

DOT NET INTERVIEW QUESTIONS AND ASNWERS

C# AND MVC QUESTIONS

ABHISHEK SHARMA

.NET FSE TRAINER

1. What is C-Sharp (C#)?

C# is a type-safe, managed and object-oriented language, which is compiled by .Net framework for generating intermediate language (IL).

2. Explain the types of comments in C#?

Below are the types of comments in C# –

- Single Line Comment Eg : //
- Multiline Comments Eg: /* */
- XML Comments Eg : ///

3. List out the differences between Array and ArrayList in C#?

Array stores the values or elements of same data type but arraylist stores values of different datatypes.

Arrays will use the fixed length but arraylist does not uses fixed length like array.

4. Why to use “using” in C#?

“Using” statement calls – “dispose” method internally, whenever any exception occurred in any method call and in “Using” statement objects are read only and cannot be reassignable or modifiable.

5. Explain namespaces in C#?

Namespaces are containers for the classes. We will use namespaces for grouping the related classes in C#. “Using” keyword can be used for using the namespace in other namespace.

6. Explain “static” keyword in C#?

“Static” keyword can be used for declaring a static member. If the class is made static then all the members of the class are also made static. If the variable is made static then it will have a single instance and the value change is updated in this instance.

7. Why to use “finally” block in C#?

“Finally,” block will be executed irrespective of exception. So while executing the code in try block when exception is occurred, control is returned to catch block and at last “finally” block will be executed. So, closing connection to database / releasing the file handlers can be kept in “finally” block.

8. Can we have only “try” block without “catch” block in C#?

Yes we can have only try block without catch block.

9. What is the difference between “out” and “ref” parameters in C#?

“out” parameter can be passed to a method and it need not be initialized where as “ref” parameter has to be initialized before it is used.

10. Explain Jagged Arrays in C#?

If the elements of an array is an array then it’s called as jagged array. The elements can be of different sizes and dimensions.

11. What you mean by inner exception in C#?

Inner exception is a property of exception class which will give you a brief insight of the exception i.e, parent exception and child exception details.

12. Explain circular reference in C#?

This is a situation where in, multiple resources are dependent on each other and this causes a lock condition and this makes the resource to be unused.

13. What you mean by delegate in C#?

Delegates are type safe pointers unlike function pointers as in C++. Delegate is used to represent the reference of the methods of some return type and parameters.

14. What are the differences between events and delegates in C#?

Main difference between event and delegate is event will provide one more of encapsulation over delegates. So when you are using events destination will listen to it but delegates are naked, which works in subscriber/destination model.

15. Is C# code is unmanaged or managed code?

C# code is managed code because the compiler – CLR will compile the code to Intermediate Language.

16. Why to use lock statement in C#?

Lock will make sure one thread will not intercept the other thread which is running the part of code. So lock statement will make the thread wait, block till the object is being released.

17. Explain Partial Class in C#?

Partial classes concept added in .Net Framework 2.0 and it allows us to split the business logic in multiple files with the same class name along with “partial” keyword.

18. What is Thread in C#?

Thread is an execution path of a program. Thread is used to define the different or unique flow of control. If our application involves some time consuming processes then it's better to use Multithreading., which involves multiple threads.

19. Can Multiple Inheritance implemented in C# ?

In C#, derived classes can inherit from one base class only. If you want to inherit from multiple base classes, use interface.

20. What are the uses of delegates in C#?

Below are the list of uses of delegates in C# –

Callback Mechanism

Asynchronous Processing

- Abstract and Encapsulate method
- Multicasting

21. What is enum in C#?

enum keyword is used for declaring an enumeration, which consists of named constants and it is called as enumerator lists. Enums are value types in C# and these can't be inherited. Below is the sample code of using Enums

Eg: enum Fruits { Apple, Orange, Banana, WaterMelon};

22. Write a sample code to write the contents to text file in C#?

Below is the sample code to write the contents to text file –

Using System.IO;

File.WriteAllText("mytextfilePath", "MyTestContent");

23. Explain Anonymous type in C#?

This is being added in C# 3.0 version. This feature enables us to create an object at compile time. Below is the sample code for the same –

Var myTestCategory = new { CategoryId = 1, CategoryName = "Category1"};

24. Which string method is used for concatenation of two strings in c#?

“Concat” method of String class is used to concatenate two strings. For example,

```
string.Concat(firstStr, secStr)
```

25. What is overriding in c# ?

To override a base class method which is defined as virtual, Override keyword is used. In the above example, method DriveType is overridden in the derived class.

