



LẬP TRÌNH C# 4

BÀI 3: ASP.NET CORE MVC

www.poly.edu.vn





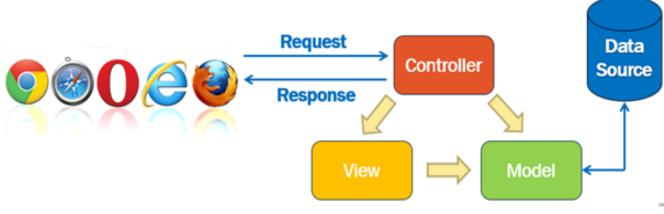
- Mô hình MVC
- Asp.net core MVC template
- Cấu hình Asp.net core MVC







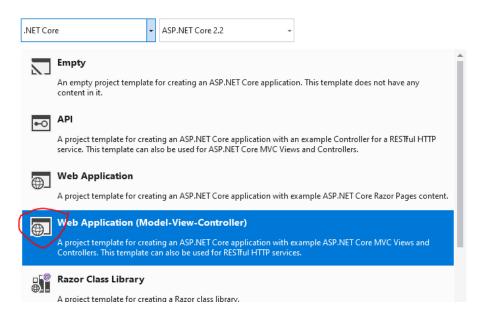
- ☐ MVC viết tắt model-view-controller
- Là một mô hình lập trình chia code thành 3 phần:
 - model triển khai các mô hình dữ liệu, truy cập dữ liệu.
 - view biểu diễn dữ liệu cho người dùng
 - controller nhận các yêu cầu browser gửi đến và trả về http response





- Tạo ứng dụng mvc từ template bằng lệnh: dotnet new mvc projectname
- ☐ Tạo ứng dụng mvc với Vs:

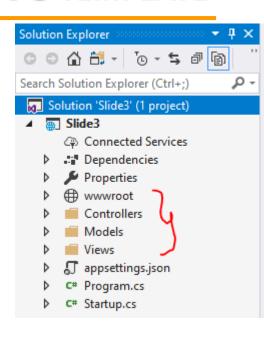
Create a new ASP.NET Core Web Application





☐ Cấu trúc ứng dụng:

- Controllers :chứa các lớp nhận yêu cầu gửi đến và trả về nội dung Http Response
- Models: nơi chứa các lớp cung cấp, xử lý dữ liệu
- ❖ Views: chứa các file .cshtml dựng HTML
- wwwroot: chứa các file tĩnh (hình ảnh, js, css ...). Trong thư mục này đã copy sẵn thư viện CSS Bootstrap và JS JQuery





Úng dụng tự động đăng ký mô hình mvc trong phương thức ConfigureServices từ Startup.cs



Úng dụng tự động tạo middleware sử dụng mô hình mvc trong phương thức Configure từ Startup.cs

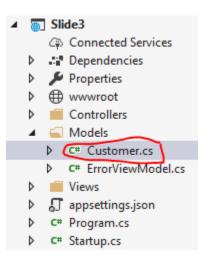
```
public void Configure(IApplicationBuilder app, IHostingEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    app.UseMvc(routes =>
    {
        routes.MapRoute(
            name: "default",
                 template: "{controller=Home}/{action=Index}/{id?}");
    }
}
```





☐ Tạo model Customer.cs

```
namespace Slide3.Models
     3 references
     public class Customer
          2 references | 0 exceptions
          public int CustomerID { get; set; }
          2 references | 0 exceptions
          public string Name { get; set; }
          2 references | 0 exceptions
          public string Address { get; set; }
          1 reference | 0 exceptions
          public Customer()
               CustomerID = 1;
               Name = "Fpoly";
               Address = "Tphcm";
```







☐ Cần hiển thị thông tin Customer nên sẽ tạo view chứa html cùng với các thông tin từ đối tượng

