

Controller trong ASP.NET MVC

- Controller là gì
- Cách thức làm việc của controller

1

1

Controller

- Quản lý dòng ứng dụng
- Chịu trách nhiệm phân loại các yêu cầu và thực thi mã lệnh tương ứng
- Kết nối các model của ứng dụng và chọn view phù hợp để đáp ứng yêu cầu
- Trong lớp C# lớp controller có namespace là System.Web.Mvc
- Phân biệt chức năng và trình bày

2

2

- Controller đáp ứng:
 - Định vị method phù hợp cho yêu cầu đầu vào
 - Chuẩn hóa dữ liệu yêu cầu trước khi gọi method yêu cầu
 - Tìm dữ liệu cần thiết và chuyển cho method
 - Điều khiển mọi vật cản (exception)
 - Hỗ trợ xuất view phù hợp

3

3

- Lớp ControllerBase của System.Web.Mvc là lớp cơ sở cho mọi controller
- Lớp Controller là mở rộng của lớp ControllerBase
- Để tạo controller sử dụng lớp Controller
- Có thể sử dụng IDE để tạo controller

4

4

- Tạo controller bằng IDE
 - Chuột phải controller
 - Add/ controller/ MVC 5 Controller-Empty/ Add
 - Đặt tên/ Add
- Tạo controller bằng câu lệnh:

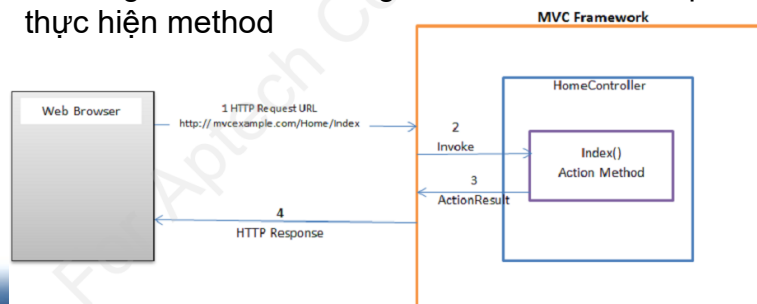

```
using System.Web.Mvc;
public class <Controller_Name>Controller:Controller
{
    //Some code
}
```

5

5

Làm việc với Action Method

- Một controller có thể chứa một hoặc nhiều action method, còn gọi là controller action
- Action method:
 - Xử lý một yêu cầu được gửi đến
 - Thường trả một đối tượng ActionResult là kết quả thực hiện method



6

6

Làm việc với Action Method

- Tạo action

```
public ActionResult <ActionMethod_Name>()  
{  
    /*Code to execute logic and return the result as  
    ActionResult*/  
}
```

7

7

Làm việc với Action Method

- VD:

```
using System.Web.Mvc;  
public class HomeController : Controller  
{  
    public ActionResult Index()  
    {  
        /*Code to execute logic and return the result as ActionResult*/  
    }  
    public ActionResult About()  
    {  
        /*Code to execute logic and return the result as ActionResult*/  
    }  
}
```

8

8

Action Result

- Action Result
 - Là một lớp nền tảng cho các lớp cung cấp các kiểu kết quả khác nhau
 - Gộp các mã HTML cùng với script hai phía server và client để đáp ứng các tương tác của người dùng

9

9

Các kiểu ActionResult

- ViewResult
- PartialViewResult
- EmptyResult
- RedirectResult
- JavascriptResult
- ContentResult
- FileContentResult
- FileStreamResult
- FilePathResult

10

10

Lời gọi Action Method

`http:// <domain_name>
/<controller_name>/<actionmethod_name>`

- `<domain_name>`
- `<controller_name>`
- `<actionmethod_name>`

- VD:

`http:// mvceexample.com/Home/Registration`

11

11

Truyền tham số

- VD:

`http://www.mvceexample.com/student/details?Id=006`

- Details:

```
public ActionResult Details(string Id)
{
    /*Return student records based on the Id
    parameter as an ActionResultobject*/
}
```

12

12

Routing Requests

- Định nghĩa cách ứng dụng xử lý và đáp ứng yêu cầu từ HTTP
- Miêu tả controller action cần cho yêu cầu gửi tới
- Routing là tiến trình ánh xạ yêu cầu đến controller action
 - Xây dựng URL đầu ra tương ứng với controller action
- Cấu hình mẫu route:
 - Tạo mẫu route
 - Ghi mẫu vào bảng route của MVC Framework

13

13

Route mặc định

- Một ứng dụng MVC yêu cầu 1 route để điều khiển yêu cầu người dùng
- Khi tạo một ứng dụng MVC trong VS2015, 1 route tự động được cấu hình trong RouteConfig.cs
- MapRoute() method:

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

14

14

Ghi thông tin route mặc định

- Trong ứng dụng ASP.NET MVC , tệp Global.asax:
 - Khởi tạo ứng dụng với các tính năng của MVC framework khi bắt đầu ứng dụng
 - Chứa lớp MvcApplication với method: Application_Start() khi đăng ký route mặc định

15

15

- Global.asax file:

```
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Http;
using System.Web.Mvc;
using System.Web.Optimization;
using System.Web.Routing;

namespace UrlsAndRoutes {
    public class MvcApplication : System.Web.HttpApplication {
        protected void Application_Start() {
            RouteConfig.RegisterRoutes(RouteTable.Routes);
            /*Code for registering other MVC components*/
        }
    }
}
```

16

16

URL

- Được xác định khi tạo một route
- Được đối chiếu với URL của yêu cầu bởi route engine của MVC
- Chứa các giá trị ngữ nghĩa và placeholder phân cách bởi dấu "/"

"{controller}/{action}/{id}"

- Tuân theo mẫu trước:
<http://www.mvcexample.com/student/records/36>
- Khi routing engine kiểm tra sự phù hợp giữa URL trước và phần URL, sẽ thực hiện :
 - Ấn định student cho {controller}
 - Ấn định records cho {action}
 - Ấn định 36 cho {id}
- Một URL có thể kết hợp giá trị và placeholder:
"student/{action}/{id}"

17

17

Sắp xếp routes

- Đôi khi cần đăng ký nhiều routes cho một ứng dụng ASP.NET MVC
- Route engine ánh xạ URL với mẫu URL

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

```
routes.MapRoute(  
    name: "manager",  
    url: "Manager/{action}",  
    defaults: new { controller = "Manager",  
        action = "Browse"
```

```
);
```

18

18

Ràng buộc route

- Routing engine cho phép áp dụng các ràng buộc giá trị placeholder

```
routes.MapRoute(  
    "Product",  
    "{controller}/{action}/{id}",  
    new { controller = "Product", action =  
        "Browse", id =  
        UrlParameter.Optional },  
    new { id = "(|Jewellery|Jeans|Mobile)" }  
);
```

19

19

Ràng buộc route

- VD: id có giá trị integer

```
routes.MapRoute(  
    name: "Product",  
    url: "{controller}/{action}/{id}",  
    defaults: new { controller =  
        "Product", action = "Browse",  
        id=UrlParameter.  
        Optional},  
    constraints: new { id = @"\d*" }  
);
```

20

20

Bỏ qua route

- Phương thức `IgnoreRoute()` của lớp `RouteTable` xác định route được bỏ qua
`routes.IgnoreRoute("{resource}.axd/{*pathInfo}");`