

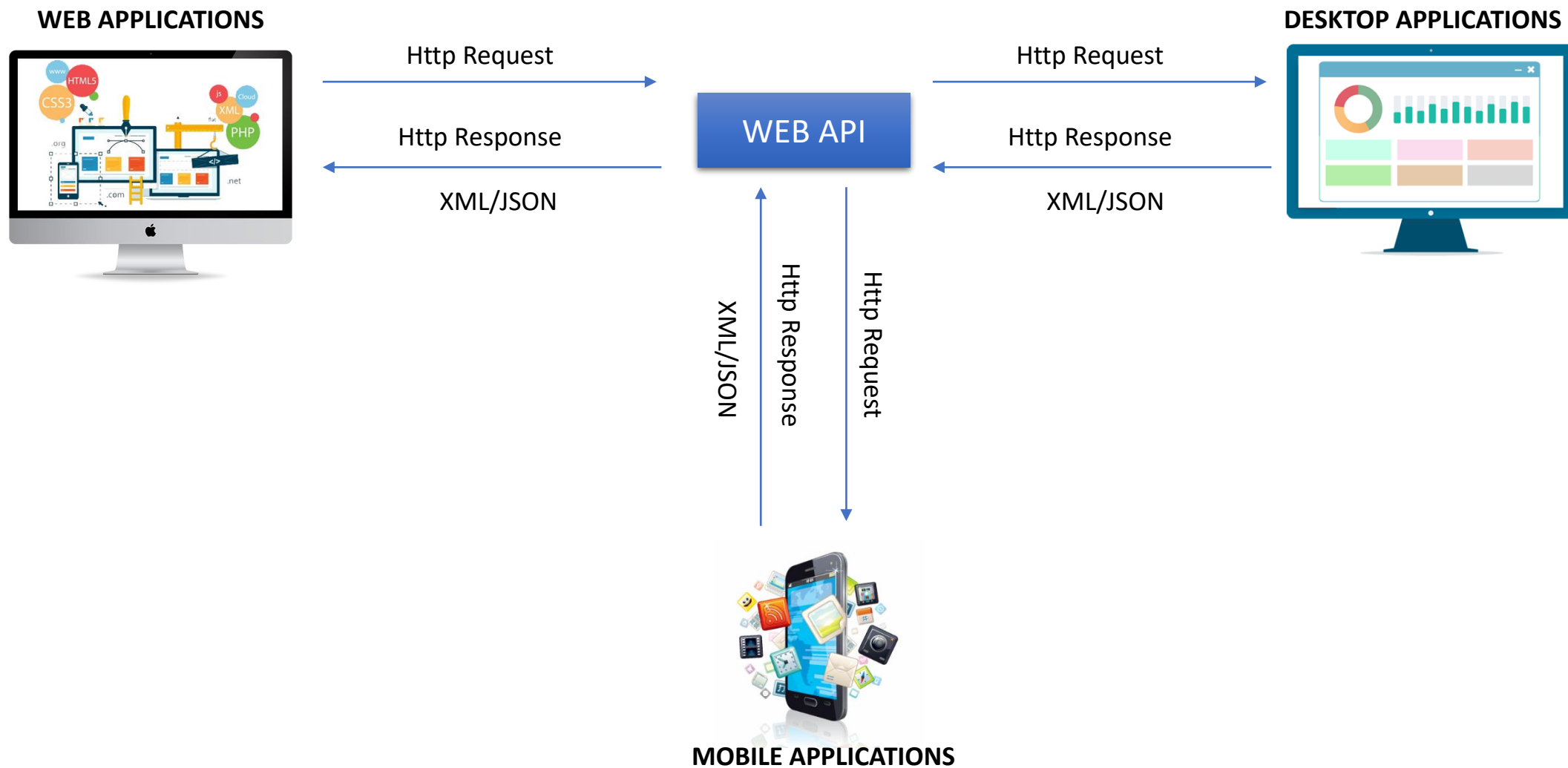


# Bài 01

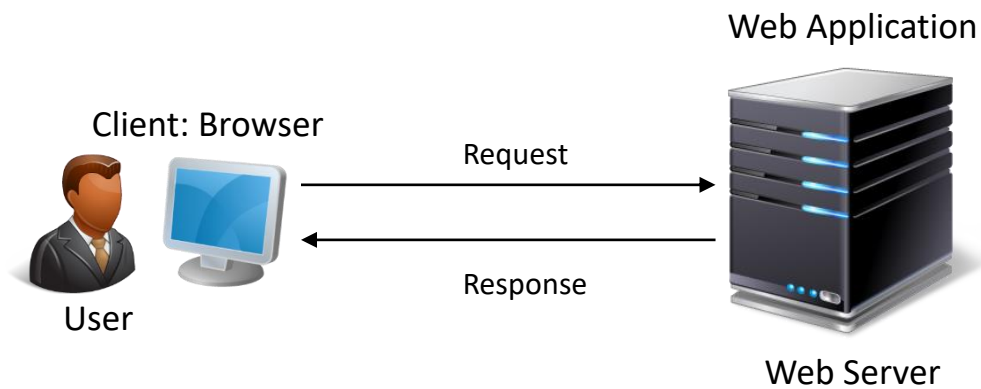
## Giới thiệu về ASP.NET Core Web API

- Giới thiệu về Web API
- So sánh Web Application với Web Service và Web API
- Tạo và gọi ASP.NET Core Web API

- **ASP.NET Core Web API** là một Framework để tạo các ứng dụng RESTful dựa trên giao thức **HTTP** với mục đích cung cấp các nghiệp vụ cho bất kỳ ứng dụng client nào bao gồm cả mobile.