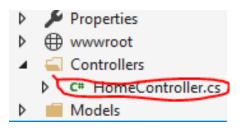
customer

```
C# ErrorViewModel.cs
Index.cshtml* + X Customer.cs
                       HomeController.cs
                                                     Views
         @model Slide3.Models.Customer
             <html>
                                                       Home
    3
                                                        Index.cshtml
             <head>
                <title>Customer Details</title>
    4
                                                        Privacy.cshtml
             </head>
                                                        Shared
             <body>
                                                     ViewImports.cshtml
                <br />
                                                     ViewStart.cshtml
                <br />
                10
                    CustomerID: 
   11
                       \@Model.CustomerID 
   12
   13
                   14
                    Name: 
   15
                       \@Model.Name
   16
                   17
   18
                    Address: 
   19
                       @Model.Address 
   20
                   21
                22
   23
             </body>
                                                                9
             </html>
   24
```





Khi run ứng dụng sẽ gởi request tới controller yêu cầu lấy dữ liệu từ model và trả về view hiển thị, tạo controller với action Index:



```
Customer.cs → X HomeController.cs* → X
Slide3

    Slide3.Controllers.l

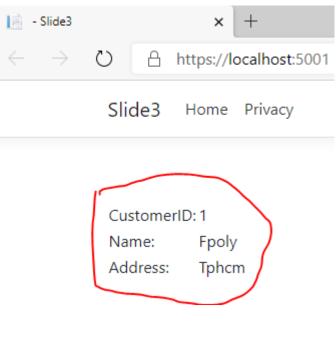
             using System.Linq;
            using System. Threading. Tasks;
             using Microsoft.AspNetCore.Mvc;
            using Slide3.Models;
           □namespace Slide3.Controllers
             {
      9
                 0 references
                 public class HomeController : Controller
     10
     11
                      0 references | 0 requests | 0 exceptions
                      public IActionResult(Index()
     12
     13
                           Customer cus = new Customer();
     14
     15
                           return View(cus);
     16
     17
```





11

Khi run ứng dụng mặc định Index.chtml chạy đầu tiên và gởi request tới controller, controller truy xuất dữ liệu từ model rồi trả ngược lại view index hiển thị:



Lập trình C#4





□ Run ứng dụng chỉ định controller và action cụ thể: localhost:port/controller/action



