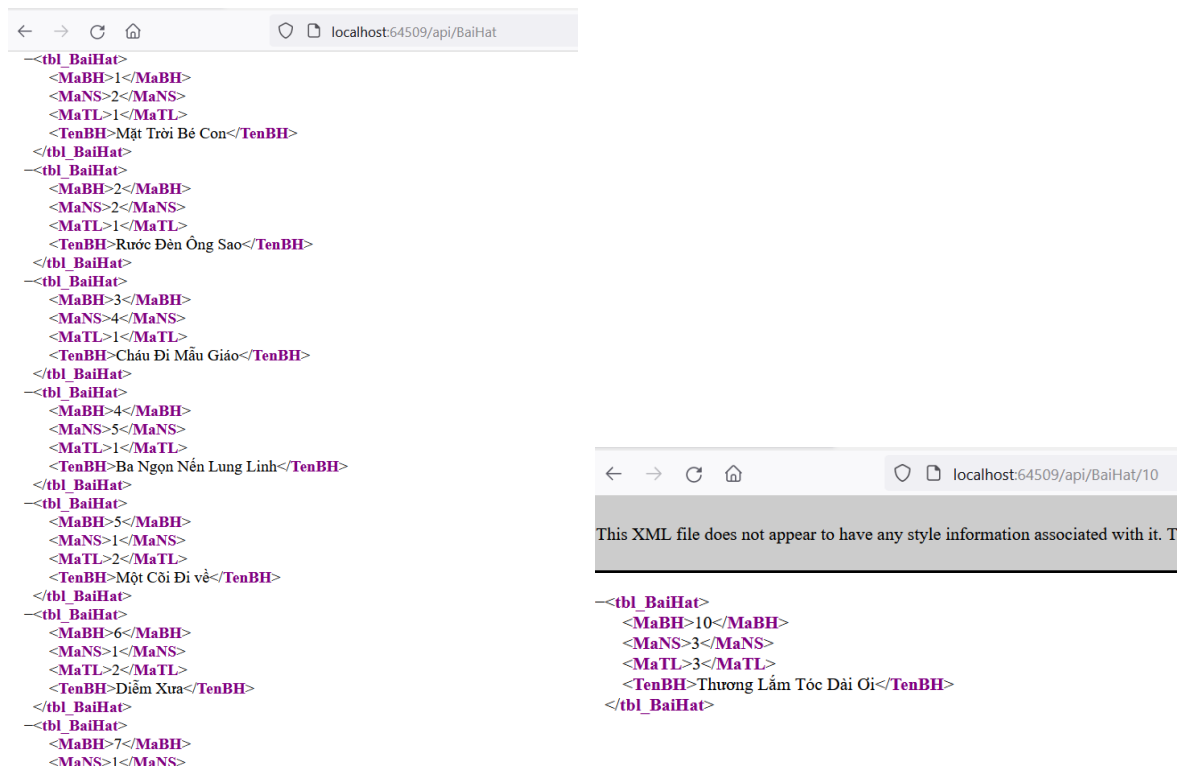
B2: Tạo Controller kế thừa từ lớp API

```
namespace MVC_Bai1_WebAPI.Controllers
{
    0 references
    public class SinhVienController : ApiController
    {
        0 references
        public SinhVien Get()
        {
            SinhVien sv = new SinhVien();
            return sv;
        }
    }
}
```