26. What is ASP.Net?

It is a framework developed by Microsoft on which we can develop new generation web sites using web forms(aspx), MVC, HTML, Javascript, CSS etc. Its successor of Microsoft Active Server Pages(ASP). Currently there is ASP.NET 4.0, which is used to develop web sites. There are various page extensions provided by Microsoft that are being used for web site development.

Eg: aspx, asmx, ascx, ashx, cs, vb, html, XML etc.

27. What's the use of Response.Output.Write()?

We can write formatted output using Response.Output.Write().

28. What is caching?

Caching is a technique used to increase performance by keeping frequently accessed data or files in memory. The request for a cached file/data will be accessed from cache instead of actual location of that file.

29. What is Cross Page Posting?

When we click submit button on a web page, the page post the data to the same page. The technique in which we post the data to different pages is called Cross Page posting. This can be achieved by setting POSTBACKURL property of the button that causes the postback. Findcontrol method of PreviousPage can be used to get the posted values on the page to which the page has been posted.

30. What is RedirectPermanent in ASP.Net?

RedirectPermanent Performs a permanent redirection from the requested URL to the specified URL. Once the redirection is done, it also returns 301 Moved Permanently responses.

31. Differentiate globalization and localization.

The globalization is a technique to identify the specific part of a Web application that is different for different languages and make separate that portion from the core of the Web application. The localization is a procedure of configuring a Web application to be supported for a specific language or locale.

32. What is a Cookie? Where is it used in ASP.NET?

Cookie is a lightweight executable program, which the server posts to client machines. Cookies store the identity of a user at the first visit of the Web site and validate them later on the next visits for their authenticity. The values of a cookie can be transferred between the user's request and the server's response.

33. What is the default timeout for a Cookie?

The default time duration for a Cookie is 30 minutes.

34. How does a content page differ from a master page?

A content page does not have complete HTML source code; whereas a master page has complete HTML source code inside its source file.

35. Explain how Cookies work. Give an example of Cookie abuse.

The server tells the browser to put some files in a cookie, and the client then sends all the cookies for the domain in each request. An example of cookie abuse is large cookies affecting the network traffic.

36. What are the advantages of the code-behind feature?

The code-behind feature of ASP.NET offers a number of advantages:

- Makes code easy to understand and debug by separating application logic from HTML tags
- Provides the isolation of effort between graphic designers and software engineers
- Removes the problems of browser incompatibility by providing code files to exist on the Web server and supporting Web pages to be compiled on demand.

37. What is an ASP.NET Web Form?

ASP.NET Web forms are designed to use controls and features that are almost as powerful as the ones used with Windows forms, and so they are called as Web forms. The Web form uses a server-side object model that allows you to create functional controls, which are executed on the server and are rendered as HTML on the client. The attribute, `runat="server"`, associated with a server control indicates that the Web form must be processed on the server.

38. What is IIS? Why is it used?

Internet Information Services (IIS) is created by Microsoft to provide Internet-based services to ASP.NET Web applications. It makes your computer to work as a Web server and provides the functionality to develop and deploy Web applications on the server. IIS handles the request and response cycle on the Web server. It also offers the services of SMTP and FrontPage server extensions. The SMTP is used to send emails and use FrontPage server extensions to get the dynamic features of IIS, such as form handler.

39. What is Query String? What are its advantages and limitations?

The Query String helps in sending the page information to the server.

The Query String has the following advantages:

- Every browser works with Query Strings.
- It does not require server resources and so does not exert any kind of burden on the server.

The following are the limitations of Query String:

- Information must be within the limit because URL does not support many characters.
- Information is clearly visible to the user, which leads to security threats.

