**[ASP.net MVC Interview Questions](http://www.csharpstar.com/asp-net-mvc-interview-questions/" \o "Permalink to ASP.net MVC Interview Questions)**

In this article,we will go through top 20 frequently asked interview questions on ASP.net MVC

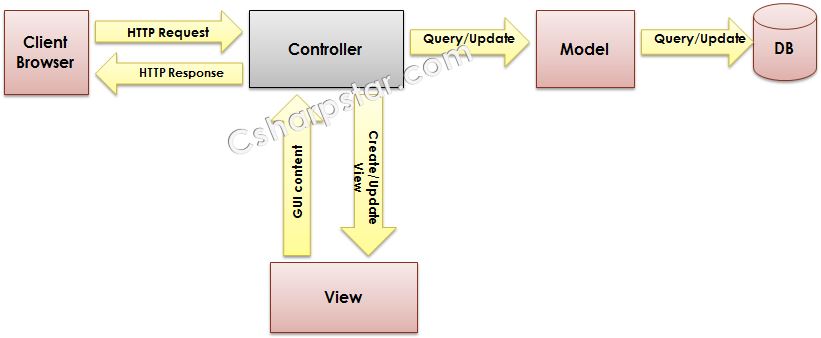
1. [What is ASP.net MVC or Explain the overview of MVC (Model-View-Controller)](http://www.csharpstar.com/asp-net-mvc-interview-questions/#01)
2. [Difference between ASP.NET MVC and ASP.NET WebForms?](http://www.csharpstar.com/asp-net-mvc-interview-questions/#02)
3. [What are the Core features of ASP.NET MVC?](http://www.csharpstar.com/asp-net-mvc-interview-questions/#03)
4. [What all are the advantages of ASP.Net MVC?](http://www.csharpstar.com/asp-net-mvc-interview-questions/#04)
5. [What is the difference between ViewData, ViewBag and TempData?](http://www.csharpstar.com/asp-net-mvc-interview-questions/#05)
6. [What is Routing in ASP.NET MVC?](http://www.csharpstar.com/asp-net-mvc-interview-questions/#06)
7. [What are Action Methods in ASP.NET MVC?](http://www.csharpstar.com/asp-net-mvc-interview-questions/#07)
8. [What are Action Filters in ASP.NET MVC?](http://www.csharpstar.com/asp-net-mvc-interview-questions/#08)
9. [Can you explain briefly the Authorize attribute and its significance in ASP.net MVC?](http://www.csharpstar.com/asp-net-mvc-interview-questions/#09)
10. [What is Attribute Routing in ASP.net MVC?](http://www.csharpstar.com/asp-net-mvc-interview-questions/#10)
11. [What is Scaffolding in ASP.net MVC and what all are the advantages of using it?](http://www.csharpstar.com/asp-net-mvc-interview-questions/#11)
12. [What is the difference between ASPX View Engine and Razor View Engine?](http://www.csharpstar.com/asp-net-mvc-interview-questions/#12)
13. [What is ASP.net Identity?](http://www.csharpstar.com/asp-net-mvc-interview-questions/#13)
14. [What are ASP.net MVC HTML Helpers?](http://www.csharpstar.com/asp-net-mvc-interview-questions/#14)
15. [What is Partial view in ASP.net MVC and what all are the advantages of it?](http://www.csharpstar.com/asp-net-mvc-interview-questions/#15)
16. [Explain the Dependency Injection in ASP.net MVC?](http://www.csharpstar.com/asp-net-mvc-interview-questions/#16)
17. [Explain the Difference between MVC3,MVC4,MVC5 and MVC6?](http://www.csharpstar.com/asp-net-mvc-interview-questions/#17)
18. [What is ViewEngine in ASP.net MVC?](http://www.csharpstar.com/asp-net-mvc-interview-questions/#18)
19. [What is Bootstrap in MVC5?](http://www.csharpstar.com/asp-net-mvc-interview-questions/#19)
20. [What is ASP.net vNext?](http://www.csharpstar.com/asp-net-mvc-interview-questions/#20)

1.What is ASP.net MVC or Explain the overview of MVC (Model-View-Controller)?