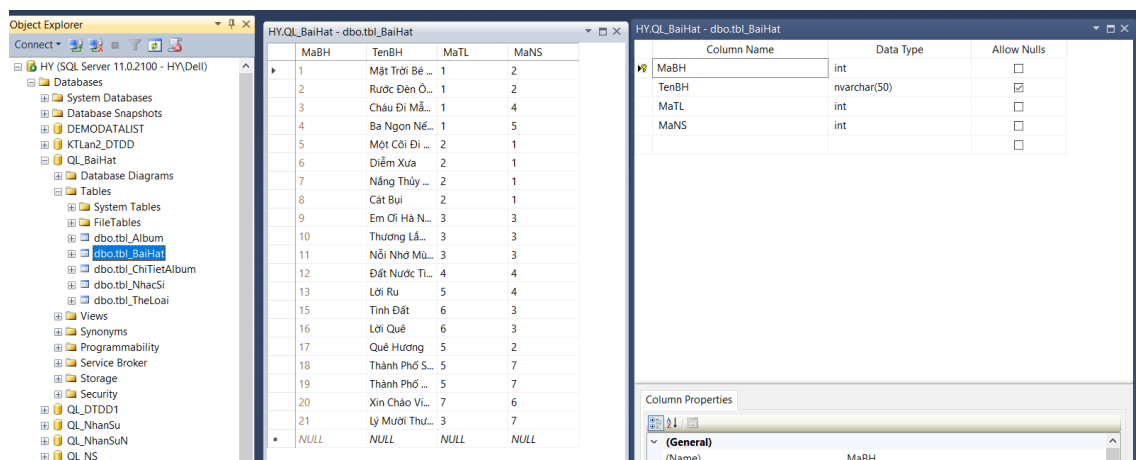
B3: Thực thi gọi API SinhVien



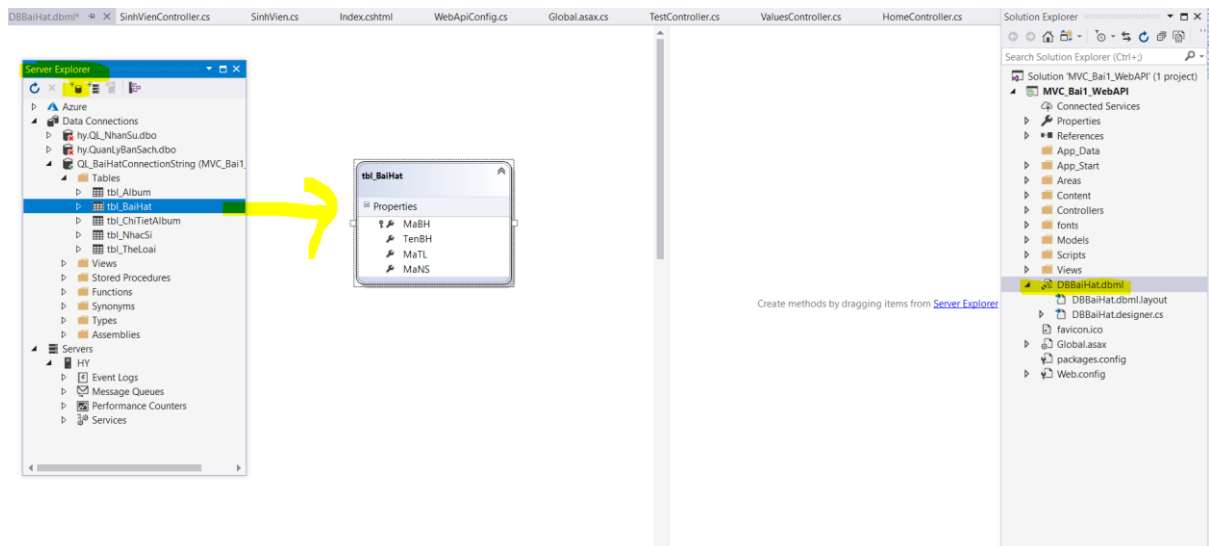
Bài 4: Tạo 1 Web API lấy dữ liệu lưu trữ từ bảng BaiHat trong SQL Server 2012



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]

```
public class BaiHatController : ApiController
{
    [HttpGet]
    0 references
    public List<tbl_BaiHat> GetBaiHatLists()
    {
        DBBaiHatDataContext db = new DBBaiHatDataContext();
        return db.tbl_BaiHats.ToList();
    }

    [HttpGet]
    0 references
    public tbl_BaiHat GetBaiHat(int id)
    {
        DBBaiHatDataContext db = new DBBaiHatDataContext();
        return db.tbl_BaiHats.FirstOrDefault(s => s.MaBH == id);
    }
}
```

