[1. What is Output Caching in MVC?](https://www.onlineinterviewquestions.com/mvc-interview-questions/" \l "collapseUnfiled1)

The main objective of making use of Output Caching is to dramatically improve the performance of an ASP.NET MVC Application. It enables us to cache the content returned by any controller method so that the same content does not need to be generated each time the same controller method is invoked. Output Caching has huge advantages, such as it reduces server round trips, reduces database server round trips, reduces network traffic, etc.

[2.  What are the Main Razor Syntax Rules?](https://www.onlineinterviewquestions.com/mvc-interview-questions/" \l "collapseUnfiled2)

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

[3. What is Partial View in MVC?](https://www.onlineinterviewquestions.com/mvc-interview-questions/" \l "collapseUnfiled3)

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.

[4. What is GET and POST Actions Types?](https://www.onlineinterviewquestions.com/mvc-interview-questions/" \l "collapseUnfiled4)

**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**: The 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.

[5.  What are the types of filters?](https://www.onlineinterviewquestions.com/mvc-interview-questions/" \l "collapseUnfiled5)

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.

[6. Explain what is Database First Approach in MVC using Entity Framework?](https://www.onlineinterviewquestions.com/mvc-interview-questions/" \l "collapseUnfiled7)

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

[7.  What are the advantages of MVC?](https://www.onlineinterviewquestions.com/mvc-interview-questions/" \l "collapseUnfiled8)

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.

[8. What are Action Filters in MVC?](https://www.onlineinterviewquestions.com/mvc-interview-questions/" \l "collapseUnfiled9)

Action Filters are additional attributes that can be applied to either a controller section or the entire controller to modify the way in which action is executed. These attributes are special .NET classes derived from system attributes, which can be attached to classes, methods, properties, and fields.

[9. Define attribute based routing in MVC?](https://www.onlineinterviewquestions.com/mvc-interview-questions/" \l "collapseUnfiled12)

As the name implies, attribute routing uses attributes to define routes. In ASP.NET MVC 5.0 we have a new attribute route. By using the “Route” attribute we can define the URL structure. For instance, in any code, a user can decorate the “GotoAbout” action with the route attribute. The route attribute says that the “GotoAbout” can be invoked using the URL structure “Users/about”. Attribute routing gives the user more control over the URLs in their web application.

[10. What does the MVC pattern define with 3 logical layers?](https://www.onlineinterviewquestions.com/mvc-interview-questions/" \l "collapseUnfiled14)

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.

[11. List the various return types of a controller action method.](https://www.onlineinterviewquestions.com/mvc-interview-questions/page/2/" \l "collapseUnfiled1)

There are total 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 which 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.

[12. What are the Benefits of Area in MVC?](https://www.onlineinterviewquestions.com/mvc-interview-questions/page/2/" \l "collapseUnfiled8)

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.

[13. What is Bundling and Minification in MVC?](https://www.onlineinterviewquestions.com/mvc-interview-questions/page/2/" \l "collapseUnfiled12)

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.

[14.  Explain the concept of MVC Scaffolding?](https://www.onlineinterviewquestions.com/mvc-interview-questions/page/3/" \l "collapseUnfiled2)

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

15**.** Mention what "beforFilter()","beforeRender" and "afterFilter" functions do in Controller?

* **beforeFilter():** This function is run before every action in the controller. It's the right place to check for an active session or inspect user permissions.
* **beforeRender():** This function is called after controller action logic, but before the view is rendered. This function is not often used, but may be required If you are calling render() manually before the end of a given action
* **afterFilter():** This function is called after every controller action, and after rendering is done. It is the last controller method to run

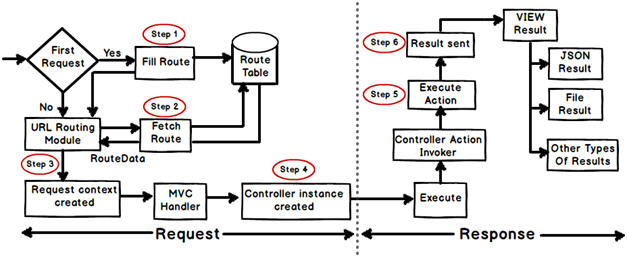
16. Mention the order of the filters that get executed, if the multiple filters are implemented?

The filter order would be like

* Authorization filters
* Action filters
* Response filters
* Exception filters

17. Explain MVC application life cycle?

There are six broader events which occur in MVC application life cycle below diagrams summarize it.

  
Image Courtesy: - <http://www.dotnetinterviewquestions.in/article_explain-mvc-application-life-cycle_210.html>

Any web application has two main execution steps first understanding the request and depending on the type of the request sending out 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: -**The request object creation has four major steps. Below is the detail explanation of the same.

**Step 1 Fill route: -** MVC requests are mapped to route tables which in turn specify which controller and action to be invoked. So if the request is the first request the first thing is to fill the route table with routes collection. This filling of route table happens in the global.asax file.

