Filters in MVC

1. In ASP.NET MVC, a user request is routed to the appropriate controller and action method.
2. However, there may be circumstances where we want to execute some logic before or after an action method executes. ASP.NET MVC provides filters for this purpose.
3. ASP.NET MVC Filter is a custom class where we can write custom logic to execute before or after an action method executes.
4. Filters can be applied to an action method or controller in a declarative or programmatic way.
5. Declarative means by applying a filter attribute to an action method or controller class.
6. Programmatic means by implementing a corresponding interface.

Types of Filters

1. **Authorization filters: -** Performs authentication and authorizes before executing action method.

**Built-in-Filter: -** [Authorize], [RequestHttps]

**Interface: -** IAuthorizationFilter

1. **Action filters: -** Performs some operation before and after an action method executes. **Interface: -** IActionFilter
2. **Result filters: -** Performs some operation before or after the execution of view result.

**Built-in-Filter: -** [OutputCache]

**Interface: -** IAuthorizationFilter

1. **Exception filters: -** Performs some operation if there is an unhandled exception thrown during the execution of the ASP.NET MVC pipeline.

**Built-in-Filter: -** [Handle Error]

**Interface: -** IExceptionFilter