

# ASP.NET MVC

ASP.NET là một framework phát triển dùng cho việc xây dựng trang web và website với HTML5, CSS3, JavaScript (Jquery, Bootstrap, Angular...) và mã kịch bản server.

ASP.NET hỗ trợ 3 mô hình phát triển:

- Web Pages, MVC (Model View Controller), và Web Forms.

# Tài liệu tham khảo

- Professional ASP.NET MVC 5.pdf
- [http://www.w3schools.com/aspnet/mvc\\_intro.asp](http://www.w3schools.com/aspnet/mvc_intro.asp)
- <http://www.asp.net/mvc/overview/getting-started/introduction/getting-started>

# Tổng quan

**Wikipedia:** “**Model–view–controller (MVC)** is a software architectural pattern mostly (but not exclusively) for implementing user interfaces on computers.”

MVC là 1 trong 3 mô hình lập trình ASP.NET

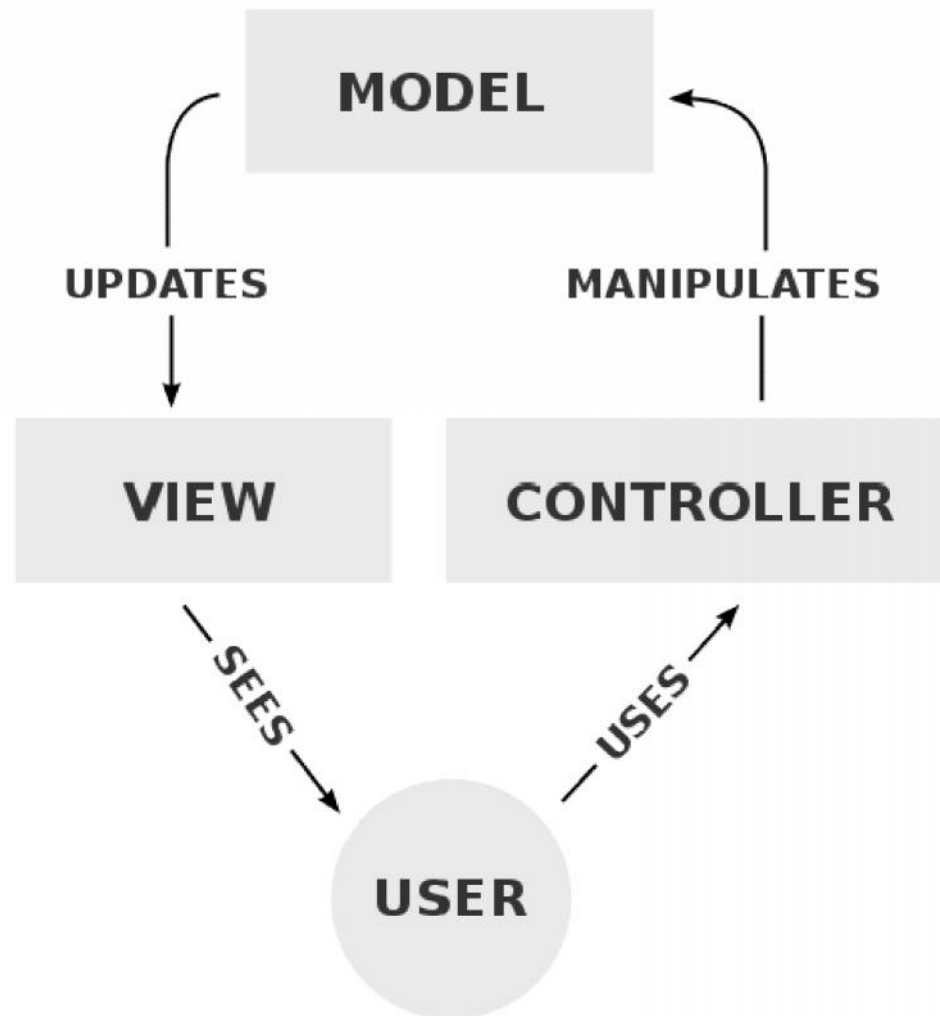
# Tổng quan

MVC là 1 framework dùng để xây dựng ứng dụng website dùng thiết kế MVC (Model View Controller)

MVC chia ứng dụng web thành 3 tầng:

- The business layer (Model logic)
- The display layer (View logic)
- The input control (Controller logic)

# Khái niệm



# Khái niệm

Ở mô hình trên:

- **Model** là lõi ứng dụng (thể hiện danh sách dữ liệu).
  - Model điều khiển logic cho các ứng dụng dữ liệu. Thông thường các đối tượng mô hình truy xuất dữ liệu (hoặc là lưu trữ dữ liệu) từ database
  - Model trực tiếp quản lý dữ liệu, logic và các luật của ứng dụng.

# Tổng quan

Ở mô hình trên:

- **View** là nơi điều khiển hiển thị dữ liệu.
  - Đa số View được tạo từ dữ liệu mô hình.
  - View hiển thị dữ liệu đầu ra của dữ liệu như biểu đồ hay mô hình.

