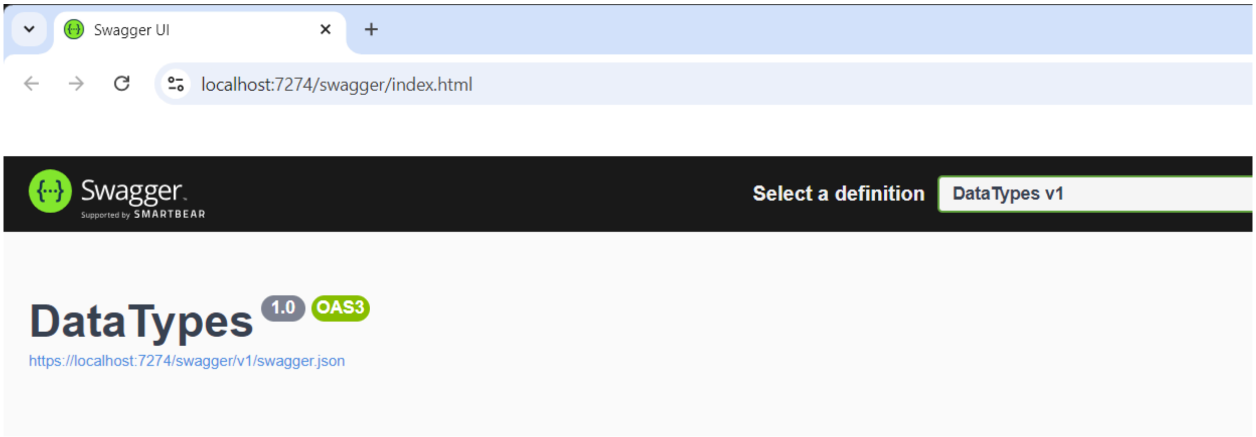
C# Assignment 0

By: Christine Bittle

This assignment aims to set up your computer for work! If you run into problems, let your instructor know as soon as possible.

Task 1: Create a .NET Core WebAPI project. Include a screenshot of the server running, and a response from your localhost environment. Step by Step instructions are provided.
Example:



Tools:
NOTE: We will use Visual Studio Community Edition for Windows in class and with examples, as it is the easiest to work with.

(Recommended) Visual Studio Community Edition Windows (Step By Step Instructions at bottom)

- Workloads ASP.NET and Web Development (Tools > Get Tools and Features)
- Make sure you have .net project and item templates.
- Create a new project
- ASP.NET Core Web API
- Enter {ProjectName}
- .NET 8.0
- [x] Configure For HTTPS
- [x] Enable OpenAPI Support
- [x] Use Controllers
- Run project with F5 or clicking the “Green Play Button”

(Option for Mac + Visual Studio Code) Visual Studio Code + .NET SDK + C# Dev Kit

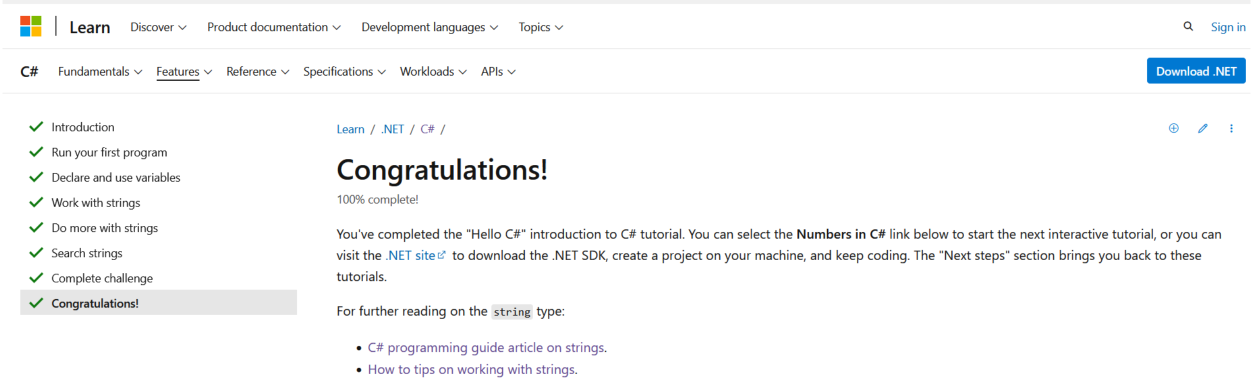
- Explorer / New Folder
- Select folder, “Trust Authors of Folder”
- View > Command Pallet
- .NET New Project
- ASP.NET Core WebAPI
- Show Template Options
- Use Controllers > True
- Enter {ProjectName}
- Create Project
- Click Program.cs
- Run and Debug

(Not Recommended) Command Prompt / Terminal with .NET SDK

- Open Powershell / Terminal
  ```
  > dotnet --version
  > mkdir {ProjectName}
  > cd {ProjectName}
  > dotnet new webapi --use-controllers --framework net8.0
  > dotnet run
  ```
- Highlight the “localhost:xx” address, right click to copy
- Put the address into your browser
- Change to localhost:xx/Swagger/Index.html

Task 2:

Complete the Hello World! Tutorial. Include a screenshot of the congratulations message.
Tools: Browser



Congratulations!

100% complete!

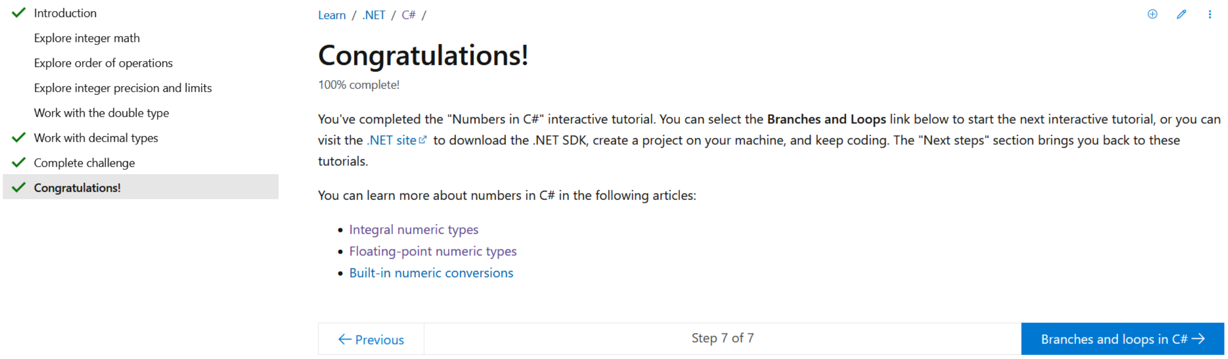
You've completed the "Hello C#" introduction to C# tutorial. You can select the **Numbers in C#** link below to start the next interactive tutorial, or you can visit the .NET site to download the .NET SDK, create a project on your machine, and keep coding. The "Next steps" section brings you back to these tutorials.

