Filters in **ASP.NET Core** allow you to run custom logic at specific points in the request processing pipeline. They help manage cross-cutting concerns like **logging, authorization, validation, exception handling, and response shaping**.

**Types of Filters:**

1. **Authorization Filters** – Run first, determining if the user is authorized.
2. **Resource Filters** – Execute before model binding and after the request pipeline.
3. **Action Filters** – Run before and after an action method is called.
4. **Exception Filters** – Handle unhandled exceptions globally.
5. **Result Filters** – Execute before and after action results.
6. **Endpoint Filters** – Introduced in .NET 7, they allow request/response modifications at the endpoint level.

Exception filters are useful for **centralized error handling**, logging, and returning custom error responses. They can be implemented using the IExceptionFilter interface or by inheriting from ExceptionFilterAttribute.