40. What is the difference between authentication and authorization?

Authentication verifies the identity of a user and authorization is a process where you can check whether or not the identity has access rights to the system. In other words, you can say that authentication is a procedure of getting some credentials from the users and verify the user's identity against those credentials.

Authorization is a procedure of granting access of particular resources to an authenticated user. You should note that authentication always takes place before authorization.

41. What are Custom User Controls in ASP.NET?

The custom user controls are the controls that are defined by developers. These controls are a mixture of custom behavior and predefined behavior. These controls work similar to other Web server controls.

42. What do you understand by aggregate dependency?

Aggregate dependency allows multiple dependencies to be aggregated for content that depends on more than one resource. In such type of dependency, you need to depend on the sum of all the defined dependencies to remove a data item from the cache.

43. How can you dynamically add user controls to a page?

User controls can be dynamically loaded by adding a Web User Control page in the application and adding the control on this page.

44. What type of code, client-side or server-side, is found in a code-behind file of a Web page?

A code-behind file contains the server-side code, which means that the code contained in a code-behind file is executed at the server.

45. What is Role-based security?

In the Role-based security, you can assign a role to every user and grant the privilege according to that role. A role is a group of principal that restricts a user's privileges. Therefore, all the organization and applications use role-based security model to determine whether a user has enough privileges to perform a requested task.

46. What is the use of Placeholder control? Can we see it at runtime?

The Placeholder control acts as a container for those controls that are dynamically generated at runtime. We cannot see it at runtime because it does not produce any visible output. It used only as a container.

47. How can you enable impersonation in the web.config file?

To enable impersonation in the web.config file, you need to include the <identity> element in the web.config file and set the impersonate attribute to true as shown in the following code snippet:

```
<identity impersonate = "true" />
```

48. Which method has been introduced in ASP.NET 4.0 to redirect a page permanently?

The RedirectPermanent() method added in ASP.NET 4.0 to redirect a page permanently. The following code snippet is an example of the RedirectPermanent() method:

```
RedirectPermanent("/path/Aboutus.aspx");
```

49. What is ViewState?

The ViewState is a feature used by ASP.NET Web page to store the value of a page and its controls just before posting the page. Once the page is posted, the first task by the page processing is to restore the ViewState to get the values of the controls.

50. Why do you use the App_Code folder in ASP.NET?

The App_Code folder is automatically present in the project. It stores the files, such as classes, typed data set, text files, and reports. If this folder is not available in the application, you can add this folder. One of the important features of the App_Code folder is that only one dll is created for the complete folder, irrespective of how many files it contains.

51. What is ASP.MVC?

The ASP.NET MVC framework is a lightweight, highly testable presentation framework that is integrated with existing ASP.NET features, such as master pages and membership-based authentication. The MVC framework is defined in the System.Web.Mvc assembly.

52. Can you explain Model, Controller and View in MVC?

Model – It's a business entity and it is used to represent the application data.

Controller – Request sent by the user always scatters through controller and it's responsibility is to redirect to the specific view using View () method.

View – It's the presentation layer of MVC.

53. Explain the new features added in version 4 of MVC (MVC4)?

Following are features added newly –

- Mobile templates
- Added ASP.NET Web API template for creating REST based services.
- Asynchronous controller task support.
- Bundling the java scripts.
- Segregating the configs for MVC routing, Web API, Bundle etc.

54. Can you explain the page life cycle of MVC?

Below are the processes followed in the sequence –

- App initialization
- Routing
- Instantiate and execute controller
- Locate and invoke controller action
- Instantiate and render view.

55. What are the advantages of MVC over ASP.NET?

- Provides a clean separation of concerns among UI (Presentation layer), model (Transfer objects/Domain Objects/Entities) and Business Logic (Controller).
- Easy to UNIT Test.
- Improved reusability of model and views. We can have multiple views which can point to the same model and vice versa.
- Improved structuring of the code.

56. What is Separation of Concerns in ASP.NET MVC?

It's the process of breaking the program into various distinct features which overlaps in functionality as little as possible. MVC pattern concerns on separating the content from presentation and data-processing from content.