For further reading on the `string` type:

- C# programming guide article on strings.
- How to tips on working with strings.

Task 3:

Complete Numbers in C# Tutorial. Include a screenshot of the congratulations message.
Tools: Browser


✓ Introduction
Explore integer math
Explore order of operations
Explore integer precision and limits
Work with the double type
✓ Work with decimal types
✓ Complete challenge
✓ Congratulations!
Learn / .NET / C# /

Congratulations!

100% complete!

You've completed the "Numbers in C#" interactive tutorial. You can select the **Branches and Loops** link below to start the next interactive tutorial, or you can visit the .NET site to download the .NET SDK, create a project on your machine, and keep coding. The "Next steps" section brings you back to these tutorials.

You can learn more about numbers in C# in the following articles:

- Integral numeric types
- Floating-point numeric types
- Built-in numeric conversions

Task 4:

Use git to clone a project from the course site. You should be able to run it the same way you ran your project in Task 1. Include a screenshot of the repository.

Tools:

(Recommended) Visual Studio Community Edition Windows
Navigate to a repository posted in the Learning Module Content
Clone a Repository
Grab the repository URL from Github (example: https://github.com/christinebittle/DataTypes.git)
Clone Project

Git

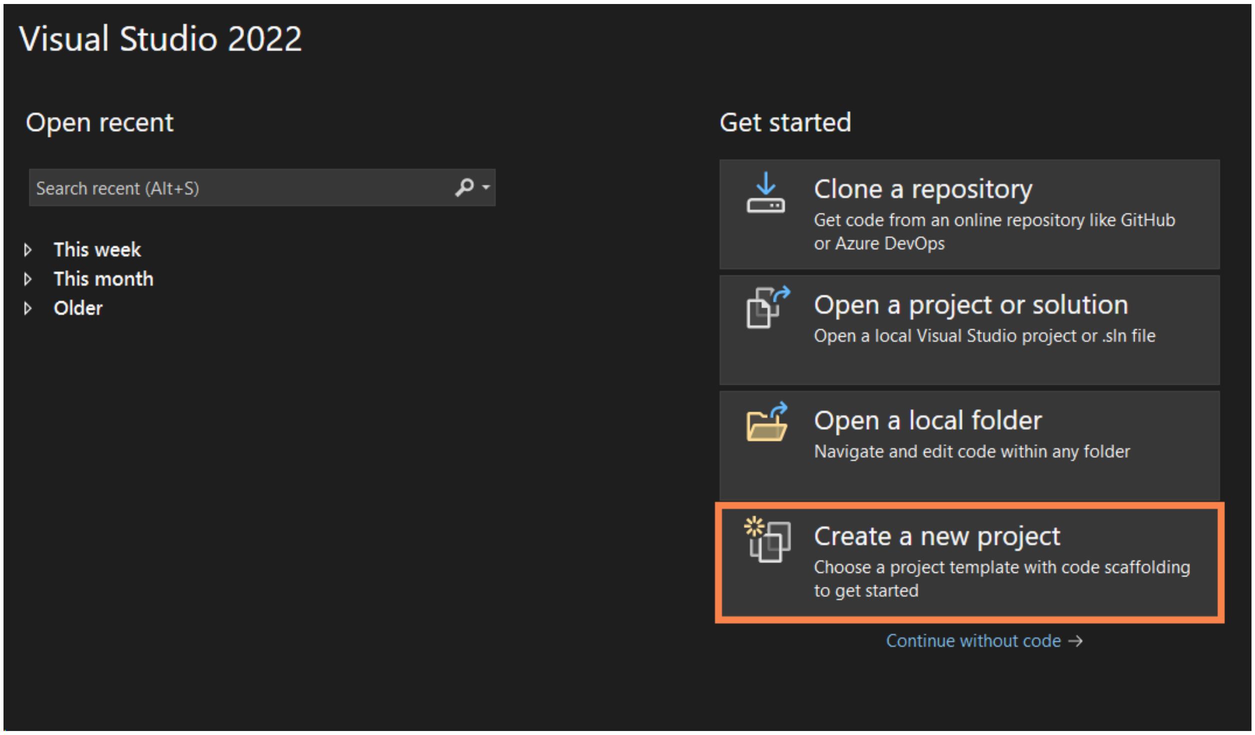
```
> git clone repository_url.git
```

Hooray! You’re all set up.

Task 1: Complete / Incomplete
Task 2: Complete / Incomplete
Task 3: Complete / Incomplete
Task 4: Complete / Incomplete

Step By Step for Visual Studio

1.



Visual Studio 2022

Open recent

Get started

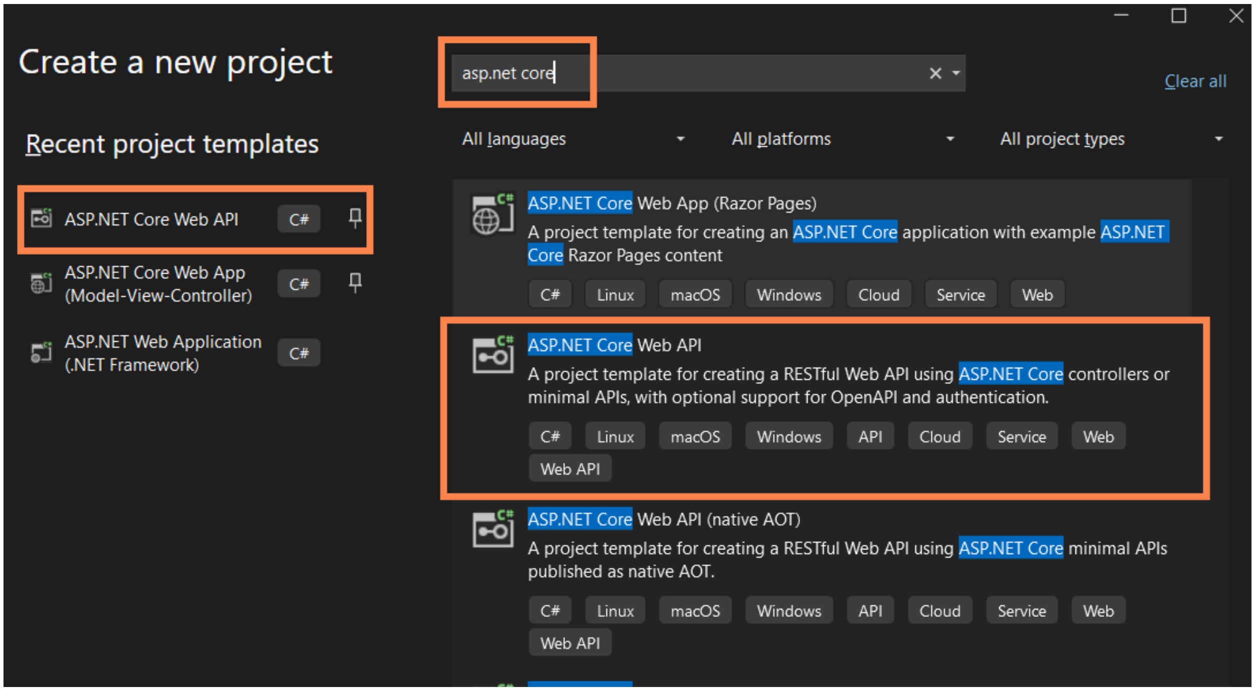
Clone a repository
Get code from an online repository like GitHub or Azure DevOps

