

BÀI THỰC HÀNH SỐ 7

XÂY DỰNG WEB API

1.1 Chuẩn đầu ra của học phần

L4. Phát triển được ứng dụng phần mềm, có sử dụng kỹ thuật Web API để tích hợp hệ thống.

1.2 Mục tiêu bài học

Sau bài học này sinh viên có khả năng sử dụng LINQ để tạo Web API với HttpGet, HttpPut, HttpPost và HttpDelete.

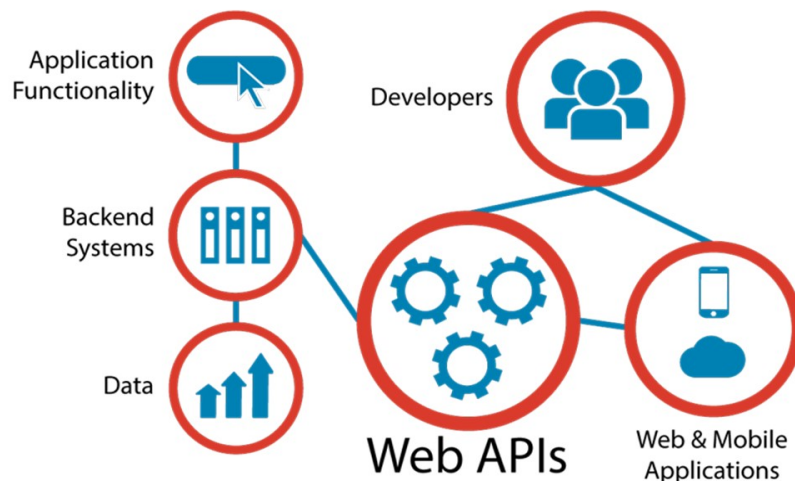
A. Cơ sở lý thuyết

1. Giới thiệu về Web API

Hiện nay API nói chung và Web API nói riêng đang được ứng dụng ngày càng nhiều. Kiến trúc ứng dụng hiện đại ngày nay ngày càng phân tán, không phụ thuộc ngôn ngữ đã thúc đẩy việc ứng dụng API. Vậy **API là gì?** Nguồn gốc và ưu điểm của nó là như thế nào?

API là gì?

API là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. Nó là viết tắt của **Application Programming Interface** – giao diện lập trình ứng dụng. API cung cấp khả năng cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó có thể trao đổi dữ liệu giữa các ứng dụng.



Hình 7.1: Mô hình hoạt động

API thường ứng dụng vào đâu?

- Web API: là hệ thống API được sử dụng trong các hệ thống website. Hầu hết các website đều ứng dụng đến Web API cho phép bạn kết nối, lấy dữ liệu hoặc cập nhật cơ sở dữ liệu. Ví dụ: Bạn thiết kế chức năng login thông Google, Facebook, Twitter, Github... Điều này có nghĩa là bạn đang gọi đến API của. Hoặc như các ứng dụng di động đều lấy dữ liệu thông qua API.
- 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.
- 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. Ví dụ bạn có thể dùng Php để yêu cầu một thư viện tạo file PDF được viết bằng C++.
- Một số khái niệm khác:
 - API hiện nay đều tuân thủ theo tiêu chuẩn REST và HTTP, tạo sự thân thiện để sử dụng với nhà phát triển. Giúp người dùng dễ dàng truy cập, dễ hiểu hơn. Web API hiện đại dùng cho các đối tượng cụ thể, chẳng hạn như mobile developer với document, version khác nhau.
 - API key: Đây là loại code (string) được truyền tải bởi các chương trình máy tính gọi là API để xác định chương trình, nhà phát triển hoặc người dùng nó tới trang web. Các API key được sử dụng với mục đích nhằm giới hạn, kiểm soát sử dụng API. Chẳng hạn như ngăn chặn sự việc lạm dụng API.

API Key thường hoạt động như một mã định danh duy nhất và mã thông báo bí mật để xác thực và thường sẽ có một bộ quyền truy cập trên API được liên kết với nó. Các API Key có thể dựa trên hệ thống định danh duy nhất toàn cầu (UUID) để đảm bảo chúng sẽ là duy nhất cho mỗi người dùng.

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.

Những điểm nổi bật của Web API

Web API hỗ trợ restful đầy đủ các phương thức: Get/Post/put/delete dữ liệu. Nó giúp bạn xây dựng các HTTP service một cách rất đơn giản và nhanh chóng. Nó cũng có khả năng hỗ trợ đầy đủ các thành phần HTTP: URI, request/response headers, caching, versioning, content format.

Tự động hóa sản phẩm: Với **web API**, chúng ta sẽ tự động hóa quản lý công việc, cập nhật luồng công việc, giúp tăng năng suất và tạo hiệu quả công việc cao hơn.

Khả năng tích hợp linh động: API cho phép lấy nội dung từ bất kỳ website hoặc ứng dụng nào một cách dễ dàng nếu được cho phép, tăng trải nghiệm người dùng. API hoạt động như một chiếc cổng, cho phép các công ty chia sẻ thông tin được chọn nhưng vẫn tránh được những yêu cầu không mong muốn.

Cập nhật thông tin thời gian thực: API có chức năng thay đổi và cập nhật thay đổi theo thời gian thực. Với công nghệ này, dữ liệu sẽ được truyền đi tốt hơn, thông tin chính xác hơn, dịch vụ cung cấp linh hoạt hơn.

Có tiêu chuẩn chung dễ sử dụng: Bất kỳ người dùng, công ty nào sử dụng cũng có thể điều chỉnh nội dung, dịch vụ mà họ sử dụng.

Hỗ trợ đầy đủ các thành phần MVC như: routing, controller, action result, filter, model binder, IoC container, dependency injection, unit test.

Web API hoạt động như thế nào?

- Đầu tiên là 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.
- 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...

Ưu và nhược điểm của Web API

Mỗi một ứng dụng bất kỳ đều có những ưu nhược điểm riêng, hỗ trợ tốt cho các ứng dụng. Vì vậy mà web API cũng không ngoại lệ:

❖ 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.
- 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.

❖ Nhược điểm

Do web API còn khá mới nên chưa thể đánh giá nhiều về nhược điểm của mô hình này. Tuy nhiên, có hai nhược điểm dễ dàng nhận thấy:

- 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.

2. Các bước xây dựng Web API

Bước 1:

Khởi động Visual Studio → tạo một project ASP.NET Web Application và chọn template Web API.

Bước 2:

Tạo Data Model sử dụng Entity Framework để Web API service có thể tương tác CRUD (Create, Read, Update, Delete) với dữ liệu được, mỗi bảng là 1 Controller.

Bước 3: Tạo ra các Web API

Nhập chuột phải vào thư mục Controllers và chọn thêm controller.

- Web API 2 Controller Empty: tự viết các phương thức từ đầu.
- Web API 2 Controller with read/write actions: phát sinh các phương thức ví dụ để bạn có thể biết cách viết các service này.

Bước 4: Chạy thử và kiểm tra.

B. Bài thực hành mẫu

Trước tiên ta tạo cơ sở dữ liệu **CSDLTest** gồm 2 bảng:

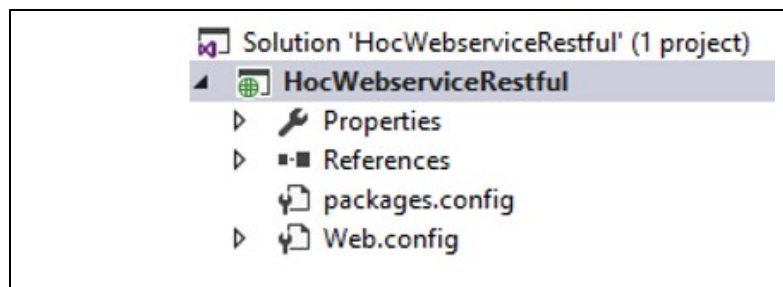
+ DanhMuc(**MaDanhMuc**, TenDanhMuc)

+ SanPham(**Ma**, Ten, DonGia, MaDanhMuc)

Nhập dữ liệu cho 2 bảng

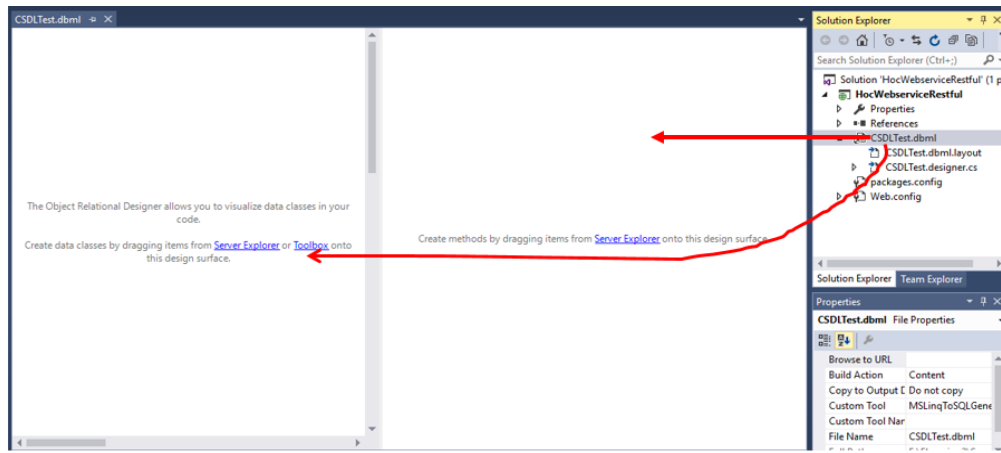
Bước 1: Khởi động Visual Studio/ vào File/ New/ chọn Project, Chọn Asp.net Web Application, chọn Empty.

