



# CÔNG NGHỆ WEB

## DỊCH VỤ WEB





# NỘI DUNG

- Giới thiệu dịch vụ web
- Làm quen với XML
- Web API service





# GIỚI THIỆU

## ■ Dịch vụ web:

- + Dịch vụ Web là một hệ thống phần mềm được thiết kế để hỗ trợ khả năng tương tác giữa các ứng dụng trên các máy tính khác nhau thông qua mạng Internet và giao diện dùng chung.
- + Dịch vụ Web là tài nguyên phần mềm có thể xác định bằng địa chỉ URL, thực hiện các chức năng và đưa ra các thông tin người dùng yêu cầu.





# GIỚI THIỆU

- Đặc điểm:
  - + Cho phép client và server tương tác được với nhau ngay cả trong những môi trường khác nhau.
  - + Một Dịch vụ Web bao gồm có nhiều moduls và có thể công bố lên mạng Internet.
  - + Là sự kết hợp của việc phát triển theo hướng từng thành phần với những lĩnh vực cụ thể và cơ sở hạ tầng Web.





# GIỚI THIỆU

- HTTP service: Dựa trên .NET Framework, xây dựng HTTP service có một số công nghệ sau:
  - + Web service
  - + WCF service
  - + WCF REST service
  - + Web API service.





## ■ Web Service:

- + Đây là công nghệ cũ của .NET Framework
- + Nó dựa trên SOAP (Simple Object Access protocol) dữ liệu trả về dạng XML
- + Chỉ hỗ trợ giao thức HTTP
- + Không phải Open Source nhưng có thể sử dụng được với bất cứ client nào hỗ trợ XML
- + Chỉ có thể host trên IIS





# GIỚI THIỆU

## ■ WCF:

- + .NET 3.0 trở lên
- + Cũng dựa trên SOAP và trả về dữ liệu dạng XML
- + Phát triển dựa trên Web service và hỗ trợ thêm rất nhiều giao thức khác nhau như: TCP, HTTP, HTTPS, Named Pipes, MSMQ.
- + Không phải Open Source nhưng có thể sử dụng bởi các client hỗ trợ XML
- + Có thể host được trong ứng dụng, trên IIS hoặc Windows Service.





## ■ WCF REST:

- + .NET 3.0 trở lên
- + Là bản nâng cấp của WCF với việc trên .NET 3.5 Microsoft bổ sung webHttpBinding để hỗ trợ RESTful service
- + Hỗ trợ 2 HTTP verb GET, POST để truyền nhận dữ liệu với 2 thuộc tính tương ứng là WebGet và WebInvoke
- + Muốn sử dụng các HTTP verb khác như PUT, DELETE cần cấu hình thêm trên IIS
- + Hỗ trợ các định dạng dữ liệu XML, ATOM, JSON.







# GIỚI THIỆU

- Web API :
  - + .NET 4.0 trở lên
  - + Đây là một framework mới giúp cho việc xây dựng các HTTP service rất đơn giản và nhanh chóng
  - + Open Source và có thể được sử dụng bởi bất kì client nào hỗ trợ XML, JSON
  - + Hỗ trợ đầy đủ các thành phần HTTP: URI, request/response headers, caching, content formats





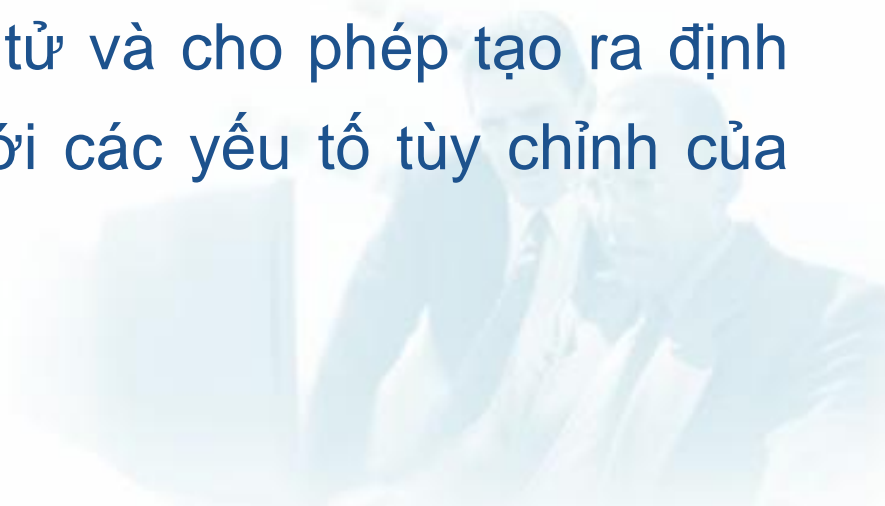
# GIỚI THIỆU

- Web API :
  - + Có thể host trong ứng dụng hoặc trên IIS
  - + Kiến trúc lý tưởng cho các thiết bị có băng thông giới hạn như smartphone, tablet
  - + Định dạng dữ liệu có thể là JSON, XML hoặc một kiểu dữ liệu bất kỳ





