Contents

[Creating a CI Pipeline in DevOps and including a Test Stage 2](#_Toc198930915)

[General Goal 2](#_Toc198930916)

[Steps to Create my TDD MsTest project in a Separate Visual Studio 2022 Solution 2](#_Toc198930917)

[Steps to Create an Azure DevOps CI Pipeline with a Test Stage 4](#_Toc198930918)

# Creating a CI Pipeline in DevOps and including a Test Stage

## General Goal

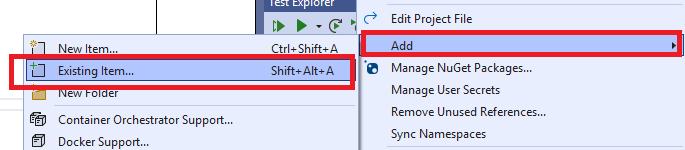
The guidance from ChatGPT was that in order to put my MsTest project in source control, I would

Have to create it in its own solution, as “Visual Studio does not support multiple Git repositories per solution via the GUI”. ChatGPT suggested I could keep the MsTest project in the same solution by using Git Submodules. However, it also mentioned that placing it in a separate solution would be the recommended action, so I proceeded with that.

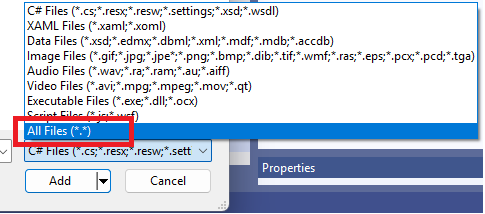
Also, once in source control I could reference it in the Test Stage of the Azure DevOps CI Pipeline I want to create.

### Steps to Create my TDD MsTest project in a Separate Visual Studio 2022 Solution

1. Create a new project in Visual Studio – and select MsTest project type
2. Name it **April3\_2025WEB\_API\_TDD**
3. This will create the project in its own solution folder
4. Then I copied the code in the Test1 file from the MsTest project that I created in the same solution as my Core Web API project, and over-wrote the basically empty Test1 file in my new MsTest project that is in its own solution. However there were several red squiggly underlines in the code indicating that there were no references to those objects, now that that code no longer existed in the same solution as the **April3\_2025WEBAPI** project.
5. To ensure that this MsTest project which exists in a separate solution than the .NET Core Web API project exists in, I needed to create a reference to the .NET Core Web API project in the MsTest project (**April3\_2025WEB\_API\_TDD**)
6. I did this by Rclicking on **April3\_2025WEB\_API\_TDD** and selecting **Add,** then **Existing Item**



1. I browsed to the **C:\Users\corry\source\repos\April3\_2025WEBAPI** folder and in the dropdown containing the extensions I clicked the down arrow and selected All files:



1. …and this allowed me to see the **April3\_2025WEBAPI.csproj** file to select for my Project Reference.
2. Once I added the reference the red squiggly lines disappeared.
3. I then ran the Test1, which passed, as it did when the MsTest project was in the same solution folder as the .NET Core Web API project, indicating that the added Reference worked.

### Steps to Create an Azure DevOps CI Pipeline with a Test Stage