Cấu trúc Project sau khi tạo thành công:

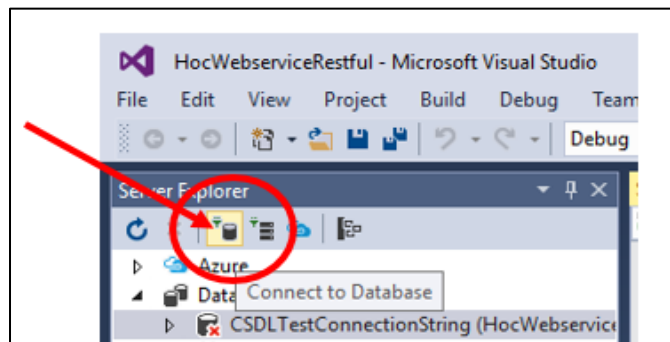


Bước 2: Tiến hành kết nối CSDL: bấm chuột phải vào Project, chọn Add/Linq to SQL Classes.

Đặt tên file LinQ, trong trường hợp này ta nên đặt cùng tên với CSDL: CSDLTest



- Vào Server Explore để kết nối CSDL



- Kéo thả các bảng dữ liệu vào LinQ

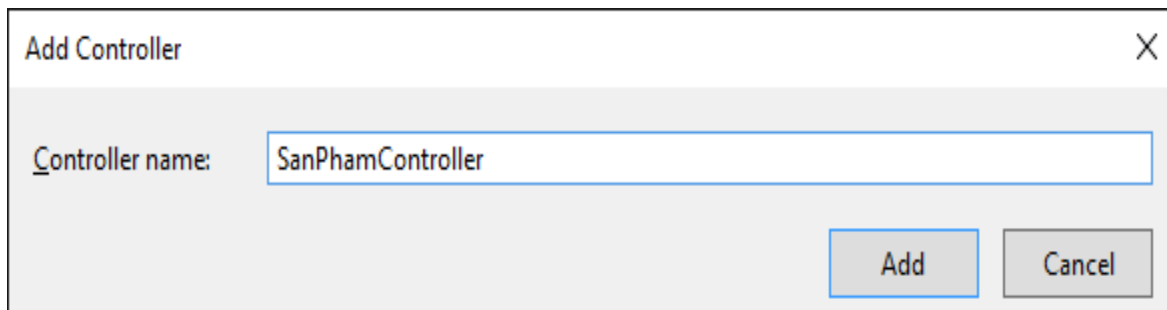
Cơ chế ORM
 (Object Relational Mapping)

Tên bảng → Tên lớp
 Tên cột → Thuộc tính
 Từng dòng dữ liệu → tập các đối tượng

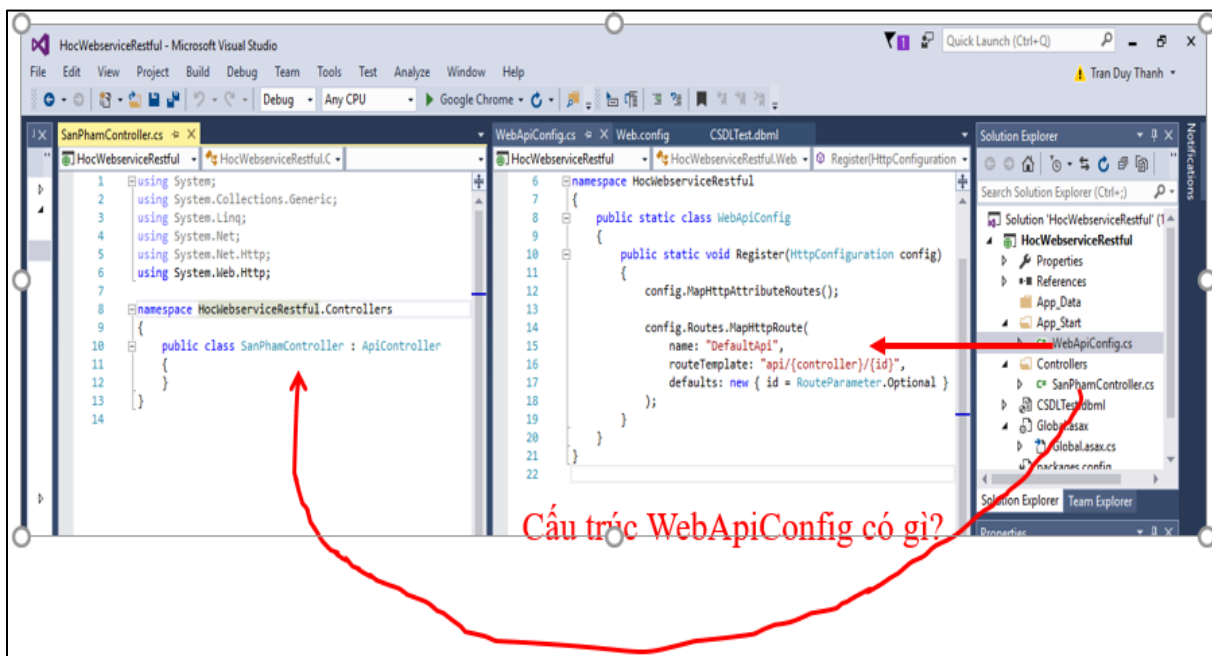
- Tạo 1 thư mục Controller.
- Muốn tương tác Bảng nào trong CSDL thì tạo từng đó Controller (mỗi bảng là 1 Controller độc lập)

- Chọn Web API 2 Controller – Empty rồi nhấn Add

Mặc định là chữ Default, bây giờ bạn đổi lại thành SanPham (thường ta làm API cho bảng nào thì lấy tên bảng đó), đó là lý do vì sao ta đặt là SanPham:



- Tập tin SanPhamController.cs sẽ được tạo ra, nhưng ta chỉ lấy SanPham(bỏ chữ Controller đằng sau đi) để tương tác (đây là cơ chế hoạt động). Ta xem cấu trúc Controller được tạo ra:



- File **WebApiConfig.cs** được sinh ra trong thư mục **App_Start**, routeTemplate: “api/{controller}/{id}”, tức là khi ta dùng thì viết: “api/sanpham” để lấy toàn bộ danh sách, hay “api/sanpham/3” để lấy chi tiết 1 Sản phẩm có mã là 3.
- File **SanPhamController.cs** kế thừa từ ApiController


```

public static class WebApiConfig
{
    public static void Register(HttpConfiguration config)
    {
        config.MapHttpAttributeRoutes();

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

