



ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

VNUHCM - UIT

LẬP TRÌNH WEB

CHƯƠNG 9 ASP.NET WEB API

NỘI DUNG

1

- Giới thiệu

2

- API là gì?

3

- Web API là gì?

4

- Web API hoạt động như thế nào?

5

- Ưu và nhược điểm của Web API

6

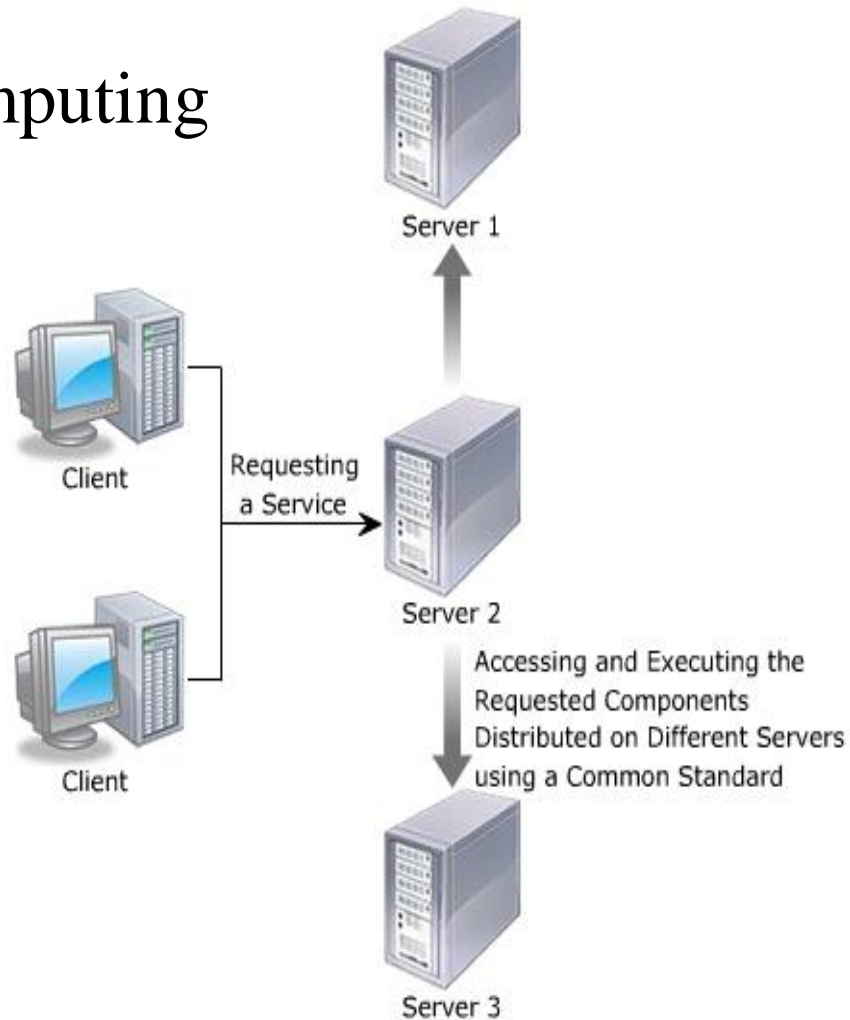
- Cách xây dựng Web API

7

- Gọi sử dụng API

1. Giới thiệu

❖ Distributed Computing

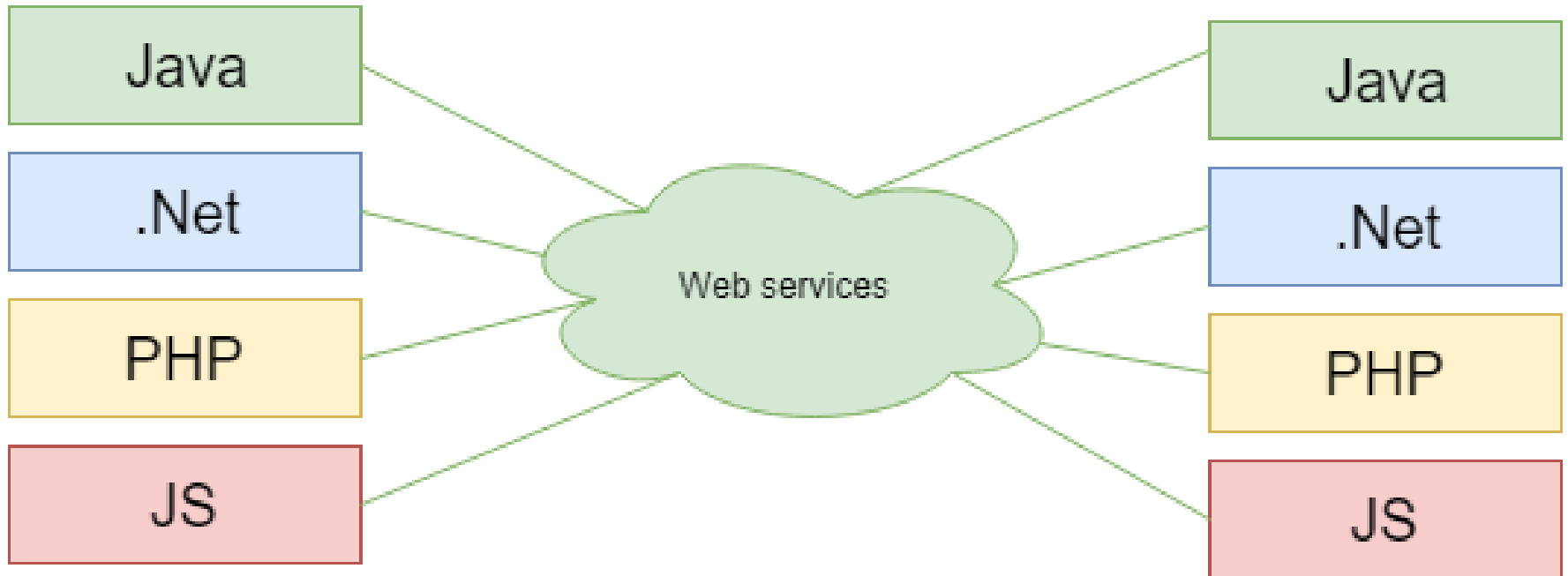


Distributed Computing Environment

1. Giới thiệu

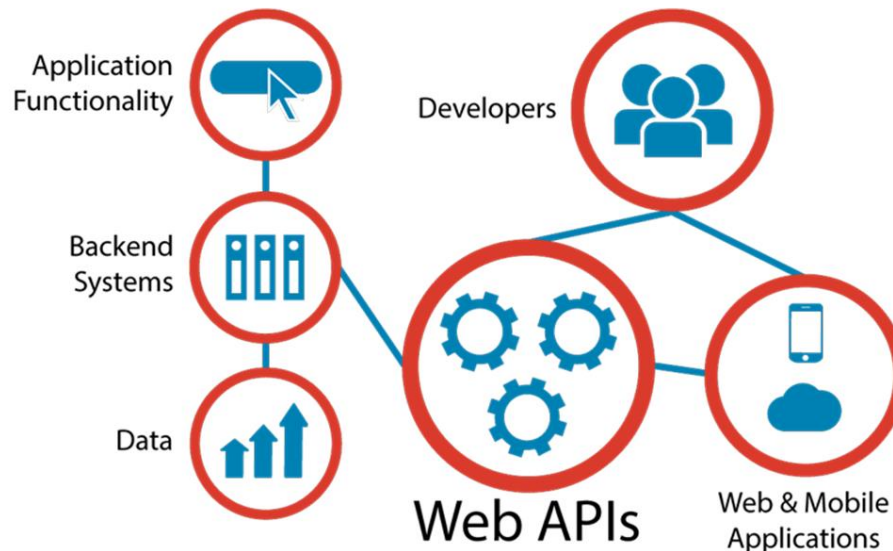
- ❖ **Web service** (dịch vụ web) là tập hợp các giao thức và tiêu chuẩn mở được sử dụng để trao đổi dữ liệu giữa các ứng dụng hoặc giữa các hệ thống.