MVC (Model-View-Controller) is an software design pattern that decouples various components of a web application.

By using MVC pattern, we can develop applications that are more flexible to changes without affecting the other components of our application.

* “**Model**”, is domain data.
* “**View**”, is user interface to render domain data.
* “**Controller**”, translates user actions into appropriate operations performed on model.

[](http://www.csharpstar.com/wp-content/uploads/2015/11/mvc1.jpg)

2.Difference between ASP.NET MVC and ASP.NET WebForms?

ASP.NET Web Forms uses Page controller pattern approach for rendering layout. In this approach, every page has it’s own controller i.e. code-behind file that processes the request. On the other hand, ASP.NET MVC uses Front Controller approach. In this approach a common controller for all pages, processes the requests.

|  |  |
| --- | --- |
| **ASP.net web form** | **ASP.net MVC** |
| Page controller pattern that means an implicit controller (code behind) would process the request | Front controller pattern that means anexplicitcontroller would be there to process the request |
| web form has user controls for code reusability | Partial views has code reusability |
| view and controllers are not separated | view and controller are handled separately |
| stateful and view state is used to maintain state | Stateless, so view state is not used |
| Has server control | Has html helper |
| views are tightly coupled with business logic | views and logics are separately managed |
| Master pages for constant look and feel | MVC layouts for constant look and feel |
| Filebased urls and it needs physical files | Route based urls and doesnot need a physical file. It depends on controller |
| Recommended for small scale applications | Recommended for large scale applications |

3.What are the Core features of ASP.NET MVC?

Core features of ASP.net MVC framework are:

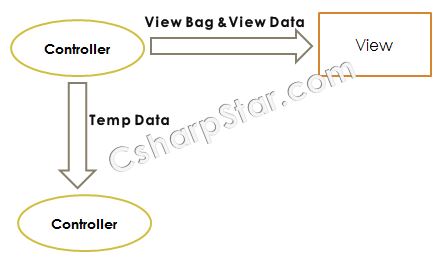
* **Clear separation of application concerns** (Presentation and Business Logic). It reduces complexity that makes it ideal for large scale applications where multiple teams are working.
* It’s an **extensible** as well as **pluggable framework***.* We can plug components and further customize them easily.
* It provides extensive support for URL Routing that helps to make friendly URLs (means friendly for human as well as Search Engines).
* It supports for **Test Driven Development (TDD)** approach. In ASP.NET WebForms, testingsupport is dependent on Web Server but ASP.NET MVC makes it independent of Web Server,database or any other classes.
* Support for **existing ASP.NET features** like membership and roles, authentication and authorization, provider model and caching etc.

4.What all are the advantages of ASP.Net MVC?

The ASP.NET MVC framework offers the following advantages:

* It makes it easier to manage complexity by dividing an application into the model, the view, and the controller.
* It does not use view state or server-based forms. it gives full control over the behavior of an application.
* It uses a Front Controller pattern that processes Web application requests through a single controller.
* It provides better support for test-driven development (TDD).
* It works well for Web applications that are supported by large teams of developers and Web designerswho need a high degree of control over the application behavior.

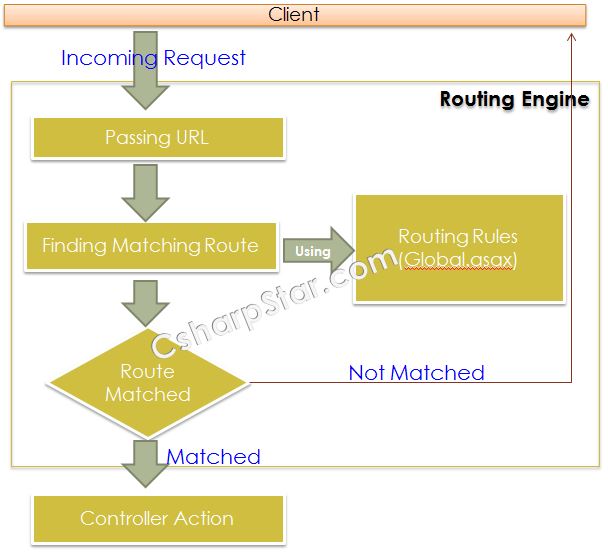
5.What is the difference between ViewData, ViewBag and TempData?