```

Tương tự như tạo Controller cho bảng DanhMuc hay bất kỳ bảng nào khác

Bước 3: Tạo các Web API

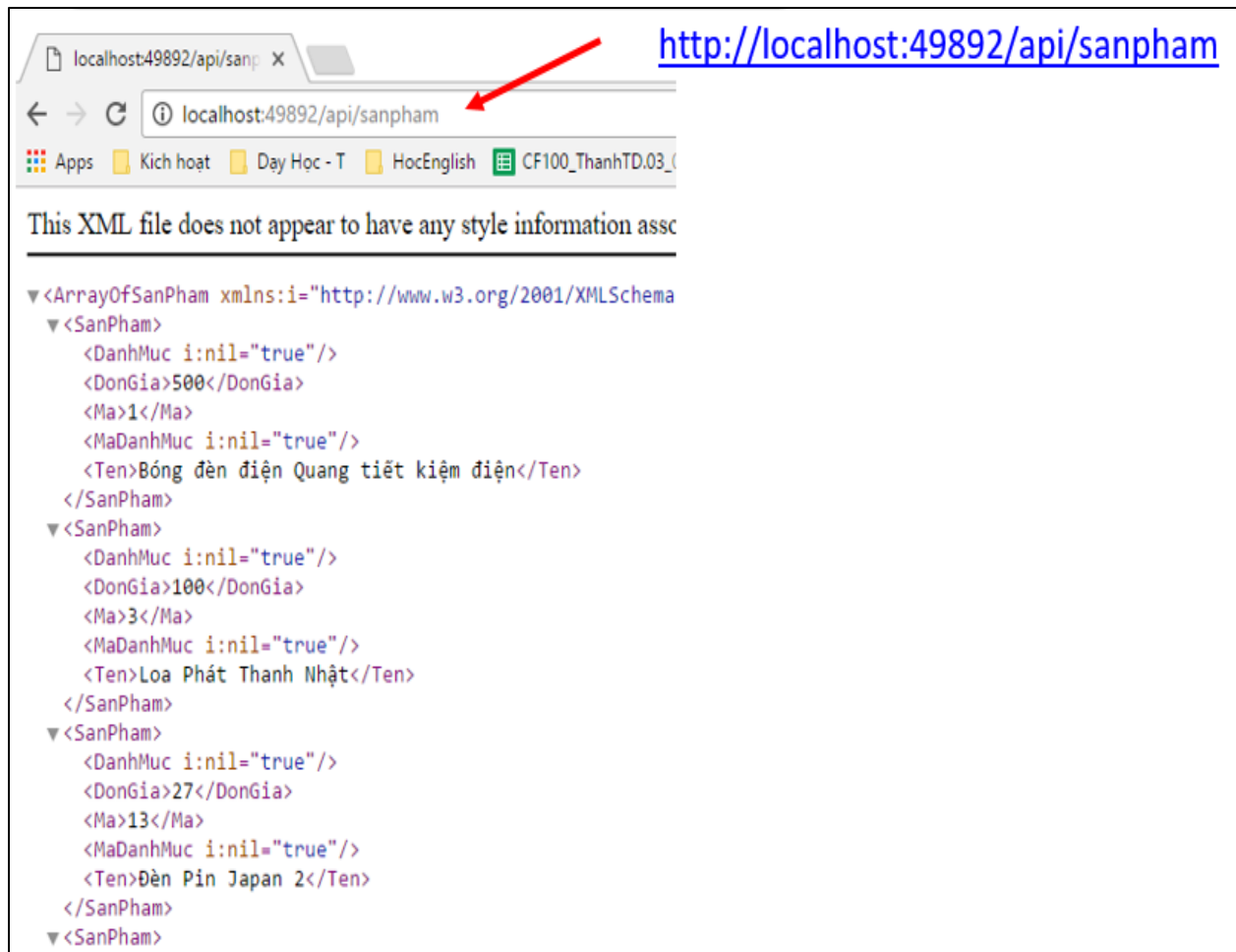
❖ Tạo Web API với HttpGet

+ Tạo Web API lấy toàn bộ danh sách Sản phẩm

```

[HttpGet]
public List<SanPham> LayToanBoSanPham()
{
    CSDLTestDataContext context = new CSDLTestDataContext();
    List<SanPham> dsSP = context.SanPhams.ToList();
    foreach (SanPham sp in dsSP)
        sp.DanhMuc = null;
    return dsSP;
}

```



+ Tạo Web API lấy chi tiết 1 Sản phẩm

```
[HttpGet]
public SanPham ChiTietSanPham(int id)
{
    CSDLTestDataContext context = new CSDLTestDataContext();
    SanPham sp = context.SanPhams
        .FirstOrDefault(x => x.Ma == id);

    if(sp!=null)
        sp.DanhMuc = null;
    return sp;
}
```

localhost:49892/api/sanpham/1

localhost:49892/api/sanpham/1

Apps Kịch hoạt Day Học - T HocEnglish CF100_1

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

<SanPham xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
 <DanhMuc i:nil="true"/>
 <DonGia>500</DonGia>
 <Ma>1</Ma>
 <MaDanhMuc i:nil="true"/>
 <Ten>Bóng đèn điện Quang tiết kiệm điện</Ten>
</SanPham>

<http://localhost:49892/api/sanpham/1>

localhost:49892/api/sanpham/?id=1

localhost:49892/api/sanpham/?id=1

Apps Kịch hoạt Day Học - T HocEnglish CF100_1

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

<SanPham xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
 <DanhMuc i:nil="true"/>
 <DonGia>500</DonGia>
 <Ma>1</Ma>
 <MaDanhMuc i:nil="true"/>
 <Ten>Bóng đèn điện Quang tiết kiệm điện</Ten>
</SanPham>

<http://localhost:49892/api/sanpham/?id=1>

❖ Tạo Web API với HttpPost

+ Tạo Web API lưu một sản phẩm: SanPhamController.cs

```
[HttpPost]
public bool LuuSanPham(int masp,string tensp,int dongia,int madm)
{
    try
    {
        CSDLTestDataContext context = new CSDLTestDataContext();
        SanPham sp = new SanPham();
        sp.Ma = masp;
        sp.Ten = tensp;
        sp.DonGia = dongia;
        sp.MaDanhMuc = madm;
        context.SanPhams.InsertOnSubmit(sp);
        context.SubmitChanges();
        return true;
    }
    catch { }
    return false;
}
```

❖ Tạo Web API với HttpPut

+ Sửa một sản phẩm: SanPhamController.cs

```
[HttpPut]
public bool SuaSanPham(int masp, string tensp, int dongia)
{
    try
    {
        CSDLTestDataContext context = new CSDLTestDataContext();
        SanPham sp = context.SanPhams
                               .FirstOrDefault(x => x.Ma == masp);
        if (sp != null)
        {
            sp.Ten = tensp;
            sp.DonGia = dongia;
            context.SubmitChanges();
            return true;
        }
    }
    catch { }
    return false;
}
```

❖ Tạo Web API với HttpDelete

+ Xóa một sản phẩm: SanPhamController.cs

```

[HttpDelete]
public bool XoaSanPham(int masp)
{
    CSDLTestDataContext context = new CSDLTestDataContext();
    SanPham sp = context.SanPhams
                        .FirstOrDefault(x => x.Ma == masp);
    if (sp != null)
    {
        context.SanPhams.DeleteOnSubmit(sp);
        context.SubmitChanges();
        return true;
    }
    return false;
}

```

C. Bài tập thực hành

Bài tập 1: Tạo cơ sở dữ liệu **CSDLQLBanHang** gồm 2 bảng:

+ DanhMuc(**MaDanhMuc**, TenDanhMuc)

+ SanPham(**Ma**, Ten, DonGia, MaDanhMuc)

Nhập dữ liệu cho 2 bảng

1.1 Dùng HttpGet tạo các Web API để:

- ✓ Lấy toàn bộ danh mục sản phẩm
- ✓ Lấy chi tiết 1 một danh mục sản phẩm

1.2 Dùng HttpPost tạo Web API để lưu 1 danh mục.

1.3 Dùng HttpPut tạo Web API để sửa 1 danh mục.

1.4 Dùng HttpDelete tạo Web API để xóa 1 danh mục.

Bài tập 2: Tạo cơ sở dữ liệu **QLLuong** gồm 2 bảng:

+ DonVi(**MaDonVi**, TenDonVi)

+ NhanVien(**Ma**, HoTen, NgaySinh, GioiTinh, Hsluong, MaDonVi)

Nhập dữ liệu cho 2 bảng

2.1 Dùng HttpGet tạo các Web API để:

- ✓ Lấy toàn bộ danh sách nhân viên
- ✓ Lấy chi tiết 1 nhân viên
- ✓ Lấy danh sách nhân viên theo đơn vị
- ✓ Lấy danh sách nhân viên theo giới tính
- ✓ Tìm danh sách nhân viên có hệ số lương trong khoảng [a ... b]
- ✓ Lấy toàn bộ danh sách đơn vị
- ✓ Lấy chi tiết 1 đơn vị

2.2 Dùng HttpPost tạo các Web API để:

- ✓ Lưu một đơn vị (DonViController.cs)
- ✓ Lưu một nhân viên (NhanVienController.cs)

2.3 Dùng HttpPut tạo các Web API để:

- ✓ Sửa một nhân viên (NhanVienController.cs)
- ✓ Sửa một đơn vị (DonviController.cs)

2.4 Dùng HttpDelete tạo các Web API để:

- ✓ Xóa một nhân viên (NhanVienController.cs)
- ✓ Xóa một đơn vị (DonviController.cs)

BÀI THỰC HÀNH SỐ 8

XÂY DỰNG ỨNG DỤNG TRIỆU GỌI WEB API

1.1 Chuẩn đầu ra của học phần

L4. Phát triển được ứng dụng phần mềm, có sử dụng kỹ thuật Web API để tích hợp hệ thống.

1.2 Mục tiêu bài học

Sau bài học này sinh viên có khả năng xây dựng ứng dụng triệu gọi các Web API với HttpGet, HttpPut, HttpPost và HttpDelete.

A. Cơ sở lý thuyết

1. Cài đặt Web API lên IIS Web Server

Bước 1: Cấu hình Application Pools:

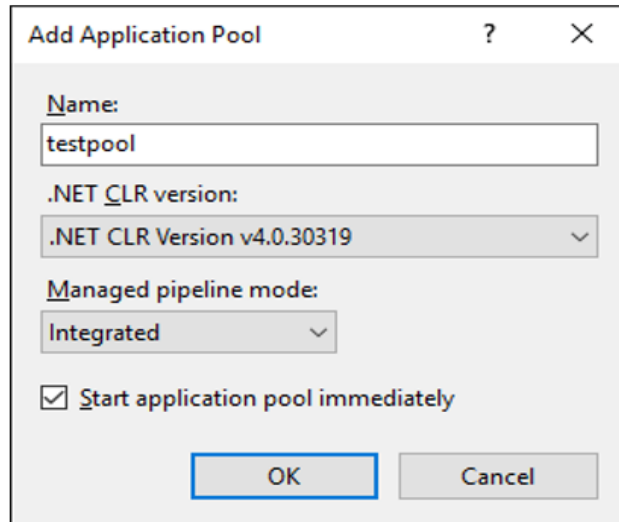
Application Pool là gì?