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]

```
public int InsertNewBaiHat(string TenBH, int MaTL, int MaNS)
{
    try
    {
        DBBaiHatDataContext db = new DBBaiHatDataContext();
        tbl_BaiHat s = new tbl_BaiHat();
        s.TenBH = TenBH;
        s.MaTL = MaTL;
        s.MaNS = MaNS;
        db.tbl_BaiHats.InsertOnSubmit(s);
        db.SubmitChanges();
        return 1;
    }
    catch
    {
        return 0;
    }
}
```

Phương thức thay đổi thông tin: [HttpPut]

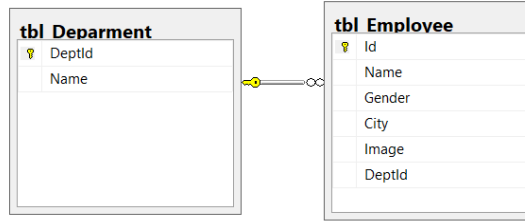
```
public bool UpdateBaiHat(int MaBH, string TenBH, int MaTL, int MaNS)
{
    try
    {
        DBBaiHatDataContext db = new DBBaiHatDataContext();
        tbl_BaiHat song = db.tbl_BaiHats.FirstOrDefault(s=>s.MaBH == MaBH);
        if (song == null) return false;
        song.TenBH = TenBH;
        song.MaTL = MaTL;
        song.MaNS = MaNS;
        db.SubmitChanges();
        return true;
    }
    catch
    {
        return false;
    }
}
```

Phương thức xóa thông tin: [HttpDelete]

```
[HttpDelete]
0 references
public bool DeleteBaiHat(int Ma)
{
    DBBaiHatDataContext db = new DBBaiHatDataContext();
    tbl_BaiHat song = db.tbl_BaiHats.FirstOrDefault(s=>s.MaBH == Ma);
    if (song == null) return false;
    db.tbl_BaiHats.DeleteOnSubmit(song);
    db.SubmitChanges();
    return true;
}
```

2. Bài tập tại lớp

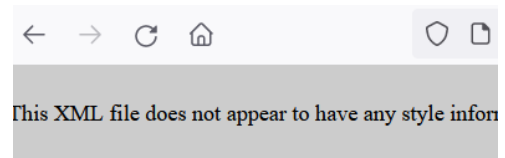
Bài 5: Cho CSDL :



Xây dựng Web API có nội dung như sau:

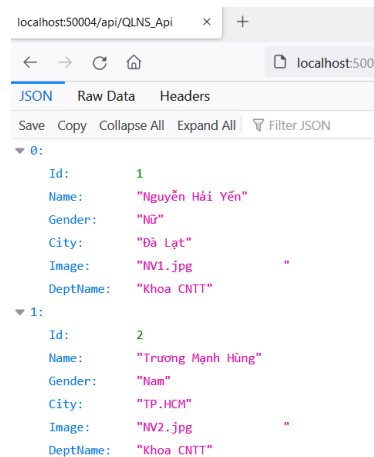
This XML file does not appear to have any style information associated with it.

```
<ArrayOfEmpDeptMix>
  <EmpDeptMix>
    <City>Đà Lạt</City>
    <DeptName>Khoa CNTT</DeptName>
    <Gender>Nữ</Gender>
    <Id>1</Id>
    <Image>NV1.jpg </Image>
    <Name>Nguyễn Hải Yến</Name>
  </EmpDeptMix>
  <EmpDeptMix>
    <City>TP.HCM</City>
    <DeptName>Khoa CNTT</DeptName>
    <Gender>Nam</Gender>
    <Id>2</Id>
    <Image>NV2.jpg </Image>
    <Name>Trương Mạnh Hùng</Name>
  </EmpDeptMix>
  <EmpDeptMix>
    <City>Thái Bình</City>
    <DeptName>Khoa Ngoại Ngữ</DeptName>
    <Gender>Nam</Gender>
    <Id>3</Id>
    <Image>NV1.jpg </Image>
    <Name>Đinh Duy Minh</Name>
  </EmpDeptMix>
  <EmpDeptMix>
    <City>Long An</City>
    <DeptName>Khoa Ngoại Ngữ</DeptName>
    <Gender>Nữ</Gender>
    <Id>4</Id>
    <Image>NV2.jpg </Image>
    <Name>Ngô Thị Nguyệt</Name>
  </EmpDeptMix>
</ArrayOfEmpDeptMix>
```



```
<EmpDeptMix>
  <City>Đà Lạt</City>
  <DeptName>Khoa CNTT</DeptName>
  <Gender>Nữ</Gender>
  <Id>1</Id>
  <Image>NV1.jpg </Image>
  <Name>Nguyễn Hải Yến</Name>
</EmpDeptMix>
```