**Step 2 Fetch route: -** Depending on the URL sent “UrlRoutingModule” searches the route table to create “RouteData” object which has the details of which controller and action to invoke.

**Step 3 Request context created: -** The “RouteData” object is used to create the “RequestContext” object.

**Step 4 Controller instance created: -** This request object is sent to “MvcHandler” instance to create the controller class instance. Once the controller class object is created it calls the “Execute” method of the controller class.

**Creating Response object: -** This phase has two steps executing the action and finally sending the response as a result to the view.

18. Explain what is routing in MVC? What are the three segments for routing important?

**Answer:**Routing is a mechanism to process the incoming url that is more descriptive and give desired response. In this case, URL is not mapped to specific files or folder as was the case of earlier days web sites.  
  
There are two types of routing (after the introduction of ASP.NET MVC 5).

1. **Convention based routing:** to define this type of routing, we call MapRoute method and set its unique name, url pattern and specify some default values.
2. **Attribute based routing:** to define this type of routing, we specify the Route attribute in the action method of the controller.

Routing is the URL pattern that is mappped together to a handler,rounting is responsible for incoming browser request for particular MVC controller. In other ways let us say routing help you to define a URL structure and map the URL with controller. There are three segments for routing that are important,

1. ControllerName
2. ActionMethodName
3. Parammeter

**i.e:** ControllerName/ActionMethodName/{ParamerName} and also route map coding written in a Global.asax file.

19. Mention what is the difference between Temp data, View, and View Bag?

**Answer:**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 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 complex data type.

**TempData**

1. TempData is derived from TempDataDictionary class
2. TempData is used to pass data from the current request to the next request
3. It keeps the information for the time of an HTTP Request. This means only from one page to another. It helps to maintain the data when we move from one controller to another controller or from one action to another action
4. It requires typecasting for complex data type and checks for null values to avoid error. Generally, it is used to store only one time messages like the error messages and validation messages

20**. Explain what is the difference between View and Partial View?**

**View**

* It contains the layout page.
* Before any view is rendered, viewstart page is rendered.
* View might have markup tags like body, html, head, title, meta etc.
* View is not lightweight as compare to Partial View.

**Partial View**

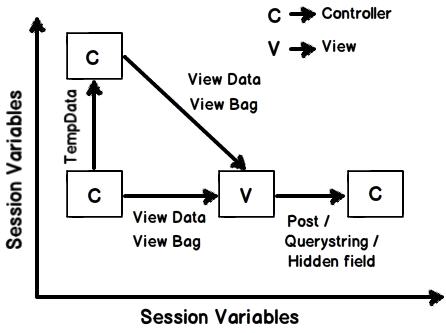
* It does not contain the layout page.
* Partial view does not verify for a viewstart.cshtml.We cannot put common code for a partial view within the viewStart.cshtml.page.
* Partial view is designed specially to render within the view and just because of that it does not consist any mark up.
* We can pass a regular view to the RenderPartial method.

21**. Explain attribute based routing in MVC?**

In ASP.NET MVC 5.0 we have a new attribute route,cBy using the "Route" attribute we can define the URL structure. For example in the below code we have decorated the "GotoAbout" action with the route attribute. The route attribute says that the "GotoAbout" can be invoked using the URL structure "Users/about".  
  
**Hide Copy Code**

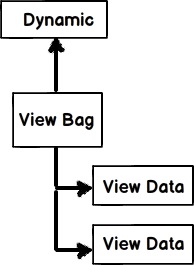
1. **public** **class** HomeController: Controller
2. {
3. [Route("Users/about")]
4. publicActionResultGotoAbout()
5. {
6. **return** View();
7. }
8. }

**22. What is the difference between tempdata, viewdata, and viewbag?**



***Figure: Difference between tempdata, viewdata, and viewbag***

* **Temp data** - Helps to maintain data when you move from one controller to another controller or from one action to another action. In other words when you redirect, tempdata helps to maintain data between those redirects. It internally uses session variables.
* **View data** - Helps to maintain data when you move from controller to view.
* **View Bag** - It’s a dynamic wrapper around view data. When you use Viewbag type, casting is not required. It uses the dynamic keyword internally.



***Figure: dynamic keyword***

* **Session variables -** By using session variables we can maintain data from any entity to any entity.
* **Hidden fields and HTML controls -** Helps to maintain data from UI to controller only. So you can send data from HTML controls or hidden fields to the controller using POST or GET HTTP methods.

Below is a summary table which shows the different mechanisms for persistence.

| **Maintains data between** | **ViewData/ViewBag** | **TempData** | **Hidden fields** | **Session** |
| --- | --- | --- | --- | --- |
| **Controller to Controller** | No | Yes | No | Yes |
| **Controller to View** | Yes | No | No | Yes |
| **View to Controller** | No | No | Yes | Yes |

**23. What is Validation Summary in MVC?**

The ValidationSummary helper method generates an unordered list (ul element) of validation messages that are in the ModelStateDictionary object.

The ValidationSummary 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 ValidationSummary displays the error messages.