Application Pool có thể chứa một hoặc nhiều ứng dụng và cho phép chúng ta cấu hình cấp độ giữa các ứng dụng web khác nhau. Ví dụ, nếu bạn muốn cô lập tất cả các ứng dụng web chạy trong cùng một máy, bạn có thể làm điều này bằng cách tạo ra Application Pool riêng biệt cho mỗi ứng dụng web và đặt chúng trong Application Pool tương ứng. Bởi vì mỗi Application Pool chạy trong quá trình làm việc riêng của mình, các lỗi trong Application Pool sẽ không ảnh hưởng đến các ứng dụng đang chạy trong Application Pool khác. Triển khai ứng dụng trong Application Pool là lợi thế chính của IIS trong quá trình làm việc ở chế độ cách ly bởi vì bạn có thể tùy chỉnh Application Pool để đạt được cấp độ tách biệt ứng dụng mà bạn cần.

Khi bạn cấu hình Application Pool để sẵn sàng tối ưu, bạn cũng nên xem xét làm thế nào để cấu hình Application Pool bảo mật ứng dụng. Ví dụ, bạn có thể cần phải tạo ra Application Pool riêng cho ứng dụng đòi hỏi mức độ bảo mật cao, trong khi cho phép các ứng dụng đòi hỏi một mức độ thấp hơn của bảo mật để chia sẻ cùng Application Pool.

Bấm chuột phải vào màn hình Application Pools/ chọn Add Application Pool...

Màn hình tạo mới Application Pool hiện lên, ta nhập Testpool rồi bấm OK:



The image shows the 'Add Application Pool' dialog box. It has a title bar with a question mark and a close button. The 'Name' field contains 'testpool'. The '.NET CLR version' dropdown is set to '.NET CLR Version v4.0.30319'. The 'Managed pipeline mode' dropdown is set to 'Integrated'. There is a checked checkbox for 'Start application pool immediately'. At the bottom are 'OK' and 'Cancel' buttons.

Add Application Pool

Name: testpool

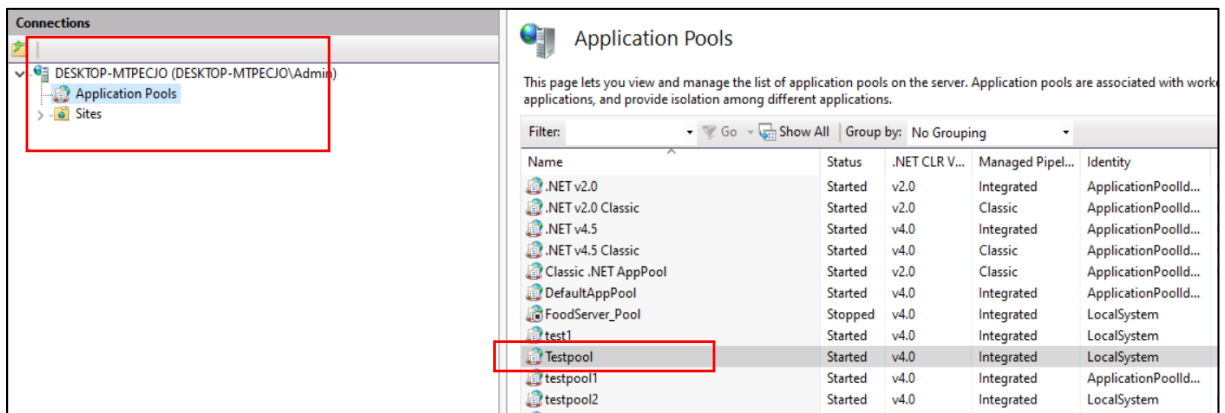
.NET CLR version: .NET CLR Version v4.0.30319

Managed pipeline mode: Integrated

☒ Start application pool immediately

OK Cancel

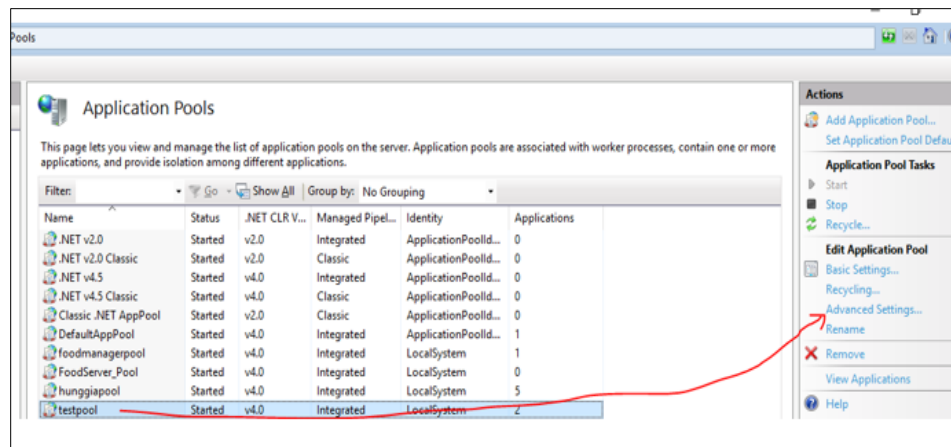
Sau khi bấm OK:



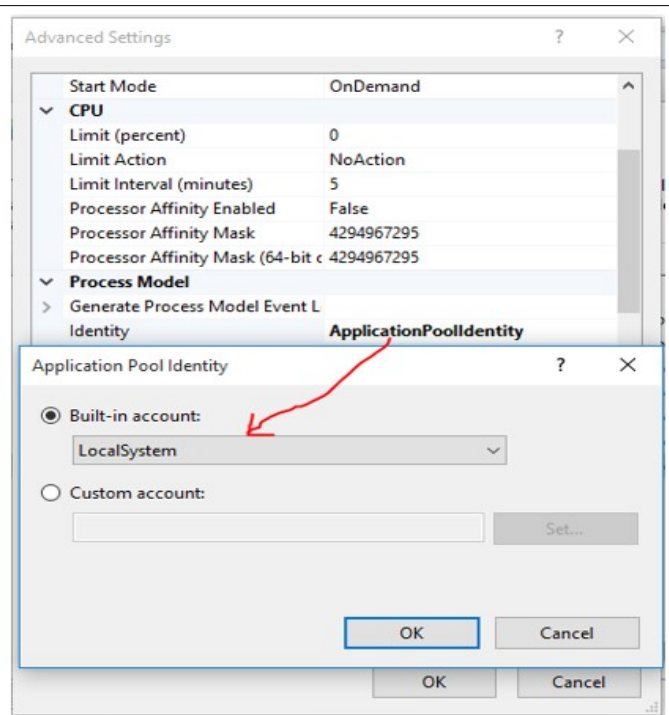
The image shows the 'Application Pools' management console. On the left, the 'Connections' pane shows the tree structure with 'Application Pools' selected. On the right, a table lists the application pools. The 'testpool' entry is highlighted with a red box.

Name	Status	.NET CLR V...	Managed Pipel...	Identity
.NET v2.0	Started	v2.0	Integrated	ApplicationPoolId...
.NET v2.0 Classic	Started	v2.0	Classic	ApplicationPoolId...
.NET v4.5	Started	v4.0	Integrated	ApplicationPoolId...
.NET v4.5 Classic	Started	v4.0	Classic	ApplicationPoolId...
Classic .NET AppPool	Started	v2.0	Classic	ApplicationPoolId...
DefaultAppPool	Started	v4.0	Integrated	ApplicationPoolId...
FoodServer_Pool	Stopped	v4.0	Integrated	LocalSystem
test1	Started	v4.0	Integrated	LocalSystem
testpool	Started	v4.0	Integrated	LocalSystem
testpool1	Started	v4.0	Integrated	ApplicationPoolId...
testpool2	Started	v4.0	Integrated	LocalSystem

Ở màn hình trên, ta chọn Advanced Settings... cho testpool:

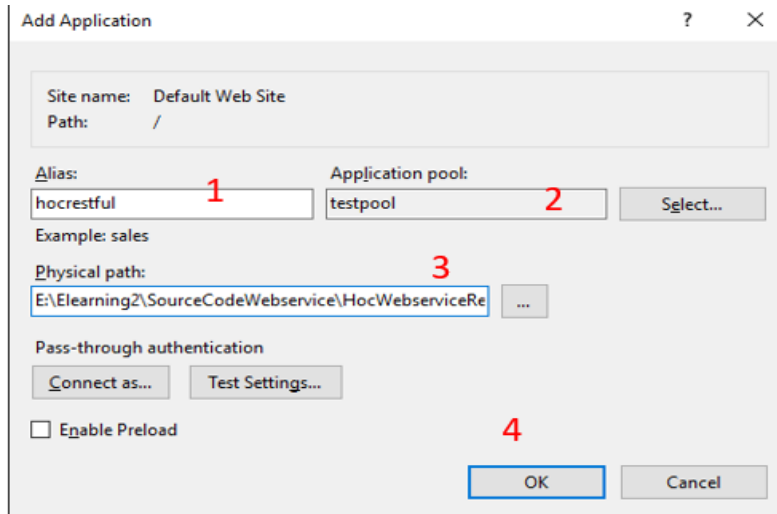


Bạn cần chỉnh
ApplicationPoolIdentity qua
LocalSystem.