## ■ Giới thiệu:

- + XML tương tự như HTML là ngôn ngữ đánh dấu với các thẻ.
  - + Khác với HTML, XML không có tập các thẻ cố định, thay vì đó XML là “metalanguage” cho phép đặt tên, sắp xếp thứ tự các phần tử và cho phép tạo ra định dạng dữ liệu của riêng với các yếu tố tùy chỉnh của riêng.
- 



```
<?xml version="1.0" encoding="utf-8" ?>
<productCatalog>
  <catalogName>Acme Fall 2008 Catalog</catalogName>
  <expiryDate>2008-01-01</expiryDate>
  <products>
    <product id="1001">
      <productName>Magic Ring</productName>
      <productPrice>342.10</productPrice>
    </product>
    <product id="1002">
      <productName>Flying Carpet</productName>
      <productPrice>982.99</productPrice>
    </product>
  </products>
</productCatalog>
```



## ■ Đặc điểm của XML:

- + Tính phổ biến
- + Khả năng mở rộng và tính linh hoạt
- + Liên quan đến một số tiêu chuẩn và công cụ
- + Mọi thẻ mở đều có thẻ đóng
- + Phần tử rỗng kết thúc với />
- + Các phần tử có thể lồng nhau nhưng không chồng chéo

```
<person>  
  <firstName>  
  </firstName>_  
</person>
```

Không được: <person>  
 <firstName>  
</person>  
 </firstName>



## ■ Cấu trúc file XML:

- + Các phần tử và các thuộc tính phải đồng nhất trong các trường hợp.
  - Ví dụ tags `<FirstName>` `</firstName>` không hợp lệ
- + Một phần tử không thể có hai thuộc tính cùng tên, nhưng một phần tử có thể chứa các phần tử lồng như cùng tên.
- + Một document chỉ có thể một phần tử root
- + Giá trị của các thuộc tính được bao trong cặp “”
- + Chú thích bao trong cặp `<!-- and -->`



- **Lược đồ XML:** Là định dạng document cho phép đưa ra các quy tắc đối với custom mark up language.
  - + Document vocabulary: Xác định tên, thuộc tính được sử dụng trong văn bản XML
  - + Document structure: Xác định cấu trúc các thẻ
  - + Supported data types: Cung cấp kiểu dữ liệu
  - + Allowed data ranges: Thiết lập các ràng buộc giới hạn dữ liệu như độ dài văn bản, khuôn dạng dữ liệu,....



```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="productCatalog">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="catalogName" type="xsd:string"/>
        <xsd:element name="expiryDate" type="xsd:date"/>
        <xsd:element name="products">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="product"
                type="productType" maxOccurs="unbounded" />
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```







# XML Schemas

```
</xsd:complexType>
  </xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:complexType name="productType">
  <xsd:sequence>
    <xsd:element name="productName" type="xsd:string"/>
    <xsd:element name="productPrice" type="xsd:decimal"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:integer" use="required"
/>
  </xsd:complexType>
</xsd:schema>
```





## ■ Giới thiệu:

- + Web API là một giao diện lập trình ứng dụng (Application Programming Interface - API) trên web (HTTP).
- + Tài nguyên của ứng dụng được thao tác trên HTTP sử dụng các phương thức HTTP chuẩn như GET, POST, PUT, and DELETE.
- + HTTP services được sử dụng trên các thiết bị khác nhau với các nền tảng khác nhau.





- Tạo Web API Application :
  - + Tính năng Web API dựa trên một controller đặc biệt đưa vào ứng dụng MVC.
  - + Controller đó gọi là API Controller, có hai đặc trưng:
    - Action methods trả về model, objects (không trả về ActionResult)
    - Action methods được chọn dựa trên HTTP method sử dụng trong request .





- Tạo Web API Application :
  - + Tạo API Controller:

A screenshot of the 'Add Controller' dialog box in Visual Studio. The dialog has a title bar with a close button (X). Inside, there's a 'Controller name:' label followed by a text box containing 'DatVeAPIController'. Below that is a 'Scaffolding options' section. It contains a 'Template:' label followed by a dropdown menu showing 'API controller with empty read/write actions'. Below the template dropdown are three more dropdowns: 'Model class:', 'Data context class:', and 'Views:'. The 'Views:' dropdown is currently set to 'None'. To the right of the 'Views:' dropdown is an 'Advanced Options...' button. At the bottom right of the dialog are two buttons: 'Add' and 'Cancel'.



## ■ Tạo Web API Application :

### + Tạo API Controller:

```
public class DatVeAPIController : ApiController {  
    // GET api/datveapi  
    public IEnumerable<DatVe> Get() {  
        return new atVeAPIFunction().TestAPIs.ToList();  
    }  
    // GET api/datveapi/5  
    public DatVe Get(string id){  
        return new DatVeAPIFunction().FindEntity(id);  
    }  
}
```





## ■ Tạo Web API Application :

### + Tạo API Controller:

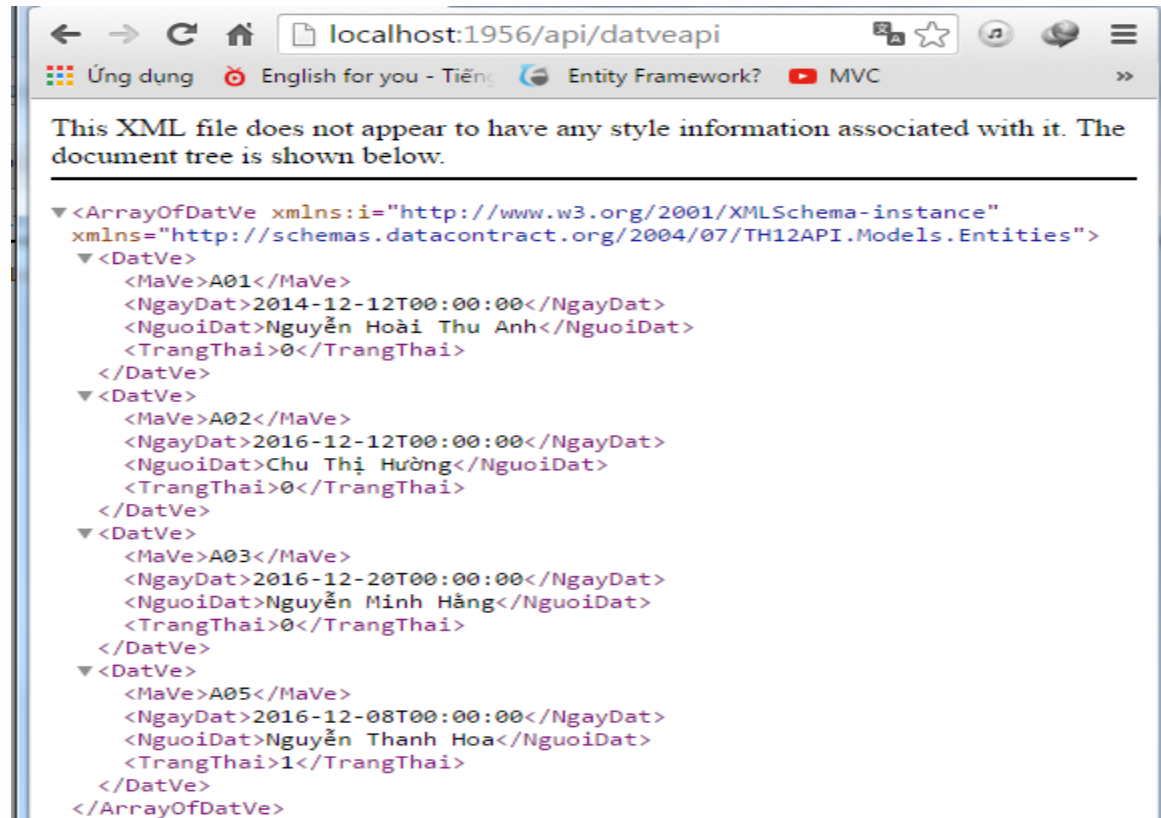
```
// POST api/datveapi
public int Post(DatVe item){
    return new DatVeAPIFunction().Insert(item);}

// PUT api/datveapi/5
public int Put(string id, DatVe item){
    return new DatVeAPIFunction().Update(id, item);}

// DELETE api/datveapi/5
public int Delete(string id){
    return new DatVeAPIFunction().Delete(id);}
}
```



- Tạo Web API Application :
  - + Kiểm tra dịch vụ: Dữ liệu XML



