Systems Integration - Practical exercises Module 2

Goal

We put our commerce registration project aside and we will implement a simple REST service with Microsoft ASP.NET Core Web API technology. The goal of this exercise is to write a small service that calculates and returns the elements of a Fibonacci sequence. The sequence starts with 1 and each element is calculated as the sum of the previous two elements, for example 1, 1, 2, 3, 5, 8, 13, …

A HTTP GET call to the URL http://localhost:xxxx/Fibonacci?length=7 should return the first seven elements as a JSON array:

[1,1,2,3,5,8,13]

Preparation

1. Go to the folder that contains the practical work for this course and create a subfolder with the name “Module2”.
2. Start Visual Studio Code.

General tips

Creating a Web API project

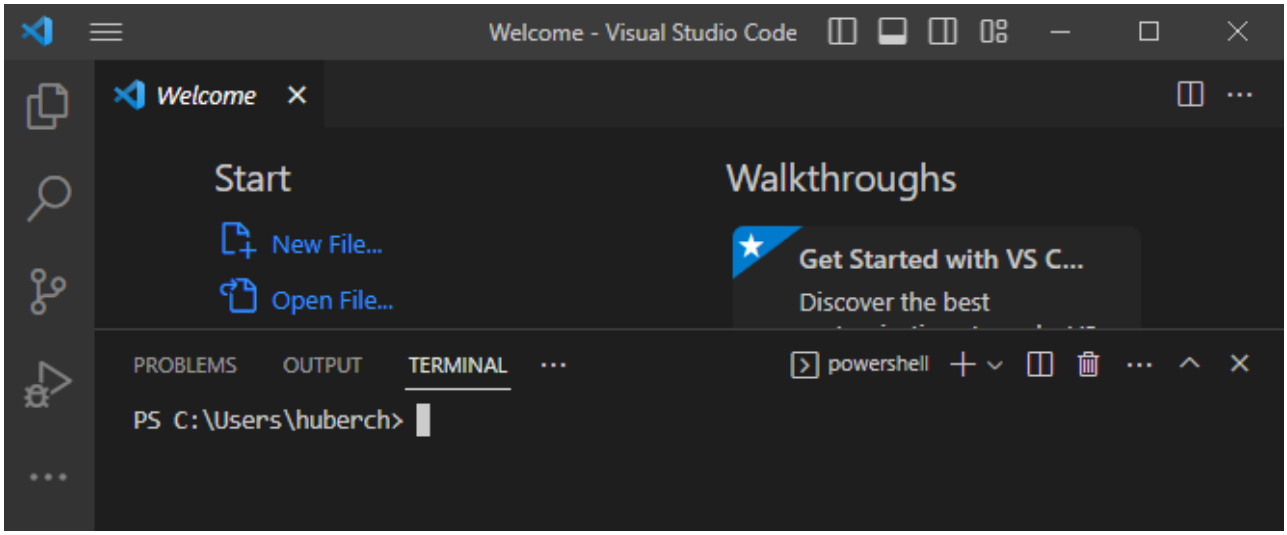
Microsoft’s official tutorial for creating a new Web API project can be found here:

https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-web-api?view=aspnetcore-9.0&tabs=visual-studio-code

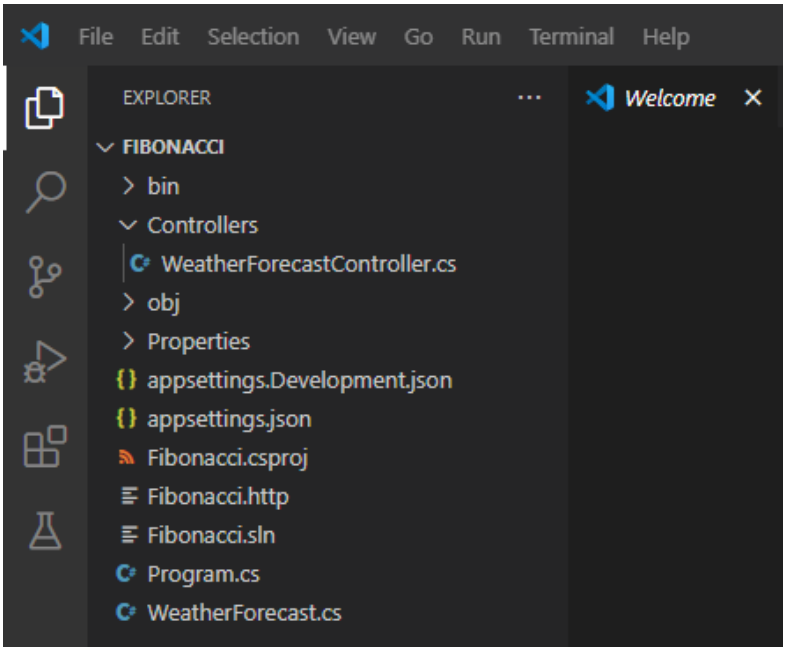
Implementation

Create a new Web API project (1 point)

