Controllers in an mvc application-

where is this mapping defined

The mapping is defined in Global.asax. Notice that in Global.asax we have RegisterRoutes() method.

RouteConfig.RegisterRoutes(RouteTable.Routes);

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 }

);

}

----------------------------------------

Controller-

// GET: Home //get request

public ActionResult Index() //Action method // by default

//{

// return View();

//}

public string Index() //Action method

{

return "Welcome to MVC";

}

path-http://localhost:3220/Home/Login

public string Login() //Action method

{

return "User login page";

}

path-http://localhost:3220/Home/Login

----------------------------------------------------

public string Login(string id) //Action method

{

return "User login page with" + id;

}

http://localhost:3220/Home/Login/10 //no error

http://localhost:3220/Home/Login/sunny //no error

http://localhost:3220/Home/Login?id=sunny //no error

http://localhost:3220/Home/Login?id=10 //no error

------------------------------------------------

It does not accept string parameter it take only int parameter but if we use it with ? then it take both string as well as int.

public string Login(string name) //Action method

{

return "User login page with"+name;

}

}

By using querystring we get the value

home/login?name="amit" //no error

http://localhost:3220/Home/Login?name=1 //no error

http://localhost:3220/Home/Login/sunny //error

http://localhost:3220/Home/Login/10 //error

--------------------------------------------------

public string Login(int id) //Action method

{

return "User login page with"+id;

}

error

......

public string Login(int? id) //Action method

{

return "User login page with"+id;

}

make it nullable

------------------------------------------------

public string Index(string id, string name)

{

return "The value of Id = " + id + " and Name = " + name;

}

http://localhost/MVCDemo/home/index/10?name=Pragim

Just like web forms, you can also use "Request.QueryString"

public string Index(string id)

{

return "The value of Id = " + id + " and Name = " + Request.QueryString["name"];

}

----------------------------------------------------

View in mvc-

public List<string> Index()

{

return new List<string>()

{

"India",

"US",

"UK",

"Canada"

};

}

Run the application and notice that the output is not as expected.

System.Collections.Generic.List`1[System.String]

.....

public ActionResult Index()

{

// Store the list of Countries in ViewBag.

ViewBag.Countries = new List<string>()

{

"India",

"US",

"UK",

"Canada"

};

// Finally return a view

return View();

}

...

@{

ViewBag.Title = "Countries List";

}

<h2>Countries List</h2>

<ul>

@foreach (string strCountry in ViewBag.Countries)

{

<li>@strCountry</li>

}

</ul>

We use "@" symbol to switch between html and c# code

---------------------------------------------------------------

ViewBag & ViewData is a mechanism to pass data from the controller to the view.

ViewData is a dictionary of objects that are stored and retrieved using strings as keys. The syntax of ViewData is very similar to that of ViewState, SessionState and ApplicationState.

"View state" is specific to a page in a session whereas "Session state" is specific to a user or browser that can be accessed across all pages in the web application.

// Storing data in ViewData

ViewData["YourData"] = "SomeData";

// Retrieving data from ViewData

string strData = ViewData["YourData"].ToString();

ViewData does not provide compile time error checking. For example, if you mis-spell the key names you wouldn't get any compile time error. You get to know about the error only at runtime.

ViewBag uses the dynamic feature that was introduced in to C# 4.0. It allows an object to have properties dynamically added to it. Using ViewBag the above code can be rewritten as below.

// Storing data in ViewBag

ViewBag.YourData = "SomeData";

// Retrieving data from ViewBag

string strData = ViewBag.YourData;

Just like ViewData, ViewBag does not provide compile time error checking. For example, if you mis-spell the property name, you wouldn't get any compile time error. You get to know about the error only at runtime.

Internally ViewBag properties are stored as name/value pairs in the ViewData dictionary.

-----------------------------------------------------------------------------------

Models

Employee.cs-Add models

public class Employee

{

public int EmployeeId { get; set; }

public string Name { get; set; }

public string Gender { get; set; }

public string City { get; set; }

}

.........

using MVCDemo.Models;

EmployeeController-Add controller

public ActionResult Details()

{

Employee employee = new Employee()

{

EmployeeId = 101,

Name = "John",

Gender = "Male",

City = "London"

};

return View();

}

..........