- ❖ Các ứng dụng phần mềm được viết bằng các ngôn ngữ lập trình khác nhau hoặc chạy trên các nền tảng khác nhau, chúng có thể sử dụng các web service để trao đổi dữ liệu qua lại theo cách tương tự như liên lạc giữa các quá trình trên một máy tính.

1. Giới thiệu



2. API là gì?

- ❖ **API** - **Application Programming Interface**.
- ❖ **API** là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác.
- ❖ API cung cấp khả năng truy xuất đến một tập các hàm hay dung, và từ đó có thể trao đổi dữ liệu giữa các ứng dụng.

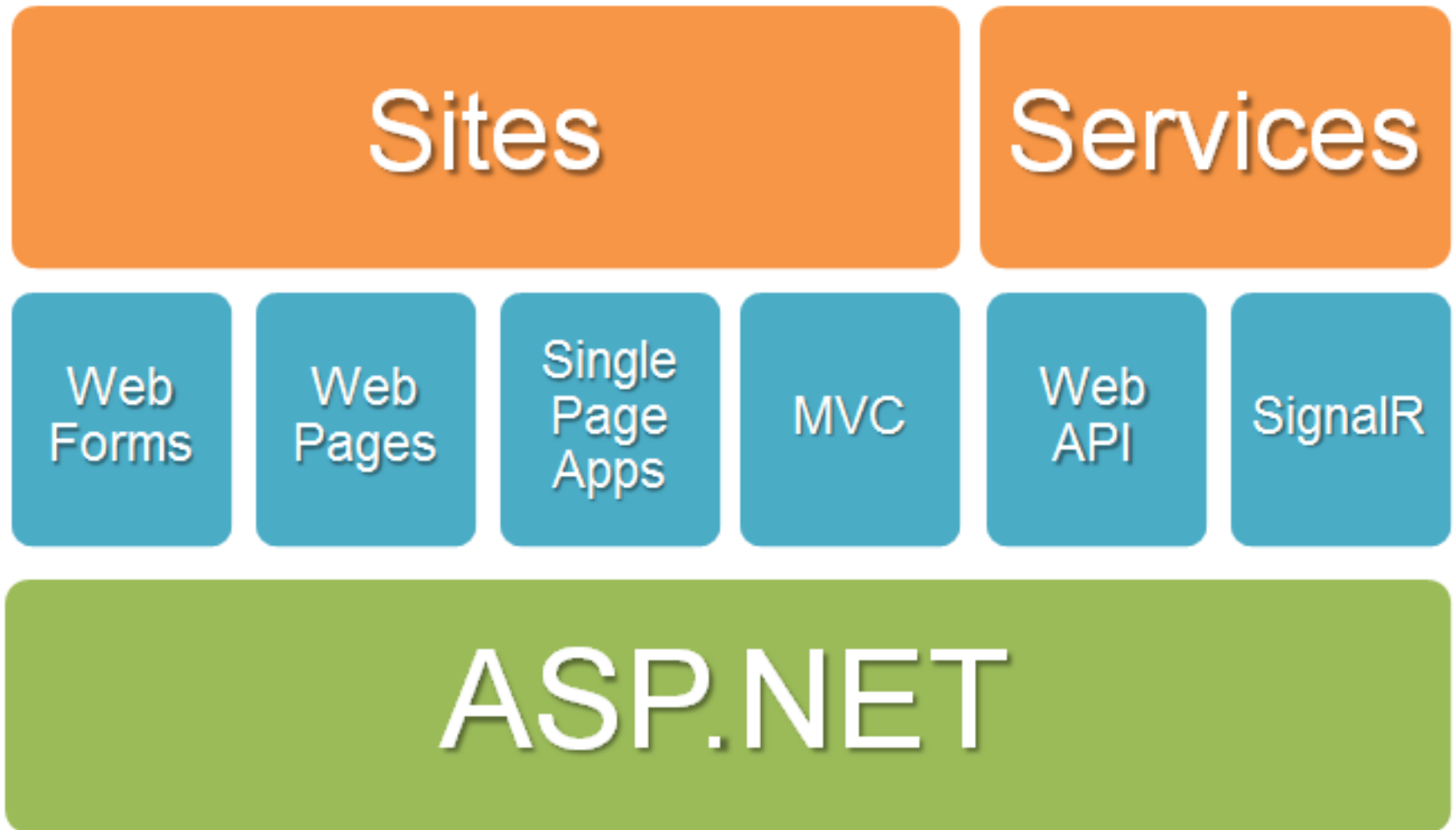


2. API là gì?

❖ API thường ứng dụng vào đâu?