Open a project or solution
Open a local Visual Studio project or .sln file

Open a local folder
Navigate and edit code within any folder

Create a new project
Choose a project template with code scaffolding to get started

2. (If you don’t see ASP.NET Core WebAPI, refer to steps 7-9)



Create a new project

asp.net core

Clear all
Recent project templates
ASP.NET Core Web API C#
ASP.NET Core Web App (Model-View-Controller) C#
ASP.NET Web Application (.NET Framework) C#
All languages
All platforms
All project types

ASP.NET Core Web App (Razor Pages)
A project template for creating an ASP.NET Core application with example ASP.NET Core Razor Pages content

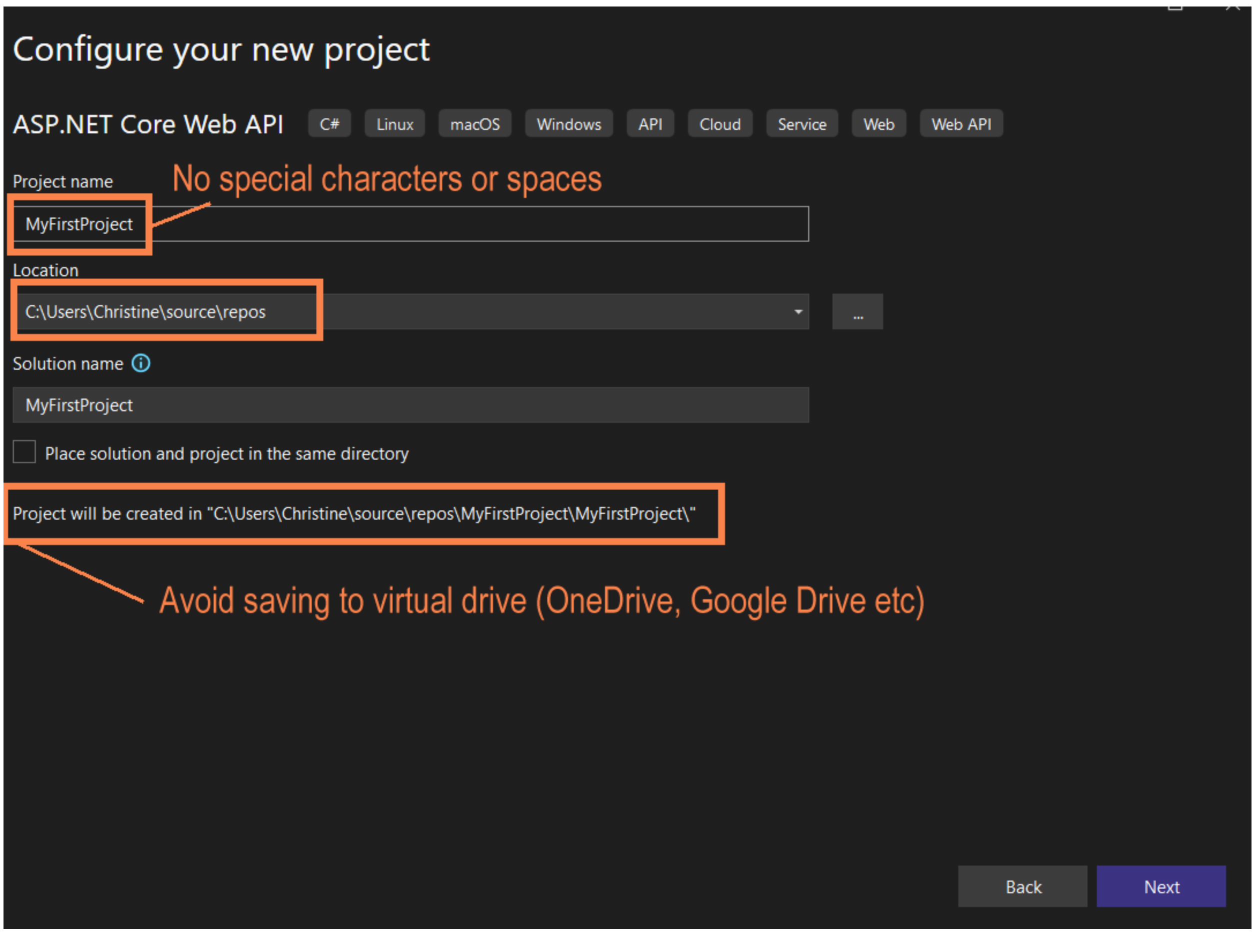
C# Linux macOS Windows Cloud Service Web

ASP.NET Core Web API
A project template for creating a RESTful Web API using ASP.NET Core controllers or minimal APIs, with optional support for OpenAPI and authentication.

C# Linux macOS Windows API Cloud Service Web Web API

ASP.NET Core Web API (native AOT)
A project template for creating a RESTful Web API using ASP.NET Core minimal APIs published as native AOT.

3.