Details.cshtml -Add view-select model employee(app name)

@{

ViewBag.Title = "Employee Details";

}

<h2>Employee Details</h2>

<table style="font-family:Arial">

<tr>

<td>

Employee ID:

</td>

<td>

@Model.EmployeeId

</td>

</tr>

<tr>

<td>

Name:

</td>

<td>

@Model.Name

</td>

</tr>

<tr>

<td>

Gender:

</td>

<td>

@Model.Gender

</td>

</tr>

<tr>

<td>

City:

</td>

<td>

@Model.City

</td>

</tr>

</table>

..........

return View(employee);

------------------------------------------------------------------------------

The controller responds to URL request, gets data from a model and hands it over to the view. The view then renders the data. Model can be entities or business objects.

------------------------------------------------

controller to view

public ActionResult Index() //Action method // by default

{

ViewBag.UserName = "Amit";

return View();

}

index.cshtml

<div>

<h1>Hello,@ViewBag.UserName ,Welcome</h1>

</div>

------------------

http://localhost:12980/Home/Index/sunny

public ActionResult Index(string id) //Action method // by default

{

ViewBag.UserName = id;

return View();

}

index.cshtml

<div>

<h1>Hello,@ViewBag.UserName ,Welcome</h1>

</div>

...........................

http://localhost:12980/Home/Index/deep?city=Noida

public ActionResult Index(string id,string city) //Action method // by default

{

ViewBag.UserName = id;

ViewData["User"] = city;

return View();

}

index.cshtml

<div>

<h1>Hello @ViewBag.UserName ,Welcome to @ViewData["User"] </h1>

</div>

------------------------------------

passing collection controller to view

public ActionResult Index(string id,string city) //Action method // by default

{

ViewBag.UserName = id;

ViewData["User"] = city;

List<string> cities = new List<string> { "delhi", "agra", "noida" }; //cities-list object

ViewBag.Cities = cities; //Cities-property name

return View();

}

..........

<div>

<h1>Hello,@ViewBag.UserName ,Welcome to @ViewData["User"] </h1>

<h2>Your favourite cities</h2>

<ul>

@foreach (var city in ViewBag.Cities)

{

<li>@city</li>

}

</ul>

</div>

-------------------------------

http://localhost:12980/Home/Contact

homecontroller.cs

public ActionResult Contact()

{

List<string> cities = new List<string> { "delhi", "agra", "noida" }; //cities-list object

return View(cities);

}

.........

@model IEnumerable<string>

@{

ViewBag.Title = "Contact";

}

contact.cshtml

<h2>Contact</h2>

<ul>

@foreach (var city in Model)

{

<li>@city</li>

}

</ul>

Add view contact

check the layout page

---------------------------

create property

prop double tab

1) Explain what is Model-View-Controller?

MVC is a software architecture pattern for developing web application. It is handled by three objects Model-View-Controller.

2) Mention what does Model-View-Controller represent in an MVC application?

In an MVC model,

Model– It represents the application data domain. In other words applications business logic is contained within the model and is responsible for maintaining data

View– It represents the user interface, with which the end users communicates. In short all the user interface logic is contained within the VIEW

Controller– It is the controller that answers to user actions. Based on the user actions, the respective controller responds within the model and choose a view to render that display the user interface. The user input logic is contained with-in the controller

3) Explain in which assembly is the MVC framework is defined?

The MVC framework is defined in System.Web.Mvc.

4) List out few different return types of a controller action method?

View Result

Javascript Result

Redirect Result

Json Result

Content Result

5) Mention what is the difference between adding routes, to a webform application and an MVC application?

To add routes to a webform application, we can use MapPageRoute() method of the RouteCollection class, where adding routes to an MVC application, you can use MapRoute() method.

mvc-design-diagram

6) Mention what are the two ways to add constraints to a route?

The two methods to add constraints to a route is

Use regular expressions

Use an object that implements IRouteConstraint Interface

7) Mention what is the advantages of MVC?