- **ASP.NET Core Web API** trao đổi dữ liệu sử dụng **content type** là **XML** hoặc JavaScript Object Notation(**JSON**).

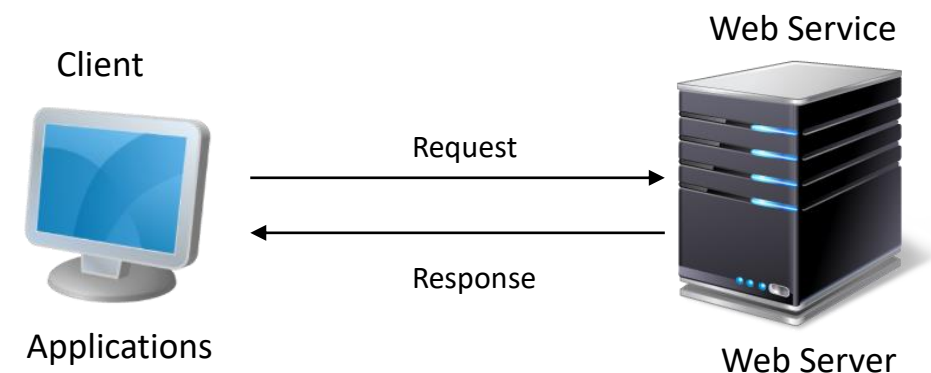


- **ASP.NET Core Web API** là kết quả của quá trình phát triển từ **Web Services** truyền thống.
- **Web Application** và **Web Service** đều nhận yêu cầu, xử lý và trả về kết quả phù hợp cho client.