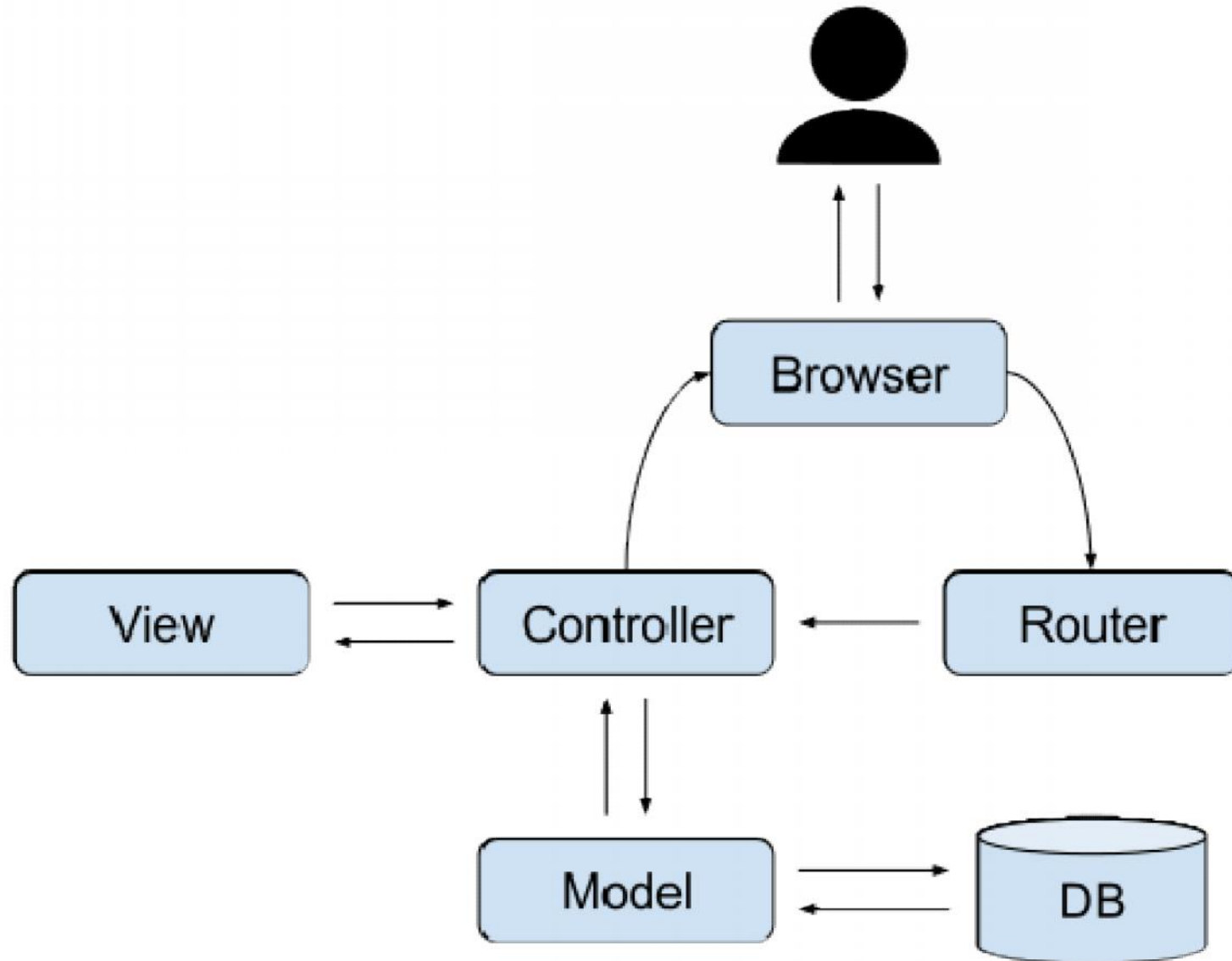
# Tổng quan

Ở mô hình trên:

- **Controller** điều khiển dữ liệu tương tác với database.
  - Controller nhận dữ liệu đầu vào và chuyển dữ liệu này thành lệnh để Model hoặc View thực thi.
  - Controller là 1 phần của ứng dụng để điều khiển tương tác giữa user và database.
  - Thông thường Controller đọc dữ liệu từ View, Input và gửi dữ liệu đến Model.



# Tổng quan



# Tổng quan

MVC nhẹ hơn ASP.NET (Web Forms), có khả năng tích hợp framework cao với các tính năng ASP.NET, chẳng hạn như:

- Master Pages
- Security
- Authentication.

# Tổng quan

## Các phiên bản ASP.NET MVC:

- 10 December 2007 ASP.NET MVC [CTP](#)
- 13 March 2009 ASP.NET MVC 1.0
- 10 March 2010 ASP.NET MVC 2
- 13 January 2011 ASP.NET MVC 3
- 15 August 2012 ASP.NET MVC 4
- 17 October 2013 ASP.NET MVC 5
- 18 November 2015 ASP.NET MVC 6.0.0-rc1

# Tổng quan

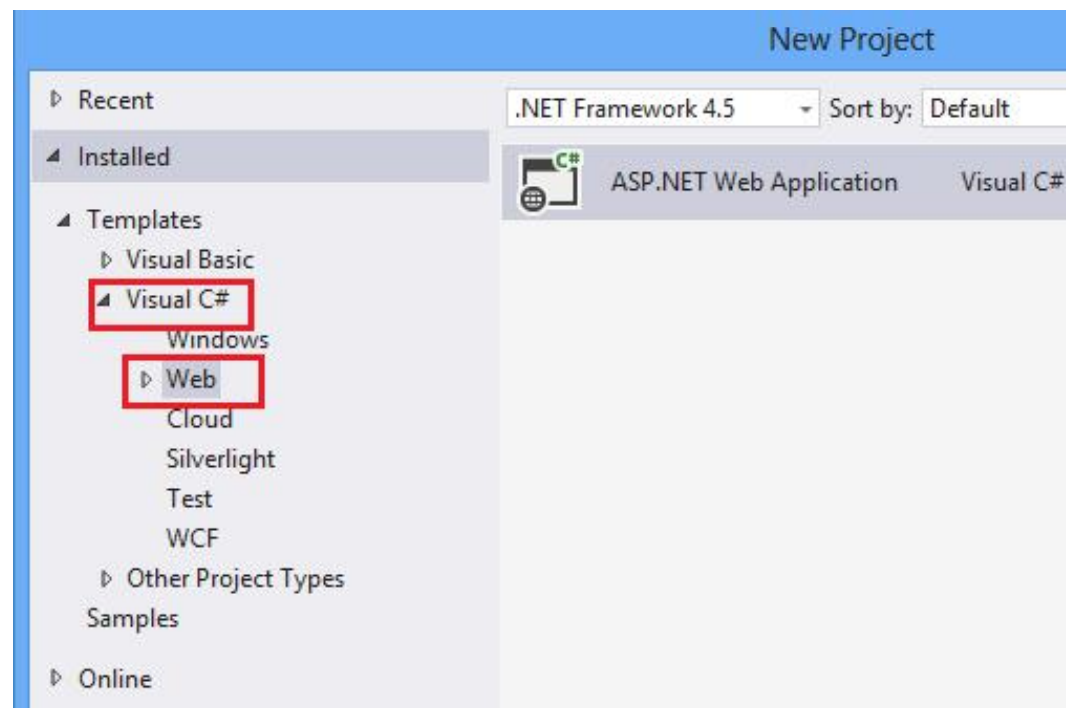
Yêu cầu:

- Start by installing and running Visual Studio Express 2013 for Web or Visual Studio 2013.

# Ứng dụng MVC cơ bản

## 1. Tạo project

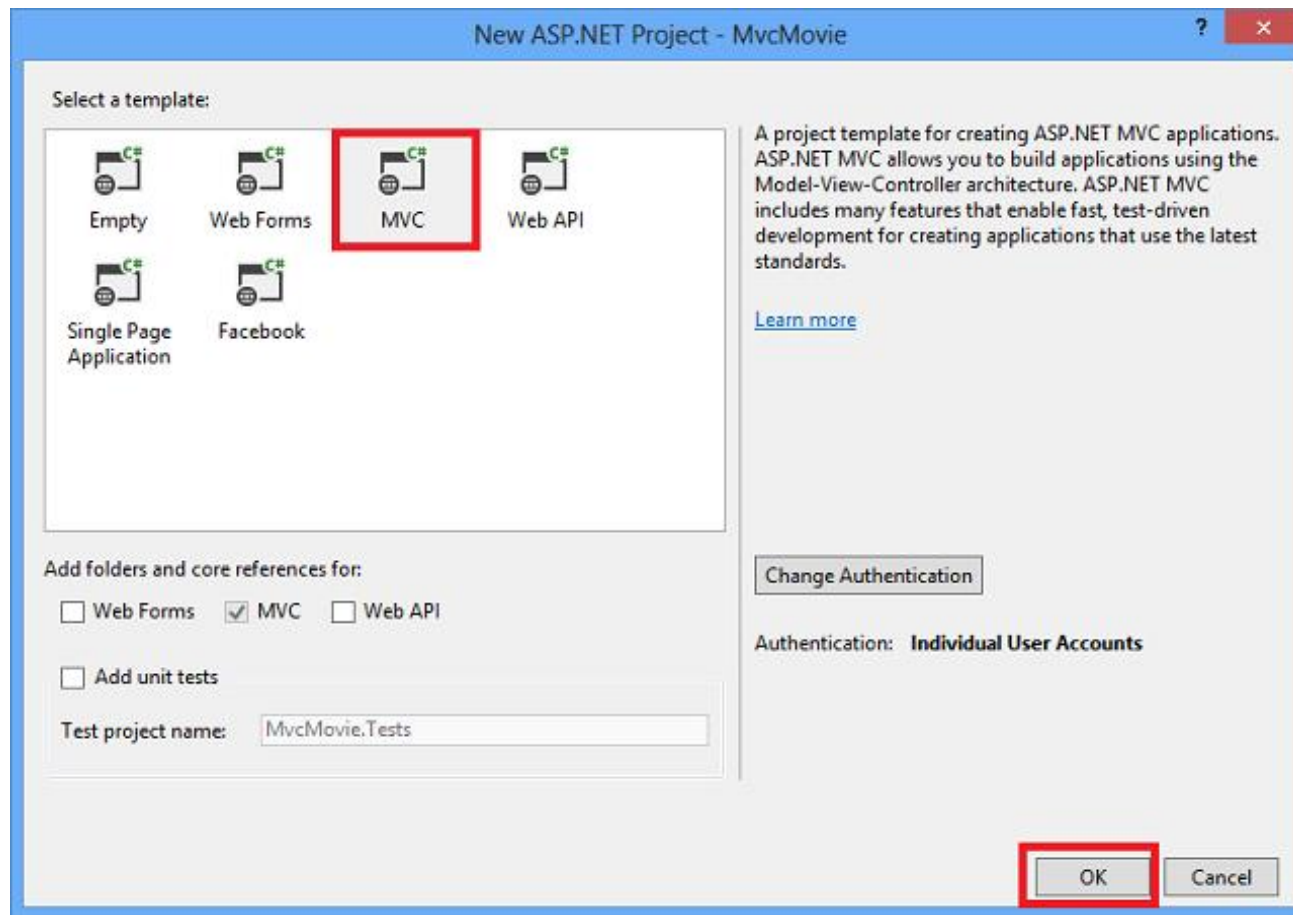
Tạo project ASP.NET Web Application đặt tên là "MvcMovie"