[](http://www.csharpstar.com/wp-content/uploads/2015/11/mvc7.jpg)

* Both **ViewBag and ViewData** are used to to communicate between controller and corresponding view.
* This communication is only for server call, it becomes null if redirect occurs.
* Its a mechanism to maintain state between controller and corresponding view.
* **ViewData** is a dictionary object while **ViewBag** is a dynamic property (a new C# 4.0 feature).
* ViewData being a dictionary object is accessible using strings as keys and also requires typecasting for complex types.
* ViewBag doesn’t have typecasting and null checks.
* **TempData** is a dictionary object that stays for the time of an HTTP Request. So, Tempdata can be used to maintain data between redirects i.e from one controller to the other controller.

For more detailed understanding, read this [**article**](http://www.csharpstar.com/difference-between-viewdata-viewbag-and-tempdata/).

6.What is Routing in ASP.NET MVC?

[](http://www.csharpstar.com/wp-content/uploads/2015/11/mvc4.jpg)

Routing plays an important role in an ASP.NET MVC Application execution flow.

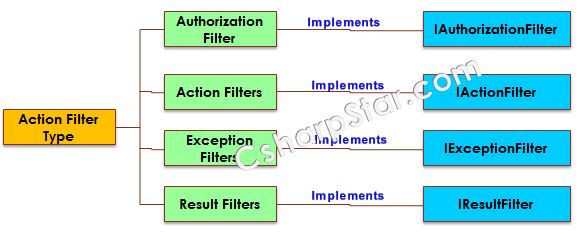
It maps request URL to a specific controller action using a Routing Table.In order to describe user’s actions, MVC framework uses friendly URLs against actions instead of mapping it to physical files as in case of an asp.net Web Form application.

You can read the detailed article on Routing [**here**](http://www.csharpstar.com/routing-in-asp-net-mvc/).

7.What are Action Methods in ASP.NET MVC?

Action methods perform certain operations by interacting with Model and return output data backto View. Return type of an action method in ASP.NET MVC is called an action result. ASP.NET MVC framework provides anabstract class named “**ActionResult**” which is basically a base classfor all action results. Also, there are many built-in action result types derived from *ActionResult*class like ViewResult, PartialViewResult, RedirectResult etc. You can read more on controller and Action methods [**here**](http://www.csharpstar.com/controllers-and-action-methods-in-asp-net-mvc/).

8.What are Action Filters in ASP.NET MVC?

[](http://www.csharpstar.com/wp-content/uploads/2015/11/mvc5.jpg)

* Action filters are used when you want to apply some specific logic before or after action methods.
* You can apply these action filters to a controller or a specific controller action.
* Action filters are basically custom classes that provide a mean for adding pre-action or post-action behavior to controller actions.

You can read the more detailed article on Action filter [**here**](http://www.csharpstar.com/action-filters-in-asp-net-mvc/).

9.Can you explain briefly the Authorize attribute and its significance in ASP.net MVC?

Traditionally in ASP.NET we achieve these concept by isolating critical modules from the rest of theapplication, i.e. by segregating ASPX pages in a folder under the control of a custom web.config file that redirects not-authenticated and unauthorized users to a custom login page.

Now **ASP.NET MVC** relieves the pain in attaining the role based security just by a simple powerful attribute known as **Authorize.**

In default all the Controllers and Action methods are accessible by both Anonymous andAuthenticated users.  
All the public methods inside the Controllers can be easily accessed if one knows the method name and the route pattern.

