**What is a Filter in MVC?**

When user (browser) sends a request to server an action method of a controller gets invoked; sometimes you may require executing a custom code before or after action method gets invoked, this custom code is called as Filter.

**What are the different types of Filters in MVC?**

a. Authorization filter

b. Action filter

c. Result filter

d. Exception filter

[Do not forget the order mentioned above as filters gets executed as per above mentioned sequence]

**What is the Request flow used for ASP.NET MVC framework?**

Request flow handles the request from the clients and passes it to the server. The Request flow is as follows:  
-Request is being taken from User to controller.  
-Controller processes the request from the user and creates a data Model of that particular request.   
-Data model that is being created is then passed to View that handles the frontend or the design.   
-View then transforms the Data Model by using its own functions in an appropriate output format.  
-The output format that is being given by the View is then gets rendered to the Browser and the View will be seen by the user.

|  |
| --- |
| **Can you describe ASP.NET MVC Request Life Cycle?** |
| Following are the steps that are executed in ASP.NET MVC Request Life Cycle.  The following table lists the stages of execution for an MVC Web project.   |  |  | | --- | --- | | **Stage** | **Details** | | Receive first request for the application | In the Global.asax file, **Route** objects are added to the [RouteTable](http://msdn.microsoft.com/en-us/library/system.web.routing.routetable(v=vs.100).aspx) object. | | Perform routing | The [UrlRoutingModule](http://msdn.microsoft.com/en-us/library/system.web.routing.urlroutingmodule(v=vs.100).aspx) module uses the first matching **Route** object in the [RouteTable](http://msdn.microsoft.com/en-us/library/system.web.routing.routetable(v=vs.100).aspx) collection to create the [RouteData](http://msdn.microsoft.com/en-us/library/system.web.routing.routedata(v=vs.100).aspx) object, which it then uses to create a [RequestContext](http://msdn.microsoft.com/en-us/library/system.web.routing.requestcontext(v=vs.100).aspx) object. | | Create MVC request handler | The [MvcRouteHandler](http://msdn.microsoft.com/en-us/library/system.web.mvc.mvcroutehandler(v=vs.100).aspx) object creates an instance of the [MvcHandler](http://msdn.microsoft.com/en-us/library/system.web.mvc.mvchandler(v=vs.100).aspx) class and passes the [RequestContext](http://msdn.microsoft.com/en-us/library/system.web.routing.requestcontext(v=vs.100).aspx) instance to the handler. | | Create controller | The [MvcHandler](http://msdn.microsoft.com/en-us/library/system.web.mvc.mvchandler(v=vs.100).aspx) object uses the [RequestContext](http://msdn.microsoft.com/en-us/library/system.web.routing.requestcontext(v=vs.100).aspx) instance to identify the [IControllerFactory](http://msdn.microsoft.com/en-us/library/system.web.mvc.icontrollerfactory(v=vs.100).aspx) object (typically an instance of the[DefaultControllerFactory](http://msdn.microsoft.com/en-us/library/system.web.mvc.defaultcontrollerfactory(v=vs.100).aspx) class) to create the controller instance with. | | Execute controller | The [MvcHandler](http://msdn.microsoft.com/en-us/library/system.web.mvc.mvchandler(v=vs.100).aspx) instance calls the controller's [Execute](http://msdn.microsoft.com/en-us/library/system.web.mvc.controllerbase.execute(v=vs.100).aspx) method. | | Invoke action | For controllers that inherit from the [ControllerBase](http://msdn.microsoft.com/en-us/library/system.web.mvc.controllerbase(v=vs.100).aspx) class, the [ControllerActionInvoker](http://msdn.microsoft.com/en-us/library/system.web.mvc.controlleractioninvoker(v=vs.100).aspx) object that is associated with the controller determines which action method of the controller class to call, and then calls that method. | | Execute result | The action method receives user input, prepares the appropriate response data, and then executes the result by returning a result type. The built-in result types that can be executed include the following: [ViewResult](http://msdn.microsoft.com/en-us/library/system.web.mvc.viewresult(v=vs.100).aspx) (which renders a view and is the most-often used result type),[RedirectToRouteResult](http://msdn.microsoft.com/en-us/library/system.web.mvc.redirecttorouteresult(v=vs.100).aspx), [RedirectResult](http://msdn.microsoft.com/en-us/library/system.web.mvc.redirectresult(v=vs.100).aspx), [ContentResult](http://msdn.microsoft.com/en-us/library/system.web.mvc.contentresult(v=vs.100).aspx), [JsonResult](http://msdn.microsoft.com/en-us/library/system.web.mvc.jsonresult(v=vs.100).aspx), [FileResult](http://msdn.microsoft.com/en-us/library/system.web.mvc.fileresult(v=vs.100).aspx), and [EmptyResult](http://msdn.microsoft.com/en-us/library/system.web.mvc.emptyresult(v=vs.100).aspx). | |

**What is the main function of URL routing system in ASP.NET MVC?**

* URL routing system provides flexibility to the system and it also enables to define new URL mapping rules that can be used with web applications.
* URL routing system is used to map the application and its routing information gets passed to right controller and action method.
* URL routing system processes and executes the method to run the application without using many designed rules.
* It is used to construct the outgoing URLs that can be used to handle the actions that have the ability to map both incoming and outgoing URLs that adds more flexibility to the application code.
* It follows the rules to execute the application globally and handle the logic that is required for the application.

To add the anti-forgery tokens to a Razor page, use the **HtmlHelper.AntiForgeryToken** helper method:

@using (Html.BeginForm("Manage", "Account")) {

@Html.AntiForgeryToken()

}

# Anti-CSRF and AJAX

The form token can be a problem for AJAX requests, because an AJAX request might send JSON data, not HTML form data. One solution is to send the tokens in a custom HTTP header. The following code uses Razor syntax to generate the tokens, and then adds the tokens to an AJAX request. The tokens are generated at the server by calling **AntiForgery.GetTokens**.

<script>

@functions{

public string TokenHeaderValue()

{

string cookieToken, formToken;

AntiForgery.GetTokens(null, out cookieToken, out formToken);

return cookieToken + ":" + formToken;

}

}

$.ajax("api/values", {

type: "post",

contentType: "application/json",

data: { }, // JSON data goes here

dataType: "json",

headers: {

'RequestVerificationToken': '@TokenHeaderValue()'

}

});

</script>

When you process the request, extract the tokens from the request header. Then call the **AntiForgery.Validate** method to validate the tokens. The **Validate** method throws an exception if the tokens are not valid.

void ValidateRequestHeader(HttpRequestMessage request)

{

string cookieToken = "";

string formToken = "";

IEnumerable<string> tokenHeaders;

if (request.Headers.TryGetValues("RequestVerificationToken", out tokenHeaders))

{

string[] tokens = tokenHeaders.First().Split(':');

if (tokens.Length == 2)

{

cookieToken = tokens[0].Trim();

formToken = tokens[1].Trim();

}

}

AntiForgery.Validate(cookieToken, formToken);

}

**What are sections?**  
Layout pages, can define sections, which can then be overriden by specific views making use of the layout. Defining and overriding sections is optional.

**When using razor views, do you have to take any special steps to proctect your asp.net mvc application from cross site scripting (XSS) attacks?**  
No, by default content emitted using a @ block is automatically HTML encoded to protect from cross site scripting (XSS) attacks.

**In razor syntax, what is the escape sequence character for @ symbol?**  
The escape sequence character for @ symbol, is another @ symbol