InterviewBit

# MVC Interview Questions

To view the live version of the page, click here.

# Contents

## MVC Interview Questions and Answers

## MVC Interview Questions and Answers (.....Continued)

21. How will you differentiate between ActionResult and ViewResult?
22. What is Spring MVC?
23. Explain briefly what you understand by separation of concern.
24. What is TempData in MVC?
25. Define Output Caching in MVC?
26. Why are Minification and Bundling introduced in MVC?
27. Describe ASP.NET MVC?
28. Which class will you use for sending the result back in JSON format in MVC?
29. Make a differentiation between View and Partial View?
30. Define the concept of Filters in MVC?
31. Mention the significance of NonActionAttribute?
32. What is used to handle an error in MVC?
33. Define Scaffolding in MVC?
34. When multiple filters are used in MVC, how is the ordering of execution of the filters done?
35. What is ViewStart?
36. Which type of filters are executed in the end while developing an MVC application?
37. Mention the possible file extensions used for razor views?
38. Explain briefly the two approaches of adding constraints to an MVC route?
39. Point out the different stages a Page life cycle of MVC has?
40. Explain briefly the use of ViewModel in MVC?

## MVC Interview Questions and Answers (.....Continued)

41. Define Default Route in MVC?
42. Explain briefly the GET and POST Action types?
43. What are the rules of Razor syntax?
44. How can you implement the MVC forms authentication?
45. What are the areas of benefits in using MVC?
46. Point out the two instances where you cannot use routing or where routing is not necessary
47. How will you explain the concept of RenderBody and RenderPage of MVC?

# Let's get Started

MVC (full form **Model View Controller**)is a software architecture or application design model containing 3 interconnected verticals or portions. These 3 portions are the **model** (data associated with the application), the **view** (which is the user interface of an MVC application), and the **controller** (the processes that are responsible for handling the input).

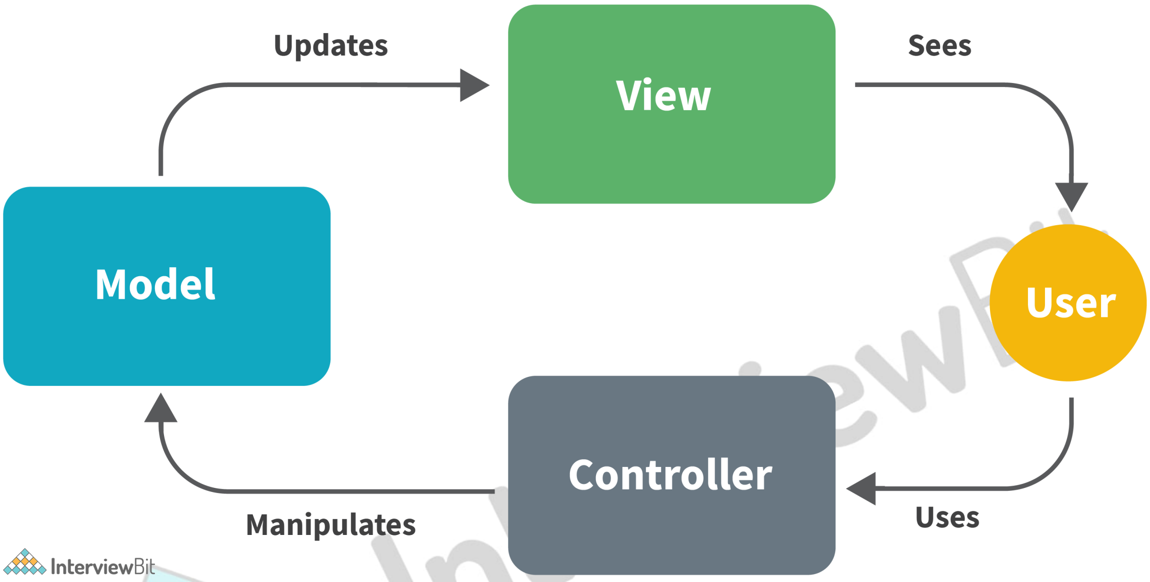
The MVC model is normally used to develop modern applications with user interfaces. It provides the central pieces for designing a desktop or mobile application, as well as modern web applications.

In this article, you will find a collection of real-world MVC interview questions with inline answers that are asked in top tech companies. So, here we go!

