This document will demonstrate how to create a repository from the ardalis/CleanArchitecture github template repository, and adjust the folder names and code to line up with the repository's name (CleanFun).
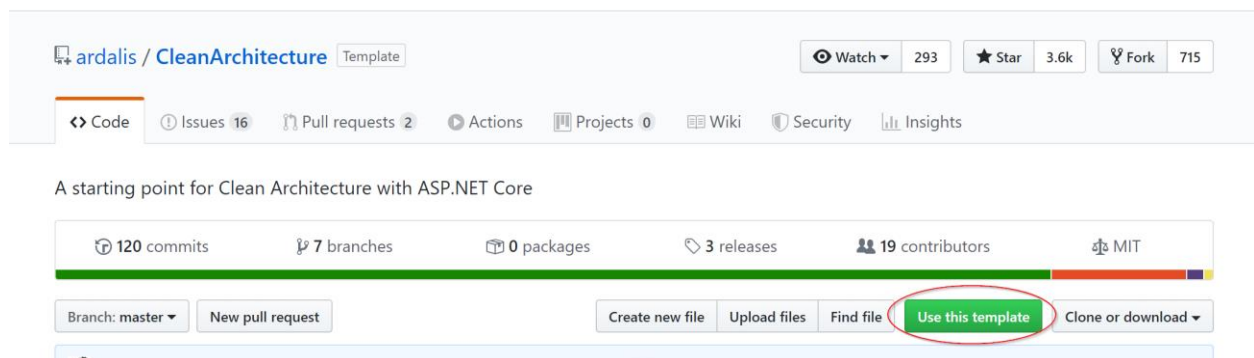
NOTE: this is a quick and dirty approach; a better way to achieve this is to create a dotnet template – I will be doing this in a future document.

We are going to create a repository named CleanFun.

Log in to github (https://github.com)

Now go the CleanArchitecture https://github.com/ardalis/CleanArchitecture/

Click the green button whose caption is "Use this template"

You will be asked for a name for your repository, put in the name CleanFun, click the green button titled "Create a repository from this template:"

Create a new repository from CleanArchitecture

The new repository will start with the same files and folders as ardalis/CleanArchitecture.

Owner

🖼️ rickhodder ▾ /

Repository name *

CleanFun ✓

Great repository names are short and memorable. Need inspiration? How about **curly-octo-waffle**?

Description (optional)

Repo for learning clean architecture and DDD based on Ardalis' template

⦿ 📖 **Public**
　　Anyone can see this repository. You choose who can commit.

○ 🔒 **Private**
　　You choose who can see and commit to this repository.

**Create repository from template**

If all goes well you will have a repository named CleanFun, and it even tells you the repository that it was generated from:
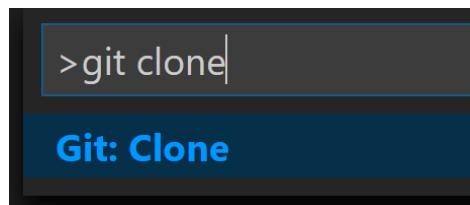
📓 rickhodder / **CleanFun**
generated from ardalis/CleanArchitecture

Next, we need to clone the repository to bring it to our PC. To do this, click on the green button titled "Clone or download", and then click the clipboard icon to copy the address of the repository to the clipboard:
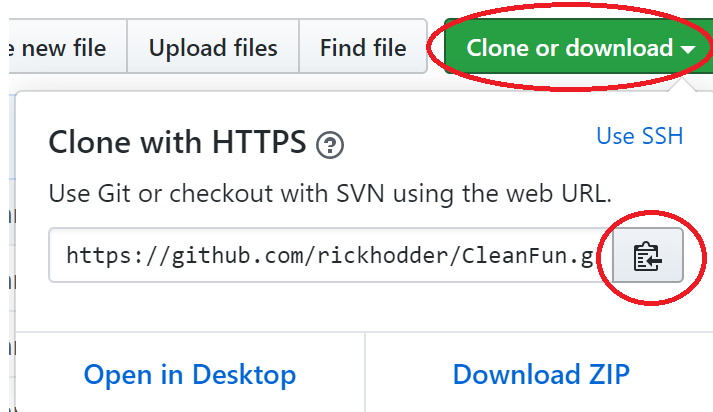
Open Visual Code, and press ctrl-shift-p, and at the prompt type "git clone" and press enter.

VSCode will ask for the Repository URL - paste from the clipboard and press Enter
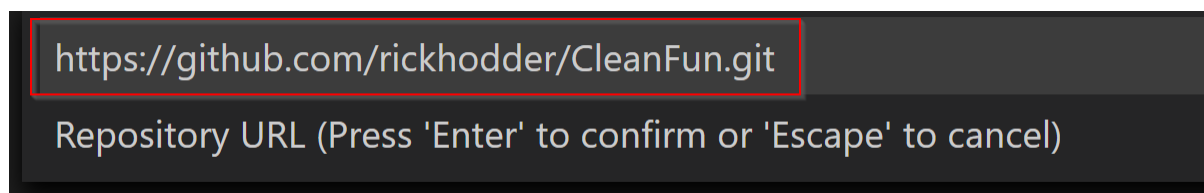

You will be asked to select your repository location. This is slightly confusing, particularly if you are new to using git.