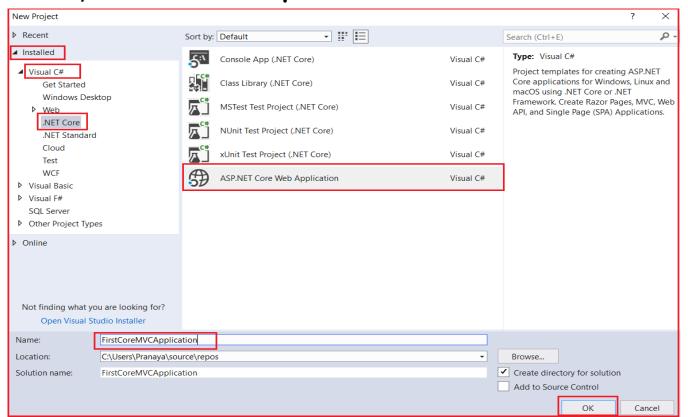






Tự TẠO ASP.NET CORE MVC

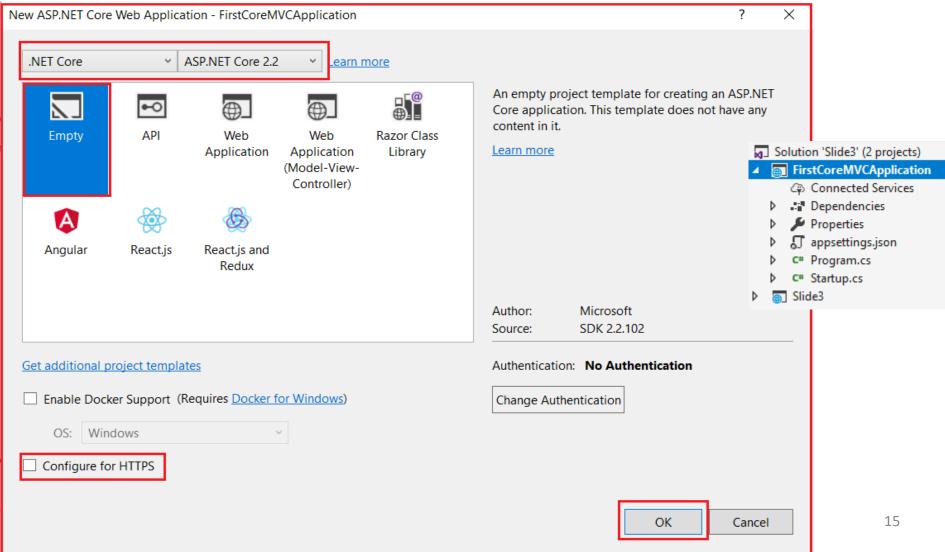
- Có thể tự tạo project Asp.net core mvc mà không cần dùng template mvc gợi ý của Vs.
- Bạn cần tự tay code thêm các middleware, modle, controller, view cho dự án





Tự TẠO ASP.NET CORE MVC

Chọn loại dự án là empty .net core





- ☐ Thêm extension method AddMvc() vào phương thức ConfigureServices trong file Startup.cs
- □ Dùng phương thức AddMvc() sẽ thêm tất cả các service liên quan đến MVC như Authorization, RazorViewEngine, Tag Helper, Data Annotaion, JsonFomatters, Cors...vào service collection

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddMvc();
}
```



THÊM MVC MIDDLEWARE

- Sử dụng phương thức UseMvcWithDefaultRoute để thêm middleware MVC vào phương thức Configure() của Startup class
- UseMvcWithDefaultRoute là một extension method được đặt trong Microsoft.AspNetCore.Mvc.Core.Package và là một phần của Microsoft.AspNetCore.App đã được cài đặt.
- ☐ Phương thức UseMvcWithDefaultRoute() cài đặt MVC Middleware với route mặc định



THÊM MVC MIDDLEWARE

```
public class Startup
    public void ConfigureServices(IServiceCollection services)
        //Configuring the MVC service to the dependency injection container
       services.AddMvc();
    // This method gets called by the runtime. Use this method to
    // configure the HTTP request pipeline.
    public void Configure(IApplicationBuilder app, IHostingEnvironment env)
        if (env.IsDevelopment())
            app.UseDeveloperExceptionPage();
        app.UseMvcwithDefaultRoute();
        app.Run(async (context) =>
            await context.Response.WriteAsync("Hello World!");
        });
```

THÊM MVC MIDDLEWARE

- UseMvcWithDefaultRoute() mặc định sử dụng controller: home và action mặc định:index
- Cho biết kết quả khi run ứng dụng??

This localhost page can't be found

No web page was found for the web address: http://localhost:52190/

Search Google for localhost 52190

HTTP ERROR 404

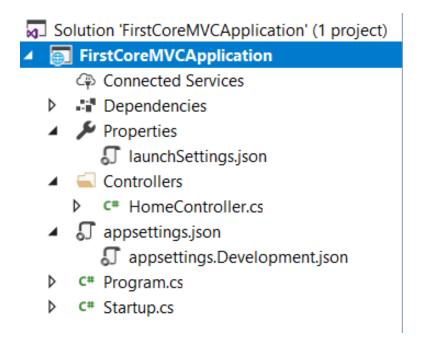
☐ Úng dụng tới thời điểm này chưa tồn tại controller và action, cần thêm controller cho ứng dụng ☐ Solution 'Slide3' (2 projects) ☐ FirstCoreMVCApplication



ADDING HOME CONTROLLER

- ☐ Thêm controller tên Home
- HomeController kế thừa từ Controller và có một phương thức public là Index. Phương thức public này trong controller gọi là Action Methods.