- API trên hệ điều hành: Windows hay Linux có rất nhiều API, họ cung cấp các tài liệu API là đặc tả các hàm, phương thức cũng như các giao thức kết nối. Nó giúp lập trình viên có thể tạo ra các phần mềm ứng dụng có thể tương tác trực tiếp với hệ điều hành một cách dễ dàng.
- API của thư viện phần mềm hay framework: API mô tả và quy định các hành động mong muốn mà các thư viện cung cấp. Một API có thể có nhiều cách triển khai khác nhau và nó cũng giúp cho một chương trình viết bằng ngôn ngữ này có thể sử dụng thư viện được viết bằng ngôn ngữ khác.

3. Web API là gì?



3. Web API là gì?

- ❖ **Web API** là một phương thức dùng để cho phép các ứng dụng khác nhau có thể giao tiếp, trao đổi dữ liệu qua lại. Dữ liệu được Web API trả lại thường ở dạng **JSON** hoặc **XML** thông qua giao thức HTTP hoặc HTTPS.
- ❖ **Web API** là một hệ thống góp phần phát triển các ứng dụng dựa trên Restfull service trong “.Net”.

3. Web API là gì?

❖ Các đặc điểm của ASP.NET Web API:

- Là một nền tảng lý tưởng để xây dựng các dịch vụ RESTful
- Ánh xạ các hành động HTTP bởi Methods name
- Hỗ trợ các loại dữ liệu trả về khác nhau (JSON, XML,...)
- Có thể thể trữ (host) trong IIS, hoặc chính nó, hoặc các máy chủ khác có hỗ trợ .NET 4.0+
- ASP.NET Web API framework tích hợp HttpClient để giao tiếp với máy chủ Web API. HttpClient có thể được sử dụng từ phía máy chủ ASP.NET MVC, ứng dụng Windows Form, Console hoặc các ứng dụng khác.

4. Web API hoạt động như thế nào?

- ❖ Đầu tiên chúng ta phải xây dựng URL API để **bên thứ ba** có thể gửi request dữ liệu đến máy chủ cung cấp nội dung, dịch vụ thông qua giao thức HTTP hoặc HTTPS.
- ❖ Tại web server cung cấp nội dung, các ứng dụng nguồn sẽ thực hiện kiểm tra xác thực nếu có và tìm đến tài nguyên thích hợp để tạo nội dung trả về kết quả.

4. Web API hoạt động như thế nào?

- ❖ Server trả về kết quả theo định dạng **JSON** hoặc **XML** thông qua giao thức HTTP/HTTPS.
- ❖ Tại nơi yêu cầu ban đầu là ứng dụng web hoặc ứng dụng di động, dữ liệu **JSON/XML** sẽ được parse để lấy data. Sau khi có được data thì thực hiện tiếp các hoạt động như lưu dữ liệu xuống Cơ sở dữ liệu, hiển thị dữ liệu...

5. Ưu và nhược điểm của Web API

❖ Ưu điểm:

- Web API được sử dụng hầu hết trên các ứng dụng desktop, ứng dụng mobile và ứng dụng website.
- Linh hoạt với các định dạng dữ liệu khi trả về client: Json, XML hay định dạng khác.
- Nhanh chóng xây dựng HTTP service: URI, request/response headers, caching, versioning, content formats và có thể host trong ứng dụng hoặc trên IIS.

5. Ưu và nhược điểm của Web API

❖ Ưu điểm:

- Mã nguồn mở, hỗ trợ chức năng RESTful đầy đủ, sử dụng bởi bất kì client nào hỗ trợ XML, Json.
- Hỗ trợ đầy đủ các thành phần MVC như: routing, controller, action result, filter, model binder, IoC container, dependency injection, unit test.
- Giao tiếp hai chiều được xác nhận trong các giao dịch, đảm bảo độ tin cậy cao.

5. Ưu và nhược điểm của Web API

❖ **Nhược điểm:**

- Web API chưa hoàn toàn phải là RESTful service, mới chỉ hỗ trợ mặc định GET, POST.
- Để sử dụng hiệu quả cần có kiến thức chuyên sâu, có kinh nghiệm backend tốt.
- Tốn thời gian và chi phí cho việc phát triển, nâng cấp và vận hành.
- Có thể gặp vấn đề về bảo mật khi hệ thống bị tấn công nếu không giới hạn điều kiện kỹ.

6. Cách xây dựng Web API

❖ Bước 1: Tạo dự án Web API

New Project

Recent

Installed

- Visual C#
 - Get Started
 - Windows Universal
 - Windows Desktop
 - Web
 - Previous Versions
- Office/SharePoint

Not finding what you are looking for?
[Open Visual Studio Installer](#)

Sort by: Default

ASP.NET Core Web Application Visual C#

ASP.NET Web Application (.NET Framework) Visual C#

Type: Visual C#

Project templates for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.

Name: DemoWebAPI

Location: D:\ĐH GTVT\01-2023\ Browse...

Solution name: DemoWebAPI

Framework: .NET Framework 4.6.1


☒ Create directory for solution


☐ Add to Source Control


OK Cancel


6. Cách xây dựng Web API


New ASP.NET Web Application - DemoWebAPI



Empty



Web Forms


MVC


Web API


Single Page Application


Azure API App


Azure Mobile App

A project template for creating RESTful HTTP services that can reach a broad range of clients including browsers and mobile devices.

