By OnlineInterviewQuestions.com

MVC interview questions

MVC Interview Questions on Asp.net

MVC is an abbreviation for **Model, View, and Controller**. The MVC architectural pattern separates an application into three components – Model, View, and Controller. In this pattern, the model represents the shape of the data and business logic. It maintains and preserves the data of the application. Model objects retrieve and store model state in a database. The view is basically and technically a user interface. The view segment displays the data-using model to the user and also enables them to modify the data. The controller is the part, which handles the user request.

Typically, a user interacts with View, which in turn raises appropriate URL requests. Finally, a controller will handle this request. The controller segment of the MVC renders the appropriate view with the model data as a response. Implementing this structure in applications and creating web apps based on this ASP.NET framework in **Azure** and other platforms is what most developers ought to know.

However, one might forget the basics when dealing with practical problems on a regular basis. So, here is a list of **common interview questions** on the basics of MVC, which might be asked in an interview for any post or designation.

Q1. What is MVC (Model view controller)?

Model-View-Controller (MVC) is a software architectural pattern for implementing user interfaces. It divides a given software application into three interconnected parts, to separate the internal representation of information from the way that information is presented to or accepted by the user.

Q2. What is Razor in MVC?

Razor is a view engine that lets you embed server-based code into web pages using C# and VB.Net. It was introduced in MVC 3 and a gneral purpose templating engine.

Q3. Explain attribute based routing in MVC?

Routing is the process of monitoring the requests with the MVC application. In MVC, the Application_Start() event registers all the routes in the routing table.

In **ASP.NET MVC 5.0**, a new type of routing is introduced, it is called **Attribute Routing**. It enables the developers to define the routing using attributes on top of the controller action method. To use the attribute routing in the MVC 5, you need to call the MapMvcAttributeRoute method. The optional parameter for the URI can be added to the route parameter using the question mark(**?**) in the attribute routing.

Q4. What is TempData in MVC?

The **TempData in the ASP.NET MVC** is used to store temporary data. It can be used in subsequent requests. This TempData will be cleared after the completion of the subsequent request. It is useful in transferring the non-sensitive data from one action method to another action method of the same or a different controller and redirects. It stores data as key-value pair.

Example

```
public ActionResult Index()
   {
       TempData["name"] = "Test data";
   }
   public ActionResult About()
   {
       string userName;
       if(TempData.ContainsKey("name"))
           userName = TempData["name"].ToString();
   }
```

Q5. What is Partial View in MVC?

A partial view is a chunk of HTML that can be safely inserted into an existing DOM. Most commonly, partial views are used to componentize Razor views and make them easier to build and update. Partial views can also be returned directly from controller methods. In this case, the browser still receives text/HTML content but not necessarily HTML content that makes up an entire page. As a result, if a URL that returns a partial view is directly invoked from the address bar of a browser, an incomplete page may be displayed. This may be something like a page that misses title, script and style sheets.

Q6. Mention what is the difference between ViewData and ViewBag?

In ASP.NET MVC there are three ways to pass/store data between the controllers and views.

ViewData

1. ViewData is used to pass data from a controller to view.
2. It is derived from ViewDataDictionary class.
3. It is available for the current request only.
4. Requires typecasting for complex data type and checks for null values to avoid error.
5. If redirection occurs, then its value becomes null.

ViewBag

1. ViewBag is also used to pass data from the controller to the respective view.
2. ViewBag is a dynamic property that takes advantage of the new dynamic features in C# 4.0
3. It is also available for the current request only.
4. If redirection occurs, then its value becomes null.
5. Doesn’t require typecasting for the complex data type.

Q7. What is Route in MVC?

A route is a URL pattern that is mapped to a handler. The handler can be a physical file, such as an .aspx file in a Web Forms application. A handler can also be a class that processes the request, such as a controller in an MVC application. To define a route, you create an instance of the route class by specifying the URL pattern, the handler, and optionally a name for the route.

Q8. Explain what is routing in MVC?

Routing is a mechanism to process the incoming URL that is more descriptive and gives the desired response. In this case, URL is not mapped to specific files or folder, as was the case of websites in the past.

Q9. What are the three segments for routing important?

Routing is the URL pattern that is mapped together to a handler and is responsible for incoming browser request for a particular MVC controller. There are a total of three segments for routing that are very significant in MVC

1. ControllerName
2. ActionMethodName
3. Parammeter

Q10. What are the Filters in MVC?

The **filter in ASP.NET MVC** is a class that is executed **before** or **after** an action method is executed. It is a place where you can write custom logic that you want to execute. It can be executed either to an action method or a controller. It is applied in two different ways, declarative and programmatic. Applying the filter to the action or controller class is called a declarative filter. Defining the interface for the filter is called the programmatic way of applying a filter.

There are four types of built-in filters in ASP.NET MVC are **authorization filter**, **action filter**, **result filter**, **exception filter**. These filters give an interface for which you can define your custom filter.

Q11. What are Action Filters in MVC?