```
public class HomeController : Controller
{
    public IActionResult Index()
    {
       return View();
    }
}
```





ADDING HOME CONTROLLER

☐ Action Methods:

- Action method sẽ gọi View với model để trả về kết quả cho người dùng.
- Action method phải là một public method
- *Action method không thể là static method hoặc một extension method.
- Constructor, getter, setter không được sử dụng.
- Các phương thức được kế thừa không được sử dụng như là một action method
- Action method không được chứa thuộc tính [NonAction]
- Ation method không thể được nạp chồng (overloaded)

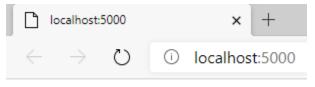


ADDING HOME CONTROLLER

Controller trong MVC pattern có trách nhiệm nhận thông tin request và phản hồi lại cho người dùng. Ví dụ action trả về một chuỗi (không trả về view):

```
public class HomeController
{
    public string Index()
    {
        return "This is Index action from MVC Controller";
    }
}
```

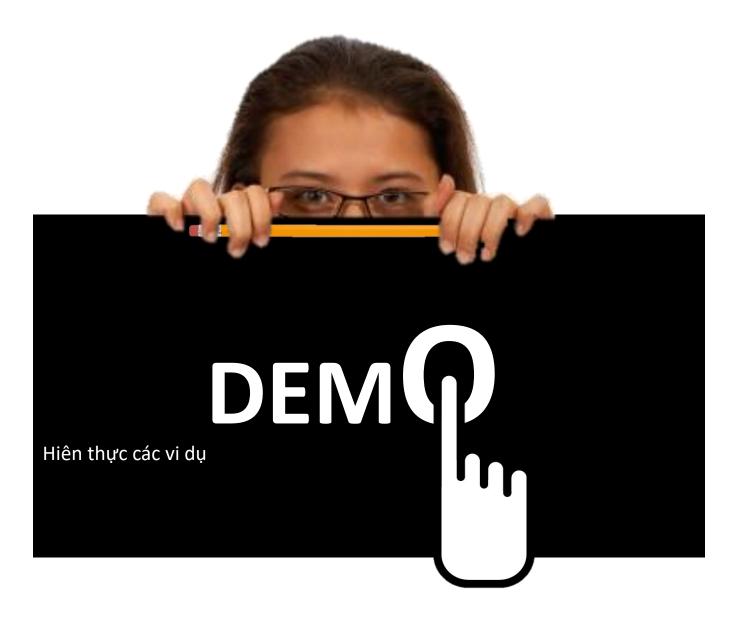
Run dự án



This is Index action from MVC Controller

□ Có thể thay đổi URL với /Home hoặc /Home/Index nó cũng trả về kết quả tương tự.

localhost:5000 home/inde



Lập trình C#4





LẬP TRÌNH C# 4

BÀI 3: ASP.NET CORE MVC (P2)

www.poly.edu.vn





- Các phương thức public của lớp Controller được gọi là Phương thức hành động (Action method)
 - Phương thức Action phải là public. Không thể private hay protected
 - Phương thức Action không thể nap chồng (overloaded)
 - Phương thức Action không thể là phương thức static

Ví dụ sau về phương thức action là Index của lớp

StudentControlle

```
Public class StudentController: Controller

Return type

// GET: Student

public ActionResult Index() Action method

{

return View();