[Learn more](#)

Add folders and core references for:

☐ Web Forms ☒ MVC ☒ Web API

☐ Enable Docker Compose support (Requires [Docker for Windows](#))

☐ Add unit tests

Test project name:

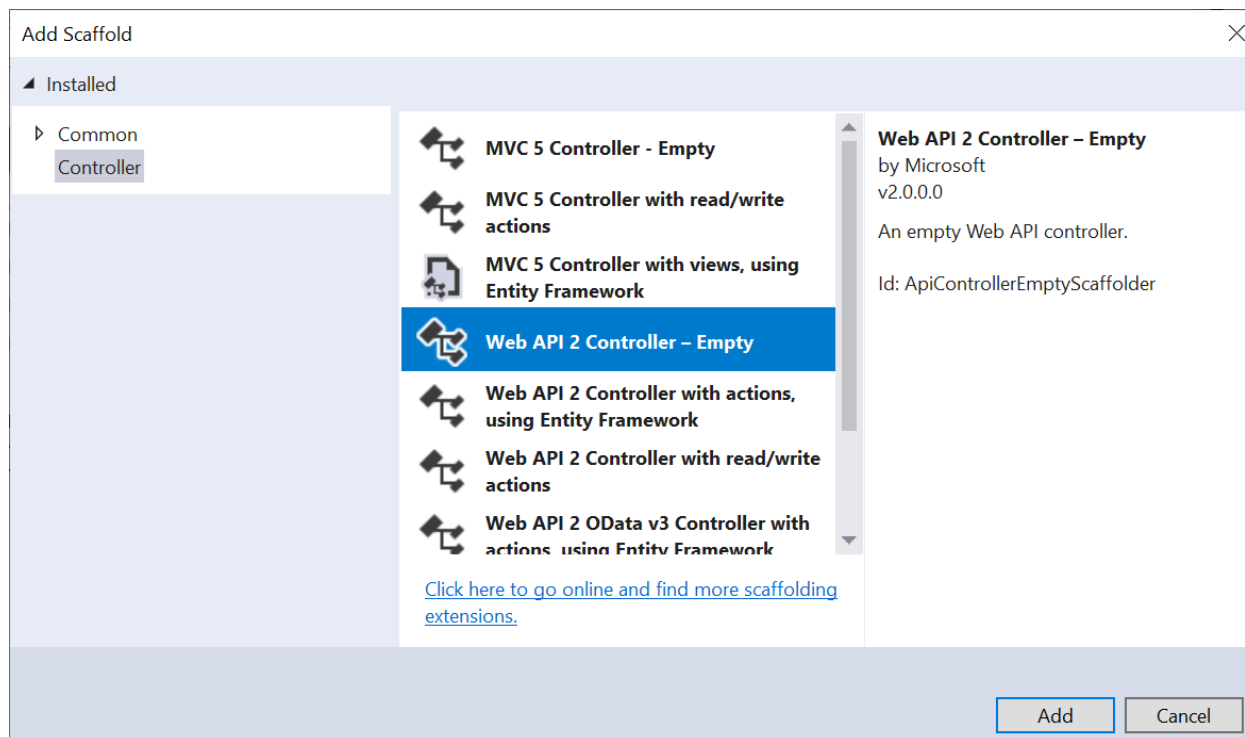
Authentication: **No Authentication**

[Change Authentication](#)

6. Cách xây dựng Web API

❖ Bước 2: Tạo Controller DemoController

- Trong Web API, Controller là nơi xử lý các yêu cầu HTTP request từ Client.



6. Cách xây dựng Web API

❖ Bước 2: Tạo Controller DemoController cung cấp 2

Action

```
namespace DemoWebAPI.Controllers
{
    public class DemoController : ApiController
    {
        [HttpGet]
        public string GetName()
        {
            return "Trần Anh Dũng";
        }
        [HttpGet]
        public int GET_Sum(int a, int b)
        {
            return (a + b);
        }
    }
}
```

6. Cách xây dựng Web API

❖ **Bước 3:** Cấu hình Root đường dẫn URL Web API.

- Để chạy một Web API hoặc gọi một Web API chúng ta cần phải cấu hình đường dẫn link URL Web API.
- Cấu hình đường dẫn link URL Web API như sau:
 - ✓ Click vào thư mục App_Start sau đó Click chọn WebApiConfig.cs
 - ✓ Thay đổi phần routeTemplate thành : routeTemplate: "api/{controller}/{action}/{id}"