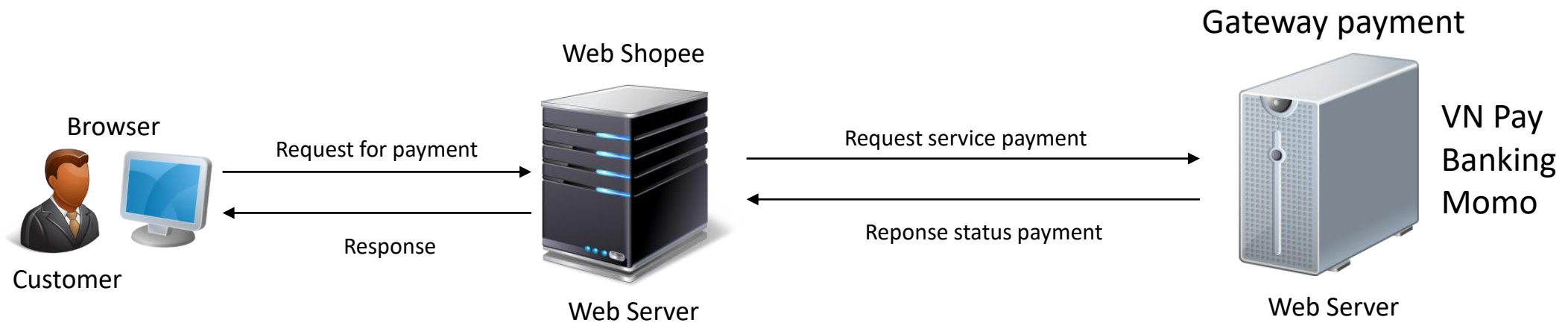
**Web Application Communication**

**VS**



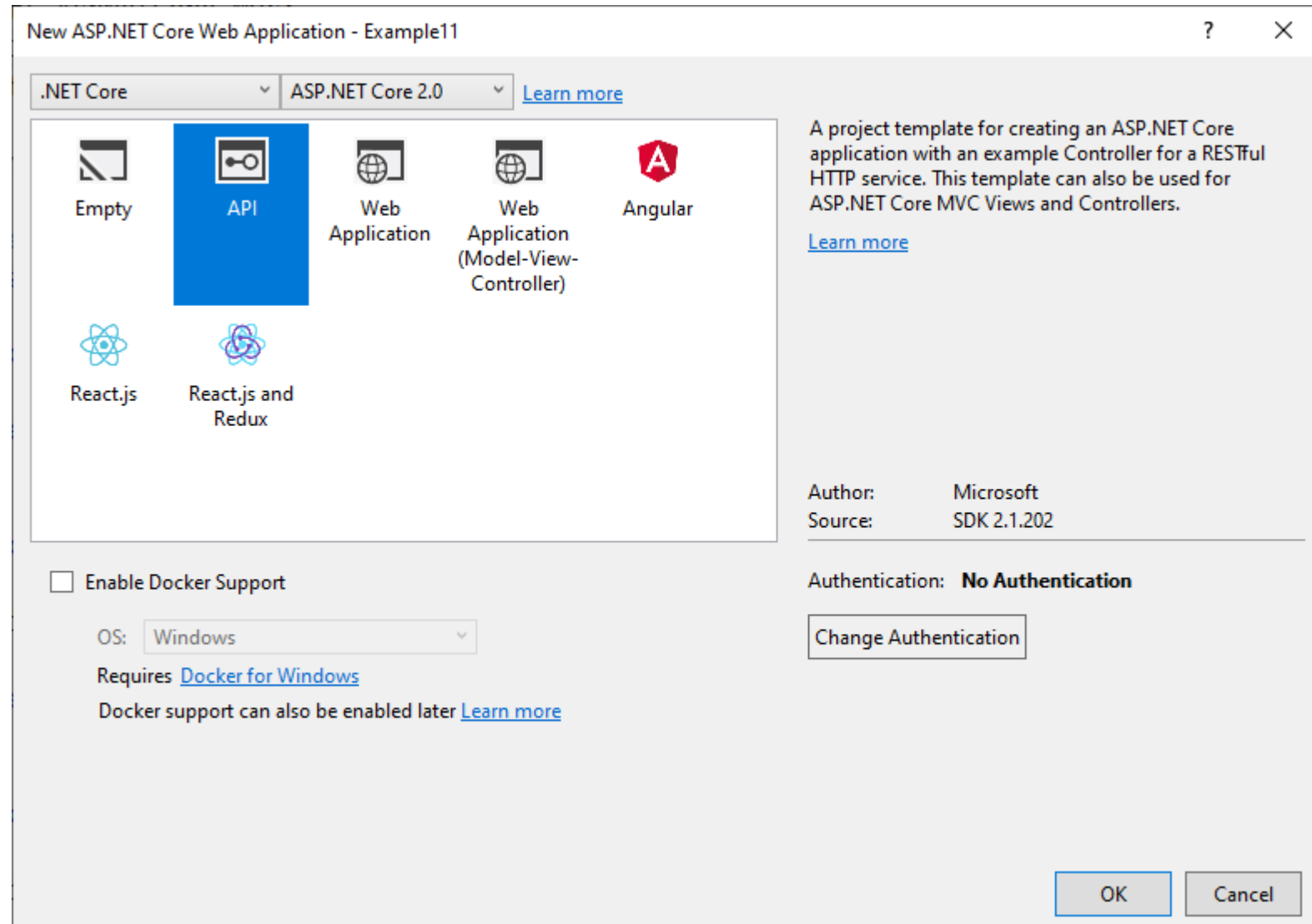
**Web Services Communication**

- Ví dụ:



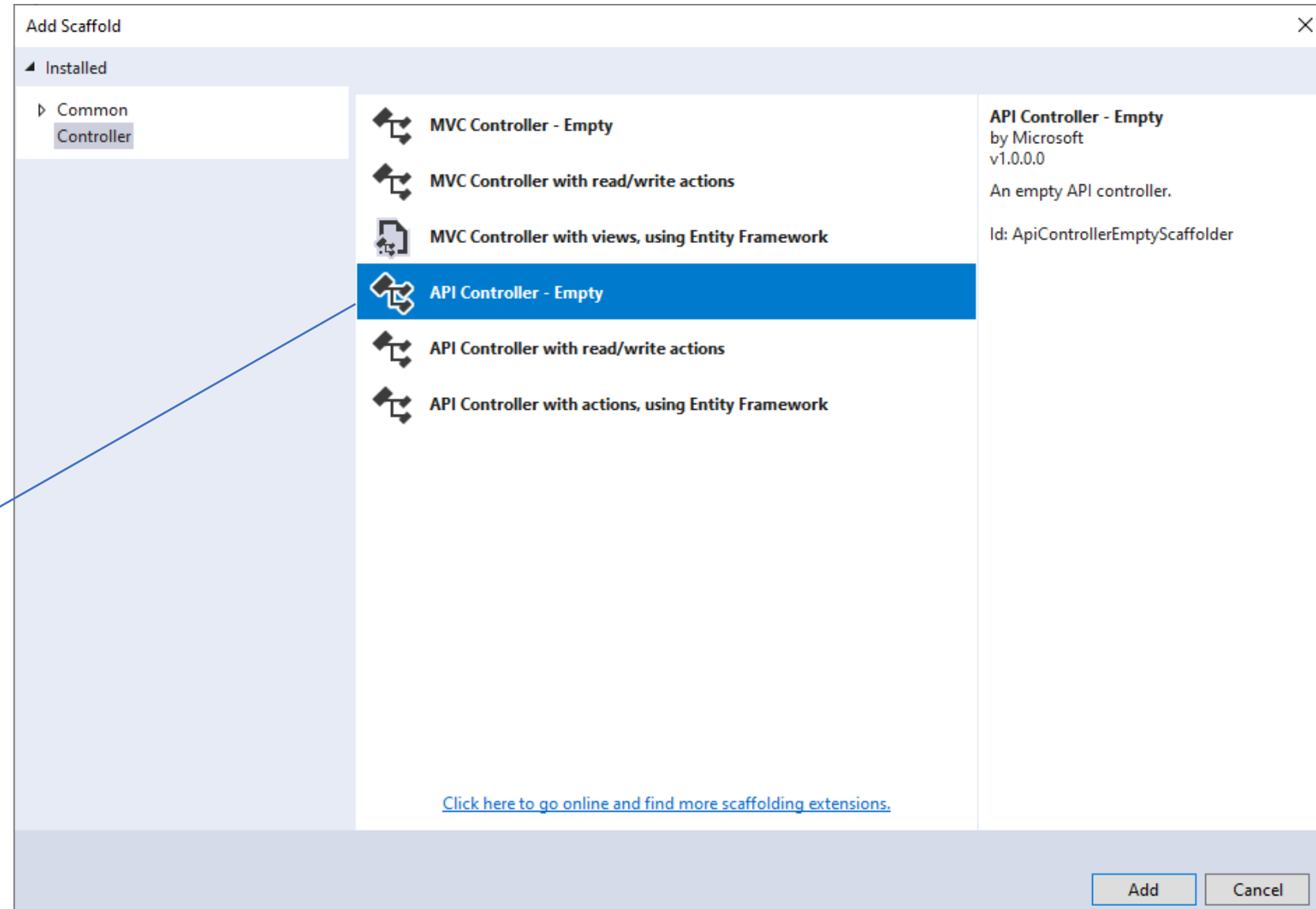
