



MVC Interview Questions and Answers

Introduction

Learning the Model-View-Controller (MVC) design pattern is a must for building scalable, maintainable, and highly organized web applications. This list of MVC interview questions and answers will help you review your skills with the basic concepts of Routing, Action Filters, Model Binding, and Dependency Injection in the ASP.NET MVC framework. Familiarity with the basic components ensures that you are able to explain how to develop robust enterprise-level applications that are flexible enough to adapt to the changing needs of the business. Are you ready to become a certified web architect? Learn the popular design pattern by taking our professional [MVC Certification Course in Chennai](#) today.

List of MVC Interview Questions for Freshers

1. What is MVC?
2. What are the features of MVC?
3. Describe the lifecycle of MVC applications.
4. What are filters in MVC?
5. What are the types of filters in MVC?

6. Explain action filters in MVC.
7. Which action filters are available for development in ASP.NET MVC?
8. What is a route in MVC?
9. How will the MVC sessions be maintained?
10. What is the partial view in MVC?
11. What distinguishes adding routes in an MVC application from a web form application?
12. Explain three logical layers in MVC.

Explore more with our [MVC course syllabus](#).

MVC Interview Questions and Answers for Freshers

1. What is MVC?

The acronym for Model-View-Controller is MVC. This separation of codes forces the domain model and controller logic to be isolated from the user interface (view). Testing and application maintenance become easier and simpler due to MVC.

2. What are the features of MVC?

Implementation of MVC has the following benefits:

- Multiple view support
- Faster development process
- SEO-friendly development
- Change accommodation
- Separation of concerns (SoC)
- High-level control over HTML, JavaScript, and CSS.
- Better testability.
- Lightweight
- Complete ASP.NET features.

3. Describe the lifecycle of MVC applications.

There are two primary phases in the execution of any web application: comprehending the request and responding appropriately based on the type of request. Request object creation and response delivery to the browser are the two primary stages of the MVC application life cycle, which is similar.

- Creating the request object
 - Fill route
 - Fetch route
 - Request context created
 - Controller instance created
- Create a response object

4. What are filters in MVC?

We frequently wish to carry out a task either before or after a specific procedure. ASP.NET MVC can add pre- and post-action behaviors like filters to the controller's action methods to accomplish this capability.

5. What are the types of filters in MVC?

Supported filters for the ASP.NET MVC framework include:

- Action filters
- Authorization filters
- Result filters
- Exception filters

6. Explain action filters in MVC.

Additional features called “action filters” can be applied to a segment of the controller or the controller as a whole to change how an action is carried out. These properties are unique .NET classes that come from the System class. Fields, properties, methods, and classes can all have this kind of attribute.

**Take your Knowledge
Test Report**

Check your Score



7. Which action filters are available for development in ASP.NET MVC?

- **Output Cache:** This action filter caches a controller action's output for a predefined amount of time.
- **Handle Error:** This action filter handles errors that arise when a controller action is being executed.
- **Authorize:** You can limit access to a specific user or position with this action filter.

8. What is a route in MVC?

A mapped URL pattern for a handler is called a route. The handler may be a physical file, like a Web Forms application's .aspx file.

9. How will the MVC sessions be maintained?

There are three alternative ways to keep an MVC session alive:

- viewdata
- temp data and
- view bag

10. What is the partial view in MVC?

A segment of HTML that has been carefully inserted into an already-existing DOM is known as a partial view.

Razor views are frequently componentized using partial views, which makes them easier to construct and maintain. Partial views can also be directly returned via controller methods.

11. What distinguishes adding routes in an MVC application from a web form application?

To add routes in a webform application, we use the MapPageRoute() function from the RouteCollection class. On the other hand, routes are added to an MVC application using the MapRoute() method.

12. Explain three logical layers in MVC.

The following definition applies to the three logical layers of MVC:

- **Model logic:** It covers a business layer.
- **View logic:** It is a representation of the display layer.
- **Controller logic:** Input control is performed via controller logic.

Begin with our [MVC tutorial for beginners](#).

