**Middleware :**

**Middleware** in ASP.**NET Core** controls how our application responds to HTTP requests. ... **Middleware** are software components that are assembled into an application pipeline to handle requests and responses.

Middleware in ASP.NET Core controls how our application responds to HTTP requests. It can also control how our application looks when there is an error, and it is a key piece in how we authenticate and authorize a user to perform specific actions.

* Middleware are software components that are assembled into an application pipeline to handle requests and responses.
* Each component chooses whether to pass the request on to the next component in the pipeline, and can perform certain actions before and after the next component is invoked in the pipeline.
* Request delegates are used to build the request pipeline. The request delegates handle each HTTP request.
* Each piece of middleware in ASP.NET Core is an object, and each piece has a very specific, focused, and limited role.
* Ultimately, we need many pieces of middleware for an application to behave appropriately

**Run, Map, Use**

Example[#](https://riptutorial.com/asp-net-core/example/20718/run--map--use#example)

**Run**

Terminates chain. No other middleware method will run after this. Should be placed at the end of any pipeline.

app.Run(async context =>

{

await context.Response.WriteAsync("Hello from " + \_environment);

});

**Use**

Performs action before and after next delegate.

app.Use(async (context, next) =>

{

//action before next delegate

await next.Invoke(); //call next middleware

//action after called middleware

});

Ilustration of how it works: [](https://i.stack.imgur.com/YXaaj.png)

**MapWhen**

Enables branching pipeline. Runs specified middleware if condition is met.

private static void HandleBranch(IApplicationBuilder app)

{

app.Run(async context =>

{

await context.Response.WriteAsync("Condition is fulfilled");

});

}

public void ConfigureMapWhen(IApplicationBuilder app)

{

app.MapWhen(context => {

return context.Request.Query.ContainsKey("somekey");

}, HandleBranch);

}

**Map**

Similar to MapWhen. Runs middleware if path requested by user equals path provided in parameter.

private static void HandleMapTest(IApplicationBuilder app)

{

app.Run(async context =>

{

await context.Response.WriteAsync("Map Test Successful");

});

}

public void ConfigureMapping(IApplicationBuilder app)

{

app.Map("/maptest", HandleMapTest);

}