An **action filter** is a type of filter in **ASP.NET MVC** that is executed to implement logic before and after a controller is executed. These provide an additional attribute that is applied to a section of the controller or the whole controller to modify it. These attributes are defined under **System.Attribute** class.

Three types of action filters in ASP.NET MVC are,

- **Output cache –** It caches the output of a controller action. The output is cached here for a defined amount of time.
- **Handle Error –** This filter is used to handle errors from the execution of controller action.
- **Authorize –** This filter is used to restrict access to the action method for a particular user.

Q12. What are the types of filters?

ASP.NET MVC structure upholds the following action filters:

- **Action Filters:** Action filters are used to implement logic that gets executed before and after a controller action executes. We will look at Action Filters in detail in this chapter.
- **Authorization Filters:** Authorization filters are used to implement authentication and authorization for controller actions.
- **Result Filters:** Result filters contain logic that is executed before and after a view result is executed. For example, you might want to modify a view result right before the view is rendered to the browser.
- **Exception Filters:** Exception filters are the last type of filter to run. You can use an exception filter to handle errors raised by either your controller actions or controller action results. You can also use exception filters to log errors.

Q13. List the various return types of a controller action method.

There is total of nine return types we can use to return results from a controller to view. The base type of all these result types is ActionResult.

1. **ViewResult (View):** This return type is used to return a webpage from an action method.
2. **PartialviewResult (Partialview):** This return type is used to send a part of a view that will be rendered in another view.
3. **RedirectResult (Redirect):** This return type is used to redirect to any other controller and action method depending on the URL.
4. **RedirectToRouteResult (RedirectToAction, RedirectToRoute):** This return type is used when we want to redirect to any other action method.
5. **ContentResult (Content):** This return type is used to return HTTP content type like text/plain as the result of the action.
6. **jsonResult (JSON):** This return type is used when we want to return a JSON message.
7. **javascriptResult (javascript):** This return type is used to return JavaScript code that will run in the browser.

8. **FileResult (File):** This return type is used to send binary output in response.
9. **EmptyResult:** This return type is used to return nothing (void) in the result.

Q14. Explain what is MVC application life cycle?

Any web application has two main execution steps, first understanding the request and depending on the type of the request sending out an appropriate response. MVC application life cycle is not different it has two main phases, first creating the request object and second sending our response to the browser.

Creating the request object includes four basic steps:

Step 1: Fill route
Step 2: Fetch route
Step 3: Request context created
Step 4: Controller instance created

Q15. What are the advantages of MVC?

Benefits or advantages of MVC are as follows:

- **Multiple view support**: Due to the separation of the model from the view, the user interface can display multiple views of the same data at the same time.
- **Change Accommodation**: User interfaces tend to change more frequently than business rules (different colors, fonts, screen layouts, and levels of support for new devices such as cell phones or PDAs)
- **SoC –** Separation of Concerns: Separation of Concerns is one of the core advantages of ASP.NET MVC. The MVC framework provides a clean separation of the UI, Business Logic, Model or Data.
- **More Control**: The ASP.NET MVC framework provides more control over HTML, JavaScript, and CSS than the traditional Web Forms.
- **Testability**: ASP.NET MVC framework provides better testability of the Web Application and good support for the test-driven development too.
- **Lightweight**: ASP.NET MVC framework doesn’t use View State and thus reduces the bandwidth of the requests to an extent.

Q16. What does the MVC pattern define with 3 logical layers?

The MVC model defines web applications with 3 logic layers:

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

The Model is the part of the application, which only handles the logic for the application data. Regularly, the model objects retrieve data (as well as store data) from a database. The View is the part of the application, which takes care of the display of the data.

Most often, the views are created from the model data, although there are other, more complicated methods of creating views.

The Controller, as the name implies, is the part of the application that handles user interaction.

Q17. What is Default Route in MVC?

Default Route: The default ASP.NET MVC project templates add a generic route that uses the following URL convention to break the URL for a given request into three named segments.

```
URL: "{controller}/{action}/{id}"
```

This route pattern is registered via a call to the MapRoute() extension method of RouteCollection.

Q18. What are HTML helpers in MVC?

In MVC HTML Helpers is a class which are used in views to render/display HTML controls like links, TextBox, TextArea, etc.

HTML Helper Example in MVC

```
@Html.ActionLink("Create New", "create")
```

The above example generates an HTML link something like <a href="/create">Create New</a>

Q19. Define attribute based routing in MVC?

Attribute Routing was introduced in ASP.NET MVC 5. It allows developers to generate urls for controllers actions by adding Route Attributes to them.

This type of MVC routing uses attributes to define routes and gives developer more control over the urls of web application.In order to use Attribute Routing first you have to enable it by calling MapMvcAttributeRoutes during configuration.

//Configuring Attribute Routing in MVC.