Bước 2: Cấu hình Web API lên IIS Server

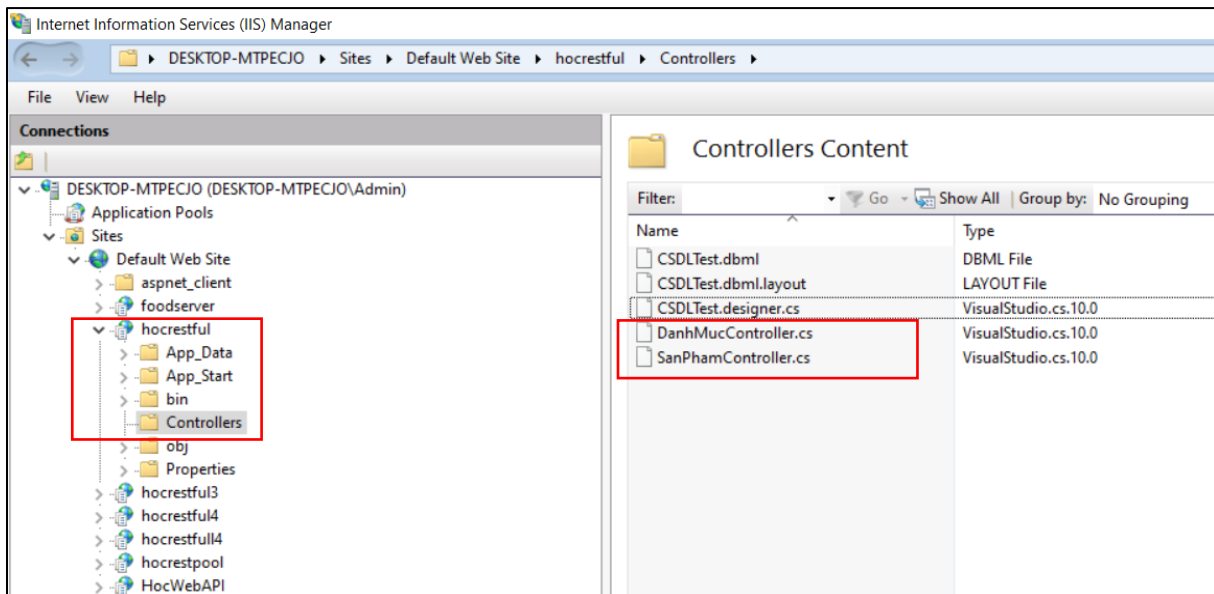
Bấm chuột vào Default Web Site/ chọn Add Application...



Mục Alias(1): Đặt tên cho Service, ở đây ta đặt là **hocrestfull**

Mục Application pool (2): nhấn nút Select... và chọn đúng **testpool**

Mục Physical path (3): Trở tới đường dẫn chứa source code của project tạo Web API, sau đó ấn OK.



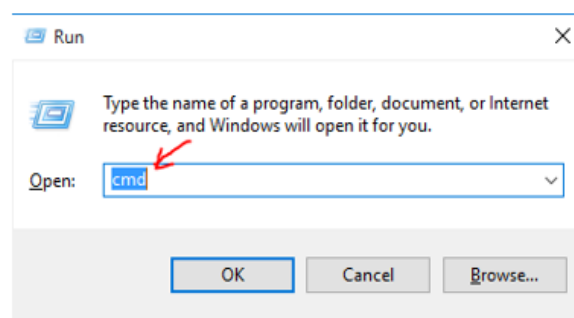
Vào trình duyệt gõ:

<http://localhost/hocrestful/api/sanpham>

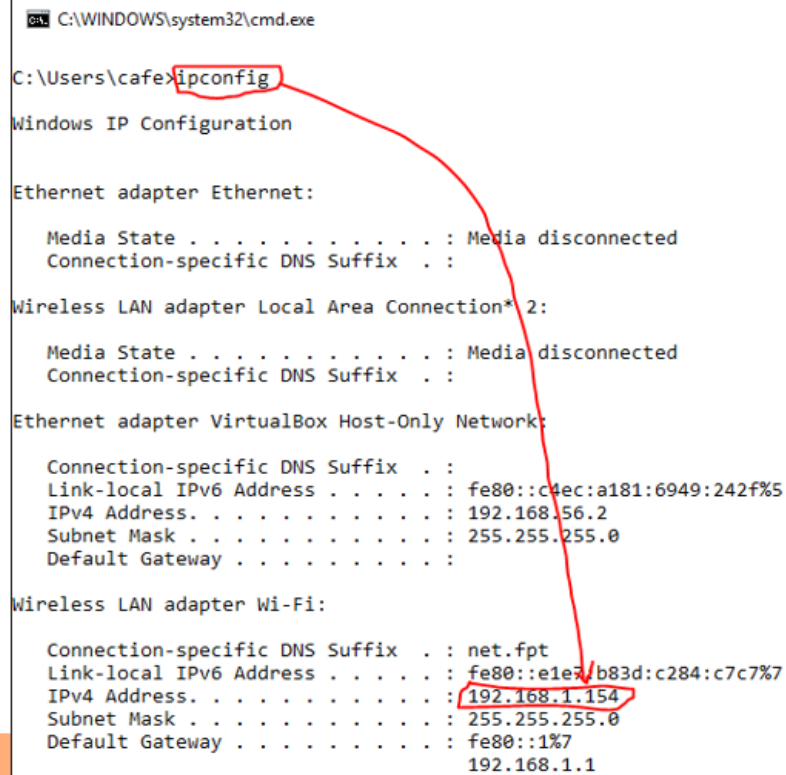
Chú ý rằng khi từ Mobile kết nối tới Web Service này thì không thể dùng localhost (vì nó làm tưởng là của chính mobile), do đó ta cần đưa localhost về 1 địa chỉ IP để test.

Cách lấy địa chỉ IP máy local của ta:

- Nhấn tổ hợp phím có biểu tượng Windows + R để mở cửa sổ Run:



– gõ lệnh **cmd**, màn hình command line hiển thị lên, tiếp tục gõ lệnh **ipconfig**



```
C:\WINDOWS\system32\cmd.exe

C:\Users\cafe>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Ethernet adapter VirtualBox Host-Only Network:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::c4ec:a181:6949:242f%5
    IPv4 Address. . . . . : 192.168.56.2
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :

Wireless LAN adapter Wi-Fi:

    Connection-specific DNS Suffix  . : net.fpt
    Link-local IPv6 Address . . . . . : fe80::e1e7:b83d:c284:c7c7%7
    IPv4 Address. . . . . : 192.168.1.154
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : fe80::1%7
                                192.168.1.1
```

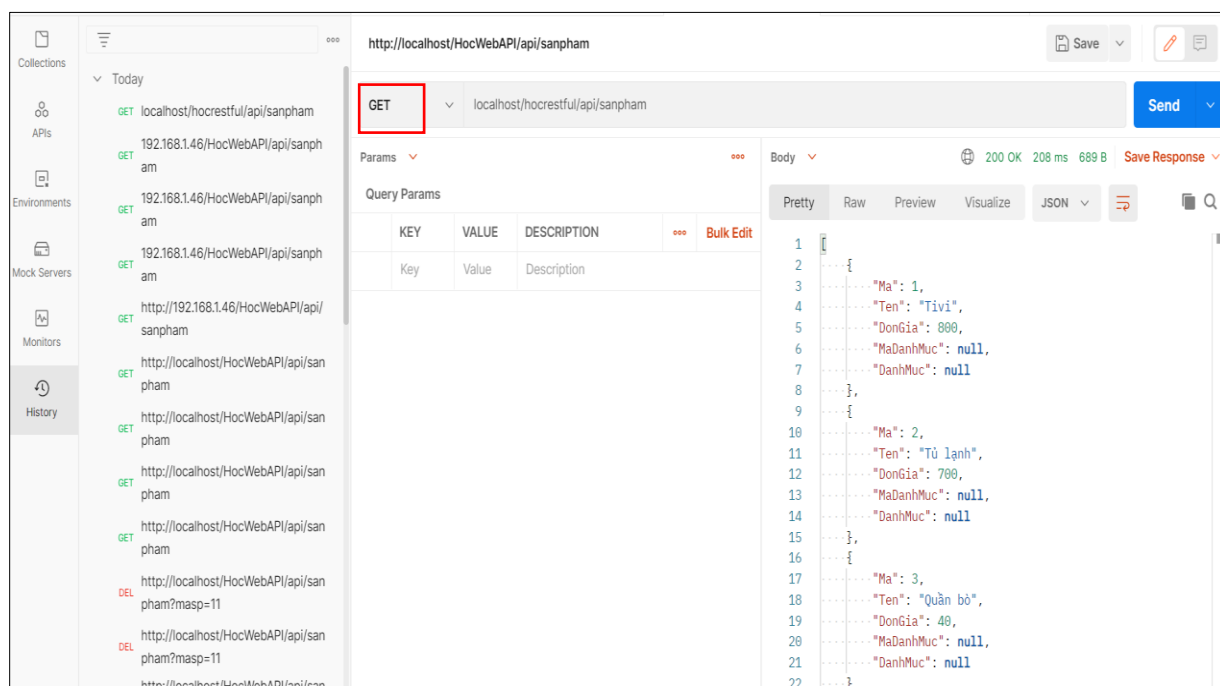
Ở trên máy local có địa chỉ IP là: **192.168.1.154**. Ta thay cho localhost:



2. Kiểm tra Web API với công cụ Postman

+ Lấy toàn bộ Sản phẩm

localhost/hocrestful/api/sanpham



+ Lưu Sản phẩm

localhost/hocrestful/api/sanpham?masp=111&tensp=abc&dongia=400&madm=1

POST localhost/hocrestful/api/sanpham?masp=111&tensp=abc&dongia=400&madm=1 Send

Params Query Params

	KEY	VALUE	DESCRIPTION		Bulk Edit
<input checked="" type="checkbox"/>	masp	111			
<input checked="" type="checkbox"/>	tensp	abc			
<input checked="" type="checkbox"/>	dongia	400			
<input checked="" type="checkbox"/>	madm	1			
	Key	Value	Description		

Body 200 OK 9 ms 262 B Save Response

Pretty Raw Preview Visualize JSON 1 true

+ Sửa Sản phẩm

localhost/hocrestful/api/sanpham?masp=111&tensp=xyz&dongia=888

PUT localhost/hocrestful/api/sanpham?masp=111&tensp=xyz&dongia=888 Send

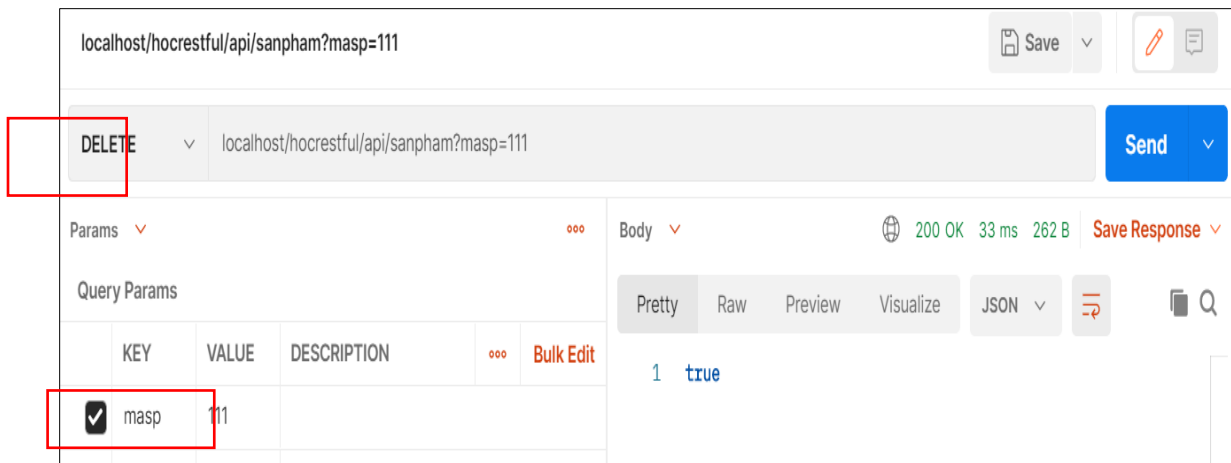
Params Query Params

	KEY	VALUE	DESCRIPTION		Bulk Edit
<input checked="" type="checkbox"/>	masp	111			X
<input checked="" type="checkbox"/>	tensp	xyz			
<input checked="" type="checkbox"/>	dongia	888			
	Key	Value	Description		

Body 200 OK 91 ms 262 B Save Response

Pretty Raw Preview Visualize JSON 1 true

+ Xóa Sản phẩm



B. Bài thực hành mẫu

Bài 1: Triệu gọi Web API lấy danh sách Sản phẩm theo danh mục

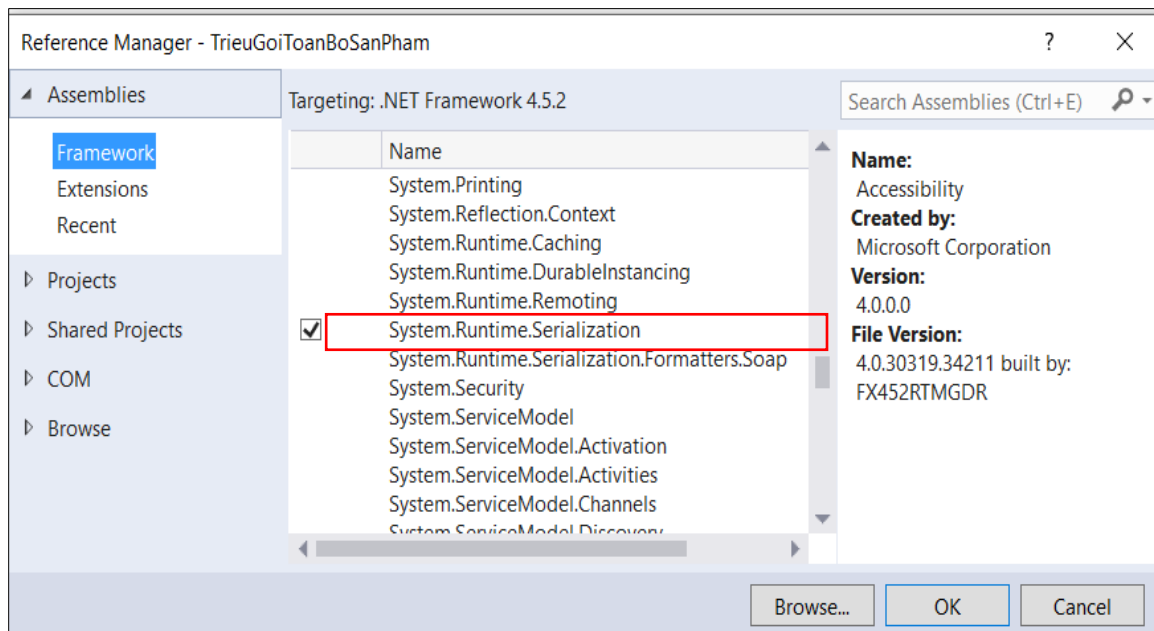
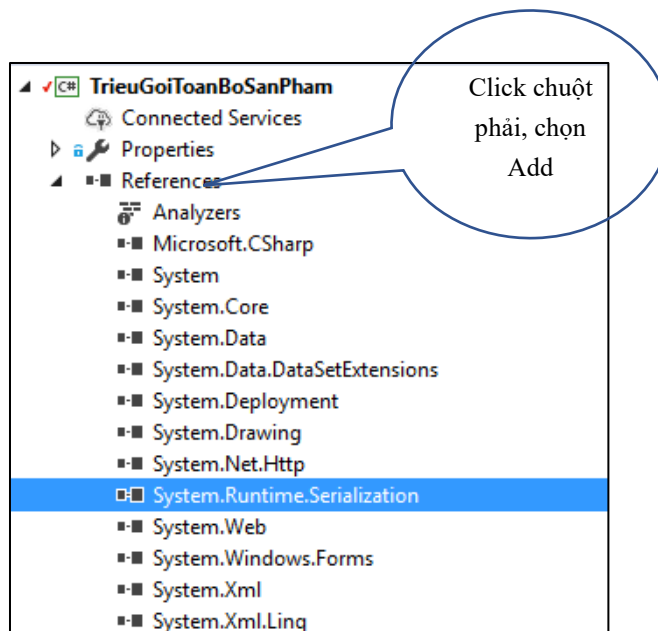
<http://192.168.1.154/hocrestful/api/sanpham?madm=1>

```
- <ArrayOfSanPham>
- <SanPham>
  <DanhMuc i:nil="true"/>
  <DonGia>599</DonGia>
  <Ma>1</Ma>
  <MaDanhMuc i:nil="true"/>
  <Ten>Bóng đèn điện quang tiết kiệm</Ten>
</SanPham>
- <SanPham>
  <DanhMuc i:nil="true"/>
  <DonGia>113</DonGia>
  <Ma>3</Ma>
  <MaDanhMuc i:nil="true"/>
  <Ten>Loa phát thanh Đức</Ten>
</SanPham>
- <SanPham>
  <DanhMuc i:nil="true"/>
  <DonGia>8000</DonGia>
  <Ma>456</Ma>
  <MaDanhMuc i:nil="true"/>
  <Ten>Máy cắt Kiếng</Ten>
</SanPham>
</ArrayOfSanPham>
```

Mã danh mục: 1 Tải Sản phẩm

Ma	Ten	DonGia
1	Bóng đèn điện quang ti...	599
3	Loa phát thanh Đức	113
456	Máy cắt Kiếng	8000

+ Thêm thư viện **System.Runtime.Serialization** vào Project (để sau gọi class JSON trong thư viện này)



+ Xây dựng lớp Mô hình cho Sản phẩm:

```
namespace TrieuGoiToanBoSanPham
{
    public class SanPham
    {
        public int Ma { get; set; }
        public string Ten { get; set; }
        public int DonGia { get; set; }
    }
}
```

+ Kết nối lấy dữ liệu từ Web API về:

```
using System.Windows.Forms;
using System.Net;
using System.Runtime.Serialization;
using System.Runtime.Serialization.Json;

namespace TrieuGoiWebAPILaySanPhamTheoDanhMuc
{
    3 references
    public partial class Form1 : Form
    {
        1 reference
        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