Configure your new project

ASP.NET Core Web API

Project name

MyFirstProject

No special characters or spaces

Location

C:\Users\Christine\source\repos

Solution name

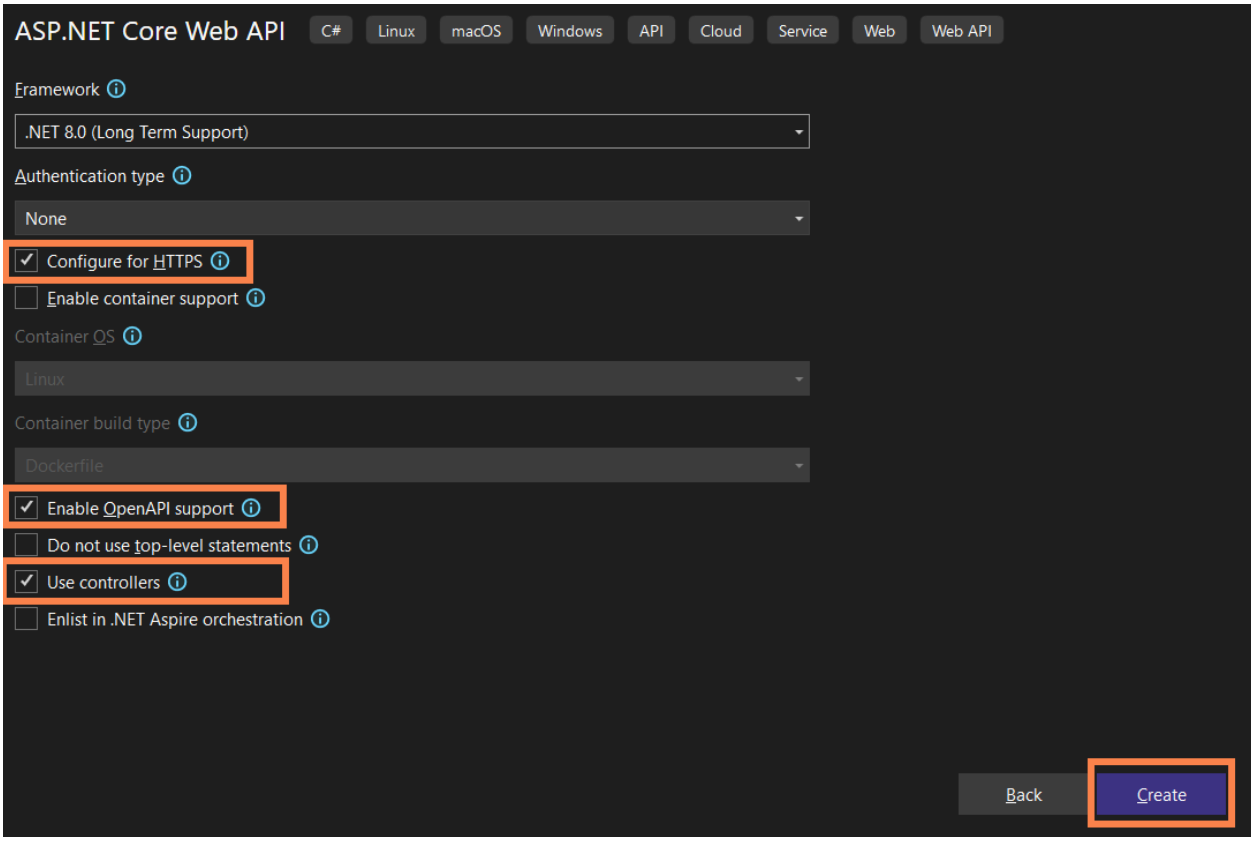
MyFirstProject

□ Place solution and project in the same directory

Project will be created in "C:\Users\Christine\source\repos\MyFirstProject\MyFirstProject\"

Avoid saving to virtual drive (OneDrive, Google Drive etc)

4. (You may be warned about a ‘self-signed certificate’, click Yes / Accept)



ASP.NET Core Web API

Framework: .NET 8.0 (Long Term Support)

Authentication type: None

☑ Configure for HTTPS

☐ Enable container support

Container OS: Linux

Container build type: Dockerfile

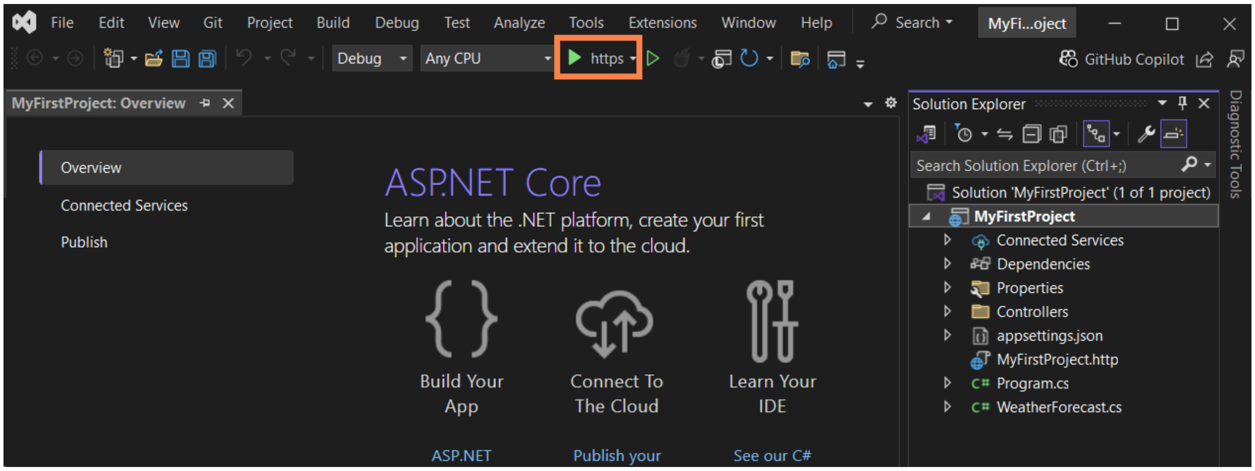
☑ Enable OpenAPI support

☐ Do not use top-level statements

☑ Use controllers

☐ Enlist in .NET Aspire orchestration

5. (You may be warned about a ‘self-signed certificate’, click Yes / Accept)