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

        routes.MapMvcAttributeRoutes();
    }
}
```

Q20. What are the Main Razor Syntax Rules?

Following are the rules for main Razor Syntax:

- Razor code blocks are enclosed in @{ ... }
- Inline expressions (variables and functions) start with @
- Code statements end with a semicolon
- Variables are declared with the var keyword
- Strings are enclosed with quotation marks
- C# code is case sensitive
- C# files have the extension .cshtml

Q21. How do you implement Forms authentication in MVC

Authentication is giving access to the user for a specific service by verifying his/her identity using his/her credentials like username and password or email and password. It assures that the correct user is authenticated or logged in for a specific service and the right service has been provided to the specific user based on their role that is nothing but authorization.

Q22. What are the Benefits of Area in MVC?

Benefits of Area in MVC

1. Allows us to organize models, views, and controllers into separate functional sections of the application, such as administration, billing, customer support and much more.
2. Easy to integrate with other Areas created by another.
3. Easy for unit testing.

Q23. Explain the need of display mode in MVC?

A display mode is a powerful tool for all mobile-based apps development. However, it is not limited to mobile web apps but it can also be used to display any alternative view, which is tied to an existing controller. Display Modes practically give you another level of flexibility on top of the default capabilities we saw in the last section. Display Modes can also be used along with the previous feature so the users will simply build off of the site they just created. It simplifies the ways to implement different versions of view for different devices.

Q24. Explain the concept of MVC Scaffolding?

ASP.NET Scaffolding is a code generation framework for ASP.NET Web applications. Visual Studio 2013 includes pre-installed code generators for MVC and Web API projects. You add scaffolding to your project when you want to quickly add code that interacts with data models. Using scaffolding can reduce the amount of time to develop standard data operations in your project.
Scaffolding consists of page templates, entity page templates, field page templates, and filter templates. These templates are called Scaffold templates and allow you to quickly build a functional data-driven Website.

Q25. What are Route Constraints in MVC?

Route constraints in the **ASP.NET** are used to control the behavior of the route. With route constraint, you can restrict your users to a set of URLs, or restrict browser requests to match a particular route. The route constraints are defined by using a regular expression. For eg: to set a constraint string to our hostname, we can use the routing constraint as the following. Open the RouteConfig.cs file and modify the routing entry as *"abc/{controller}/{action}/{id}"*. Now the URL with string **"abc"** as the prefix will only be able to access the site.

Q26. What is Output Caching in MVC?

The **Output cache in the ASP.NET MVC** will increase the performance of your application significantly. It is used to cache the value that is returned by the controller action. So, when every time the controller action is invoked, the same value need not be generated each time. To enable the output cache, use the OutputCache attribute to the individual controller action or the entire controller class. You can also define the duration for the content to be cached in the output cache.

Example:

```
using System.Web.Mvc;
namespace MvcApplication1.Controllers
{
   [HandleError]
   public class HomeController : Controller
   {
       [OutputCache(Duration=10, VaryByParam="none")]
       public ActionResult Index()
       {
           return View();
       }
   }
}
```

In the above example, the output view is cached in the output cache for about 10 seconds.

Q27. What is Bundling and Minification in MVC?

Bundling and minification are two new techniques introduced to improve request load time. It improves load time by reducing the number of requests to the server and reducing the size of requested assets (such as CSS and JavaScript).

Bundling: It lets us combine multiple JavaScript (.js) files or multiple cascading style sheet (.css) files so that they can be downloaded as a unit, rather than making individual HTTP requests.

Minification: It extracts the whitespace and performs other types of compression to make the downloaded files as small as possible. At runtime, the process recognizes the agent of the user, for example, IE, Mozilla, etc. and then removes whatever is specific to Mozilla when the request comes from IE.

Q28. What is Validation Summary in MVC?

The Validation Summary helper method generates an unordered list (UL element) of validation messages that are in the Model State Dictionary object.

The Validation Summary can be used to display all the error messages for all the fields. It can also be used to display custom error messages. The following figure shows how Validation Summary displays the error messages.

Q29. Explain what is Database First Approach in MVC using Entity Framework?

Database First Approach is an alternative or substitutes to the Code First and Model First approaches to the Entity Data Model. The Entity Data Model creates model codes (classes, properties, DbContext, etc.) from the database in the project and that class behaves as the link between database and controller.

There are the following approaches, which are used to connect the database with the application.

- Database First
- Model First
- Code First

Q30. What is GET and POST Actions Types?

GET Action Type: GET is used to request data from a specified resource. With all the GET requests, we pass the URL, which is compulsory; however, it can take up the following overloads.

POST Action Type: Tthe POST is used to submit data to be processed to a specified resource. With all the POST requests, we pass the URL, which is essential and the data. However, it can take up the following overloads.

Q31. What’s new in the latest version of MVC?

In MVC 6 Microsoft removed the dependency of System.Web.Dll from MVC6 because it is extremely expensive, as it typically used to consume 30K of memory per request and response, whereas now MVC 6 only requires 2K of memory per request and the response is really small memory consumption.

Q32. How to render raw html in Asp.net MVC view?

You can use **Html.Raw** helper to render raw Html in Asp.net MVC

Syntax

```
@Html.Raw('<h1>Raw Example </h1>');
```

Please Visit OnlineInterviewquestions.com to download more pdfs