# Ứng dụng MVC cơ bản

## 1. Tạo project

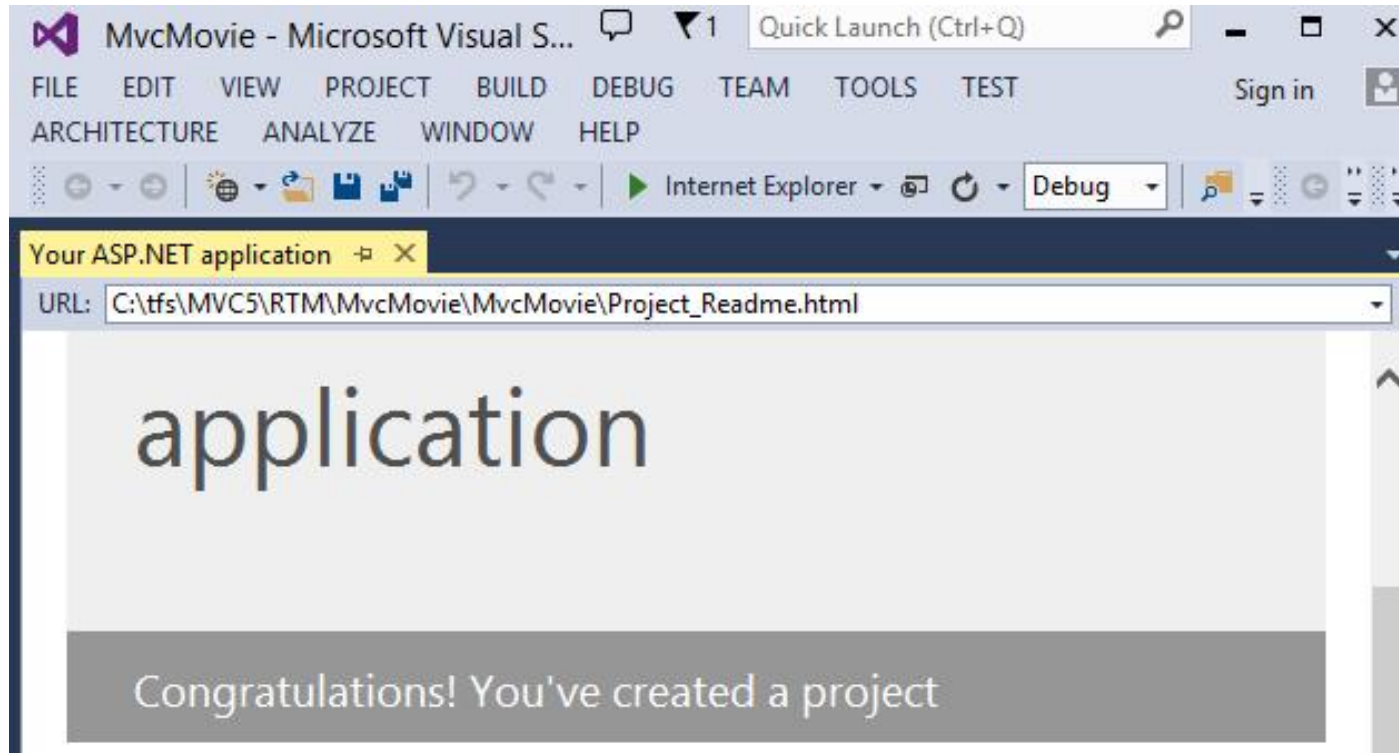
Chọn kiểu dự án Web là MVC



# Ứng dụng MVC cơ bản

## 1. Tạo project

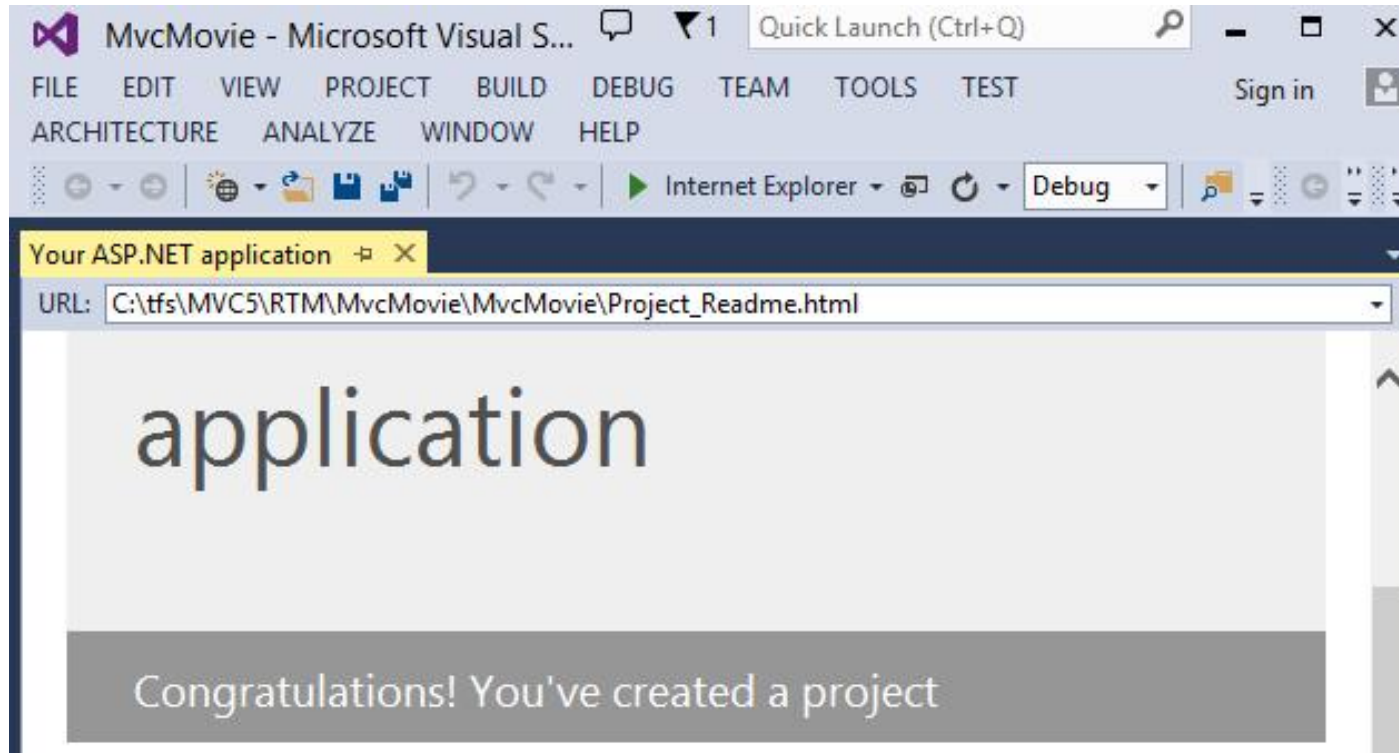
Visual Studio để tạo mặc định template cho ASP.NET MVC.