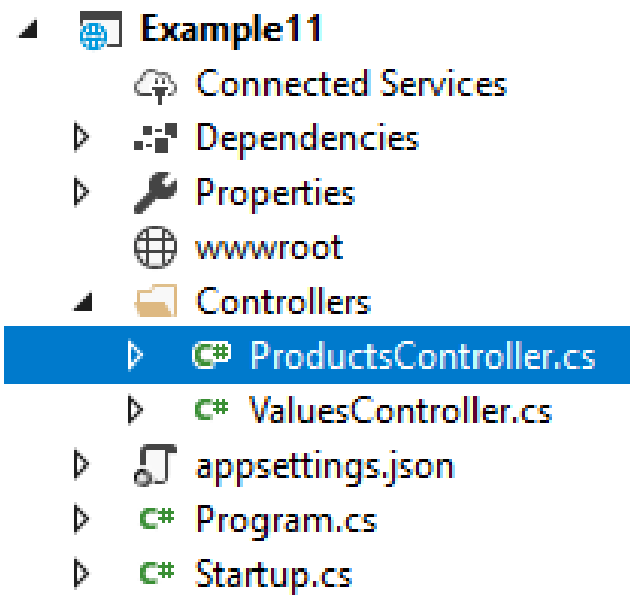
	Web Applications	Web Services	Web API
Protocol	Http/Https	Http/Https based SOAP	Http/Https
Request	Send any thing data	Send any thing data	Send any thing data
Response	HTML data	XML data wrap by SOAP messages	XML or JSON Data
Method	GET, POST	GET, POST	GET, POST, PUT, DELETE
Target	Provide Service Directly to User	Provide Service to other Applications	Provide Service to other Applications