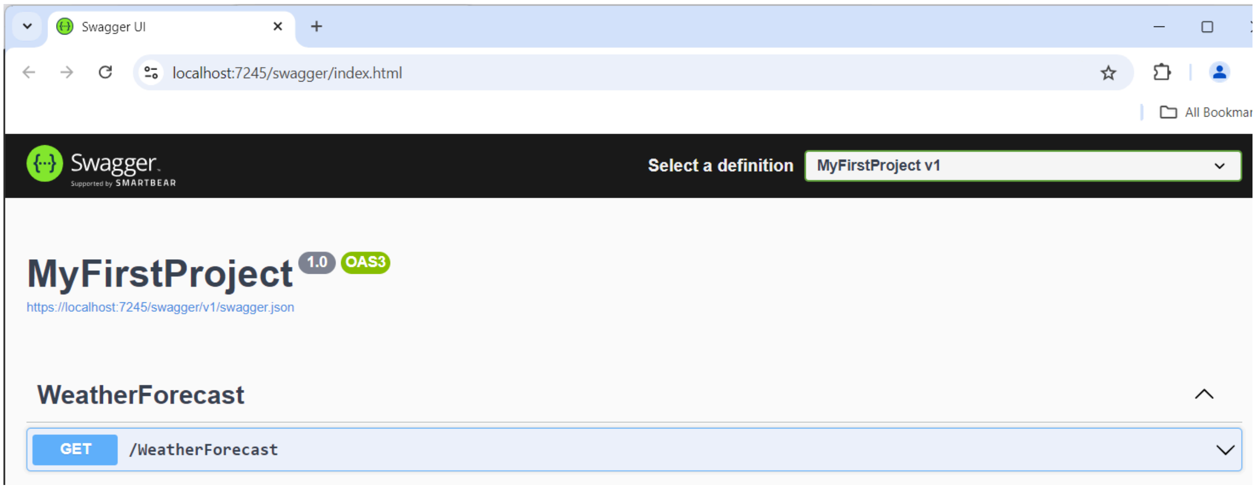
ASP.NET Core

Learn about the .NET platform, create your first application and extend it to the cloud.

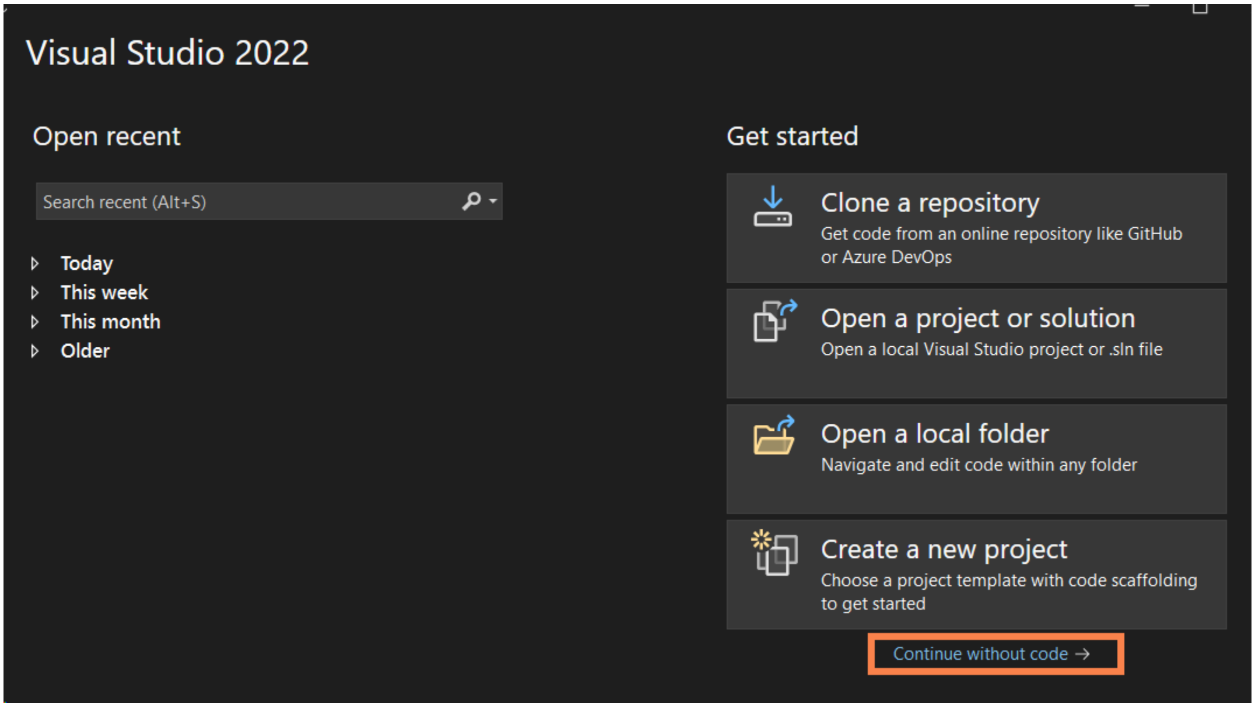
Solution 'MyFirstProject' (1 of 1 project)

- MyFirstProject
  - Connected Services
  - Dependencies
  - Properties
  - Controllers
  - appsettings.json
  - MyFirstProject.http
  - Program.cs
  - WeatherForecast.cs

6. (You may be warned about a ‘self-signed certificate’, click Yes / Accept)



7. (If you do not see ASP.NET Core WebAPI as a project template)



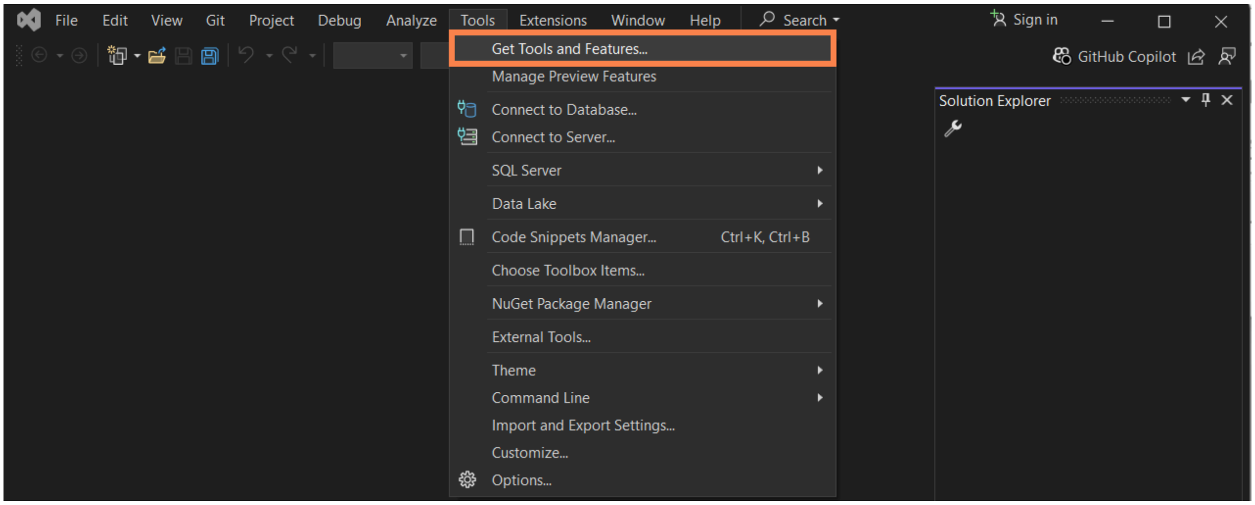
Get code from an online repository like GitHub or Azure DevOps