# Ứng dụng MVC cơ bản

## 1. Tạo project

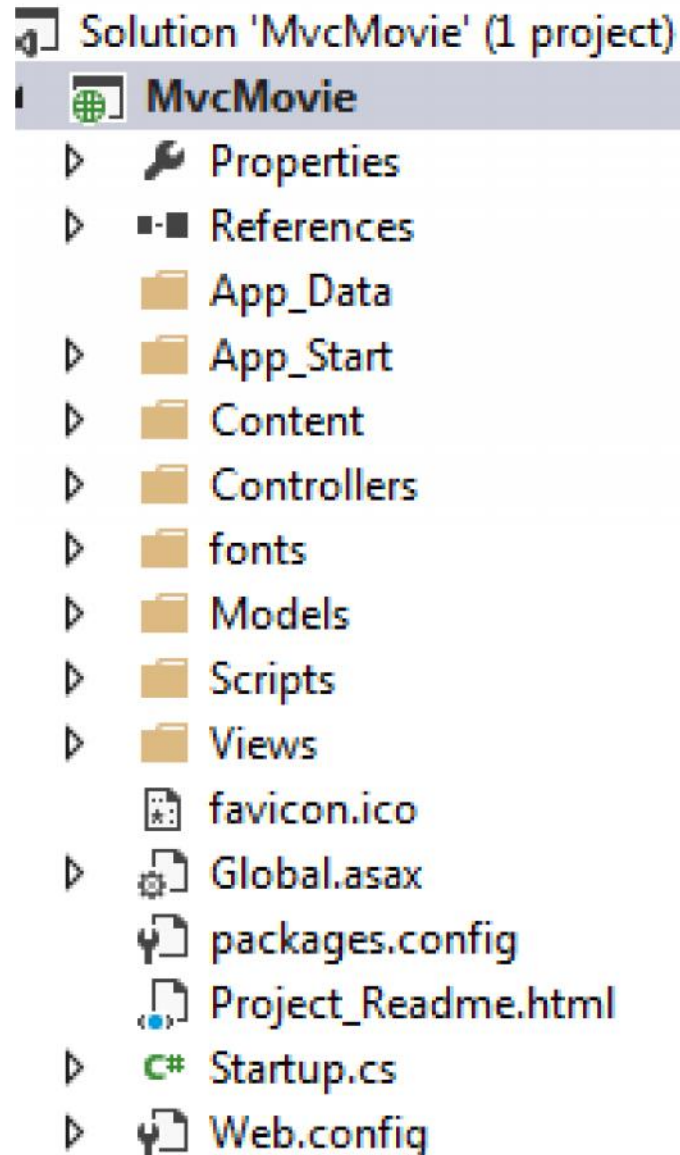
Visual Studio để tạo mặc định template cho ASP.NET MVC.