- Tạo mới Project ASP.NET Core Web Application. Chọn loại ứng dụng API -> OK





- Tạo API Controller mới có tên ProductsController trong thực mục Controllers



- Tạo thư mục Models -> tạo lớp Product trong Models

## Example11

- Connected Services
- Dependencies
- Properties
- wwwroot
- Controllers
  - ProductsController.cs
  - ValuesController.cs
- Models
  - Product.cs
- appsettings.json
- Program.cs
- Startup.cs

```
public class Product
{
    public string Id { get; set; }
    public string Name { get; set; }
    public double Price { get; set; }
    public string Description { get; set; }
}
```

```
[Produces("application/json")]
[Route("api/Products")]//url API
public class ProductsController : Controller
{
    //khởi tạo bộ dữ liệu ban đầu để test
    public static List<Product> products = new List<Product>...;

    [HttpGet]
    public IEnumerable<Product> Get()...

    [HttpGet("{id}")] //Lấy theo id
    public IEnumerable<Product> Get(string id)...

    [HttpGet("search/{name}")] //tìm kiếm theo tên
    public IEnumerable<Product> Search(string name)...

    [HttpDelete("{id}")]//xóa theo id
    public JsonResult Delete(string id)...

    [HttpPost]//thêm mới
    [Consumes("application/json")]
    public JsonResult Post(Product p)...

    [HttpPut("{id}")]//cập nhật
    [Consumes("application/json")]
    public JsonResult Put(string id, Product product)...
}
```

```
//khởi tạo bộ dữ liệu ban đầu để test
public static List<Product> products = new List<Product>
{
    new Product{Id="IXUS-185",Name="Máy ảnh Canon IXUS 185",
        Price =2500000,Description="Hàng chính hãng"},
    new Product{Id="DSC-W830",Name="Máy ảnh Sony DSC-W830",
        Price =2675000,Description="Hàng chính hãng bảo hành 2 năm"},