View() defined in base

Controller class
```



- ☐ Tạo action trả về view:
 - Kiểu trả về của các action nên là IActionResult IActionResult hoặc các kế thừa của nó

- ■Tạo view
 - Tất cả các View sẽ nằm trong thư mục Views thuộc thư mục gốc
 - Mỗi Action Method trong Controller sẽ được tạo ra một file view tương ứng giống tên của Action Method

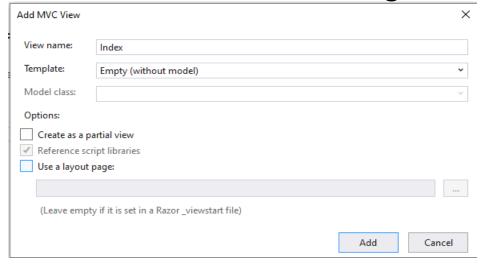


ACTION TRẢ VỀ VIEW

■ Tạo view

Ví dụ HomeController sẽ có một file Index.cshtml trong thư

muc Views/Home





- ■Tạo view
 - Ví dụ thay đổi nội dung Index.cshtml

```
<!DOCTYPE html>
| <a href="html">
| <a href="htm
```

Run ứng dụng, Action method gọi View bằng cách sử dụng phương thức View(), trường hợp này action index tìm view

index.chtml



FPoly Hello from the View



FirstCoreMVCApplicati

Connected Services
Dependencies
Properties
www.root

√img1.jpg

Controllers

Models

- ☐ Ví dụ tạo project hiển thị thông tin các Customer gồm: Image, CustomerID, Name và Address
- □ B1: tạo model với dữ liệu mẫu

```
7 references | 0 exceptions
                                                                                                        C# TestCustomer.cs
                                                       public int CustomerID { get; set; }
public class TestCustomer
                                                       6 references | 0 exceptions
                                                       public string Name { get; set; }
    public List<Customer> customerList;
                                                       6 references | 0 exceptions
                                                       public string Address { get; set; }
    1 reference | 0 exceptions
                                                       2 references | 0 exceptions
    public TestCustomer()
                                                       public string Image { get; set; }
         customerList = new List<Customer>()
             new Customer() { CustomerID = 1, Name = "FpolyHN", Address= "hn", Image = "/Images/img1.jpg"},
             new Customer() { CustomerID = 2, Name = "FpolyHCM", Address= "hcm" },
             new Customer() { CustomerID = 3, Name = "FpolyDN", Address= "dn" },
             new Customer() { CustomerID = 4, Name = "FpolyCT", Address= "ct" },
             new Customer() { CustomerID = 5, Name = "FpolyTN",Address= "tn" }
        };
```

public class Customer



□ B2: tạo CustomerControlle trả về danh sách customer

□ B3:Tạo view và khai báo dùng model

```
@model Slide3.Models.TestCustomer
```

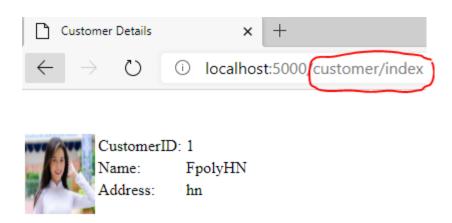


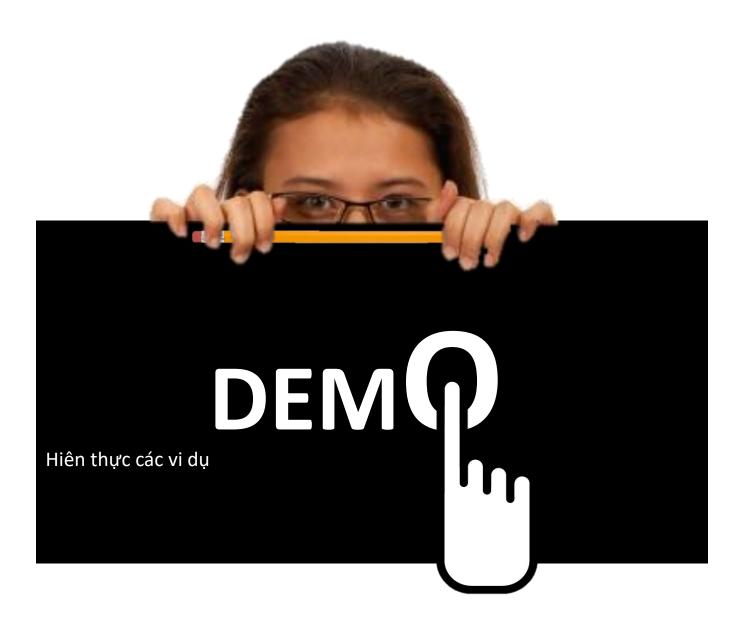
☐ B4: Tạo view hiển thị danh sách customer