List of MVC Interview Questions for Experienced

1. What is Spring MVC?
2. Why is MVC introducing bundling and minification?
3. In MVC, how is an error handled?
4. Define Scaffolding in MVC
5. Describe ViewStart in MVC
6. Identify the various phases of the MVC page life cycle.
7. What are the GET and POST action kinds?
8. How can the authentication for MVC forms be implemented?

Explore our [MVC project ideas](#) to learn more.

MVC Technical Interview Questions and Answers for Experienced

1. What is Spring MVC?

- A “Model View Controller” (MVC) architecture in the application and ready-to-use components designed for creating customizable and adaptive web applications are what make up the Spring MVC or Spring Web MVC framework.
- It’s a framework for creating web apps that run on Java. It utilizes the Model-View-Controller architecture paradigm.
- Additionally, this framework utilizes all of the fundamental features of the core Spring Framework, including inversion of control, lightweight development, dependency injection, and framework integration.
- DispatcherServlet provides a respectable way for Spring MVC to build MVC within the Spring Framework.

2. Why is MVC introducing bundling and minification?

MVC now includes two new techniques called bundling and minification, whose main purpose is to accelerate the request load time. By sending fewer queries to the server and requesting fewer assets (in this case, CSS and JavaScript), the load time is accelerated.

3. In MVC, how is an error handled?

Exception handling is typically used for handling errors in applications, whether they are web or Windows Forms-based.

Utilizing the `HandleError` property aids in the provision of integrated exception filters. Both the action method and the controller at the global level can use the ASP.NET `HandleError` attribute.

Example:

```
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);
```

```
}
```

4. Define Scaffolding in MVC.

The code-generation mechanism for ASP.NET web applications is known as scaffolding. The amount of time needed to develop a standard data operation in the application can also be reduced with scaffolding.

People who wish to quickly improve the code in their MVC apps that interacts with the data model can use scaffolding.

Take your Knowledge
Test Report

Check your Score



5. Describe ViewStart in MVC.

The Razor View Engine introduces a new layout called `_ViewStart`, which is automatically applied to all views. At the very beginning, ViewStart is executed, followed by additional views and the start of rendering.

Example:

```
@ {
```

```
    Layout = "~/Views/Shared/_
```

```
    file.cshtml";
```

```
}
```

```
<html>
```

```
    <head>
```

```
        <meta name="viewport" />
```

<title> InitialView </title> </head>

<body>

</body>

</html>

6. Identify the various phases of the MVC page life cycle.

The many phases of the MVC page life-cycle are:

- Initialization of the app and routing.
- After the object has been instantiated, run the controller.
- Find and execute the controller action.
- Rendering the view after it has been instantiated.

7. What are the GET and POST action kinds?

When requesting data from a specific resource, the GET Action Type is used. Developers can use these GET queries to provide the required URL.

The POST action type is used to provide data to a specific resource that needs to be processed. A developer can work with the URL, which is crucial, and the data by using these POST requests.

8. How can the authentication for MVC forms be implemented?

Forms with authentication add an extra degree of security to allow access to the user for a particular service. Verifying the user's identity via credentials, such as a password-protected email address or username, is how authentication is carried out.

This is how the code snippet will appear:

<system.web>

```
<authentication mode = "Forms" >  
  
    <formsloginUrl = "Login.aspx" protection = "All"  
    timeout = "30" name = ".ASPXAUTH" path = "/" requireSSL  
    = "false" defaultUrl = "default.aspx" cookieless =  
    "UseDeviceProfile" />  
  
</authentication>  
  
</system.web>
```

Conclusion

In an MVC interview, you must show your expertise in the application of Separation of Concerns, Routing, and Model Binding. The employer is seeking a developer who can take his/her skills beyond basic controllers, such as Dependency Injection, Action Filters, and Partial Views. The ability to explain the intricacies of Strongly Typed Views and Custom Validation Attributes will show your worth as a web application developer. As the world of web development is becoming more advanced, your expertise in ASP.NET Core MVC and Web API integration will make you stand out as a high-level full-stack architect. Ready to become a certified web developer? Learn the most popular web application design pattern in our [software training institute in Chennai](#).