MVC segregates your project into a different segment, and it becomes easy for developers to work on

It is easy to edit or change some part of your project that makes project less development and maintenance cost

MVC makes your project more systematic

8) Mention what “beforFilter()”,“beforeRender” and “afterFilter” functions do in Controller?

beforeFilter(): This function is run before every action in the controller. It’s the right place to check for an active session or inspect user permissions.

beforeRender(): This function is called after controller action logic, but before the view is rendered. This function is not often used, but may be required If you are calling render() manually before the end of a given action

afterFilter(): This function is called after every controller action, and after rendering is done. It is the last controller method to run

9) Explain the role of components Presentation, Abstraction and Control in MVC?

Presentation: It is the visual representation of a specific abstraction within the application

Abstraction: It is the business domain functionality within the application

Control: It is a component that keeps consistency between the abstraction within the system and their presentation to the user in addition to communicating with other controls within the system

10) Mention the advantages and disadvantages of MVC model?

Advantages Disadvantages

It represents clear separation between business logic and presentation logic

Each MVC object has different responsibilities

The development progresses in parallel

Easy to manage and maintain

All classes and object are independent of each other

Disadvantages

The model pattern is little complex

Inefficiency of data access in view

With modern user interface, it is difficult to use MVC

You need multiple programmers for parallel development

Multiple technologies knowledge is required

11) Explain the role of “ActionFilters” in MVC?

In MVC “ ActionFilters” help you to execute logic while MVC action is executed or its executing.

12) Explain what are the steps for the execution of an MVC project?

The steps for the execution of an MVC project includes

Receive first request for the application

Performs routing

Creates MVC request handler

Create Controller

Execute Controller

Invoke action

Execute Result

13) Explain what is routing? What are the three segments for routing is important?

Routing helps you to decide a URL structure and map the URL with the Controller.

The three segments that are important for routing is

ControllerName

ActionMethodName

Parameter

14) Explain how routing is done in MVC pattern?

There is a group of routes called the RouteCollection, which consists of registered routes in the application.

The RegisterRoutes method records the routes in this collection. A route defines a URL pattern and a handler to use if the request matches the pattern. The first parameter to the MapRoute method is the name of the route. The second parameter will be the pattern to which the URL matches. The third parameter might be the default values for the placeholders if they are not determined.

15) Explain using hyperlink how you can navigate from one view to other view?

By using “ActionLink” method as shown in the below code. The below code will make a simple URL which help to navigate to the “Home” controller and invoke the “GotoHome” action.

Collapse / Copy Code

<%= Html.ActionLink(“Home”, “Gotohome”) %>

16) Mention how can maintain session in MVC?

Session can be maintained in MVC by three ways tempdata, viewdata, and viewbag.

17) Mention what is the difference between Temp data, View, and View Bag?

Temp data: It helps to maintain data when you shift from one controller to other controller.

View data: It helps to maintain data when you move from controller to view

View Bag: It’s a dynamic wrapper around view data

18) What is partial view in MVC?

Partial view in MVC renders a portion of view content. It is helpful in reducing code duplication. In simple terms, partial view allows to render a view within the parent view.

19) Explain how you can implement Ajax in MVC?

In Ajax, MVC can be implemented in two ways

Ajax libraries

Jquery

20) Mention what is the difference between “ActionResult” and “ViewResult” ?

“ActionResult” is an abstract class while “ViewResult” is derived from “AbstractResult” class. “ActionResult” has a number of derived classes like “JsonResult”, “FileStreamResult” and “ViewResult” .

“ActionResult” is best if you are deriving different types of view dynamically.

21) Explain how you can send the result back in JSON format in MVC?

In order to send the result back in JSON format in MVC, you can use “JSONRESULT” class.

22) Explain what is the difference between View and Partial View?

View Partial View

It contains the layout page

Before any view is rendered, viewstart page is rendered

View might have markup tags like body, html, head, title, meta etc.

View is not lightweight as compare to Partial View

It does not contain the layout page

Partial view does not verify for a viewstart.cshtml. We cannot put common code for a partial view within the viewStart.cshtml.page

Partial view is designed specially to render within the view and just because of that it does not consist any mark up

We can pass a regular view to the RenderPartial method

23) List out the types of result in MVC?

In MVC, there are twelve types of results in MVC where “ActionResult” class is the main class while the 11 are their sub-types

ViewResult

PartialViewResult

EmptyResult

RedirectResult

RedirectToRouteResult

JsonResult

JavaScriptResult

ContentResult

FileContentResult

FileStreamResult

FilePathResult

24) Mention what is the importance of NonActionAttribute?