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



Hướng dẫn:

Vào App_Start → WebApiConfig.cs

```
// Set JSON formatter
// config.Formatters.JsonFormatter.SupportedMediaTypes.Add(new MediaTypeHeaderValue("text/html"));





//var appXmlType = config.Formatters.XmlFormatter.SupportedMediaTypes.FirstOrDefault(t => t.MediaType == "application/xml");
//config.Formatters.XmlFormatter.SupportedMediaTypes.Remove(appXmlType);

// Remove the XML formatter
config.Formatters.Remove(config.Formatters.XmlFormatter);
```

Bài 7: Hiển thị Web API lên trang

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


localhost:50004/QLNS/ShowDataEmpDept 90% ☆

Name	Gender	City	Image	DeptName	
Nguyễn Hải Yến	Nữ	Đà Lạt		Khoa CNTT	Edit Details Delete
Trương Mạnh Hùng	Nam	TP.HCM		Khoa CNTT	Edit Details Delete
Đinh Duy Minh	Nam	Thái Bình		Khoa Ngoại Ngữ	Edit Details Delete
Ngô Thị Nguyệt	Nữ	Long An		Khoa Ngoại Ngữ	Edit Details Delete

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

localhost:50004/QLNS/ShowDetailEmpDept/1 90% ☆

	Tên NV	Nguyễn Hải Yến
	Giới tính	Nữ
	Thành Phố	Đà Lạt
	Phòng	Khoa CNTT

[Edit](#) | [Back to List](#)

Câu 9:

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

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

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

localhost:55261/TiGiaVCB/Index 90% ☆

TỈ GIÁ NGOẠI TỆ

Nguồn: <http://www.Vietcombank.com.vn/ExchangeRates>

CurrencyCode	CurrencyName	Buy	Transfer	Sell
AUD	AUSTRALIAN DOLLAR	16,482.72	16,649.21	17,171.43
CAD	CANADIAN DOLLAR	17,843.47	18,023.71	18,589.04
CHF	SWISS FRANC	24,242.55	24,487.42	25,255.49
CNY	YUAN RENMINBI	3,474.49	3,509.58	3,620.21
DKK	DANISH KRONE	-	3,474.67	3,605.23
EUR	EURO	25,648.14	25,907.21	27,062.04
GBP	POUND STERLING	30,226.42	30,531.74	31,489.40
HKD	HONGKONG DOLLAR	2,841.23	2,869.93	2,959.95
INR	INDIAN RUPEE	-	303.89	315.82
JPY	YEN	192.76	194.70	203.87
KRW	KOREAN WON	16.62	18.47	20.24
KWD	KUWAITI DINAR	-	75,090.44	78,038.35
MYR	MALAYSIAN RINGGIT	-	5,405.96	5,520.04

Hướng dẫn:

B1: Tạo Class lưu trữ thông tin Exrate và ExrateList

```
public class Exrate
{
    [XmlAttribute(AttributeName = "CurrencyCode")]
    0 references
    public string CurrencyCode { get; set; }

    [XmlAttribute(AttributeName = "CurrencyName")]
    0 references
    public string CurrencyName { get; set; }

    [XmlAttribute(AttributeName = "Buy")]
    0 references
    public string Buy { get; set; }

    [XmlAttribute(AttributeName = "Transfer")]
    0 references
    public string Transfer { get; set; }

    [XmlAttribute(AttributeName = "Sell")]
    0 references
    public string Sell { get; set; }
}
```

```

[XmlRoot(ElementName = "ExrateList")]
0 references
public class ExrateList
{
    [XmlElement(ElementName = "DateTime")]
    0 references
    public string DateTime { get; set; }

    [XmlElement(ElementName = "Exrate")]
    0 references
    public List<Exrate> Exrates { get; set; }

    [XmlElement(ElementName = "Source")]
    0 references
    public string Source { get; set; }
}

```

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

```

public class TiGiaVCBController : Controller
{
    // GET: TiGiaVCB
    0 references
    public ActionResult Index()
    {
        string siteContent = string.Empty;

        //Link XML của VietComBank
        string url = "https://www.vietcombank.com.vn/exchangerates/ExrateXML.aspx";

        //Dùng HTTPWebRequest
        HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
        request.AutomaticDecompression = DecompressionMethods.GZip;

        // Lấy đối tượng Response
        HttpWebResponse response = (HttpWebResponse)request.GetResponse();

        // Gọi hàm GetResponseStream() để trả về đối tượng Stream
        Stream responseStream = response.GetResponseStream();

        //Convert từ file XML qua C# Model
        XmlSerializer serializer = new XmlSerializer(typeof(ExrateList));
        ExrateList exrateList = (ExrateList)serializer.Deserialize(responseStream);

        // Lấy danh sách Exrates truyền qua View
        return View(exrateList.Exrates);
    }
}

```

B3: Tạo View Hiển Thị Dữ liệu ExrateList

```

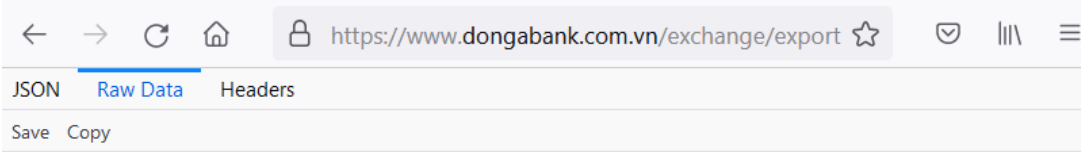
<div class="container">
  <h2 style="color:blue;font-weight:bold; text-align:center;">TỈ GIÁ NGOẠI TỆ </h2>
  <h6 style="color:blue; text-align:center;">Nguồn: http://www.Vietcombank.com.vn/ExchangeRates </h6>
  <table class="table table-bordered">
    <tr style="background-color:antiquewhite;">
      <th>@Html.DisplayNameFor(model => model.CurrencyCode)</th>
      <th>@Html.DisplayNameFor(model => model.CurrencyName)</th>
      <th>@Html.DisplayNameFor(model => model.Buy)</th>
      <th>@Html.DisplayNameFor(model => model.Transfer)</th>
      <th>@Html.DisplayNameFor(model => model.Sell)</th>
    </tr>
    @foreach (var item in Model)
    {
      <tr>
        <td>@Html.DisplayFor(modelItem => item.CurrencyCode)</td>
        <td>@Html.DisplayFor(modelItem => item.CurrencyName)</td>
        <td>@Html.DisplayFor(modelItem => item.Buy)</td>
        <td>@Html.DisplayFor(modelItem => item.Transfer)</td>
        <td>@Html.DisplayFor(modelItem => item.Sell)</td>
      </tr>
    }
  </table>
</div>