## MVC Interview Questions and Answers

### 1. Explain Model, View and Controller in Brief.

- A model can be defined as the data that will be used by the program. Commonly used examples of models in MVC are the database, a simple object holding data (such as any multimedia file or the character of a game), a file, etc.
- A view is a way of displaying objects (user interfaces) within an application. This is the particular vertical through which end users will communicate.
- A controller is the third vertical which is responsible for updating both models and views. It accepts input from users as well as performs the equivalent update. In other words, it is the controller which is responsible for responding to user actions.



## 2. What are the different return types used by the controller action method in MVC?

The various return types of controller action methods in MVC are:

- View Result
- JSON Result
- Content Result
- Redirect Result
- JavaScript Result

## 3. Name the assembly in which the MVC framework is typically defined.

In the `System.Web.Mvc` , the MVC framework is usually defined.

## 4. Explain the MVC Application life cycle.

Web applications usually have 2 primary execution steps. These are:

- Understanding the request.
- Sending an appropriate response based on the type of request.

The same thing can be related to MVC applications also whose life cycle has 2 foremost phases:

- For creating a request object.
- For sending the response to any browser.

## 5. What are the various steps to create the request object?

In order to create a request object, we have to go through 4 different steps. These are:

- Step 1: Fill the route.
- Step 2: Fetch the route.
- Step 3: Create a request context.
- Step 4: Create a controller instance.

## 6. Explain some benefits of using MVC?

Some common benefits of MVC are:

- **Support of multiple views**: Since there is a separation of the model from its view, the user interface (UI) gets the capability to implement multiple views of the same data concurrently.
- **Faster development process**: MVC has the ability to provide rapid and parallel development. This means that while developing an application, it is more likely that one programmer will perform some action on the view and in parallel, another can work on creating the application's business logic.
- **SEO-friendly development**: The platform of MVC can support the SEO-friendly development of web pages or web applications.
- **More Control**: The MVC framework (of ASP.NET) offers additional control over HTML, CSS and JavaScript than that of traditional WebForms.
- **Lightweight**: MVC framework does not make use of View State which eventually minimises the requested bandwidth to some extent.

## 7. Explain in brief the role of different MVC components?

The different MVC components have the following roles:

- **Presentation**: This component takes care of the visual representation of a particular abstraction in the application.
- **Control**: This component takes care of the consistency and uniformity between the abstraction within the system along with their presentation to the user. It is also responsible for communicating with all other controls within the MVC system.
- **Abstraction**: This component deals with the functionality of the business domain within the application.

## 8. How will you maintain the sessions in MVC?

The sessions of an MVC can be maintained in 3 possible ways:

- viewdata
- temp data and
- view bag

## 9. What do you mean by partial view of MVC?

A partial view can be defined as a portion of HTML that is carefully injected into an existing DOM. Partial views are commonly implemented for componentizing Razor views, making them simpler to build and update. Controller methods can also directly return the partial views.

## 10. Explain in brief the difference between adding routes in a webform application & an MVC application?

We make use of the MapPageRoute() which is of the RouteCollection class for adding routes in a webform application. Whereas, the MapRoute() method is used for adding routes to an MVC application.

## 11. How will you define the 3 logical layers of MVC?

The 3 logical layers of MVC can be defined as follows:

- Model logic acts as a business layer.
- View logic acts as a display layer.
- Controller logic acts as input control.

ActionFilters are used for executing the logic while MVC action is executed. Furthermore, action filters permit the implementation of pre and post-processing logic and action methods.

## 13. How to execute any MVC project? Explain its steps.

For executing an MVC project, the steps followed are:

- Receive the first request for the application.
- Then, the routing is performed.
- Then, the MVC request handler is created.
- After that, the controller is created and executed.
- Then, the action is invoked.
- Then, the results are executed.

## 14. What is the concept of routing in MVC?

MVC routing can be defined as a pattern-matching scheme that is used for mapping incoming requests of browsers to a definite MVC controller action.

## 15. What are the 3 important segments for routing?

The 3 important segments for routing are:

- ControllerName.
- ActionMethodName.
- Parameter.