6. Cách xây dựng Web API

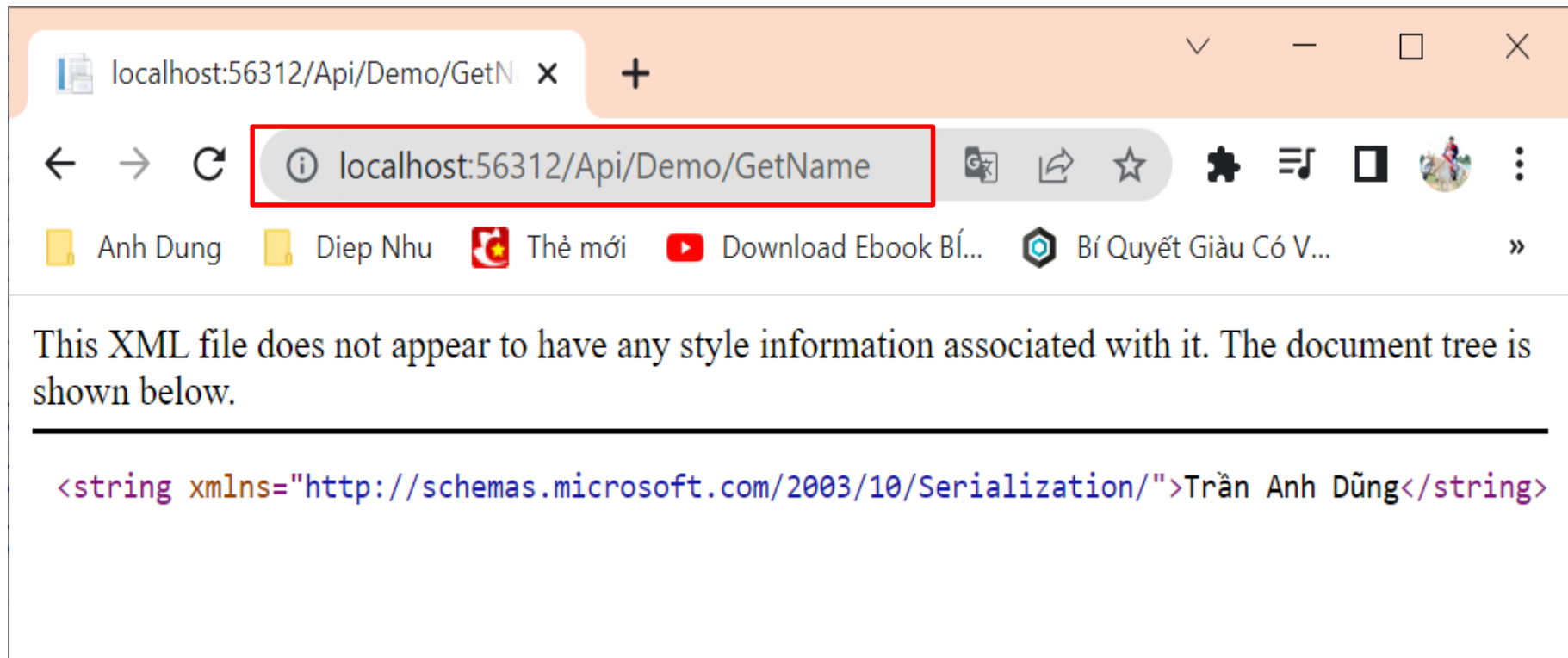
```
public static class WebApiConfig
{
    public static void Register(HttpConfiguration config)
    {
        // Web API configuration and services

        // Web API routes
        config.MapHttpAttributeRoutes();

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

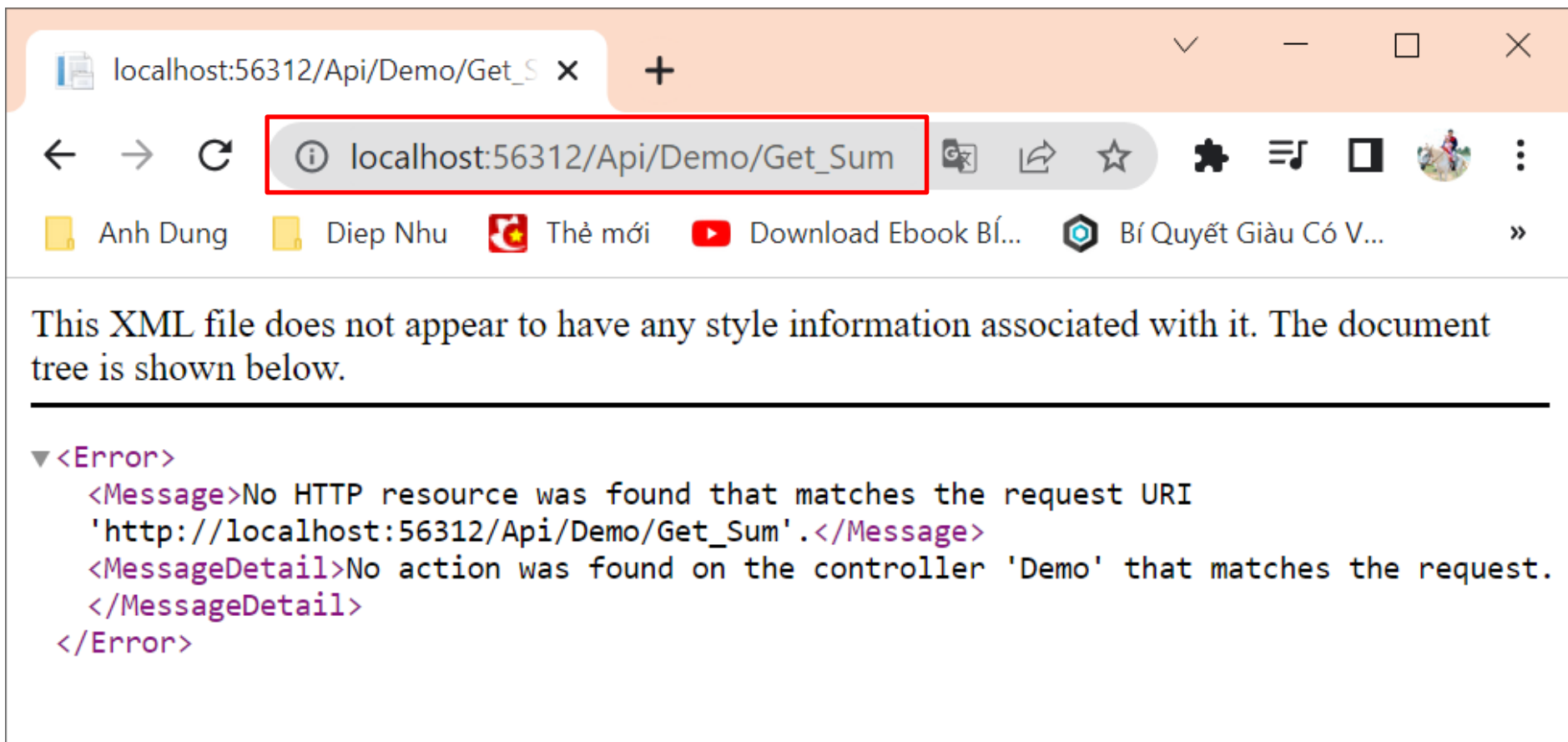
6. Cách xây dựng Web API

❖ Thực thi và xem kết quả:



6. Cách xây dựng Web API

❖ Thực thi và xem kết quả:

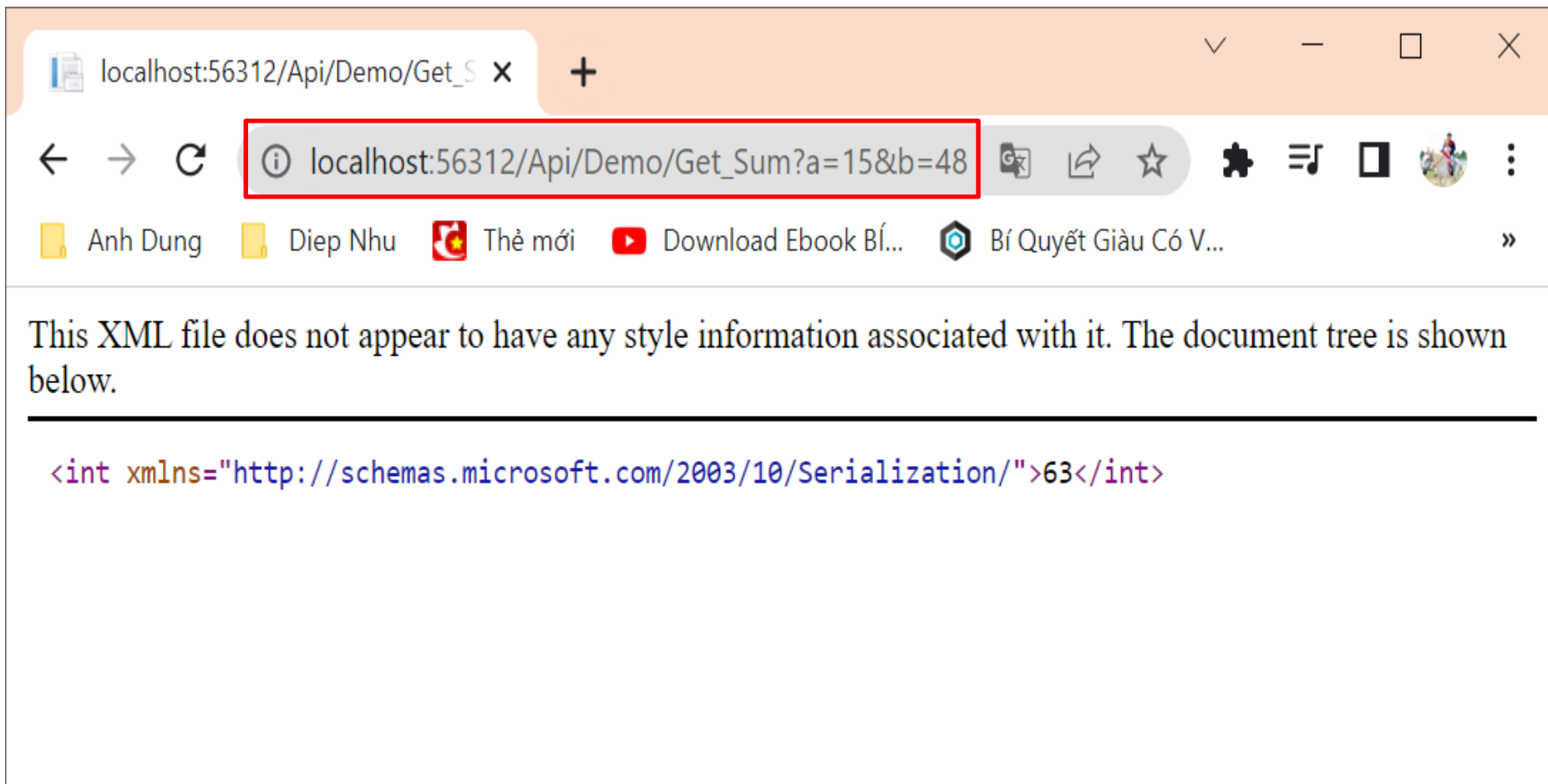


The screenshot shows a web browser window with the address bar containing `localhost:56312/Api/Demo/Get_Sum`, which is highlighted with a red rectangle. Below the address bar, there are several bookmarks including "Anh Dung", "Diep Nhu", "Thẻ mới", "Download Ebook Bí...", and "Bí Quyết Giàu Có V...". The main content area displays the message: "This XML file does not appear to have any style information associated with it. The document tree is shown below." Below this message, the XML error is displayed in a code block:

```
<?xml version="1.0" encoding="utf-8" />
<Error>
  <Message>No HTTP resource was found that matches the request URI
'http://localhost:56312/Api/Demo/Get_Sum'.</Message>
  <MessageDetail>No action was found on the controller 'Demo' that matches the request.
</MessageDetail>
</Error>
```

6. Cách xây dựng Web API

❖ Thực thi và xem kết quả:



6. Cách xây dựng Web API

- ❖ Trường hợp các kiểu dữ liệu của action không phải là các kiểu dữ liệu cơ sở?

```
[HttpGet]
public string GetName()
{
    return "Trần Anh Dũng";
}

[HttpGet]
public int GET_Sum(int a, int b)
{
    return (a + b);
}
```

- ❖ Thêm ví dụ làm việc với kiểu dữ liệu phức tạp, chẳng hạn kiểu **object**

6. Cách xây dựng Web API

❖ Tạo Model Employee:

```
public class Employee
{
    public string EmployeeID{
        get; set;
    }
    public string EmployeeName{
        get; set;
    }
    public Employee(){
    }
    public Employee(string empID, string empName){
        this.EmployeeID = empID;
        this.EmployeeName = empName;
    }
}
```

6. Cách xây dựng Web API

- ❖ Giả sử yêu cầu bài toán là lấy danh sách các nhân viên và tạo danh nhân viên thông qua Web API.
- ❖ Như vậy chúng ta cần xây dựng hàm lấy danh sách nhân viên (**GET_ALLEmployees**) và tạo danh sách nhân viên (**POST_CreateEmployees**)
- ❖ Tạo **Controller Employee** cung cấp 2 hàm trên

```
public class EmployeeController : ApiController
{
    // GET api/Employee/GET_ALLEmployees
    [HttpGet]
    public HttpResponseMessage GET_ALLEmployees()
    {
```

6. Cách xây dựng Web API