```

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



```

({
  "items": [
    {
      "type": "AUD",
      "imageUrl": "https://www.dongabank.com.vn/images/flag/AUD.gif",
      "muatienmat": "16660",
      "muack": "16760",
      "bantienmat": "16980",
      "banck": "16970"
    },
    {
      "type": "CAD",
      "imageUrl": "https://www.dongabank.com.vn/images/flag/CAD.gif",
      "muatienmat": "18060",
      "muack": "18170",
      "bantienmat": "18410",
      "banck": "18400"
    },
    {
      "type": "CHF",
      "imageUrl": "https://www.dongabank.com.vn/images/flag/CHF.gif",
      "muatienmat": "22610",
      "muack": "24680",
      "bantienmat": "23080",
      "banck": "25020"
    },
    {
      "type": "CNY",
      "imageUrl": "https://www.dongabank.com.vn/images/flag/CNY.gif",
      "muatienmat": "3000",
      "muack": "3000",
      "bantienmat": "3500",
      "banck": "3500"
    },
    {
      "type": "EUR",
      "imageUrl": "https://www.dongabank.com.vn/images/flag/EUR.gif",
      "muatienmat": "26000",
      "muack": "26120",
      "bantienmat": "26460",
      "banck": "26450"
    },
    {
      "type": "GBP",
      "imageUrl": "https://www.dongabank.com.vn/images/flag/GBP.gif",
      "muatienmat": "30660",
      "muack": "30790",
      "bantienmat": "31200",
      "banck": "31190"
    },
    {
      "type": "HKD",
      "imageUrl": "https://www.dongabank.com.vn/images/flag/HKD.gif",
      "muatienmat": "2410",
      "muack": "2900",
      "bantienmat": "2920",
      "banck": "2940"
    },
    {
      "type": "JPY",
      "imageUrl": "https://www.dongabank.com.vn/images/flag/JPY.gif",
      "muatienmat": "193.6",
      "muack": "197.5",
      "bantienmat": "199.7",
      "banck": "200"
    },
    {
      "type": "NZD",
      "imageUrl": "https://www.dongabank.com.vn/images/flag/NZD.gif",
      "muatienmat": "",
      "muack": "16100",
      "bantienmat": "",
      "banck": "16460"
    },
    {
      "type": "PNJ_DAB",
      "imageUrl": "https://www.dongabank.com.vn/images/flag/PNJ_DAB.gif",
      "muatienmat": "5150000",
      "muack": "5150000",
      "bantienmat": "5250000",
      "banck": "5250000"
    },
    {
      "type": "SGD",
      "imageUrl": "https://www.dongabank.com.vn/images/flag/SGD.gif",
      "muatienmat": "16560",
      "muack": "16710",
      "bantienmat": "16920",
      "banck": "16920"
    },
    {
      "type": "THB",
      "imageUrl": "https://www.dongabank.com.vn/images/flag/THB.gif",
      "muatienmat": "610",
      "muack": "660",
      "bantienmat": "680",
      "banck": "680"
    },
    {
      "type": "USD",
      "imageUrl": "https://www.dongabank.com.vn/images/flag/USD.gif",
      "muatienmat": "22600",
      "muack": "22600",
      "bantienmat": "22780",
      "banck": "22780"
    },
    {
      "type": "XAU",
      "imageUrl": "https://www.dongabank.com.vn/images/flag/XAU.gif",
      "muatienmat": "5780000",
      "muack": "5815000",
      "bantienmat": "5850000",
      "banck": "5815000"
    }
  ]
})

```

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 này 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

localhost:55261/TiGiaDongA/Index

TỈ GIÁ NGOẠI TỆ

Nguồn: <http://www.dongabank.com.vn/Exchange/Export>

type	imageurl	muatienmat	muack	bantienmat	banck
AUD		16660	16760	16990	16980
CAD		18080	18180	18420	18410
CHF		22610	24670	23080	25010
CNY		3000	3000	3500	3500
EUR		25940	26050	26390	26380
GBP		30630	30770	31170	31160
HKD		2410	2900	2920	2940
JPY		193.8	197.7	200	200.3
NZD			16090		16450
PNJ_DAB		5130000	5130000	5230000	5230000
SGD		16550	16700	16910	16910
THB		610	660	680	680
USD		22610	22610	22770	22770

Hướng dẫn:

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

```

namespace MVC_Bai10_WebApi.Models
{
    0 references
    public class Item
    {
        0 references
        public string type { get; set; }
        0 references
        public string imageurl { get; set; }
        0 references
        public string muatienmat { get; set; }
        0 references
        public string muack { get; set; }
        0 references
        public string bantienmat { get; set; }
        0 references
        public string banck { get; set; }
    }
}

```

dongabank.com.vn/exchange/expo

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

JSON Raw Data Headers

Save Copy

```

{
  "items": [
    {
      "type": "AUD",
      "imageurl": "https://www.dongabank.com.vn/images/flag/AUD.gif",
      "muatienmat": "16660",
      "muack": "16760",
      "bantienmat": "16980",
      "banck": "16970"
    },
    {
      "type": "CAD",
      "imageurl": "https://www.dongabank.com.vn/images/flag/CAD.gif",
      "muatienmat": "18080",
      "muack": "18170",
      "bantienmat": "18410",
      "banck": "18400"
    },
    {
      "type": "CHF",
      "imageurl": "https://www.dongabank.com.vn/images/flag/CHF.gif",
      "muatienmat": "22610",
      "muack": "24680",
      "bantienmat": "23080",
      "banck": "25020"
    },
    {
      "type": "CNY",
      "imageurl": "https://www.dongabank.com.vn/images/flag/CNY.gif",
      "muatienmat": "3000",
      "muack": "3000",
      "bantienmat": "3500",
      "banck": "3500"
    },
    {
      "type": "EUR",
      "imageurl": "https://www.dongabank.com.vn/images/flag/EUR.gif",
      "muatienmat": "26000",
      "muack": "26120",
      "bantienmat": "26460",
      "banck": "26450"
    },
    {
      "type": "GBP",
      "imageurl": "https://www.dongabank.com.vn/images/flag/GBP.gif",
      "muatienmat": "30660",
      "muack": "30790",
      "bantienmat": "31200",
      "banck": "31190"
    },
    {
      "type": "HKD",
      "imageurl": "https://www.dongabank.com.vn/images/flag/HKD.gif",
      "muatienmat": "2410",
      "muack": "2900",
      "bantienmat": "2920",
      "banck": "2940"
    },
    {
      "type": "JPY",
      "imageurl": "https://www.dongabank.com.vn/images/flag/JPY.gif",
      "muatienmat": "193.6",
      "muack": "197.5",
      "bantienmat": "199.7",
      "banck": "200"
    }
  ]
}

```



```

0 references
public class TiGiaDongA
{
    0 references
    public List<Item> items { get; set; }
}

```

B2: Tạo Controller để đọc dữ liệu kiểu JSON

```

0 references
public class TiGiaDongAController : Controller
{
    // GET: TiGiaDongA
    0 references
    public ActionResult Index()
    {
        string siteContent = string.Empty;

        //Link JSON của Dong A
        string url = "http://www.dongabank.com.vn/exchange/export";

        //Dùng HTTPWebRequest
        HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
        // Xử lý thêm cho Dong A
        request.Headers["UserAgent"] = "Mozilla/5.0 (compatible)";
        request.Accept = "*/*";
        request.AutomaticDecompression = DecompressionMethods.GZip;

        //Lấy đối tượng Response
        HttpWebResponse response = (HttpWebResponse)request.GetResponse();

        // Gọi hàm GetResponseStream() để trả về đối tượng Stream
        Stream responseStream = response.GetResponseStream();
        StreamReader reader = new StreamReader(responseStream);
        string data = reader.ReadToEnd();
        //lọc bỏ dấu ngoặc tròn
        data = data.Replace("(", "").Replace(")", "");

        //Chuyển dữ liệu JSON qua C# Class
        TiGiaDongA tiGia = (TiGiaDongA)JsonConvert.DeserializeObject(data, typeof(TiGiaDongA));
        // Trả về cho View danh sách các Item (các dòng tỉ giá)
        return View(tiGia.items);
    }
}

```

B3: Tạo View hiển thị Tỉ giá

3. Bài tập về nhà

Bài 1.

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