57. What is Razor View Engine?

Razor is the first major update to render HTML in MVC 3. Razor was designed specifically for view engine syntax. Main focus of this would be to simplify and code-focused templating for HTML generation.

Below is the sample of using Razor:

```
@model MvcMusicStore.Models.Customer
```

```
@{ViewBag.Title = "Get Customers";}
```

```
<div class="cust"> <h3><em>@Model.CustomerName</em> </h3>
```

58. What is the meaning of Unobtrusive JavaScript?

This is a general term that conveys a general philosophy, similar to the term REST (Representational State Transfer). Unobtrusive JavaScript doesn't intermix JavaScript code in your page markup.

Eg : Instead of using events like onclick and onsubmit, the unobtrusive JavaScript attaches to elements by their ID or class based on the HTML5 data- attributes.

59. What is the use of ViewModel in MVC?

ViewModel is a plain class with properties, which is used to bind it to strongly typed view. ViewModel can have the validation rules defined for its properties using data annotations.

60. What you mean by Routing in MVC?

Routing is a pattern matching mechanism of incoming requests to the URL patterns which are registered in route table. Class – "UrlRoutingModule" is used for the same process

61. What are Actions in MVC?

Actions are the methods in Controller class which is responsible for returning the view or json data. Action will mainly have return type – "ActionResult" and it will be invoked from method – "InvokeAction()" called by controller.

62. What is Attribute Routing in MVC?

ASP.NET Web API supports this type routing. This is introduced in MVC5. In this type of routing, attributes are being used to define the routes. This type of routing gives more control over classic URI Routing. Attribute Routing can be defined at controller level or at Action level like –

```
[Route("{action = TestCategoryList}")] – Controller Level
```

```
[Route("customers/{TestCategoryId:int:min(10)}")] – Action Level
```

63. How to enable Attribute Routing?

Just add the method – "MapMvcAttributeRoutes()" to enable attribute routing as shown below

```
public static void RegisterRoutes(RouteCollection routes)
```

```
{
```

```
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");  
    //enabling attribute routing
```

```
routes.MapMvcAttributeRoutes();

//convention-based routing

routes.MapRoute

(

name: "Default",

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

defaults: new { controller = "Customer", action = "GetCustomerList", id = UrlParameter.Optional }

);

}
```

64. Explain JSON Binding?

JavaScript Object Notation (JSON) binding support started from MVC3 onwards via the new `JsonValueProviderFactory`, which allows the action methods to accept and model-bind data in JSON format. This is useful in Ajax scenarios like client templates and data binding that need to post data back to the server.

65. Explain Dependency Resolution?

Dependency Resolver again has been introduced in MVC3 and it is greatly simplified the use of dependency injection in your applications. This turn to be easier and useful for decoupling the application components and making them easier to test and more configurable.

66. Explain Bundle.Config in MVC4?

"BundleConfig.cs" in MVC4 is used to register the bundles by the bundling and minification system. Many bundles are added by default including jQuery libraries like – `jquery.validate`, `Modernizr`, and default CSS references.

67. How route table has been created in ASP.NET MVC?

Method – "`RegisterRoutes()`" is used for registering the routes which will be added in "`Application_Start()`" method of `global.asax` file, which is fired when the application is loaded or started.

68. Which are the important namespaces used in MVC?

Below are the important namespaces used in MVC –

- `Web.Mvc`
- `Web.Mvc.Ajax`

- Web.Mvc.Html
- Web.Mvc.Async

69. What is ViewData?

Viewdata contains the key, value pairs as dictionary and this is derived from class – “ViewDataDictionary”. In action method we are setting the value for viewdata and in view the value will be fetched by typecasting.

70. What is the difference between ViewBag and ViewData in MVC?

ViewBag is a wrapper around ViewData, which allows to create dynamic properties. Advantage of viewbag over viewdata will be –

In ViewBag no need to typecast the objects as in ViewData.

ViewBag will take advantage of dynamic keyword which is introduced in version 4.0. But before using ViewBag we have to keep in mind that ViewBag is slower than ViewData.

71. Explain TempData in MVC?

TempData is again a key, value pair as ViewData. This is derived from “TempDataDictionary” class. TempData is used when the data is to be used in two consecutive requests, this could be between the actions or between the controllers. This requires typecasting in view.