```
// GET api/Employee/GET_ALLEmployees
[HttpGet]
public HttpResponseMessage GET_ALLEmployees()
{
    List<Employee> list_employees = new List<Employee>();
    Employee emp = new Employee("E001", "Nguyễn Tấn Dũng");
    list_employees.Add(emp);
    emp = new Employee("E002", "Trần Minh Triết");
    list_employees.Add(emp);
    emp = new Employee("E003", "Nguyễn Xuân Phúc");
    list_employees.Add(emp);
    emp = new Employee("E004", "Nguyễn Thanh Phước");
    list_employees.Add(emp);
    emp = new Employee("E005", "Nguyễn Thanh Bình");
    list_employees.Add(emp);
    if (list_employees != null)
        return Request.CreateResponse(HttpStatusCode.OK, list_employees);
    else
        return Request.CreateResponse(HttpStatusCode.NotFound);
}
```

6. Cách xây dựng Web API

```
// POST api/Employee/POST_CreateEmployees
[HttpPost]
public HttpResponseMessage POST_CreateEmployees(Employee emp_Object)
{
    List<Employee> list_employees = new List<Employee>();
    for (int i = 0; i <= 5; i++)
    {
        Employee emp = new Employee();
        emp.EmployeeID = emp_Object.EmployeeID + i;
        emp.EmployeeName = emp_Object.EmployeeName + i;
        list_employees.Add(emp);
    }
    if (list_employees != null)
        return Request.CreateResponse(HttpStatusCode.OK, list_employees);
    else
        return Request.CreateResponse(HttpStatusCode.NotFound);
}
```

6. Cách xây dựng Web API

localhost:56312/Api/Employee/G x +

localhost:56312/Api/Employee/GET_ALLEmployees

Anh Dung Diep Nhu Thẻ mới Download Ebook Bí Quyết Giàu Có V... Download and Inst... Download

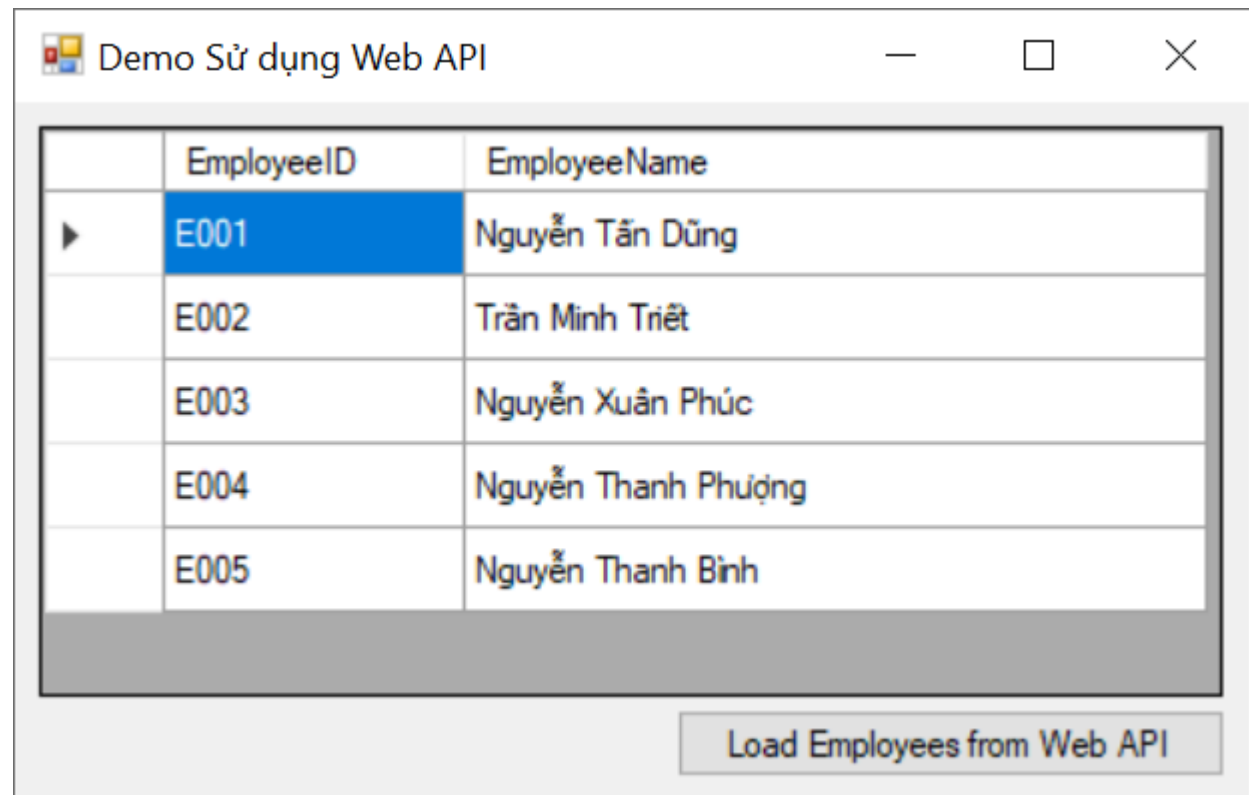
This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<ArrayOfEmployee xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://schemas.datacontract.org/2004/07/System.Data.DataContractSerializer">
  <Employee>
    <_x003C_EmployeeID_x003E_k__BackingField>E001</_x003C_EmployeeID_x003E_k__BackingField>
    <_x003C_EmployeeName_x003E_k__BackingField>Nguyễn Tấn Dũng</_x003C_EmployeeName_x003E_k__BackingField>
  </Employee>
  <Employee>
    <_x003C_EmployeeID_x003E_k__BackingField>E002</_x003C_EmployeeID_x003E_k__BackingField>
    <_x003C_EmployeeName_x003E_k__BackingField>Trần Minh Triết</_x003C_EmployeeName_x003E_k__BackingField>
  </Employee>
  <Employee>
    <_x003C_EmployeeID_x003E_k__BackingField>E003</_x003C_EmployeeID_x003E_k__BackingField>
    <_x003C_EmployeeName_x003E_k__BackingField>Nguyễn Xuân Phúc</_x003C_EmployeeName_x003E_k__BackingField>
  </Employee>
  <Employee>
    <_x003C_EmployeeID_x003E_k__BackingField>E004</_x003C_EmployeeID_x003E_k__BackingField>
    <_x003C_EmployeeName_x003E_k__BackingField>Nguyễn Thanh Phú</_x003C_EmployeeName_x003E_k__BackingField>
  </Employee>
  <Employee>
    <_x003C_EmployeeID_x003E_k__BackingField>E005</_x003C_EmployeeID_x003E_k__BackingField>
    <_x003C_EmployeeName_x003E_k__BackingField>Nguyễn Thanh Bình</_x003C_EmployeeName_x003E_k__BackingField>
  </Employee>
</ArrayOfEmployee>
```