Open a local Visual Studio project or .sln file

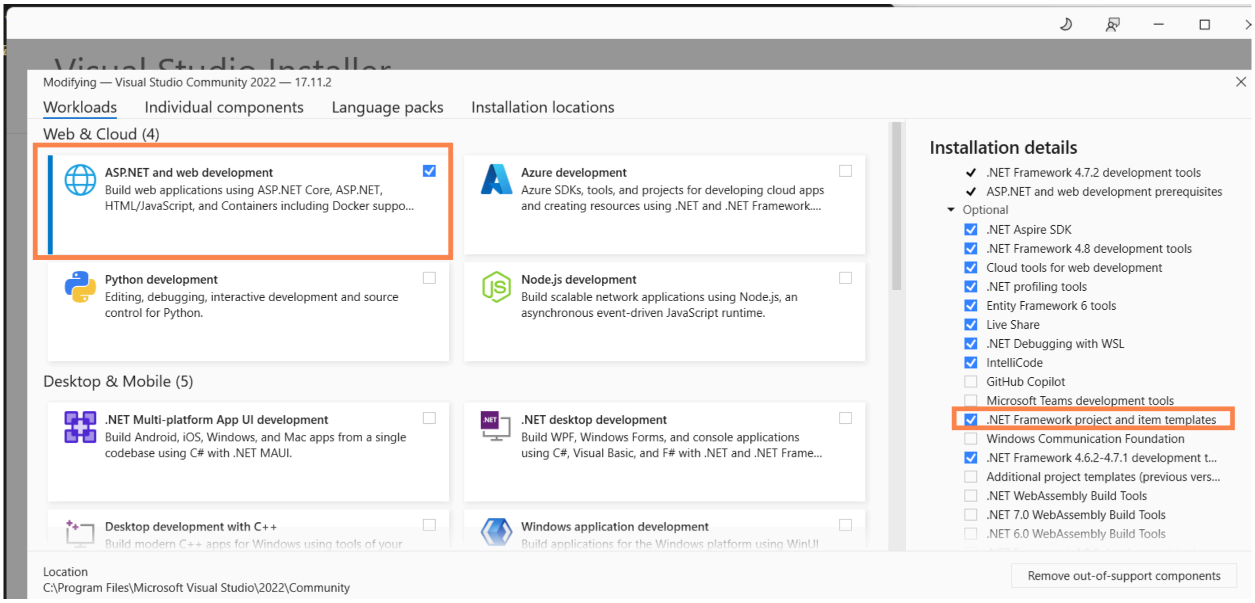
Navigate and edit code within any folder

Choose a project template with code scaffolding to get started

8.



9.



Modifying — Visual Studio Community 2022 — 17.11.2

Web & Cloud (4)

☑ **ASP.NET and web development**
Build web applications using ASP.NET Core, ASP.NET, HTML/JavaScript, and Containers including Docker suppo...

□ **Azure development**
Azure SDKs, tools, and projects for developing cloud apps and creating resources using .NET and .NET Framework....

□ **Python development**
Editing, debugging, interactive development and source control for Python.

□ **Node.js development**
Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.

Desktop & Mobile (5)

□ **.NET Multi-platform App UI development**
Build Android, iOS, Windows, and Mac apps from a single codebase using C# with .NET MAUI.

□ **.NET desktop development**
Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET and .NET Frame...

□ **Desktop development with C++**
Build modern C++ apps for Windows using tools of your

□ **Windows application development**
Build applications for the Windows platform using WinUI

Installation details

- ✓ .NET Framework 4.7.2 development tools
- ✓ ASP.NET and web development prerequisites