    new Product{Id="A6000",Name="Máy Ảnh Sony Alpha A6000",
        Price =12500000,Description="Bonus Kit 16-50mm"},
    new Product{Id="X-A3",Name="FUJIFILM X-A3 +kit 16-50mm",
        Price =8490000,Description="Tặng thẻ nhớ 16GB, Filter, túi Mirroless"}
};

[HttpGet]//lấy tất cả sản phẩm
public IEnumerable<Product> Get()
{
    return products.AsEnumerable();
}

[HttpGet("{id}")] //Lấy theo id
public IEnumerable<Product> Get(string id)
{
    return products.Where(x=>x.Id==id).AsEnumerable();
}
```

```
[HttpGet("search/{name}")] //tìm kiếm theo tên
public IEnumerable<Product> Search(string name)
{
    return products.Where(x => x.Name.Contains(name)).AsEnumerable();
}

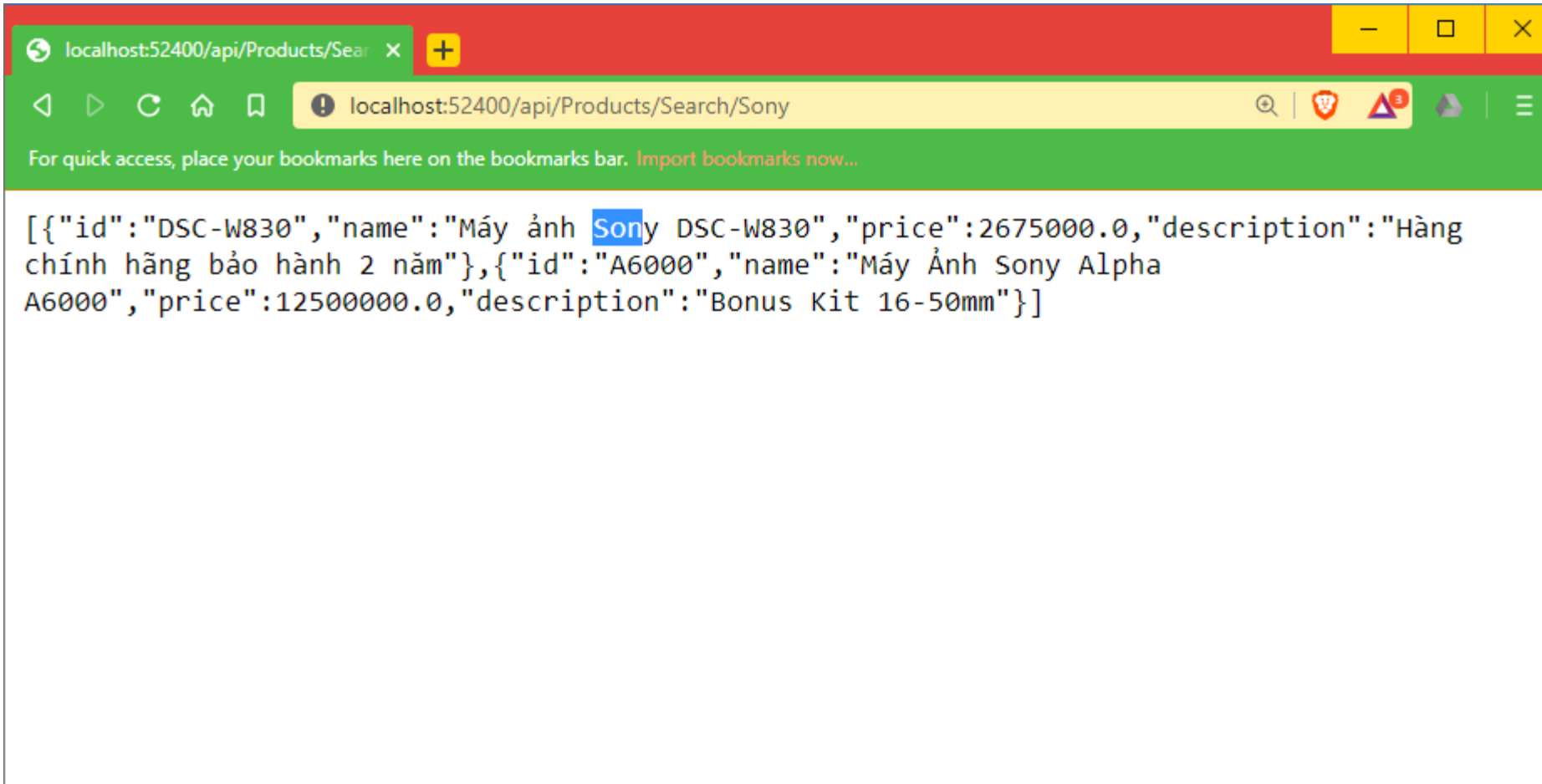
[HttpDelete("{id}")]//xóa theo id
public JsonResult Delete(string id)
{
    var p = products.SingleOrDefault(x => x.Id == id);
    if (p != null)
    {
        products.Remove(p);
        return Json(new { Message = "Xóa thành công" });
    }
    else
    {
        return Json(new { Message = "Không tồn tại sản phẩm bạn xóa" });
    }
}
```

```
[HttpPost]//thêm mới
[Consumes("application/json")]
public JsonResult Post(Product p)
{
    products.Add(p);
    return Json(new { Message = "Thêm thành công" });
}

[HttpPut("{id}")]//cập nhật
[Consumes("application/json")]
public JsonResult Put(string id, Product product)
{
    var p = products.SingleOrDefault(x => x.Id == id);
    if (p != null) {
        p.Name = product.Name;
        p.Price = product.Price;
        p.Description = product.Description;
        return Json(new { Message = "Sửa thành công" });
    }
    else
    {
        return Json(new { Message = "Không tồn tại sản phẩm bạn sửa" });
    }
}
```