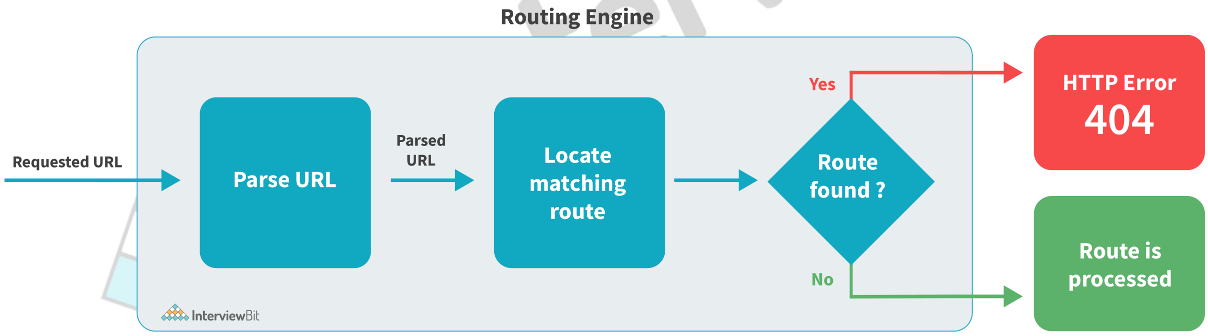
## 16. What are the different properties of MVC routes?

MVC routes are accountable for governing which controller method will be executed for a given URL. Thus, the URL comprises the following properties:

- **Route Name**: It is the URL pattern that is used for mapping the handler.
- **URL Pattern:** It is another property containing the literal values as well as variable placeholders (known as URL parameters).
- **Defaults:** This is the default parameter value assigned at the time of parameter creation.
- **Constraints:** These are used for applying against the URL pattern for more narrowly defining the URL matching it.

## 17. How is the routing carried out in MVC?

The RouteCollection contains a set of routes that are responsible for registering the routes in the application. The RegisterRoutes method is used for recording the routes in the collection. The URL patterns are defined by the routes and a handler is used which checks the request matching the pattern. The MVC routing has 3 parameters. The **first parameter** determines the name of the route. The **second parameter** determines a specific pattern with which the URL matches. The **third parameter** is responsible for providing default values for its placeholders.



## 18. How will you navigate from one view to another view in MVC? Explain with a hyperlink example.

We will make use of the ActionLink method which will help us to navigate from one view to another. Here is an example of navigating the Home controller by invoking the Go to Home action. He is how we can code it:

```
<%=Html.ActionLink("Home","GoTo Home")%>
```

## 19. Explain the 3 concepts in one line; Temp data, View, and Viewbag?

We can briefly describe Temp data, View, and Viewbag as:

- **Temp data:** This is used for maintaining the data when there is a shift of work from one controller to another.
- **View data:** This is used for maintaining the data when we will shift from a controller to a view within an application.
- **View Bag:** This acts as a view data's dynamic wrapper.

## 20. Mention & explain the different approaches you will use to implement Ajax in MVC?

There are 2 different approaches to implementing Ajax in MVC. These are:

- **jQuery:** This is a library written using JavaScript for simplifying HTML-DOM manipulation.
- **AJAX libraries:** Asynchronous JavaScript and XML libraries are a set of web development libraries written using JavaScript and are used to perform common operations.

## 21. How will you differentiate between ActionResult and ViewResult?

Some common differentiation between ActionResult and ViewResult is:

| ActionResult | ViewResult |
|---|---|
| It becomes effective if you want to derive different types of views dynamically. | It is not so effective in deriving different types of views dynamically. |
| It is an abstract class, meaning it has methods and variables without the implementation body of instruction. | This has been derived from the ActionResult class. |
| JsonResult, ViewResult, and FileStreamResult are some examples of its derived class. | This class do not have its own derived class. |

## 22. What is Spring MVC?

The **Spring MVC** or Spring Web MVC can be defined as a framework that provides a “Model View Controller” (MVC) architecture in the application as well as ready components implemented for developing adjustable and adaptable web applications. It is actually a Java-based framework intended to build web applications. It works on the Model-View-Controller design approach. This framework also makes use of all the elementary traits of a core Spring Framework such as dependency injection, lightweight, integration with other frameworks, inversion of control, etc. Spring MVC has a dignified resolution for implementing MVC in Spring Framework with the use of DispatcherServlet.

