

Bryce Mack

1/30/26

Guide on how to make an api endpoint say Hello World.

1. Create the API project

First, open the terminal in the folder where you want your project.

Type this command and press Enter:

```
dotnet new webapi --use-controllers
```

2. Add Swagger (for testing your API)

Swagger lets you test your API in the browser.

1. Open your .csproj file to check what version of .NET you are using.
2. Open NuGet Package Manager
3. Search for Swashbuckle
4. Install these three packages (make sure the versions match your .NET version):
 - Swashbuckle.AspNetCore
 - Swashbuckle.AspNetCore.SwaggerGen
 - Swashbuckle.AspNetCore.SwaggerUI

When finished, all three should appear in your .csproj file.

3. Enable Swagger to open automatically

1. Open launchSettings.json (inside the Properties folder)

2. Change both `"launchBrowser"` values from `false` to `true`

3. Under each one, add:

```
"launchUrl": "swagger"
```

4. Update `Program.cs`

Open `Program.cs` and make these changes:

Replace this:

```
builder.Services.AddOpenApi();
```

With this:

```
builder.Services.AddEndpointsApiExplorer();  
builder.Services.AddSwaggerGen();
```

Replace this:

```
app.MapOpenApi();
```

With this:

```
app.UseSwagger();  
app.UseSwaggerUI();
```

5. Create a Services folder

1. Right-click your project
2. Create a folder named Services
3. Make sure it is not inside another folder

If your API is inside its own folder, the Services folder should be there too.

6. Create the service logic

1. Right-click the Services folder
2. Add a new Class
3. Name it `HelloWorldService`

Inside the class, add this method:

```
public string HelloWorld()  
{  
    return "Hello, World!";  
}
```

This is the code that returns the message.

7. Create the controller

1. Open the Controllers folder
2. Add a new API Controller
3. Name it `HelloWorldController`

Inside the controller:

Connect the service:

```
private readonly HelloWorldService _helloWorldService;
```

Add a constructor:

```
public HelloWorldController(HelloWorldService helloWorldService)
{
    _helloWorldService = helloWorldService;
}
```

8. Create the endpoint

Below the constructor, add:

```
[HttpGet]
[Route("HelloWorld")]
public string HelloWorld()
{
    return _helloWorldService.HelloWorld();
}
```

This creates a GET endpoint that returns "Hello, World!"

9. Register the service

Go back to `Program.cs` and add:

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

Put this under the other `builder.Services` lines.

10. Run the API

In the terminal, run:

```
dotnet watch run
```

If your API is inside a folder, first do:

```
cd FolderName
```

Swagger should open automatically in your browser.

11. Test the endpoint

1. Find HelloWorld in Swagger
2. Open the GET dropdown
3. Click Try it out
4. Click Execute

You should see the response:

`Hello, World!`