**FILTERS IN ASP.NET MVC 5**

**MVC APPLICATION WORKFLOW**

**CLIENT**

**ROUTE / ROUTING**

**CONTROLLER**

**ACTION METHOD**

**VIEW**

In an ASP.NET MVC application there might be situations where you need to implement some functionality before or after the execution of an action method.

**BEFORE EXECUTING ACTION METHOD**

**ACTION METHOD**

**AFTER EXECUTING ACTION METHOD**

In such situations, you need to use filters.

**There are five types of Filters in ASP.NET MVC 5.**

* Authentication Filters
* Authorization Filters
* Action Filters
* Result Filters
* Exception Filters

**Note:** Types of Filters in ASP.NET MVC and their Sequence of Execution

**Note:** We can use built-in filters and custom filters in MVC.

**While developing an ASP.NET MVC application, you can use filters at the following levels:**

* **Action method Level:** When you use filters in an action method, the filter will execute only when the associated action method is accessed.
* **Controller Level:** When you use filters in controller, the filter will execute for all the actions methods defined in the controller.
* **Application Level:** When you use filters in an application, the filter will execute for all the actions methods and controllers present in the application.

**EXCEPTION FILTERS IN ASP.NET MVC 5**

* In an ASP.NET MVC application, you can use exception filters to handle exceptions that the application throws at runtime.
* Exception filters are additional exception handling component of MVC Framework besides the built-in .NET Framework exception handling mechanism comprising try-catch block.
* The MVC Framework provides a built-in exception filter through the HandleError filter that the HandleErrorAttribute class implements.
* Like other filters, you can use the HandleError filter on an action method or a controller.
* Using HandleError we can avoid try catch.
* The HandleError filter handles the exceptions that are raised by the controller actions and other filters applied to the action.

**TO USE HandleError FILTER IN MVC APPLICATION, YOU HAVE TO DO 3 STEPS:**

1. This filter returns a view named Error.cshtml which by default is in the shared folder of the application that’s why we have to create Error.cshtml view in shared folder.
2. To use the HandleError filter, you need to configure the web.config file by adding a customErrors attribute inside the <system.web> element. In this code, the “On” value of the mode attribute in the <customErrors> element enables exception handling using the HandleError filter.
3. Using HandlerError on Action Method / Controller / Global.