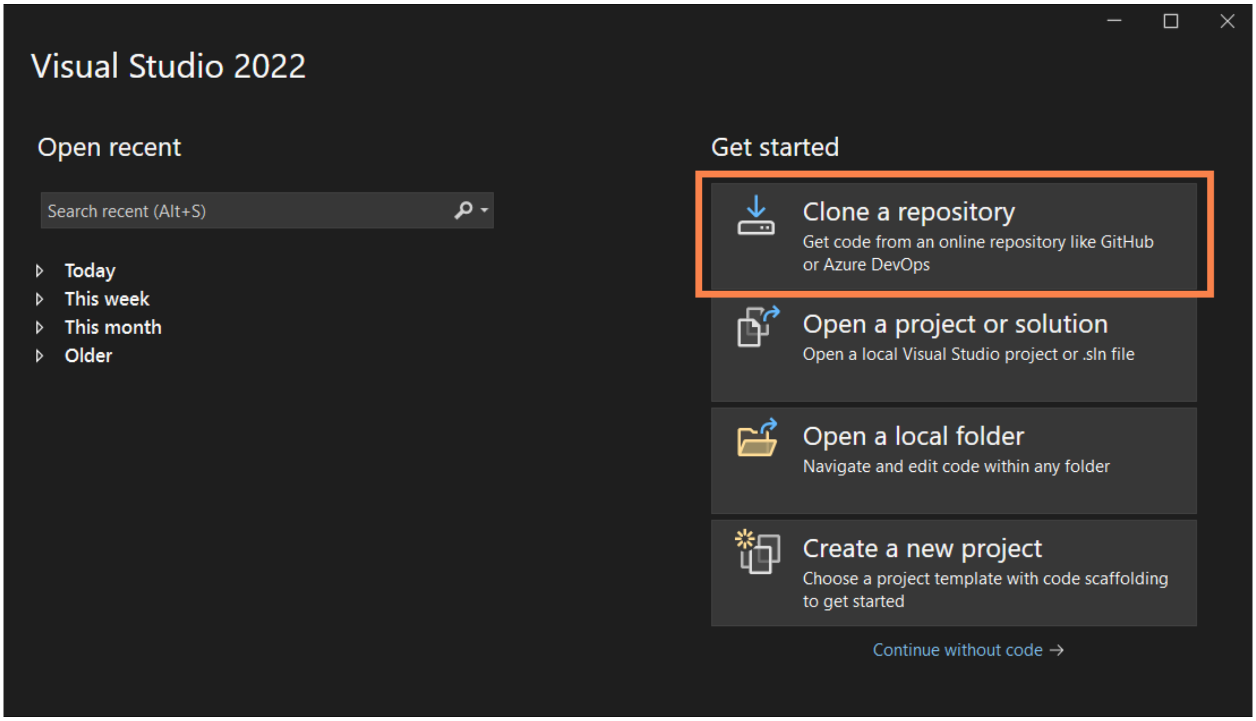
Optional

- ☑ .NET Aspire SDK
- ☑ .NET Framework 4.8 development tools
- ☑ Cloud tools for web development
- ☑ .NET profiling tools
- ☑ Entity Framework 6 tools
- ☑ Live Share
- ☑ .NET Debugging with WSL
- ☑ IntelliCode
- □ GitHub Copilot
- □ Microsoft Teams development tools
- ☑ .NET Framework project and item templates
- □ Windows Communication Foundation
- ☑ .NET Framework 4.6.2-4.7.1 development t...
- □ Additional project templates (previous vers...
- □ .NET WebAssembly Build Tools
- □ .NET 7.0 WebAssembly Build Tools
- □ .NET 6.0 WebAssembly Build Tools

Location
C:\Program Files\Microsoft Visual Studio\2022\Community

Clone a project through Visual Studio

1.



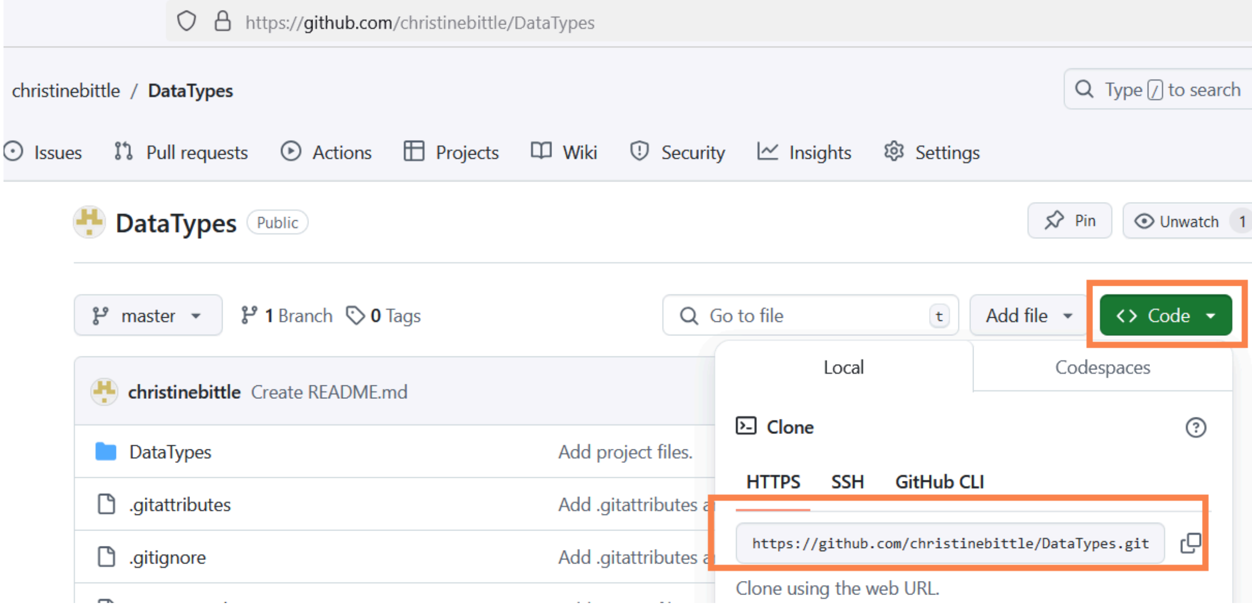
Get code from an online repository like GitHub or Azure DevOps

Open a local Visual Studio project or .sln file

Navigate and edit code within any folder

Choose a project template with code scaffolding to get started

2 (Navigate to github page)