72. What are HTML Helpers in MVC?

HTML Helpers are like controls in traditional web forms. But HTML helpers are more lightweight compared to web controls as it does not hold viewstate and events.

HTML Helpers returns the HTML string which can be directly rendered to HTML page. Custom HTML Helpers also can be created by overriding “HtmlHelper” class.

73. What are AJAX Helpers in MVC?

AJAX Helpers are used to create AJAX enabled elements like as Ajax enabled forms and links which performs the request asynchronously and these are extension methods of AJAXHelper class which exists in namespace – System.Web.Mvc.

74. What are the options can be configured in AJAX helpers?

Below are the options in AJAX helpers –

Url – This is the request URL.

Confirm – This is used to specify the message which is to be displayed in confirm box.

OnBegin – Javascript method name to be given here and this will be called before the AJAX request.

OnComplete – Javascript method name to be given here and this will be called at the end of AJAX request.

OnSuccess – Javascript method name to be given here and this will be called when AJAX request is successful.

OnFailure – Javascript method name to be given here and this will be called when AJAX request is failed.

UpdateTargetId – Target element which is populated from the action returning HTML.

75. What is Layout in MVC?

Layout pages are similar to master pages in traditional web forms. This is used to set the common look across multiple pages. In each child page we can find – /p>

```
@{
```

```
Layout = "~/Views/Shared/TestLayout1.cshtml";
```

```
}
```

This indicates child page uses TestLayout page as it's master page.

76. Explain Sections in MVC?

Section are the part of HTML which is to be rendered in layout page. In Layout page we will use the below syntax for rendering the HTML –

```
@RenderSection("TestSection")
```

And in child pages we are defining these sections as shown below –

```
@section TestSection{
```

```
<h1>Test Content</h1>
```

```
}
```

If any child page does not have this section defined then error will be thrown so to avoid that we can render the HTML like this –

```
@RenderSection("TestSection", required: false)
```

77. Can you explain RenderBody and RenderPage in MVC?

RenderBody is like ContentPlaceHolder in web forms. This will exist in layout page and it will render the child pages/views. Layout page will have only one RenderBody() method. RenderPage also exists in Layout page and multiple RenderPage() can be there in Layout page.

78. What is ViewStart Page in MVC?

This page is used to make sure common layout page will be used for multiple views. Code written in this file will be executed first when application is being loaded.

79. Explain the methods used to render the views in MVC?

Below are the methods used to render the views from action –

View() – To return the view from action.

PartialView() – To return the partial view from action.

RedirectToAction() – To Redirect to different action which can be in same controller or in different controller.

Redirect() – Similar to “Response.Redirect()” in webforms, used to redirect to specified URL.

RedirectToRoute() – Redirect to action from the specified URL but URL in the route table has been matched.

80. What are the sub types of ActionResult?

ActionResult is used to represent the action method result. Below are the subtypes of ActionResult –

- ViewResult
- PartialViewResult
- RedirectToRouteResult
- RedirectResult
- JavascriptResult
- JsonResult
- FileResult
- HttpStatusCodeResult

81. What are Non Action methods in MVC?

In MVC all public methods have been treated as Actions. So if you are creating a method and if you do not want to use it as an action method then the method has to be decorated with “NonAction” attributes are

[NonAction]

```
public void TestMethod()
```

```
{
```

```
// Method logic
```

```
}
```

82. How to change the action name in MVC?

“ActionName” attribute can be used for changing the action name. Below is the sample code snippet to demonstrate more –

```
[ActionName("TestActionNew")]
```

```
public ActionResult TestAction()
```

```
{
```

```
return View();
```

```
}
```

So in the above code snippet “TestAction” is the original action name and in “ActionName” attribute, name – “TestActionNew” is given. So the caller of this action method will use the name “TestActionNew” to call this action.

83. What are Code Blocks in Views?

Unlike code expressions that are evaluated and sent to the response, it is the blocks of code that are executed. This is useful for declaring variables which we may be required to be used later.

```
@{
```

```
int x = 123;
```

```
string y = "aa";
```

```
}
```

84. What is the “HelperPage.IsAjax” Property?

The HelperPage.IsAjax property gets a value that indicates whether Ajax is being used during the request of the Web page.