```
private void butTaiSP_Click(object sender, EventArgs e)
{
    string madm = txtMaDanhMuc.Text;
    string link = "http://localhost/HocWebAPI/api/sanpham?madm=" + madm;
    HttpWebRequest request = HttpWebRequest.CreateHttp(link);
    WebResponse response = request.GetResponse();
    DataContractJsonSerializer js = new
DataContractJsonSerializer(typeof(SanPham[]));
    object data = js.ReadObject(response.GetResponseStream());
    SanPham[] arrSanPham = data as SanPham[];
    dataGridView1.DataSource = arrSanPham;
}
```

=> Chạy xem kết quả

Mã danh mục: 1 Tải Sản phẩm

Ma	Ten	DonGia
1	Bóng đèn điện quang ti...	599
3	Loa phát thanh Đức	113
456	Máy cắt Kiếng	8000

Bài 2: Triệu gọi Web API lưu một sản phẩm

<http://192.168.1.154/hocrestful/api/sanpham>

```
[HttpPost]
public bool LuuSanPham(int masp, string tensp, int dongia, int madm)
{
    try
    {
        CSDLTestDataContext context = new CSDLTestDataContext();
        SanPham sp = new SanPham();
        sp.Ma = masp;
        sp.Ten = tensp;
        sp.DonGia = dongia;
        sp.MaDanhMuc = madm;
        context.SanPhams.InsertOnSubmit(sp);
        context.SubmitChanges();
        return true;
    }
    catch { }
    return false;
}
```

➤ Xây dựng ứng dụng WinForm với giao diện sau:

The screenshot shows a Windows application window titled 'Form1'. It contains four input fields stacked vertically: 'Mã SP:' with the value '738', 'Tên SP:' with the value 'Đèn Laze', 'Đơn giá:' with the value '1123', and 'Danh mục:' with a dropdown menu showing 'Hàng Điện Tử'. Below these fields is a button labeled 'Lưu sản phẩm'.

Phải hiển thị
danh sách
Danh Mục cho
người dùng
chọn

- Thêm thư viện `System.Runtime.Serialization` vào Project
- Xây dựng lớp mô hình cho danh mục:

```
public class DanhMuc
{
    public int MaDanhMuc { get; set; }
    public string TenDanhMuc { get; set; }
}
```

- Khai báo thêm các thư viện:

```
using System.Net;
using System.Runtime.Serialization.Json;
using System.IO;
```

- Gọi API lấy danh sách Danh mục hiển thị lên ComboBox:

```
private void Form1_Load(object sender, EventArgs e)
{
    string link = "http://localhost/HocWebAPI1/api/danhmuc";
    HttpRequest request = WebRequest.CreateHttp(link);
    WebResponse response = request.GetResponse();
    DataContractJsonSerializer js = new DataContractJsonSerializer(typeof(DanhMuc[]));
    object Data = js.ReadObject(response.GetResponseStream());
    DanhMuc[] dsDM = Data as DanhMuc[];
    cboDanhMuc.DataSource = dsDM;
    cboDanhMuc.ValueMember = "MaDanhMuc";
    cboDanhMuc.DisplayMember = "TenDanhMuc";
}
```

➤ Gọi API Lưu sản phẩm để lưu sản phẩm mới vào CSDL

```
private void btnLuuSP_Click(object sender, EventArgs e)
{
    string postString = string.Format("?masp={0}&tensp={1}&dongia={2}&madm={3}",
        txtMaSP.Text, txtTenSP.Text, txtDonGia.Text, cboDanhMuc.SelectedValue);
    //Link gọi API lưu 1 sản phẩm
    string link = "http://localhost/HocWebAPI/api/sanpham/" + postString;
    HttpRequest request = HttpWebRequest.CreateHttp(link);
    request.Method = "POST";
    request.ContentType = "application/json;charset=UTF-8";
    byte[] byteArray = Encoding.UTF8.GetBytes(postString);
    request.ContentLength = byteArray.Length;
    Stream dataStream = request.GetRequestStream();
    dataStream.Write(byteArray, 0, byteArray.Length);
    dataStream.Close();
    DataContractJsonSerializer js = new DataContractJsonSerializer(typeof(bool));
    object data = js.ReadObject(request.GetResponse().GetResponseStream());
    bool kq = (bool)data;
    if (kq)
        MessageBox.Show("Thêm sản phẩm thành công");
    else
        MessageBox.Show("Thêm sản phẩm thất bại");
}
```

Bài 3: Triệu gọi Web API Sửa một sản phẩm

- Xây dựng ứng dụng WinForm với giao diện sau:

Form1

Danh Sách Sản phẩm:

- Bóng đèn điện quang tiết kiệm
- Loa phát thanh Đức
- Diệt muỗi Châu Phi
- Chuối Chien Gion
- Giày Vina cao cấp**
- Dép Hồng Hạnh cao cấp
- Bia 333
- Máy cắt Kiếng
- Nước Mía
- Không có gì Gĩ trắng
- Đèn Laze
- Máy giặt
- Bia Tiger

Chi tiết Sản phẩm:

Mã sản phẩm:

133

Tên sản phẩm:

Giày Vina cao cấp

Đơn Giá:

7000

Sửa sản phẩm

<http://192.168.1.154/hocrestful/api/sanpham>

```
- <ArrayOfSanPham>
+ <SanPham></SanPham>
+ <SanPham></SanPham>
- <SanPham>
  <DanhMuc i:nil="true"/>
  <DonGia>27</DonGia>
  <Ma>13</Ma>
  <MaDanhMuc i:nil="true"/>
  <Ten>Đèn Pin Japan 2</Ten>
</SanPham>
+ <SanPham></SanPham>
+ <SanPham></SanPham>
- <SanPham>
  <DanhMuc i:nil="true"/>
  <DonGia>5355</DonGia>
  <Ma>38</Ma>
  <MaDanhMuc i:nil="true"/>
  <Ten>Bom Bi</Ten>
</SanPham>
+ <SanPham></SanPham>
+ <SanPham></SanPham>
</ArrayOfSanPham>
```

- Thêm thư viện System.Runtime.Serialization vào Project (như bài trước)
- Xây dựng lớp mô hình cho sản phẩm:

```
public class SanPham
{
    public int Ma { get; set; }
    public string Ten { get; set; }
    public int DonGia { get; set; }
}
```

- Khai báo thêm các thư viện:

```
using System.Net;
using System.Runtime.Serialization.Json;
using System.IO;
```

- Gọi API lấy danh sách sản phẩm hiển thị lên ListBox:

```
private void Form1_Load(object sender, EventArgs e)
{
    HienThiDanhSachSanPham();
}
private void HienThiDanhSachSanPham()
{
    HttpWebRequest request = WebRequest.CreateHttp
("http://192.168.1.154/hocrestful/api/sanpham");
    WebResponse response = request.GetResponse();
    DataContractJsonSerializer jsonSerializer =
        new DataContractJsonSerializer(typeof(SanPham[]));
    object objResponse = jsonSerializer.ReadObject
(response.GetResponseStream());
    SanPham[] arr = objResponse as SanPham[];
    lstSanPham.DataSource = arr;
    lstSanPham.ValueMember = "Ma";
    lstSanPham.DisplayMember = "Ten";
}
```

- Lấy chi tiết Sản phẩm

```

private void lstSanPham_SelectedIndexChanged(object sender,
EventArgs e)
{
    if (lstSanPham.SelectedIndex == -1) return;
    SanPham sp = lstSanPham.SelectedItem as SanPham;
    txtMa.Text = sp.Ma + "";
    txtTen.Text = sp.Ten;
    txtDonGia.Text = sp.DonGia + "";
}

```

➤ Triệu gọi API Sửa một sản phẩm