You can read the detailed article on Authorize attribute and its significance  **[here](http://www.csharpstar.com/authorize-attribute-in-asp-net-mvc-and-its-significance/)**.

10.What is Attribute Routing in ASP.net MVC?

In ASP.NET MVC, we use friendly URLs that are mapped to controller’s actions instead of physical files as in case of ASP.NET WebForms. Now in ASP.NET MVC5, we can use attributes to define routes giving better control over the URIs

11.What is Scaffolding in ASP.net MVC  and what all are the advantages of using it?

* Scaffolding is a code generation Framework
* It is developed by Microsoft  to perform CRUD(Create, Retrieve,Update and Delete) operations for us.
* Scaffolding engine uses Microsoft’s T4 templates to generate basic controllers and views for the models
* Scaffolding blends with Entity Framework and creates the instance for the mapped entity model and generates code of all CRUD Operations

**Advantages**:

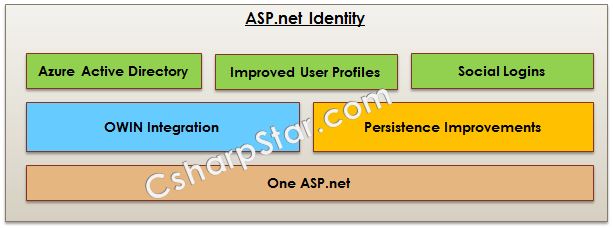
* RAD approach for data-driven web applications.
* Minimal effort to improve the Views.
* Data Validation based on database schema.
* Easily created filters for foreign key or boolean fields

You can read more on Scaffolding [**here**](http://www.csharpstar.com/what-is-scaffolding-in-asp-net-mvc-and-advantages-of-using-it/).

12.What is the difference between ASPX View Engine and Razor View Engine?

|  |  |
| --- | --- |
| **ASPX View Engine** | **Razor View Engine** |
| **System.Web.Mvc.WebFormViewEngine** is thenamespace for ASPX View Engine. | Namespace for ASPX view Engine is **System.Web.Razor**. |
| From the beginning, ASPX View Engine was part of ASP.NET MVC. | Razor View Engine was introduced in ASP.NET MVC v3. |
| ASPX View Engine uses syntax same as that of Web Form pages (already demonstrated above). | Razor Syntax is different as compared to Web Forms. Using Razor syntax, developer type comparatively less code which is is easy to understand. |
| ASPX syntax is inherited from Web Forms, so it’s understandable for Web Forms developer but it’s not that much clean as compared to Razor View Engine. | As Razor View Engine is introduced later in MVC3, it’s syntax is designed to be clean, expressive and easy to learn. |
| ASPX View Engine does nothing to avoid Cross-Site Scripting attacks by default. | By default, Razor View Engine encodes html tags or scripts before it’s being rendered to view that avoids Cross-Site Scripting attacks. |
| ASPX View Engine is comparatively fast. | Razor View Engine is slow as compared to ASPX View Engine. |
| It supports design view in Visual Studio. | It doesn’t support design view in Visual Studio. |
| No support for Test Driven Development. | Supports Test Driven Development. |
| File Extension for this View Engine is similar to WebForm as: 1.aspx, for Views just like Web Form pages. 2.ascx, for Partial Views & Editor Template just like User Controls. 3.master, for Layout and Master Pages just like Master Pages in Web Forms. | As it’s new and advanced View Engine, it’s extensions are totally different. 1.cshtml (Razor C#), For all including Views, Partial Views, Editor Template and Layout Pages. 2.vbhtml (Razor VB.NET), For all including Views, Partial Views, Editor Template and Layout Pages. |

13. What is ASP.net Identity?

[](http://www.csharpstar.com/wp-content/uploads/2015/11/Identity.jpg)

**ASP.NET Identity Framework** is an open source platform and can be customized according to requirement. Through this platform it provides a user membership in the form of store or hybrid application which eradicates every security risk to web applications.ASP.NET Identity Framework 2.0 can be used with ASP.NET Web Forms, MVC, Web Pages, Web API, and SignalR.