```
<head>
   <title>Customer Details</title>
   <style>
      img {float:left;
          width: 70px;
           height: 80px;
   </style>
</head>
I<body>
   <br/>
<br/>
<!--hiển thị custumer đâu tiên trong ds-->
   <img src="@Model. customerList[0].Image" />
   CustomerID: 
          aModel. customerList[0].CustomerID 
      Name: 
          @Model. customerList[0].Name
      Address: 
          @Model. customerList[0].Address 
       <!--Sinh viên code hien thi các customer còn lai-->
</body>
</html>
```



■ B5: chạy ứng dụng với định dạng localhost:port/controller/action



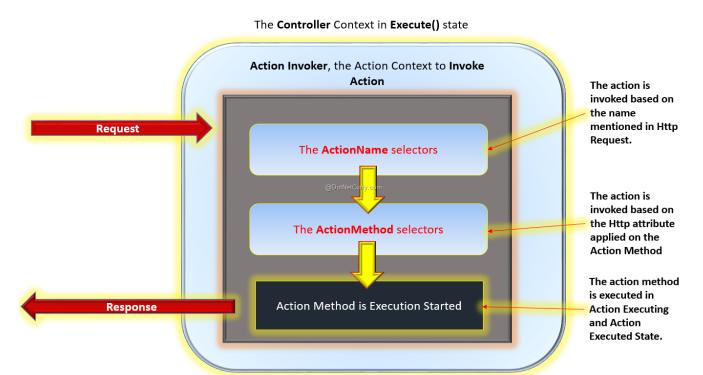


Lập trình C#4





□ Action selector là một attribute có thể được áp dụng vào controller action. Các attribute này giúp Routing Engine chọn đúng action method để xử lý request. ASP.NET Core bao gồm 3 kiểu Action Selector: Action Name, Non Action, Action Verbs





Định nghĩa tên của một action. Routing engine sẽ sử dụng tên này thay vì tên phương thức để match với action name trong routing

```
[ActionName("find")]
public ActionResult GetById(int id)
{
    // get student from the database
    return View();
}
```

□ Ví dụ trên áp dụng ActionName("find") cho action method GetByld→khi run ứng dụng sẽ dùng Action Name: http://localhost/student/find/1



Định một phương thức trong controller nhưng phương thức này không là một action method

```
[NonAction]
public Student GetStudent(int id)
{
   return studentList.Where(s => s.StudentId == id).FirstOrDefault();
}
```

GetStudent(int id) là một phương thức trong controller nhưng nó không phải là action method được gọi trong routing





- Một phương thức action khi được định nghĩa có thể được gọi theo Get và Post dựa trên HTTP Request
- Cho phép lập trình viên có thể định nghĩa hai phương thức khác nhau nhưng cùng tên, một phương thức có thể dùng Http Get và phương thức kia thì có thể dùng HttpPost
- Asp.Net core h\(\tilde{\tilde{0}}\) tr\(\tilde{\tilde{0}}\) nh\(\tilde{0}\) HttpPost, HttpPut, HttpDelete, HttpOptions & HttpPatch.
- Mặc định nếu không sử dụng ActionVerb nào, thì MVC Framework sẽ tự hiểu đó hà HttpGet.



☐ Ví dụ phương thức Edit với HttpGet trả về một Get Request và tạo ra Edit Form. Phương thức Edit với HttpPost nhận Post request và cập nhật database.

Tổng kết bài học

- Mô hình MVC
- Asp.net core MVC template
- ⊙Cấu hình Asp.net core MVC