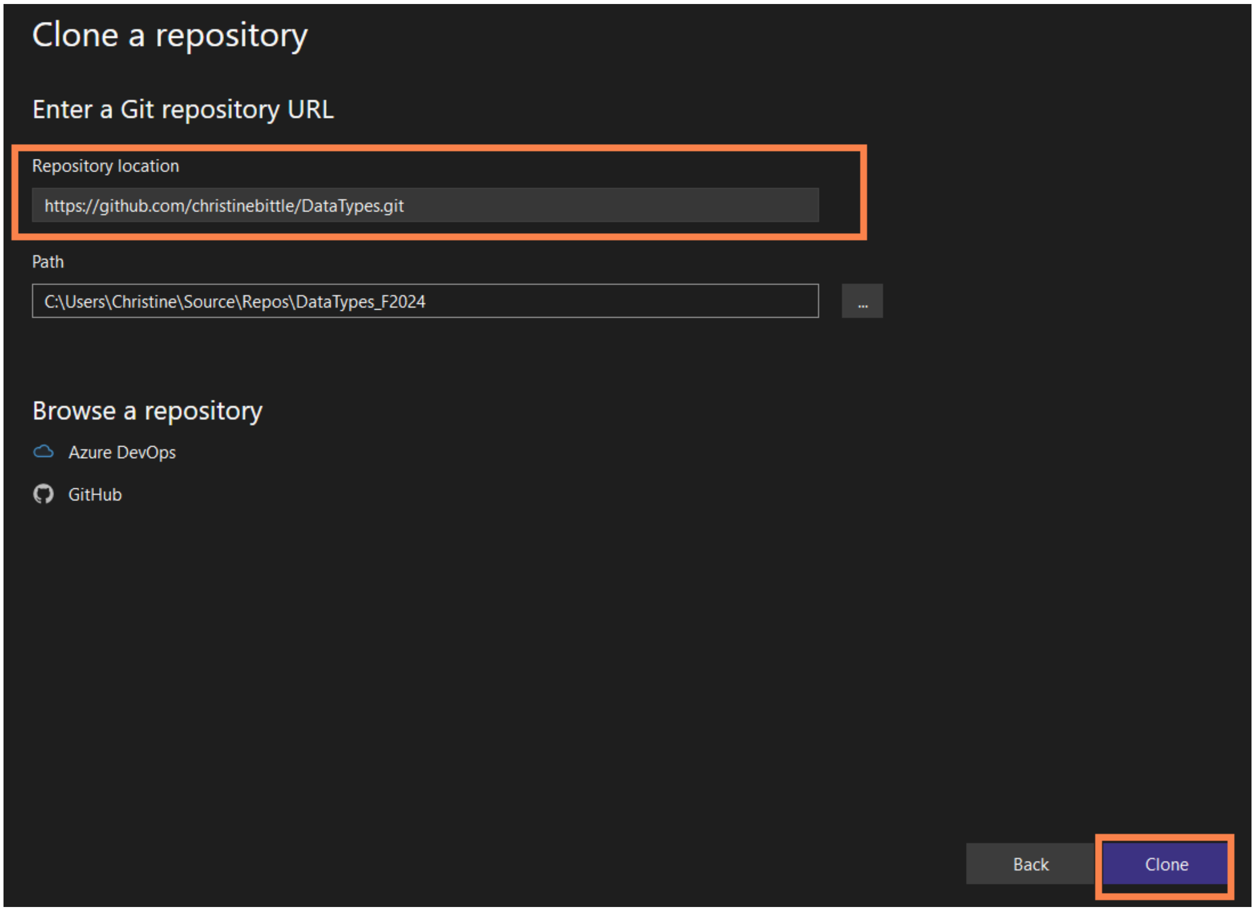


https://github.com/christinebittle/DataTypes

https://github.com/christinebittle/DataTypes.git

Clone using the web URL.

3.



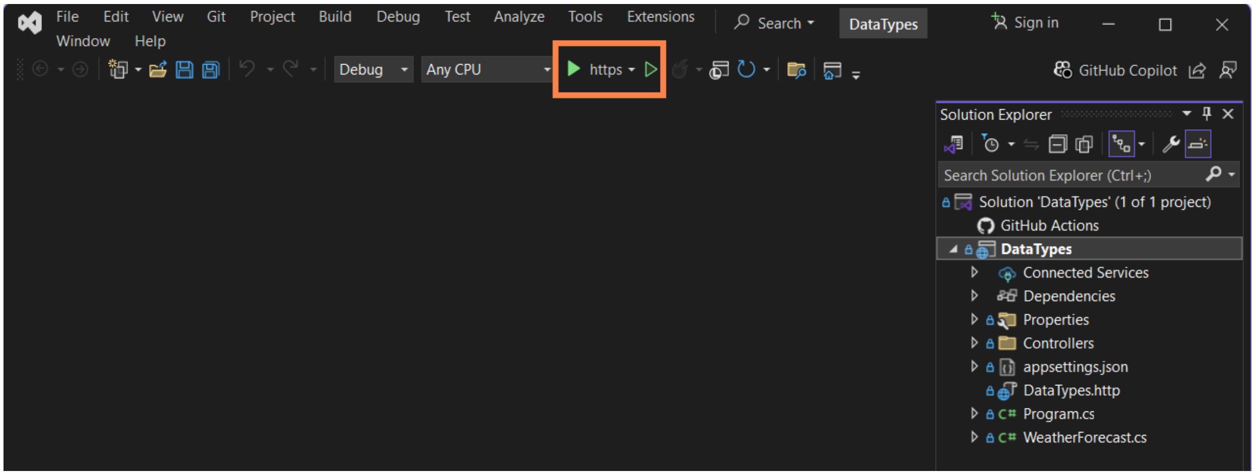
Clone a repository

Enter a Git repository URL

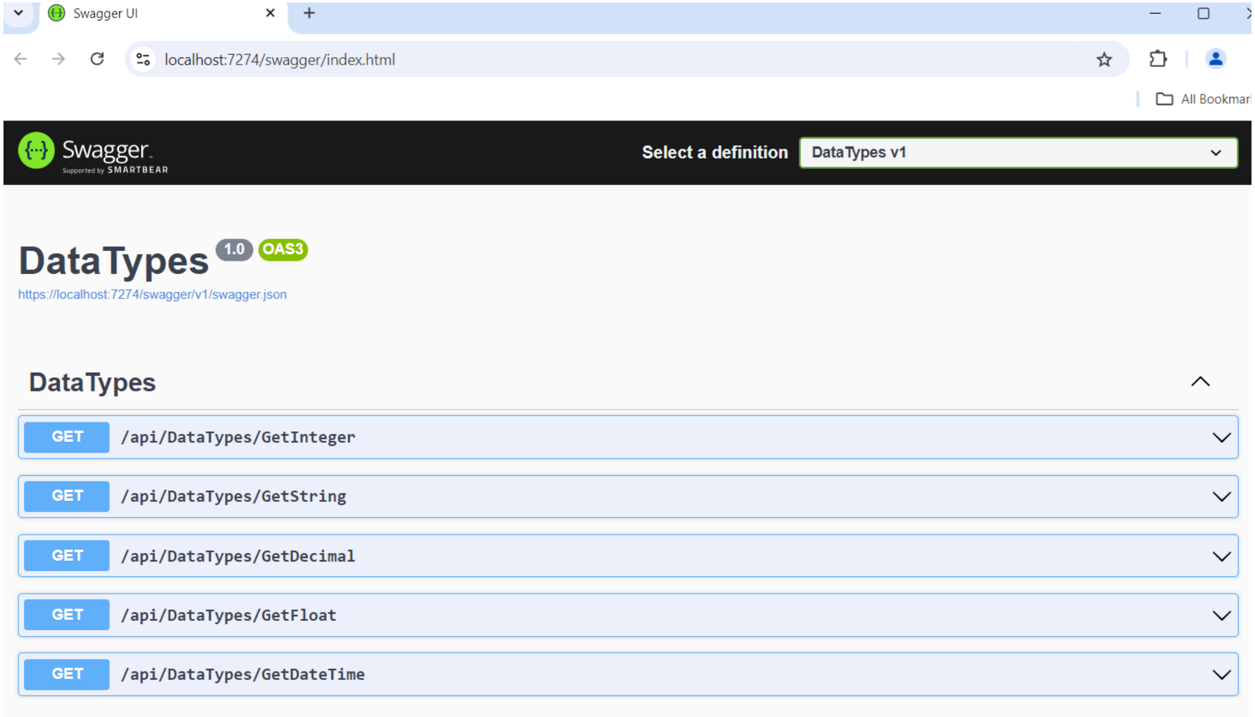
Repository location: https://github.com/christinebittle/DataTypes.git

Path: C:\Users\Christine\Source\Repos\DataTypes_F2024

4.



5.



DataTypes 1.0 OAS3

https://localhost:7274/swagger/v1/swagger.json

DataTypes

GET /api/DataTypes/GetInteger

GET /api/DataTypes/GetString

GET /api/DataTypes/GetDecimal

GET /api/DataTypes/GetFloat

GET /api/DataTypes/GetDateTime