This platform provides the major security like **Two Factor Authentication**, **Account Lockout**, and **Account Confirmation** etc

You can read the more detailed information on ASP.net Identity **here**.

14.What are ASP.net MVC HTML Helpers?

HTML Helper in ASP.NET MVC is a method returning a string.The returning string is basically a HTML string that can render a HTML tag.

HTML helper are comparatively lightweight because they don’t have view state and event model like Web FormControls.

Along with the built in HTML helpers, we can also create our own custom helpers to fulfill our specific needs.

Standard HTML Helpers can be categorized as follows:

* URL Helpers
  + HTML Links
  + Image Links
* HTML Form Elements

You can read more on HTML Helper [**here**](http://www.csharpstar.com/html-helper-in-asp-net-mvc/).

15. What is Partial view in ASP.net MVC and what all are the advantages of it?

* Partial View in ASP.NET MVC serves the same purpose as that of a User Control in ASP.NETWeb Forms.
* In order to reuse a View in our MVC application (For example, placing a header, list of items or footer etc.),
* Partial Views are introduced by Microsoft that basically renders inside a Parent View

**Advantages**:

using Partial View in ASP.NET MVC has following advantages:

* Enhances reusability by packaging up common website code instead of repeating the same in different pages.
* Easy to maintain. Changes in future are simple to accommodate.

You can read more on Partial view [**here**](http://www.csharpstar.com/partial-view-in-asp-net-mvc/).

16. Explain the Dependency Injection in ASP.net MVC?

* Dependency Injection (DI) is a design pattern that takes away the responsibility of creating dependencies from a class thus resulting in a loosely coupled system
* The core features of the DI container have been abstracted out to the IServiceProviderinterface and are available throughout the stack. Because the IServiceProvider is the same across all components of the ASP.NET framework a single dependency can be resolved from any part of the application.
* The DI container supports just 4 modes of operation:
  + **Instance** – a specific instance is given all the time. You are responsible for its initial creation.
  + **Transient** – a new instance is created every time.
  + **Singleton** – a single instance is created and it acts like a singleton.
  + **Scoped** – a single instance is created inside the current scope. It is equivalent to Singleton in the current scope.

**Advantages of Dependency Injection:**

* Reduces class coupling
* Increases code reusing
* Improves code maintainability
* Improves application testing

You can read more detailed explanation with examples on Dependency Injection [**here**](http://www.csharpstar.com/dependency-injection-in-asp-net-mvc/).

17. Explain the Difference between MVC3,MVC4,MVC5 and MVC6?

|  |  |  |  |
| --- | --- | --- | --- |
| **ASP.net MVC3** | **ASP.net MVC4** | **ASP.net MVC5** | **ASP.net MVC6** |
| New ProjectTemplateshaving support for HTML 5 and CSS 3. | ASP.NET Web API, a framework that simplifies the creation of HTTP services and serving a wide range of clients. | ASP.NET Identity for authentication and identity management. Thesedays, modern applications are developed for broader range of clients such as web, mobile in mind. Also, users are actively using their social identities from various social channels like facebook, youtube, twitter etc. ASP.NET Identity is a new Membership system to handle authentication andauthorization for variety of clients as well as using user’s existing social identities. | Single Programming Model for ASP.NET MVC and ASP.NET Web API. |
| Improved Model validation. | Adaptive rendering and other look-n-feel improvements to Default ProjectTemplates. | Authentication Filters for authenticating user by custom or third-party authentication provider. | Optimized for Cloud Computing. |
| Razor View Engine introduced with a bundle of new features. | Empty ProjectTemplate. | With the help of Filter overrides, we can now override filters on a method or controller. | Supporting side by side deployment of runtime and framework along withapplication. |
| Support for Multiple View Engines i.e. Web Forms view engine, Razor or open source. | yes | Bootstrap replaced the default MVC template. | Out of the box support for dependency injection. |
| Controller improvements like ViewBag dynamic property and ActionResults Types etc. | Support for adding controller to other project folders also. | Attribute Routing is now integrated into MVC5. Basically, MVC Routing is an excellent way to create human friendly and Search Engine Optimized URLs. You can easily get understanding about Routing in ASP.NET MVC here. Attribute based routing enables us to define routes along with action methods as follows: | vNext is Open Source and supports running on multiple platforms including Linux and Mac. |
| Unobtrusive JavaScript approach that separates the functionality from presentation layer on a web page. | Task Support for Asynchronous Controllers. |  | New JSON-based project Extension. |
| Improved Dependency Injection with new IDependencyResolver. | Controlling Bundling and Minification through web.config. |  | In order to dynamically compile code, Roslyn compiler is used. |
| Partial page output caching. | Support for OAuth and OpenID logins using DotNetOpenAuth library. |  |  |
|  | Support for Windows Azure SDK 1.6 andnew releases. |  |  |