All public methods of a controller class are treated as the action method if you want to prevent this default method then you have to assign the public method with NonActionAttribute.

25) Mention what is the use of the default route {resource}.axd/{\*pathinfo} ?

This default route prevents request for a web resource file such as Webresource.axd or ScriptResource.axd from being passed to the controller.

26) Mention the order of the filters that get executed, if the multiple filters are implemented?

The filter order would be like

Authorization filters

Action filters

Response filters

Exception filters

27) Mention what filters are executed in the end?

In the end “Exception Filters” are executed.

28) Mention what are the file extensions for razor views?

For razor views the file extensions are

.cshtml: If C# is the programming language

.vbhtml: If VB is the programming language

29) Mention what are the two ways for adding constraints to a route?

Two methods for adding constraints to route is

Using regular expressions

Using an object that implements IRouteConstraint interface

30) Mention two instances where routing is not implemented or required?

Two instance where routing is not required are

When a physical file is found that matches the URL pattern

When routing is disabled for a URL pattern

31) Mention what are main benefits of using MVC?

There are two key benefits of using MVC

As the code is moved behind a separate class file, you can use the code to a great extent

As behind code is simply moved to.NET class, it is possible to automate UI testing. This gives an opportunity to automate manual testing and write unit tests.

The main advantages of ASP.net MVC are:

Enables the full control over the rendered HTML.

Provides clean separation of concerns(SoC).

Enables Test Driven Development (TDD).

Easy integration with JavaScript frameworks.

Following the design of stateless nature of the web.

RESTful urls that enables SEO.

No ViewState and PostBack events

The main advantage of ASP.net Web Form are:

It provides RAD development

Easy development model for developers those coming from winform development.

Asp.Net Web Forms Asp.Net MVC

Asp.Net Web Form follow a traditional event driven development model. Asp.Net MVC is a lightweight and follow MVC (Model, View, Controller) pattern based development model.

Asp.Net Web Form has server controls. Asp.Net MVC has html helpers.

Asp.Net Web Form supports view state for state management at client side. Asp.Net MVC does not support view state.

Asp.Net Web Form has file-based URLs means file name exist in the URLs must have its physically existence. Asp.Net MVC has route-based URLs means URLs are divided into controllers and actions and moreover it is based on controller not on physical file.

Asp.Net Web Form follows Web Forms Syntax Asp.Net MVC follow customizable syntax (Razor as default)

In Asp.Net Web Form, Web Forms(ASPX) i.e. views are tightly coupled to Code behind(ASPX.CS) i.e. logic. In Asp.Net MVC, Views and logic are kept separately.

Asp.Net Web Form has Master Pages for consistent look and feels. Asp.Net MVC has Layouts for consistent look and feels.

Asp.Net Web Form has User Controls for code re-usability. Asp.Net MVC has Partial Views for code re-usability.

Asp.Net Web Form has built-in data controls and best for rapid development with powerful data access. Asp.Net MVC is lightweight, provide full control over markup and support many features that allow fast & agile development. Hence it is best for developing interactive web application with latest web standards.

Asp.Net Web Form is not Open Source. Asp.Net Web MVC is an Open Source.

http://localhost:54350

http://localhost:54350/Home/index

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace WebApplication1.Controllers

{

public class HomeController : Controller

{

// GET: Home

public string Index()

{

return "Hello World, this is ASP.Net MVC Tutorials";

}

public string GetCurrentTime()

{

return DateTime.Now.ToString("T");

}

}

}

--------------------------------------------------------------------

Green-class

white-method

RouteConfig.RegisterRoutes

---------------------------------------------

routes.MapRoute(

"Process", "Process/{action}/{id}",

defaults: new{

controller = " Process", action = "List ", id =

UrlParameter.Optional});

routes.MapRoute(

name: "Default", url: "{controller}/{action}/{id}",

defaults: new{

controller = "Home", action = "Index", id =

UrlParameter.Optional});

------------------------------------------------------

routes.MapRoute(

"Employee", "Employee/{name}", new{

controller = "Employee", action = "Search", name = UrlParameter.Optional });

...................