# Ứng dụng MVC cơ bản

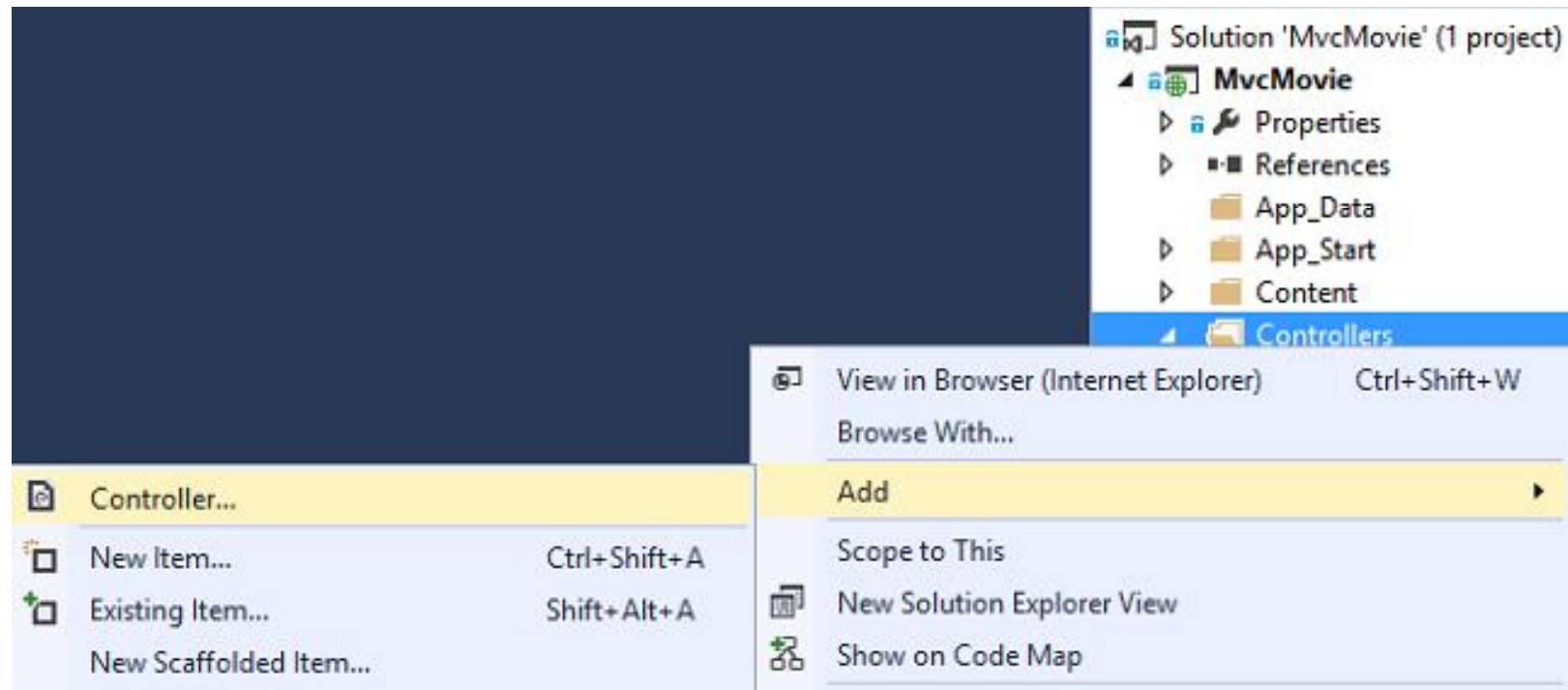
## 1. Tạo project



# Ứng dụng MVC cơ bản

## 2. Thêm Controller

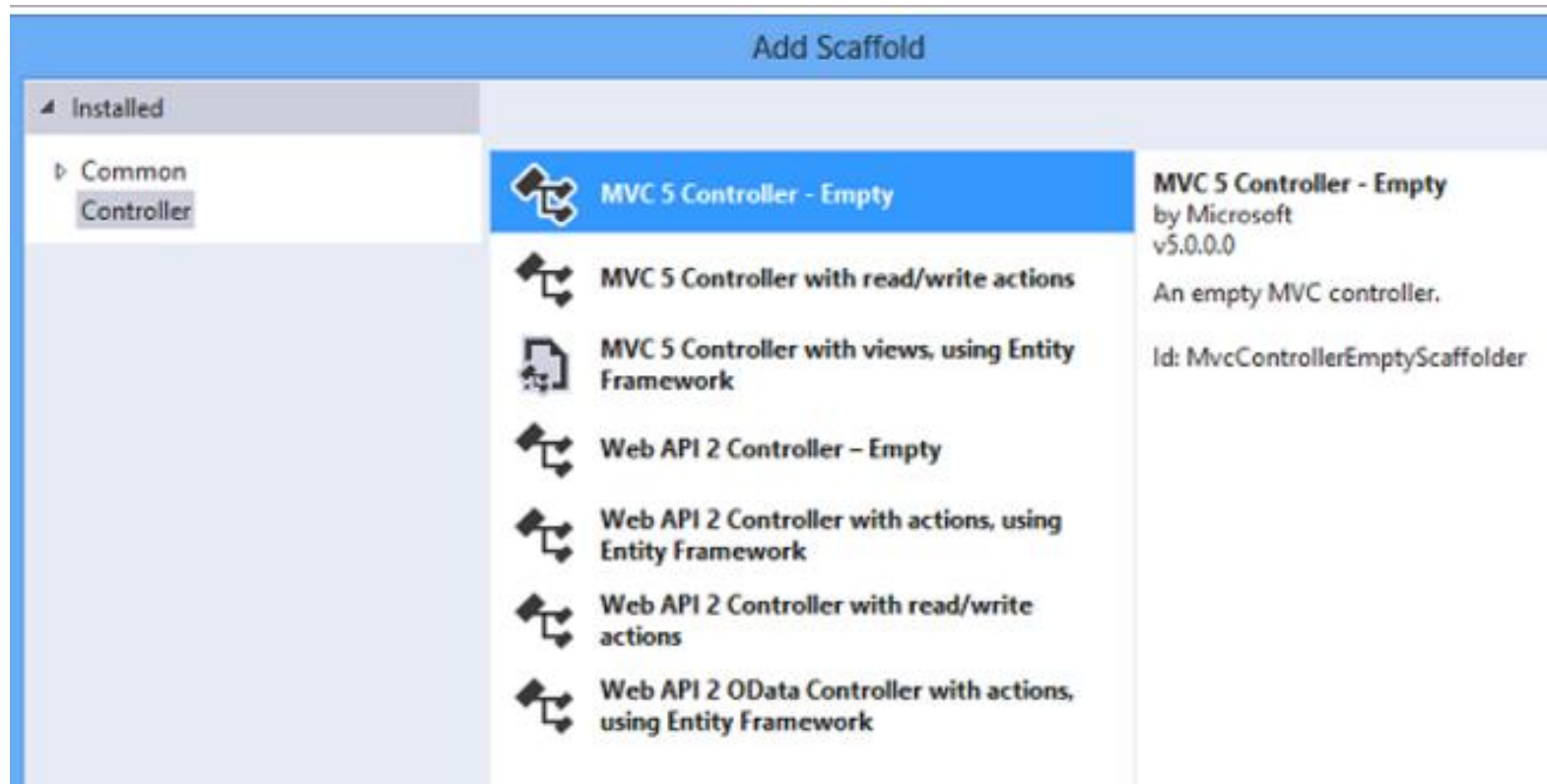
Ở **Solution Explorer**, chuột phải thư mục *Controllers* và nhấn **Add**, sau đó **Controller**.



# Ứng dụng MVC cơ bản

## 2. Thêm Controller

Ở hộp thoại **Add Scaffold**, chọn **MVC 5 Controller – Empty**, sau đó **Add**



# Ứng dụng MVC cơ bản

## 2. Thêm Controller

ASP.NET Scaffolding là framework gieo mã cho các ứng dụng ASP.NET Web.

Visual Studio 2013 chứa bộ gieo mã được cài đặt dành cho MVC và các ứng dụng Web API.

