

### Download Swagger:

Go to nuget in the terminal by clicking the three dots and clicking nuget and then type in swash and download swashbuckle to test your endpoints you need three swashbuckle packages and make sure they are version 9.0.0.

### Download the following:

```
<PackageReference Include="Swashbuckle.AspNetCore" Version="9.0.0" />
<PackageReference Include="Swashbuckle.AspNetCore.SwaggerGen"
Version="9.0.0" />
<PackageReference Include="Swashbuckle.AspNetCore.SwaggerUI"
Version="9.0.0" />
```

Then make a Services Folder and create a HelloWorldServices class in the services folder  
To create a class right click on the service or controller folder and click class or for controller do a api controller.

In the HelloWorldServices add the code to run in the controller later.

Should look like this:

```
public class HelloWorldServices
{
    public string SayHelloWorld()
    {
        return $"Hello World";
    }
}
```

Now we go to the [program.cs](#) and we can now scope our services and swagger.

### We add the following:

```
builder.Services.AddScoped<HelloWorldServices>();
```

Make sure its before the var app = builder.build(); because it has to read the code before it builds the endpoints.

Now we add swagger calls.

So what is under the comment replace it to have the following two lines of code.

```
// Learn more about configuring OpenAPI at https://aka.ms/aspnet/openapi
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();
```

And in the app.environment if statement replace whats in it with the useswagger and useswagger ui code.

```
if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}
```

Now create a controller folder and create a HelloWorldController and some code to connect the services and return back the hello world response.

```
[ApiController]
[Route("[controller]")]
public class HelloWorldController : ControllerBase
{
    private readonly HelloWorldServices _helloWorldServices;

    public HelloWorldController(HelloWorldServices helloWorldServices)
    {
        _helloWorldServices = helloWorldServices;
    }

    [HttpGet]
    [Route("SayHelloWorld")]
    public string SayHelloWorld()
    {
        return _helloWorldServices.SayHelloWorld();
    }
}
```

Now we can run it to test our code and endpoints so open up the terminal and type in dotnet watch run and then when it opens up test the endpoints.