Employeecontroller.cs

public ActionResult Search(string name){

var input = Server.HtmlEncode(name);

return Content(input);

//Server.HtmlEncode will simply convert any kind of malicious script in plain text.

http://localhost:61465/Employee/Mark

---------------------------------------------------------

customercontroller

public string GetAllCustomers(){

return @"<ul>

<li>Ali Raza</li>

<li>Mark Upston</li>

<li>Allan Bommer</li>

<li>Greg Jerry</li>

</ul>";

}

http://localhost:61465/Customer/GetAllCustomers.

public class HomeController : Controller{

// GET: Home

public ActionResult Index(){

return RedirectToAction("GetAllCustomers","Customer");

--------------------------------------------------------------------------

ActionName attribute(filter)-

http://localhost:54350/Home/currentTime

[ActionName("CurrentTime")]

public string GetCurrentTime()

{

return DateTime.Now.ToString("T");

}

-----------------------------------------------------------------------------

[ActionName("CurrentTime")]

public string GetCurrentTime(){

return TimeString();

}

[NonAction]

public string TimeString(){

return "Time is " + DateTime.Now.ToString("T");

}

http://localhost:62833/Home/CurrentTime ->working

http://localhost:62833/Home/TimeString ->not working

+---------------------------------------------------------------------------------

httpverbs give yo choice which action you want to perform with the same name methods

public ActionResult Search(string name)

{

var input = Server.HtmlEncode(name);

return Content(input);

}

[HttpGet]

public ActionResult Search()

{

var input = "Another Search action";

return Content(input);

}

--------------------------------------------------------------------------------

we have built **Employee**entity.  
public class Employee  
{  
    public int EmployeeId { get; set; }  
    public string Name { get; set; }  
    public string Gender { get; set; }  
    public string City { get; set; }  
}   
  
  
  
In this video, we will discuss, retrieving data from a database table **tblEmployee**using entity framework. In a later video, we will discuss using business objects as our model.  
  
**Step 1:** Install entity framework, if you don't have it installed already on your computer. At the time of this recording the latest version is 5.0.0.0. Using nuget package manager, is the easiest way to install. A reference to EntityFramework.dll is automatically added.  
Open visual studio > Tools > Library Package Manager > Manage NuGet Packages for Solution  
  
**Step 2:** Add **EmployeeContext.cs** class file to the Models folder. Add the following **"using"** declaration.  
using System.Data.Entity;  
  
**Copy & paste the following code in EmployeeContext.cs**  
public class EmployeeContext : DbContext  
{  
    public DbSet<Employee> Employees {get; set;}  
}  
  
**EmployeeContext**class derives from **DbContext** class, and is responsible for establishing a connection to the database. So the next step, is to include connection string in web.config file.  
  
**Step 3:** Add a connection string, to the web.config file, in the root directory.  
<connectionStrings>  
  <add name="EmployeeContext"   
        connectionString="server=.; database=Sample; integrated security=SSPI"  
        providerName="System.Data.SqlClient"/>  
</connectionStrings>  
  
**Step 4:** Map **"Employee"** model class to the database table, tblEmployee using **"Table"** attribute as shown below.  
[Table("tblEmployee")]  
public class Employee  
{  
    public int EmployeeId { get; set; }  
    public string Name { get; set; }  
    public string Gender { get; set; }  
    public string City { get; set; }  
}  
  
**Note:** "Table" attribute is present in "System.ComponentModel.DataAnnotations.Schema" namespace.  
  
**Step 5:** Make the changes to "Details()" action method in "EmployeeController" as shown below.  
public ActionResult Details(int id)  
{  
    EmployeeContext employeeContext = new EmployeeContext();  
    Employee employee = employeeContext.Employees.Single(x => x.EmployeeId == id);  
  
    return View(employee);  
}  
  
**Step 6:** Finally, copy and paste the following code in **Application\_Start**() function, in **Global.asax** file. Database class is present "in System.Data.Entity" namespace. Existing databases do not need, database initializer so it can be turned off.  
Database.SetInitializer<MVCDemo.Models.EmployeeContext>(null);  
  
**That's it**, run the application and navigate to the following URL's and notice that the relevant employee details are displayed as expected  
http://localhost/MVCDemo/Employee/details/1  
http://localhost/MVCDemo/Employee/details/2