```
<?xml version='1.0'>
<ArrayOfDatVe xmlns:i='http://www.w3.org/2001/XMLSchema-instance'
xmlns='http://schemas.datacontract.org/2004/07/TH12API.Models.Entities'>
  <DatVe>
    <MaVe>A01</MaVe>
    <NgayDat>2014-12-12T00:00:00</NgayDat>
    <NguoiDat>Nguyễn Hoài Thu Anh</NguoiDat>
    <TrangThai>0</TrangThai>
  </DatVe>
  <DatVe>
    <MaVe>A02</MaVe>
    <NgayDat>2016-12-12T00:00:00</NgayDat>
    <NguoiDat>Chu Thị Hường</NguoiDat>
    <TrangThai>0</TrangThai>
  </DatVe>
  <DatVe>
    <MaVe>A03</MaVe>
    <NgayDat>2016-12-20T00:00:00</NgayDat>
    <NguoiDat>Nguyễn Minh Hằng</NguoiDat>
    <TrangThai>0</TrangThai>
  </DatVe>
  <DatVe>
    <MaVe>A05</MaVe>
    <NgayDat>2016-12-08T00:00:00</NgayDat>
    <NguoiDat>Nguyễn Thanh Hoa</NguoiDat>
    <TrangThai>1</TrangThai>
  </DatVe>
</ArrayOfDatVe>
```





## ■ Tạo Web API Application :

+ Kiểm tra dịch vụ: Cấu hình cung cấp dữ liệu Json:

```
public static class WebApiConfig {  
    public static void Register(HttpConfiguration config){  
        config.Routes.MapHttpRoute(name: "DefaultApi",  
            routeTemplate: "api/{controller}/{id}",  
            defaults: new { id = RouteParameter.Optional } );
```

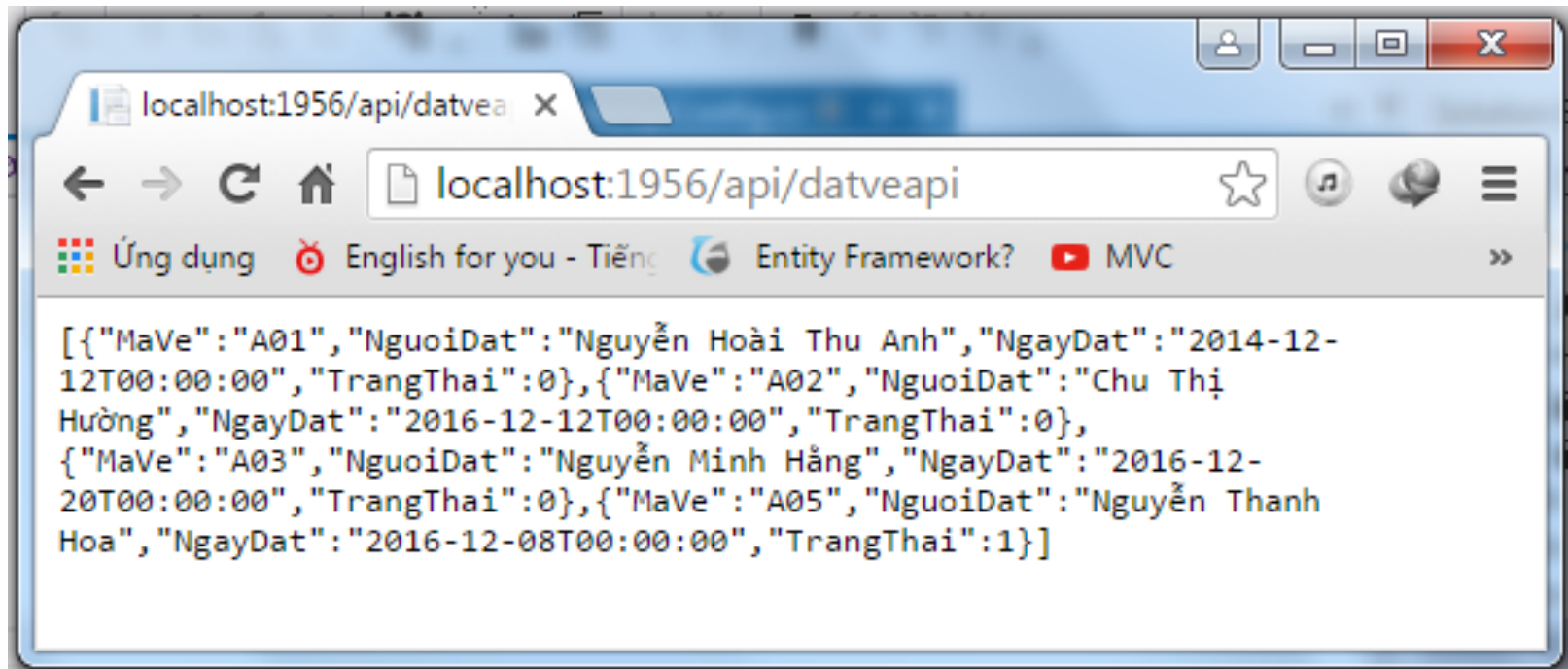
```
        var appXmlType =  
config.Formatters.XmlFormatter.SupportedMediaTypes.FirstOrDefault(  
t => t.MediaType == "application/xml");  
config.Formatters.XmlFormatter.SupportedMediaTypes.Remove(appXml  
Type);  
    }  
}
```

}





- Tạo Web API Application :
  - + Kiểm tra dịch vụ: Dữ liệu Json





## ■ Sử dụng Web API Application : + Dùng Console Application/MVC:

```
public ActionResult Index(){  
    HttpClient client = new HttpClient();  
    client.BaseAddress = new Uri("http://localhost:1956/");  
    client.DefaultRequestHeaders.Accept.Add(new  
        MediaTypeWithQualityHeaderValue("application/json"));  
    HttpResponseMessage response = client.GetAsync("api/DatVeAPI").Result;  
    if (response.IsSuccessStatusCode){  
        var model =  
            response.Content.ReadAsAsync<IEnumerable<DatVe>>().Result;  
        return View(model);  
    }  
    return View();  
}
```





## ■ Sử dụng Web API Application : + Dùng Console Application/MVC:

```
public ActionResult Details(string id){  
    HttpClient client = new HttpClient();  
    client.BaseAddress = new Uri("http://localhost:1956/");  
    client.DefaultRequestHeaders.Accept.Add(new  
        MediaTypeWithQualityHeaderValue("application/json"));  
    HttpResponseMessage response =  
        client.GetAsync("api/DatVeAPI/"+id).Result;  
    DatVe model = new DatVe();  
    if (response.IsSuccessStatusCode){  
        model = response.Content.ReadAsAsync<DatVe>().Result;  
    }  
    return View(model);  
}
```





## ■ Sử dụng Web API Application : + Dùng Console Application/MVC:

```
// GET: /Test/Edit/5
public ActionResult Edit(string id){
    HttpClient client = new HttpClient();
    client.BaseAddress = new Uri("http://localhost:1956/");
    client.DefaultRequestHeaders.Accept.Add(new
        MediaTypeWithQualityHeaderValue("application/json"));
    HttpResponseMessage response =
        client.GetAsync("api/DatVeAPI/" + id).Result;
    DatVe model = new DatVe();
    if (response.IsSuccessStatusCode){
        model = response.Content.ReadAsAsync<DatVe>().Result;
    }
    return View(model);
}
```





## ■ Sử dụng Web API Application : + Dùng Console Application/MVC:

[HttpPost]

```
public ActionResult Edit(string id, DatVe item){  
    HttpClient client = new HttpClient();  
    client.BaseAddress = new Uri("http://localhost:1956/");  
    client.DefaultRequestHeaders.Accept.Add(new  
        MediaTypeWithQualityHeaderValue("application/json"));  
    HttpResponseMessage response =  
        client.PutAsJsonAsync<DatVe>("api/DatVeAPI/" + id, item).Result;  
    if (response.IsSuccessStatusCode){  
        return RedirectToAction("Index");  
    }  
    return View();  
}
```





- Sử dụng Web API Application :
  - + Dùng Javascript:
    - View [Index và Javascript](#)

## Đặt vé

Vé	Người đặt	Ngày đặt	Chức năng
A01	Nguyễn Hoài Thu Anh	2014-12-12T00:00:00	<a href="#">Chi tiết</a>
A02	Chu Thị Hường	2016-12-12T00:00:00	<a href="#">Chi tiết</a>
A03	Nguyễn Minh Hằng	2016-12-20T00:00:00	<a href="#">Chi tiết</a>
A05	Nguyễn Thanh Hoa	2016-12-08T00:00:00	<a href="#">Chi tiết</a>

Add



- Sử dụng Web API Application :
  - + Dùng Javascript:
    - View [Details và Javascript](#)

## Edit Reservation

Ngày đặt

Người đặt



- Sử dụng Web API Application :
  - + Dùng Javascript:
    - View [Create và Javascript](#)

## Add New Reservation

Số vé:

Ngày đặt:

Người đặt:

Submit







# THẢO LUẬN – CÂU HỎI



Biên soạn: Chu Thị Hường – Bộ môn HTTT – Khoa CNTT