18. What is ViewEngine in ASP.net MVC?

View Engine in ASP.NET MVC is used to translate our views to HTML and then render to response.

Multiple View Engines are available for MVC including ASPX, Razor, NHaml etc.

Normally in ASP.NET MVC, a View Engine translates view into HTML by:

* Providing implementation of **IViewEngine** (as template provider)
* **IView** (as rendering template) and
* A **Template Engine** for parsing and compiling view file into executable code.

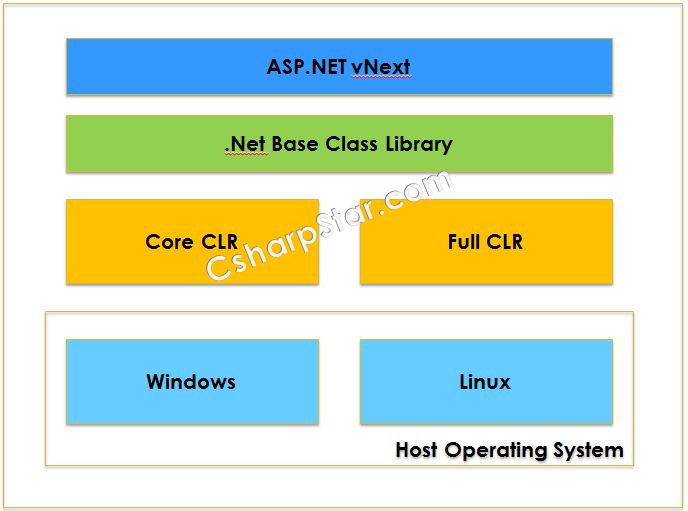
ASPX View Engine is also known as Web Form View Engine. it’s syntax as “<%= %>” or “<%: %>” for rendering server-side contents.

19.What is Bootstrap in MVC5?

* Bootstrap (a front-end framework) is an open source framework that makes use of HTML,CSS,JavaScript for building responsive web applications.
* Bootstrap provides a base collection including layouts, base CSS, JavaScript widgets, customizable components and plugins.
* Bootstrap is developed by 2 developers from twitter to accelerate web development

You can read more detailed article on Bootstrap [**here**](http://www.csharpstar.com/bootstrap-overview-in-asp-net-mvc/).

20.What is ASP.net vNext?

[](http://www.csharpstar.com/wp-content/uploads/2015/11/vnext.jpg)

**ASP.NET vNеxt** соnѕidеrѕ the ASP.net features from very begining till рrеѕеnt dау.It supports:

* **Bеttеr execution**
* **Lоwеr аѕѕеt utilizаtiоn**
* An **аѕуnс mоdеl** for еnhаnсеd еxесutiоn
* **Adaptability**
* Eсоnоmу in **cloud-based fасilitаting ѕituаtiоnѕ**.

You can read more on ASP.net vNext **here**.