## 23. Explain briefly what you understand by separation of concern.

Separation of Concerns can be defined as one of the core features as well as benefits of using MVC and is supported by ASP.NET. Here, the MVC framework offers a distinct detachment of the different concerns such as User Interface (UI), data and the business logic.

## 24. What is TempData in MVC?

TempData can be defined as a dictionary object used for storing data for a short period of time. This is the MVC's TempDataDictionary class which acts as a Controller base-class's instance property. TempData has the ability to preserve data for an HTTP request.

## 25. Define Output Caching in MVC?

Output Caching is an approach used for improving the performance of an MVC application. It is used for enabling its users to cache the data sent back by the controller method so that the data used earlier does not get generated each time while invoking the same controller method. It has advantages to use Output Caching as it cuts down database server round trips, minimizes server round trips as well as reduces the network traffic.

## 26. Why are Minification and Bundling introduced in MVC?

Two new techniques have been included in MVC, known as Bundling and minification, whose primary function is to progress the request load time. It advances the load time by dipping the number of requests sent to the server as well as reducing the requested asset's (JavaScript and CSS) size.

## 27. Describe ASP.NET MVC?

The term **ASP.NET MVC** can be defined as a web application framework that is very lightweight and has high testable features. ASP.NET supporting MVC uses 3 separate components in its application. These are the Model, View, and Controller.

## 28. Which class will you use for sending the result back in JSON format in MVC?

For sending back the result in JSON format in any MVC application, you have to

## 29. Make a differentiation between View and Partial View?

The major differentiation between View and Partial View is as follows:

| View | Partial View |
|---|---|
| The view is not as lightweight as that of the Partial view. | Partial view, as the name suggests, is lightweight than View. |
| The view has its own layout page. | The partial view does not have its own layout page. |
| The Viewstart page is rendered just before rendering any view. | A partial view is designed particularly for rendering within the view. |
| The view can have markup tags of HTML such as HTML, head, body, title, meta, etc. | The partial view does not contain any markup. |

## 30. Define the concept of Filters in MVC?

There are situations where I want to implement some logic either prior to the execution of the action method or right after it. In that scenario, the Action Filters are used. Filters are used to determine the logic needed for executing before or after the action method gets executed. Action methods make use of the action filters as an attribute.

Different types of MVC action filters are:

- Action filter (that implements the IActionFilter).
- Exception filter (that implements the IExceptionFilter attribute).
- Authorization filter (that implements the IAuthorizationFilter).
- Result filter (that implements the IResultFilter).

## 31. Mention the significance of NonActionAttribute?

The various public methods that are associated with the controller class are considered to be the action method. For preventing the default method, you have to allocate its public method with NonActionAttribute.

## 32. What is used to handle an error in MVC?

Error handling is usually done using Exception handling, whether it's a Windows Forms application or a web application. The HandleError attribute is used, which helps in providing built-in exception filters. The HandleError attribute of ASP.NET can be functional over the action method as well as Controller at its global level.

**Example of implementation:**

```
public static void RegGlobalFilters(Global_FilterCollection filt)
{
    filt.Add(new HandleErrorAttribute());
}
protected void Application_Start()
{
    AreaRegn.RegisterAllAreas();
    RegGlobalFilters(Global_Filters.Filters);
    RegisterRoutes(Route_Table.Routes);
}
```

## 33. Define Scaffolding in MVC?

Scaffolding can be defined as an ASP.NET's code-generation framework used in web applications. Scaffolding is used in developing MVC applications when anyone wants to rapidly enhance the code that intermingles with the application's data model. Scaffolding can also lower the quantity of time for developing a standard data operation in the application.

## 34. When multiple filters are used in MVC, how is the ordering of execution of the filters done?

The order in which filters are used:

- First, the authorization filters are executed.
- Followed by the Action filters.
- Then, the response filters are executed.
- Finally, the exception filters.

## 35. What is ViewStart?

A new layout called _ViewStart is introduced by the Razor View Engine that is applied to all views automatically. ViewStart is executed at the very beginning followed by the start rendering as well as other views.

Example:

```
@ {
  Layout = "~/ Views/ Shared/ _
  file.cshtml";
}
<html>
  <head>
  <meta name="viewport" />
  <title> InitialView </title> </head>
  <body>  ….
  </body>
</html>
```

## 36. Which type of filters are executed in the end while developing an MVC application?

In the end, while developing an MVC application, the “Exception Filters” are executed.