- Ctrl + F5 để test những api hỗ trợ method “GET”

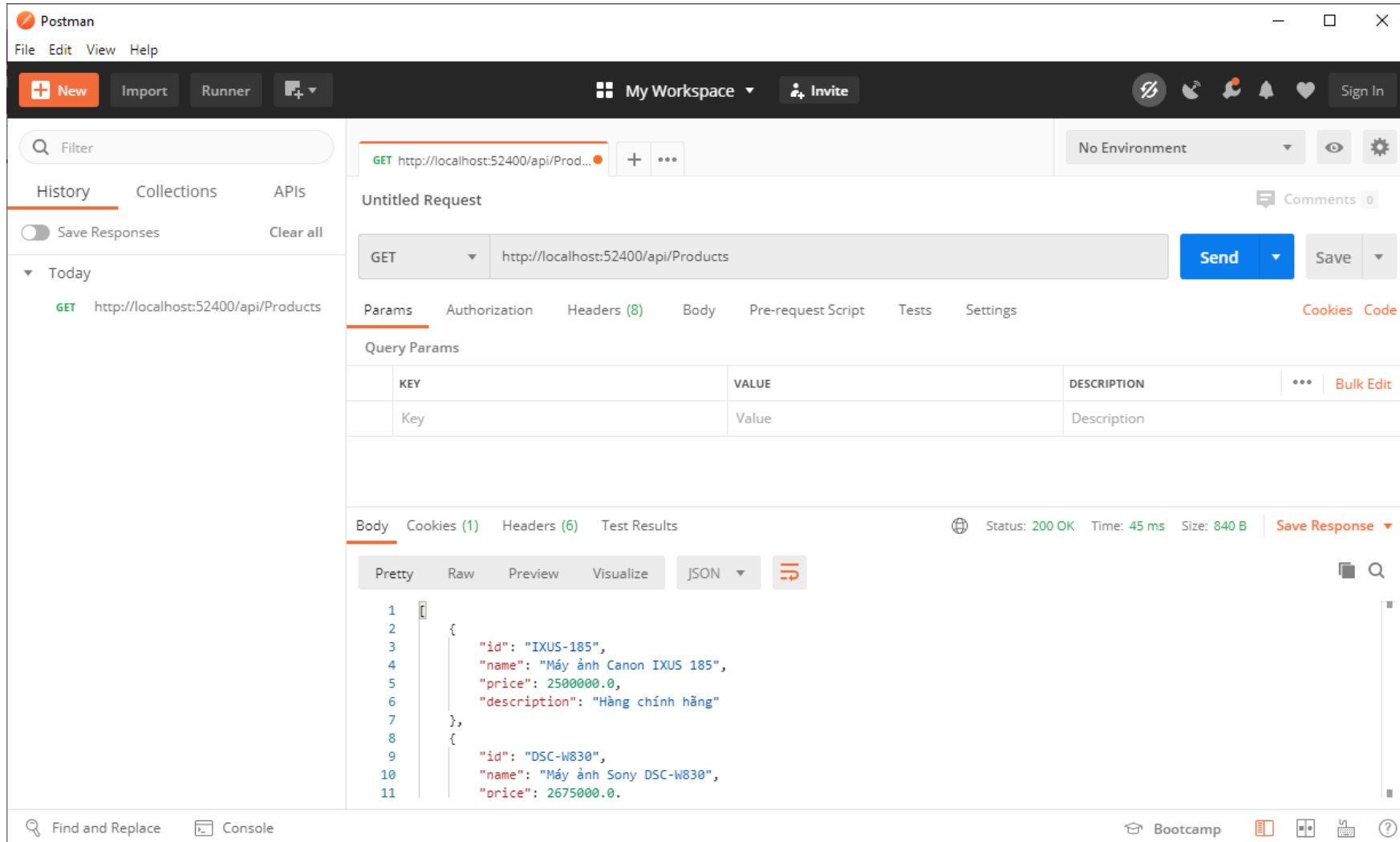








- Test các API bằng phần mềm Postman



The screenshot displays the Postman application interface. On the left sidebar, the 'History' tab is active, showing a list of recent requests: a POST request to `http://localhost:52400/api/Products` and a GET request to the same endpoint. The main workspace shows an 'Untitled Request' for a POST method to `http://localhost:52400/api/Products`. The 'Body' tab is selected, showing a JSON payload: 

```
{  "id": "Iphone-X",  "name": "Điện thoại Iphone X 64g - 256g",  "price": 6690000,  "description": "98% full box"}
```

. Below the request editor, the 'Response' section shows a status of '200 OK' with a time of '53 ms' and a size of '404 B'. The response body is displayed in 'Pretty' JSON format: 

```
{  "message": "Thêm thành công"}
```

. The bottom of the interface includes a 'Find and Replace' search bar, a 'Console' tab, and a 'Bootcamp' link.

# HỎI ĐÁP





# TRẢI NGHIỆM THỰC HÀNH



## TRÂN TRỌNG CẢM ƠN!



238 Hoàng Quốc Việt, Bắc Từ Liêm, Hà Nội



0968.27.6996



[tuyensinh@bachkhoa-apttech.edu.vn](mailto:tuyensinh@bachkhoa-apttech.edu.vn)



[www.bachkhoa-apttech.edu.vn](http://www.bachkhoa-apttech.edu.vn)