1. Open the integrated terminal by selecting Terminal -> New Terminal in the Visual Studio menu:



2. Change directories (cd) to the folder “Module2” that we created in the preparation steps:
   `cd <FolderPathModule2>`
3. Create the new project and open it.
   a. Run the command to create the project:
      `dotnet new webapi --use-controllers -o Fibonacci`
   b. Open the project:
      `code -r Fibonacci`
4. The project folder should now be open in Visual Studio Code.



Prepare project for the implementation (1 point)

1. Change the name of controller from WeatherForecastController.cs to FibonacciController.cs.
2. Open the FibonacciController.cs and replace WeatherForecastController with FibonacciController in the code.
3. Delete the Summaries array in the FibonacciController.
4. Change the HttpGet method to
```
[HttpGet(Name = "GetFibonacciSequence")]
public IEnumerable<long> Index(int length)
```
5. Delete the WeatherForecast.cs class.

Implement the creation of the Fibonacci sequence (1 point)

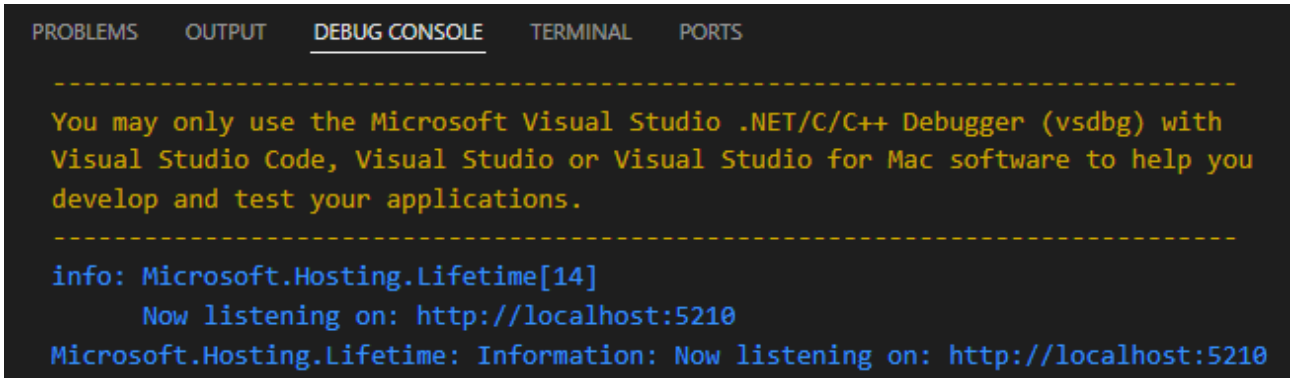
1. Implement the logic for the Fibonacci sequence in the Index method of the Fibonacci controller using C# as the programming language.
2. Take the length parameter of the Index method as the information on how many elements of the sequence should be created.
3. Create the sequence and return it as the IEnumerable<long> of the Index method.

Test the implementation using the Swagger interface (1 point)

1. Go to the terminal window and run: `dotnet dev-certs https --trust`
2. Install package for Swagger tooling.
    a. Go to the terminal window and run: `dotnet add package NSwag.AspNetCore`
3. Configure Swagger interface
    a. Add the highlighted Code to the Program.cs:

```
if (app.Environment.IsDevelopment())
{
      app.MapOpenApi();
      app.UseSwaggerUi(options =>
      {
            options.DocumentPath = "/openapi/v1.json";
      });
}
```

4. Start the debugging of the project:
    a. Go to Run -> Start Debugging
    b. Select C# as debugger
    c. Select Launch Configuration: C#: Launch Setup Project Fibonacci
    d. Select C# Startup Project: Fibonacci
    e. Debugging will be started
    f. You can find the URL where the project can be reached in the debug console



5. Open the swagger URL in a browser: http://localhost:xxxx/swagger
6. Test the service using the “Try it out” function of the swagger interface.