7. Sử dụng Web API

❖ Ví dụ demo sử dụng Web API đã xây dựng ở trên vào ứng dụng **Windows Forms**

➤ Tạo ứng dụng



7. Sử dụng Web API

❖ Định nghĩa class **Employee**:

```
namespace DemWindowsFormsAPI
{
    [Serializable]
    class Employee
    {
        public string EmployeeID{ get; set; }
        public string EmployeeName{ get; set; }
        public Employee(){
        }
        public Employee(string empID, string empName){
            this.EmployeeID = empID;
            this.EmployeeName = empName;
        }
    }
}
```

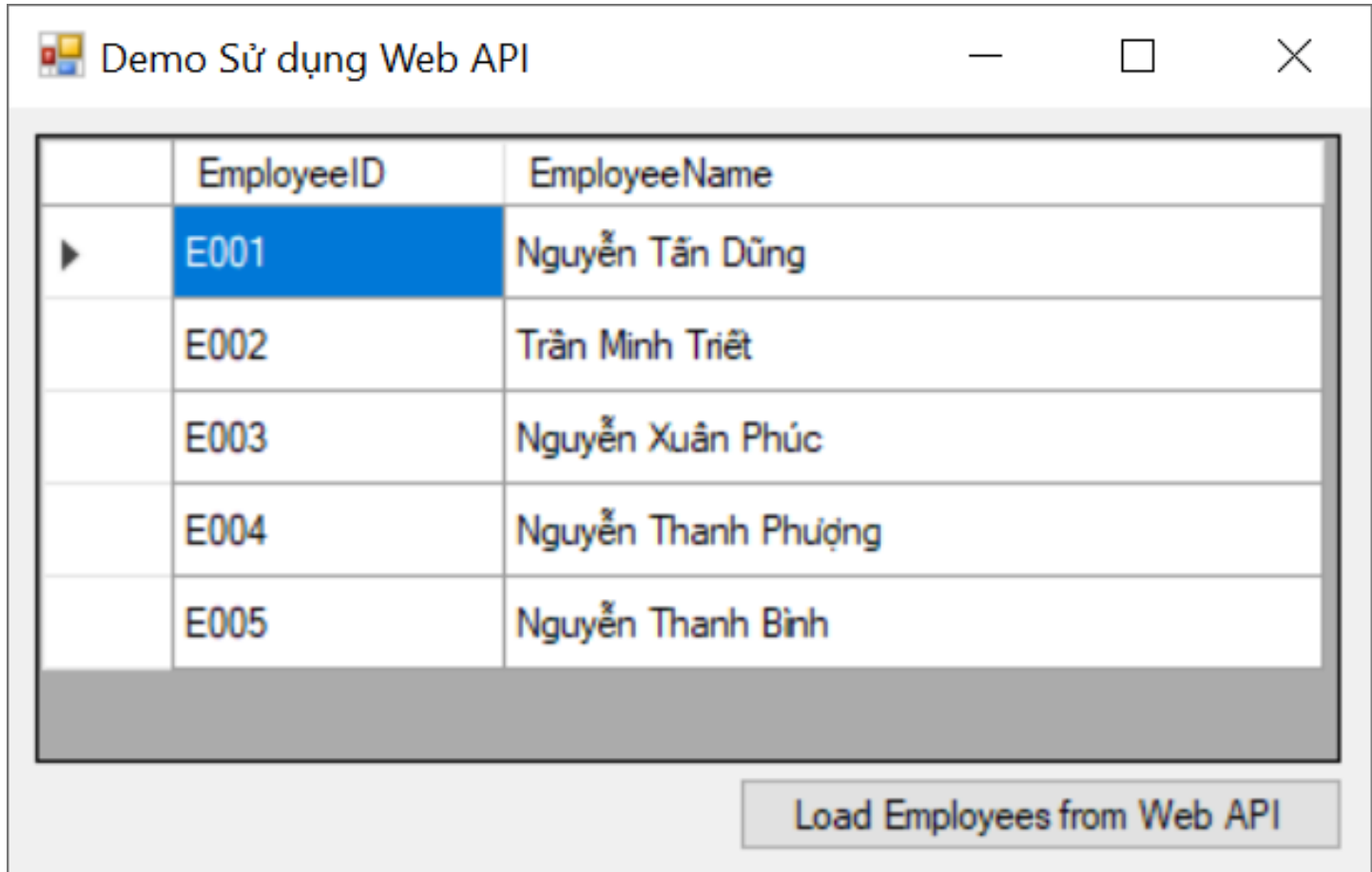

7. Sử dụng Web API

❖ Xử lý khi người dùng click button “*Load Employees from Web API*”

```
private void button1_Click(object sender, EventArgs e)
{
    string link = "http://localhost:56312/Api/Employee/GET\_ALLEmployees";
    HttpRequest req = HttpRequest.CreateHttp(link);
    WebResponse res = req.GetResponse();
    DataContractJsonSerializer js = new DataContractJsonSerializer(typeof(Employee[]));
    object data = js.ReadObject(res.GetResponseStream());
    Employee[] listEmp = (Employee[])data;
    dataGridView1.DataSource = listEmp;
}
```

7. Sử dụng Web API

- ❖ Kết quả khi chạy chương trình và get dữ liệu từ Web API:



	EmployeeID	EmployeeName
▶	E001	Nguyễn Tấn Dũng
	E002	Trần Minh Triết
	E003	Nguyễn Xuân Phúc
	E004	Nguyễn Thanh Phú
	E005	Nguyễn Thanh Bình

Load Employees from Web API