# Ứng dụng MVC cơ bản

## 2. Thêm Controller

Đặt tên controller là "HelloWorldController" và sau đó nhấn nút **Add**.

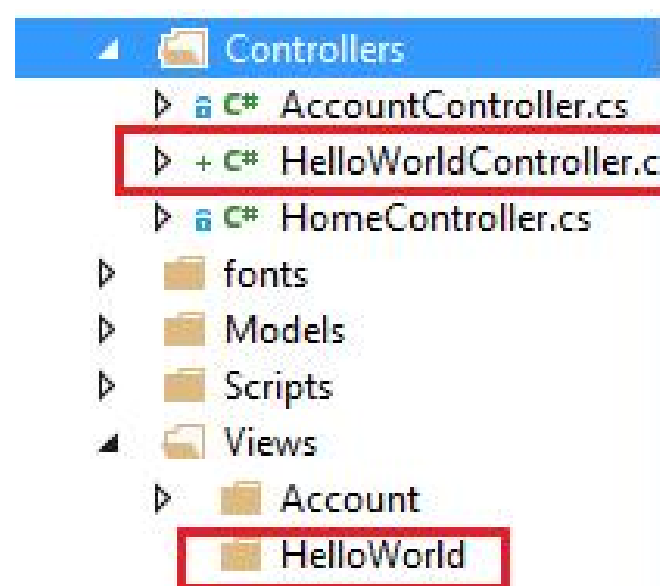


The image shows a standard Windows-style dialog box titled "Add Controller". It features a close button (an 'x' icon) in the top right corner. The main content area contains a label "Controller name:" followed by a text input field. The text "HelloWorldController" is entered into this field. At the bottom right of the dialog, there are two buttons: "Add" and "Cancel".

# Ứng dụng MVC cơ bản

## 2. Thêm Controller

Ở giao diện, **HelloWorldController.cs** và **Views\HelloWorld** được tạo



# Ứng dụng MVC cơ bản

## 2. Thêm Controller

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace MvcMovie.Controllers
{
    public class HelloWorldController : Controller
    {
        //
        // GET: /HelloWorld/
        public ActionResult Index()
        {
            return View();
        }
    }
}
```

# Ứng dụng MVC cơ bản

## 2. Thêm Controller

Thay nội dung **HelloController.cs** bằng đoạn code

```
using System.Web;
using System.Web.Mvc;
namespace MvcMovie.Controllers
{
    public class HelloWorldController : Controller
    {
```



# Ứng dụng MVC cơ bản

## 2. Thêm Controller