85. How we can call a JavaScript function on the change of a Dropdown List in MVC?

Create a JavaScript method:

```
<script type="text/javascript">
```

```
function DrpIndexChanged() { }
```

```
</script>
```

Invoke the method:

```
<bt;%:Html.DropDownListFor(x => x.SelectedProduct, new SelectList(Model.Customers, "Value", "Text"),  
"Please Select a Customer", new { id = "ddlCustomers", onchange=" DrpIndexChanged ()" })%>
```

86. What are Validation Annotations?

Data annotations are attributes which can be found in the "System.ComponentModel.DataAnnotations" namespace. These attributes will be used for server-side validation and client-side validation is also supported. Four attributes – Required, String Length, Regular Expression and Range are used to cover the common validation scenarios.

87. Why to use Html.Partial in MVC?

This method is used to render the specified partial view as an HTML string. This method does not depend on any action methods. We can use this like below –

```
<blol.Partial("TestPartialView")
```

88. What is Html.RenderPartial?

Result of the method – "RenderPartial" is directly written to the HTML response. This method does not return anything (void). This method also does not depend on action methods. RenderPartial() method calls "Write()" internally and we have to make sure that "RenderPartial" method is enclosed in the bracket. Below is the sample code snippet –@{Html.RenderPartial("TestPartialView"); }

89. What is RouteConfig.cs in MVC 4?

"RouteConfig.cs" holds the routing configuration for MVC. RouteConfig will be initialized on Application_Start event registered in Global.asax.

90. What are Scaffold templates in MVC?

Scaffolding in ASP.NET MVC is used to generate the Controllers, Model and Views for create, read, update, and delete (CRUD) functionality in an application. The scaffolding will be knowing the naming conventions used for models and controllers and views.

91. Explain the types of Scaffoldings.

Below are the types of scaffoldings –

- Empty
- Create
- Delete
- Details
- Edit
- List

92. Can a view be shared across multiple controllers? If Yes, How we can do that?

Yes, we can share a view across multiple controllers. We can put the view in the “Shared” folder. When we create a new MVC Project we can see the Layout page will be added in the shared folder, which is because it is used by multiple child pages.

93. What are the components required to create a route in MVC?

Name – This is the name of the route.

URL Pattern – Placeholders will be given to match the request URL pattern.

Defaults –When loading the application which controller, action to be loaded along with the parameter.

94. Why to use “{resource}.axd/{*pathInfo}” in routing in MVC?

Using this default route – {resource}.axd/{*pathInfo}, we can prevent the requests for the web resources files like – WebResource.axd or ScriptResource.axd from passing to a controller.

95. Can we add constraints to the route? If yes, explain how we can do it?

Yes we can add constraints to route in following ways –

- Using Regular Expressions
- Using object which implements interface – IRouteConstraint.

96. What are the possible Razor view extensions?

Below are the two types of extensions razor view can have –

- .cshtml – In C# programming language this extension will be used.
- .vbhtml – In VB programming language this extension will be used.

97. What is PartialView in MVC?

PartialView is similar to UserControls in traditional web forms. For re-usability purpose partial views are used. Since it's been shared with multiple views these are kept in shared folder. Partial Views can be rendered in following ways –

Html.Partial()

Html.RenderPartial()

98. How we can add the CSS in MVC?

Below is the sample code snippet to add css to razor views –

```
<link rel="StyleSheet" href="@Href("~/Content/Site.css")" type="text/css"/>
```

99. Can I add MVC Testcases in Visual Studio Express?

No. We cannot add the test cases in Visual Studio Express edition it can be added only in Professional and Ultimate versions of Visual Studio.

100. What is the use. Glimpse in MVC?

Glimpse is an open source tool for debugging the routes in MVC. It is the client-side debugger. Glimpse has to be turned on by visiting to local url link –

`http://localhost:portname//glimpse.axd`

This is a popular and useful tool for debugging which tracks the speed details, url details etc.