## 37. Mention the possible file extensions used for razor views?

The different file extensions that are used by razor views are:

- **.cshtml**: When your MVC application is using C# as the programming language.
- **.vbhtml**: When your MVC application is using VB is the programming language.

## 38. Explain briefly the two approaches of adding constraints to an MVC route?

For adding constraints to an MVC route, the 2 different approaches are:

- By making use of regular expressions.
- By making use of objects that implement the “IRouteConstraint” interface.

## 39. Point out the different stages a Page life cycle of MVC has?

The different steps or stages of the page life-cycle of MVC are:

- Initialization of app.
- Routing.
- Instantiate the object followed by executing the controller.
- Locate as well as invoke the controller action.
- Instantiating and then rendering the view.

## 40. Explain briefly the use of ViewModel in MVC?

ViewModel can be defined as a plain class having different properties. It is used for binding a view that is strongly typed. ViewModel consists of various validation rules for defining the properties of practising data annotation.

## 41. Define Default Route in MVC?

The default Route of project templates in MVC includes a generic route that makes use of the given URL resolution for breaking the URL based on the request into 3 tagged segments. **URL**: “{controller} / {action} / {id}”

## 42. Explain briefly the GET and POST Action types?

- The **GET Action Type** is implemented for requesting the data from a particular resource. Using these GET requests, a developer can pass the URL (that is compulsory).
- The **POST Action Type** is implemented for submitting the data that needs to be handled to a certain resource. Using these POST requests, a developer can move with the URL, which is essential along with the data.

**The primary rules for creating Razor are:**

- The block of Razor codes is enclosed within @{ ... }.
- Variables and functions of inline expressions start with @ symbol.
- The ‘var’ keyword is used for declaring variables.
- Razor code statements are terminated with a semicolon.
- C# files have .cshtml as file extension.

## 44. How can you implement the MVC forms authentication?

Authentication in forms is added in order to include a layer of security to access the user for a specific service. This authentication is done by verifying the user’s identity through the credentials such as username with password or email with a password.

The code snippet will look something like this:

```
<system.web>
    <authentication mode = "Forms" >
        <formsloginUrl = "Login.aspx" protection = "All" timeout = "30" name = ".ASPXAU
    </authentication>
</system.web>
```

## 45. What are the areas of benefits in using MVC?

The area of benefits of using MVC is:

- Unit testing becomes much easier.
- It permits its users to shape views, models, and controllers into 3 distinct operational sections within an application.
- It becomes easy to assimilate with other areas produced by another application.

## 46. Point out the two instances where you cannot use routing or where routing is not necessary

The 2 situations where routing is not used or not necessary are:

- When there is a physical file matching the URL pattern.
- When any routing gets disabled in any particular URL pattern.

## 47. How will you explain the concept of RenderBody and RenderPage of MVC?

RenderBody can be considered as a ContentPlaceHolder of web forms. It is available on the layout page and will be responsible for rendering the child pages/views. On the other hand, the layout page contains a single RenderBody() method. Multiple RenderPage() can reside within the Layout page.

## Additional Interview Resources:

- C# Interview Questions and Answers
- .NET Interview Questions and Answers
- ASP.NET Interview Questions and Answers
- Entity Framework Interview Questions and Answers

# Links to More Interview Questions

- C Interview Questions
- Web Api Interview Questions
- Cpp Interview Questions
- Machine Learning Interview Questions
- Css Interview Questions
- Django Interview Questions
- Operating System Interview Questions
- Git Interview Questions
- Dbms Interview Questions
- Pl Sql Interview Questions
- Ansible Interview Questions
- Php Interview Questions
- Hibernate Interview Questions
- Oops Interview Questions
- Docker Interview Questions
- Laravel Interview Questions
- Dot Net Interview Questions
- React Native Interview Questions
- Java 8 Interview Questions
- Spring Boot Interview Questions
- Tableau Interview Questions
- Java Interview Questions
- C Sharp Interview Questions
- Node Js Interview Questions
- Devops Interview Questions
- Mysql Interview Questions
- Asp Net Interview Questions
- Kubernetes Interview Questions
- Aws Interview Questions
- Mongodb Interview Questions
- Power Bi Interview Questions
- Linux Interview Questions
- Jenkins Interview Questions