```
//
```

```
// GET: /HelloWorld/
```

```
public string Index()
```

```
{ return "This is my <b>default</b> action..."; }
```

```
//
```

```
// GET: /HelloWorld/Welcome/
```

```
public string Welcome()
```

```
{ return "This is the Welcome action method..."; }
```

```
}}
```

# Ứng dụng MVC cơ bản

## 2. Thêm Controller

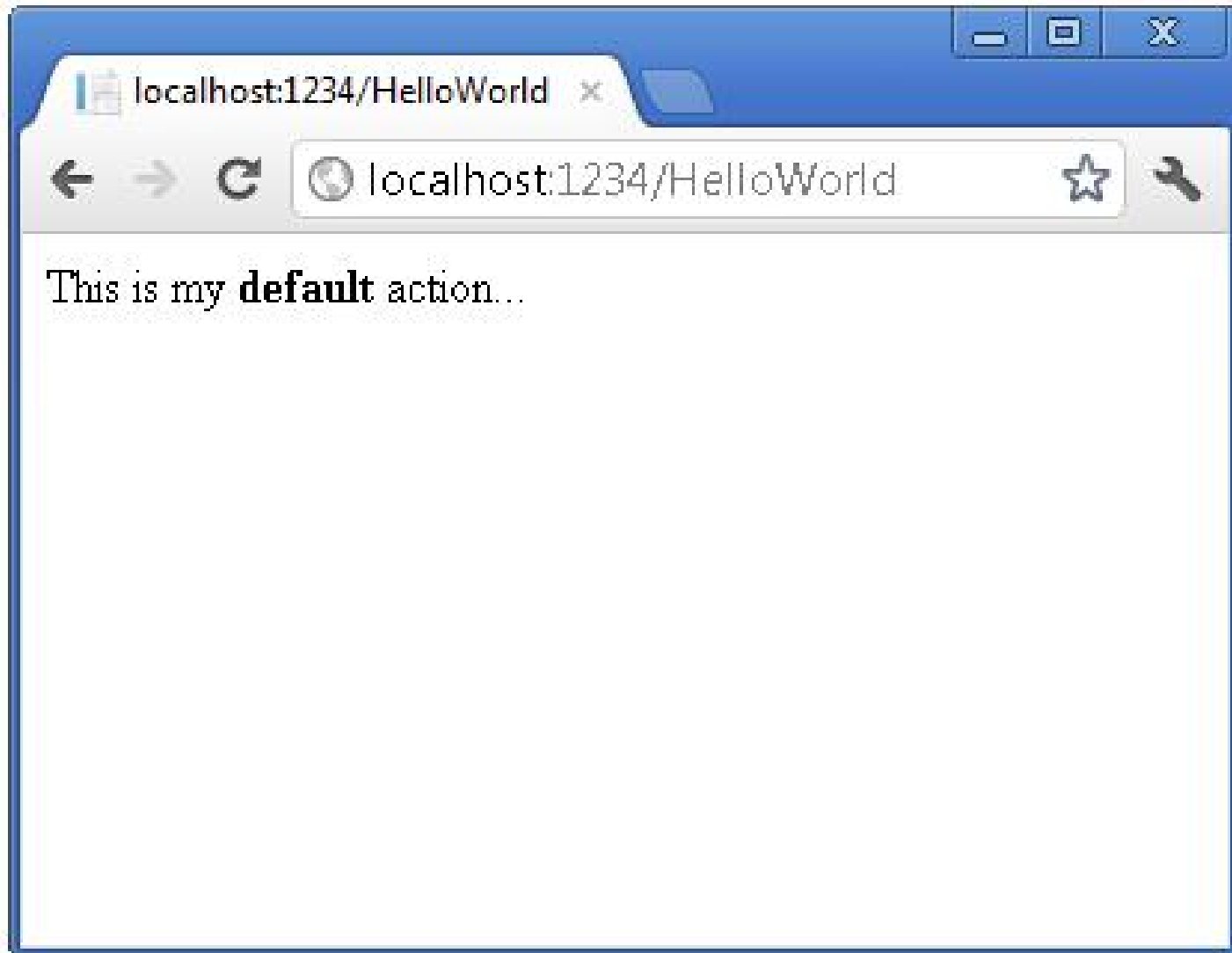
Các phương thức sẽ trả về một chuỗi HTML như đoạn mã trên.

Controller có tên HelloWorldController với phương thức đầu tiên (mặc định) có tên là Index.

Nhấn F5 hoặc Ctrl + F5 để đường dẫn:

<http://localhost:1234/HelloWorld>

# Ứng dụng MVC cơ bản



# Ứng dụng MVC cơ bản

## 1. Thêm Controller

Giải thích cách chạy của đường dẫn

- <http://localhost:1234/HelloWorld>



# Ứng dụng MVC cơ bản

## 2. Thêm Controller

Giải thích cách chạy của đường dẫn

- <http://localhost:1234/HelloWorld>

ASP.NET MVC gọi các lớp controller khác nhau tùy theo URL. URL mặc định dùng bởi ASP.NET MVC như sau:

- **`/[Controller]/[ActionName]/[Parameters]`**

# Ứng dụng MVC cơ bản

## 2. Thêm Controller

Có thể chỉnh lại định dạng đường dẫn ở tập tin **App\_Start/RouteConfig.cs**.

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

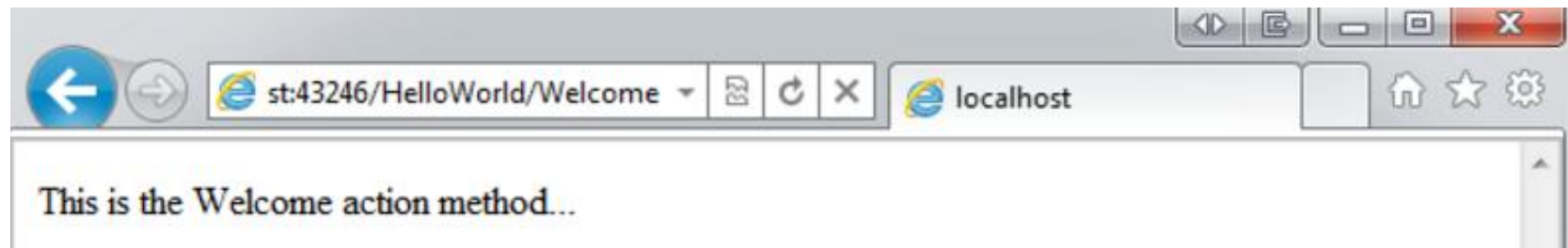
    routes.MapRoute(
        name: "Default",
        url: "{controller}/{action}/{id}",
        defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
    );
}
```

# Ứng dụng MVC cơ bản

## 2. Thêm Controller

Tiếp tục chạy

<http://localhost:xxxx/HelloWorld/Welcome>



# Ứng dụng MVC cơ bản

## 2. Thêm Controller

Thay đổi phương thức Welcome để có thể sử dụng tham số **[Parameters]**

```
public string Welcome(string name, int numTimes = 1)
{
    return HttpUtility.HtmlEncode("Hello " + name +
    ", NumTimes is: " + numTimes);
}
```



# Ứng dụng MVC cơ bản

## 2. Thêm Controller

HttpServerUtility.HtmlEncode dùng để bảo vệ ứng dụng từ các đầu vào nguy hiểm (JavaScript).

# Ứng dụng MVC cơ bản

## 2. Thêm Controller

Lúc này chạy đường dẫn

<http://localhost:xxxx/HelloWorld/Welcome?name=Scott&numtimes=4>



# Ứng dụng MVC cơ bản

## 2. Thêm Controller

Thay phương thức Welcome bằng đoạn code

```
public string Welcome(string name, int ID = 1)
{
    return HttpUtility.HtmlEncode("Hello " +
name + ", ID: " + ID);
}
```

# Ứng dụng MVC cơ bản

## 2. Thêm Controller

Chạy đường dẫn

<http://localhost:xxx/HelloWorld/Welcome/3?name=Rick>



# Ứng dụng MVC cơ bản

## 2. Thêm Controller

Đoạn URL trên khớp tham số ID được đặc tả trong phương thức RegisterRoutes.

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

    routes.MapRoute(
        name: "Default",
        url: "{controller}/{action}/{id}",
        defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
    );
}
```

# Ứng dụng MVC cơ bản

## 2. Thêm Controller

ASP.NET MVC cho phép tùy biến ở RouteConfig để sử dụng các tham số đầu vào khác nhau cho các Controller ở **App\_Start\RouteConfig.cs**.

Thêm 1 đoạn mã ở phương thức **RouteConfig()**

```
routes.MapRoute(  
    name: "Hello",  
    url: "{controller}/{action}/{name}/{id}"  
);
```

# Ứng dụng MVC cơ bản

## 2. Thêm Controller

Sau đó chạy đường dẫn:

`/localhost:XXX/HelloWorld/Welcome/Scott/3`

