1. **ASP.NET Core Request Processing Pipeline**

# Middleware

* Middleware is a software component that sits between the web server and the application. It intercepts incoming requests and outgoing responses and allows you to modify them. Middleware can be used to perform a wide range of tasks such as authentication, logging, compression, and caching.
* Middleware works by using a pipeline model. When a request comes in, it is passed through a series of middleware components before it reaches the application. Each middleware component can modify the request before passing it on to the next component. Once the request has passed through all the middleware components, it reaches the application. Similarly, when a response is sent back to the client, it is passed through a series of middleware components before it is sent back to the client.
* There are two types of middleware in .NET Core : terminal and non-terminal middleware. Terminal middleware is the final middleware component in the pipeline. It is responsible for sending the response back to the client. Non-terminal middleware is any middleware component that is not the final component in the pipeline.
* Terminal Middleware
* The terminal middleware is responsible for sending the response back to the client. It is the final middleware component in the pipeline. Terminal middleware can be used to modify the outgoing response before it is sent back to the client.
* Some examples of terminal middleware include:
  + Static files middleware: This middleware is used to serve static files such as CSS, JavaScript, and images.
  + File server middleware: This middleware is used to serve files from a specified directory.
  + MVC middleware: This middleware is used to handle requests for MVC endpoints.
* Non-terminal Middleware
* Non-terminal middleware is any middleware component that is not the final component in the pipeline. Non-terminal middleware can be used to modify incoming requests and outgoing responses.
* Some examples of non-terminal middleware include:
  + Authentication middleware: This middleware is used to authenticate users.
  + Authorization middleware: This middleware is used to authorize users to access certain resources.
  + Response compression middleware: This middleware is used to compress the response before it is sent back to the client.
  + Request logging middleware: This middleware is used to log requests.
  + Routing middleware: This middleware is used to route requests to the appropriate endpoint.
* Custom Middleware
  + In addition to the built-in middleware components that are available in .NET Core, you can also create your own custom middleware components. Creating custom middleware allows you to add functionality to your application that is specific to your needs.
  + To create custom middleware, you need to create a class that implements the IMiddleware interface. The IMiddleware interface has a single method called InvokeAsync that is called when the middleware component is invoked.

# Routing

# Filters

# Controller Initialization

# Action Method