```

private void butSua_Click(object sender, EventArgs e)
{
    string putString = string.Format("?masp={0}&tensp={1}&dongia={2}",
        txtMa.Text, txtTen.Text, txtDonGia.Text);
    string link = "http://localhost/hocrestful/api/sanpham/" + putString;
    HttpRequest request = HttpRequest.CreateHttp(link);
    request.Method = "PUT";
    request.ContentType = "application/json;charset=UTF-8";
    byte[] byteArray = Encoding.UTF8.GetBytes(putString);
    request.ContentLength = byteArray.Length;
    Stream dataStream = request.GetRequestStream();
    dataStream.Write(byteArray, 0, byteArray.Length);
    dataStream.Close();
    DataContractJsonSerializer js = new DataContractJsonSerializer(typeof(bool));
    object data = js.ReadObject(request.GetResponse().GetResponseStream());
    bool kq = (bool)data;
    if (kq)
    {
        MessageBox.Show("Sửa sản phẩm thành công");
        HienThiDanhSachSanPham();
    }
    else
        MessageBox.Show("Sửa sản phẩm thất bại");
}

```

Kết quả:

Form1

Danh Sách Sản phẩm:

- Bóng đèn điện quang tiết kiệm
- Loa phát thanh Đức
- Diệt muỗi Châu Phi
- Chui Chien Gion
- Giày Vina cao cấp
- Dép Hồng Hạnh cao cấp
- Bia 333
- Máy cắt Kiếng
- Nước Mía
- Không có gì Gi trắng
- Đèn Laze
- Máy giặt
- Bia Tiger

Chi tiết Sản phẩm:

Mã sản phẩm: 1

Tên sản phẩm: Bóng đèn điện quang tiết kiệm điện

Đơn Giá: 599

Sửa sản phẩm

Form1

Danh Sách Sản phẩm:

- Bóng đèn điện quang
- Loa phát thanh Đức
- Diệt muỗi Châu Phi
- Chui Chien Gion
- Giày Vina cao cấp
- Dép Hồng Hạnh cao cấp
- Bia 333
- Máy cắt Kiếng
- Nước Mía
- Không có gì Gi trắng
- Đèn Laze
- Máy giặt
- Bia Tiger

Chi tiết Sản phẩm:

Mã sản phẩm: 1

Tên sản phẩm: Bóng đèn điện quang

Đơn Giá: 600

Sửa sản phẩm

Bài 4: Triệu gọi Web API xóa một sản phẩm

➤ Xây dựng ứng dụng WinForm với giao diện sau:

Form1

Triệu gọi toàn bộ sản phẩm

Xóa sản phẩm

	Ma	Ten	DonGia
▶	1	Bóng đèn điện q...	600
	3	Loa phát thanh Đ...	113
	15	Diệt muỗi Châu Phi	24
	84	Chui Chien Gion	15
	133	Giày Vina cao cấp	7000
	134	Dép Hồng Hạnh ...	3500
	333	Bia 333	233
	456	Máy cắt Kiếng	8000
	555	Nước Mía	100
	564	Không có gì Gĩ tr...	33
	738	Đèn Laze	1123
	789	Máy giặt	3500
	899	Bia Tiger	355

<http://192.168.1.154/hocrestful/api/sanpham>

```
-<ArrayOfSanPham>
+<SanPham></SanPham>
+<SanPham></SanPham>
-<SanPham>
  <DanhMuc i:nil="true"/>
  <DonGia>27</DonGia>
  <Ma>13</Ma>
  <MaDanhMuc i:nil="true"/>
  <Ten>Đền Pin Japan 2</Ten>
</SanPham>
+<SanPham></SanPham>
+<SanPham></SanPham>
-<SanPham>
  <DanhMuc i:nil="true"/>
  <DonGia>5355</DonGia>
  <Ma>38</Ma>
  <MaDanhMuc i:nil="true"/>
  <Ten>Bom Bi</Ten>
</SanPham>
+<SanPham></SanPham>
+<SanPham></SanPham>
</ArrayOfSanPham>
```

- Thêm thư viện System.Runtime.Serialization vào Project
- Xây dựng lớp mô hình cho sản phẩm:

```
public class SanPham
{
    public int Ma { get; set; }
    public string Ten { get; set; }
    public int DonGia { get; set; }
}
```

- Khai báo thêm các thư viện:

```
using System.Net;
using System.Runtime.Serialization.Json;
using System.IO;
```

- Gọi API lấy danh sách sản phẩm hiển thị lên DataGridView:

```
private void butLoadSP_Click(object sender, EventArgs e)
{
    string link = "http://localhost/hocrestful/api/sanpham";
    HttpRequest request = HttpRequest.CreateHttp(link);
    WebResponse response = request.GetResponse();
    DataContractJsonSerializer js = new DataContractJsonSerializer(typeof(SanPham[]));
    object data = js.ReadObject(response.GetResponseStream());
    SanPham[] arr = data as SanPham[];
    dataGridView1.DataSource = arr;
}
```

➤ Triệu gọi API Xóa một sản phẩm

```
private void butXoaSP_Click(object sender, EventArgs e)
{
    if (dataGridView1.SelectedRows.Count == 0) return;
    DataGridViewRow row = dataGridView1.SelectedRows[0];
    string masp = row.Cells[0].Value.ToString();
    string postString = string.Format("?Masp={0}", masp);
    string link = "http://localhost/hocrestful/api/sanpham";
    HttpRequest request = HttpRequest.CreateHttp(link + postString);
    request.Method = "DELETE";
    request.ContentType = "application/json;charset=UTF-8";
    byte[] byteArr = Encoding.UTF8.GetBytes(postString);
    request.ContentLength = byteArr.Length;
    Stream dataStream = request.GetRequestStream();
    dataStream.Write(byteArr, 0, byteArr.Length);
    dataStream.Close();
    DataContractJsonSerializer js = new DataContractJsonSerializer(typeof(bool));
    object data = js.ReadObject(request.GetResponse().GetResponseStream());
    bool kq = (bool)data;
    if (kq)
    {
        butLoadSP.PerformClick();
        MessageBox.Show("Xóa sản phẩm thành công");
    }
}
```

C. Bài tập thực hành

Bài tập 1: Tạo cơ sở dữ liệu QLLuong gồm 2 bảng:

+ DonVi(MaDonVi, TenDonVi)

+ NhanVien(Ma, HoTen, NgaySinh, GioiTinh, Hsluong, MaDonVi)

Nhập dữ liệu cho 2 bảng

1.1. Xây dựng ứng dụng Windows Form gọi Web API lấy toàn bộ danh sách nhân viên, hiển thị dữ liệu lên điều khiển dữ liệu dạng bảng.

1.2. Xây dựng ứng dụng Windows Form gọi Web API lấy chi tiết 1 nhân viên, hiển thị dữ liệu lên điều khiển dữ liệu dạng bảng.

1.3. Xây dựng ứng dụng Windows Form gọi Web API lấy danh sách nhân viên theo đơn vị, hiển thị dữ liệu lên điều khiển dữ liệu dạng bảng.

1.4. Xây dựng ứng dụng Windows Form gọi Web API lấy danh sách nhân viên theo giới tính, hiển thị dữ liệu lên điều khiển dữ liệu dạng bảng.

1.5. Xây dựng ứng dụng Windows Form gọi Web API tìm danh sách nhân viên có hệ số lương trong khoảng $[a \dots b]$, hiển thị dữ liệu lên điều khiển dữ liệu dạng bảng.

1.6. Xây dựng ứng dụng Windows Form gọi Web API lấy toàn bộ danh sách đơn vị, hiển thị dữ liệu lên điều khiển dữ liệu dạng bảng.

1.7. Xây dựng ứng dụng Windows Form gọi Web API lấy chi tiết 1 đơn vị, hiển thị dữ liệu lên điều khiển dữ liệu dạng bảng.

Bài tập 2: Tạo cơ sở dữ liệu **QLLuong** gồm 2 bảng:

+ DonVi(**MaDonVi**, TenDonVi)

+ NhanVien(**Ma**, HoTen, NgaySinh, GioiTinh, Hsluong, MaDonVi)

Nhập dữ liệu cho 2 bảng

2.1. Xây dựng ứng dụng Windows Form gọi các Web API với giao diện và các chức năng như sau:

Form cập nhật sản phẩm

Mã SP:

Tên SP:

Đơn giá:

Danh mục:

Lưu sản phẩm Xóa sản phẩm Sửa sản phẩm

	Mã SP	Tên sản phẩm	Đơn giá
*			

2.2. Xây dựng ứng dụng Web Form gọi các Web API với giao diện và các chức năng như sau:

Mã sản phẩm:	1																																
Tên sản phẩm:	Tivi																																
Đơn giá:	800																																
Danh mục:	Điện tử																																
<div>Thêm Sửa Xóa</div>																																	
	<table border="1"> <thead> <tr> <th></th> <th>Mã SP</th> <th>Tên SP</th> <th>Đơn giá</th> </tr> </thead> <tbody> <tr> <td>Chọn dòng</td> <td>1</td> <td>Tivi</td> <td>800</td> </tr> <tr> <td>Chọn dòng</td> <td>2</td> <td>Tủ lạnh</td> <td>700</td> </tr> <tr> <td>Chọn dòng</td> <td>3</td> <td>Quần bò</td> <td>45</td> </tr> <tr> <td>Chọn dòng</td> <td>4</td> <td>Áo cộc tay</td> <td>30</td> </tr> <tr> <td>Chọn dòng</td> <td>5</td> <td>Váy</td> <td>50</td> </tr> <tr> <td>Chọn dòng</td> <td>11</td> <td>Quần bò</td> <td>45</td> </tr> <tr> <td>Chọn dòng</td> <td>12</td> <td>Váy</td> <td>50</td> </tr> </tbody> </table>		Mã SP	Tên SP	Đơn giá	Chọn dòng	1	Tivi	800	Chọn dòng	2	Tủ lạnh	700	Chọn dòng	3	Quần bò	45	Chọn dòng	4	Áo cộc tay	30	Chọn dòng	5	Váy	50	Chọn dòng	11	Quần bò	45	Chọn dòng	12	Váy	50
	Mã SP	Tên SP	Đơn giá																														
Chọn dòng	1	Tivi	800																														
Chọn dòng	2	Tủ lạnh	700																														
Chọn dòng	3	Quần bò	45																														
Chọn dòng	4	Áo cộc tay	30																														
Chọn dòng	5	Váy	50																														
Chọn dòng	11	Quần bò	45																														
Chọn dòng	12	Váy	50																														