The default location for github repositories in windows is c:\users\<yourlogin>\Source\Repos.

You might think that you should add "CleanFun" to the end of this location, but what really happens is that the name of the repository with .git removed (CleanFun.git) is the name of the folder created in the Repos folder.

So you choose the Repos folder and click on button titled "Select Repository Location"
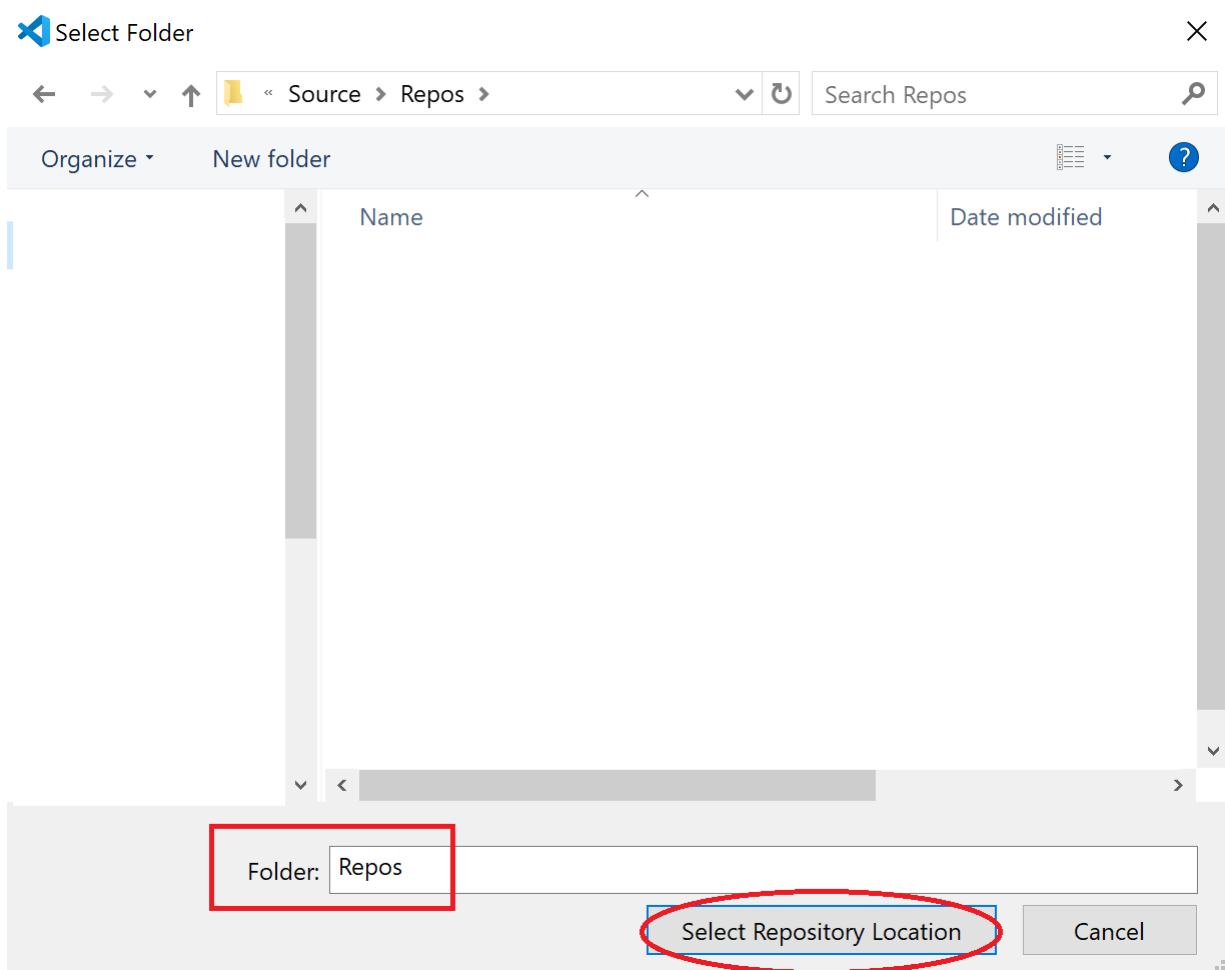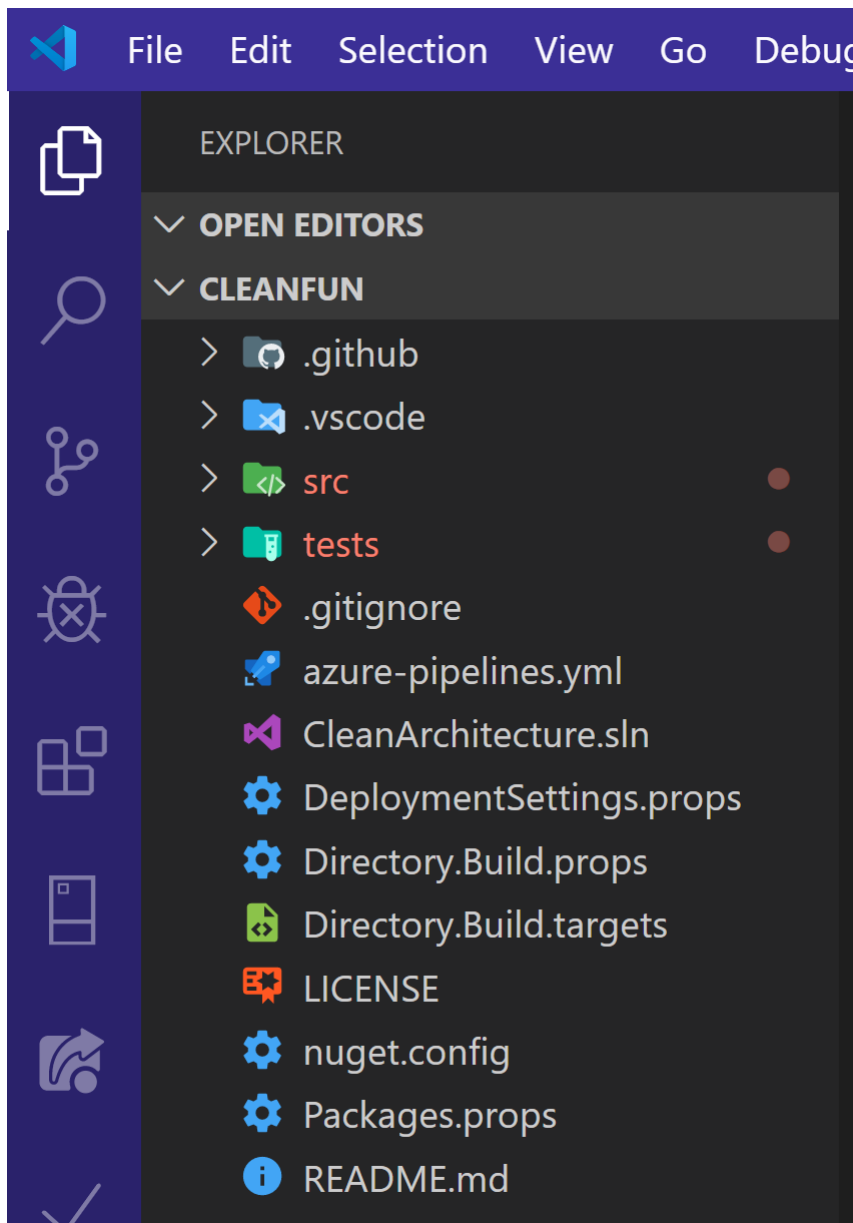
After the repository is cloned, VS Code will ask if you want to open the cloned repository, click the blue button titled "Open in New Window"
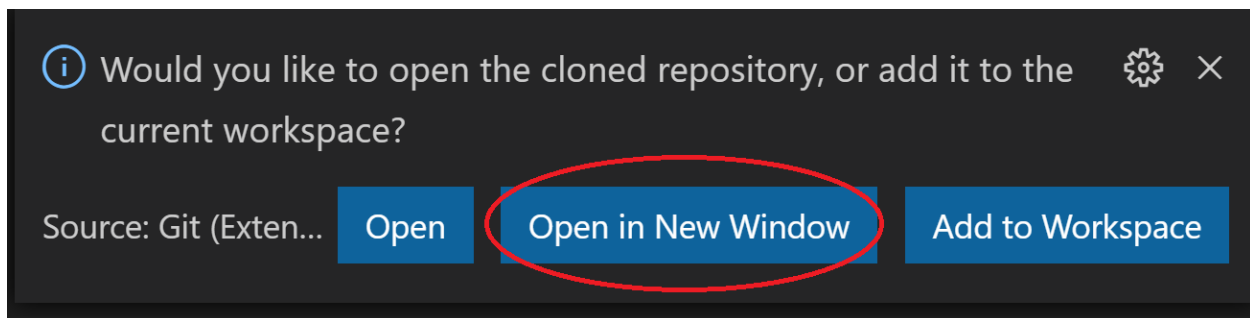
You may see other pop up windows but you can close and ignore those for now.

You should see the following in the Explorer pane:

Before moving on, we should make sure that the web project builds and works.

Open the terminal window (ctrl-~)

Use the following command to build all of the code in the repository:

```
Dotnet build
```

```
PROBLEMS       OUTPUT    DEBUG CONSOLE   TERMINAL        1: cmd          ﹀

Microsoft Windows [Version 10.0.18362.592]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\rickh\Source\Repos\CleanFun>dotnet build
```

If all goes well, you should see the following in the Terminal window

```
Build succeeded.
    0 Warning(s)
    0 Error(s)
```

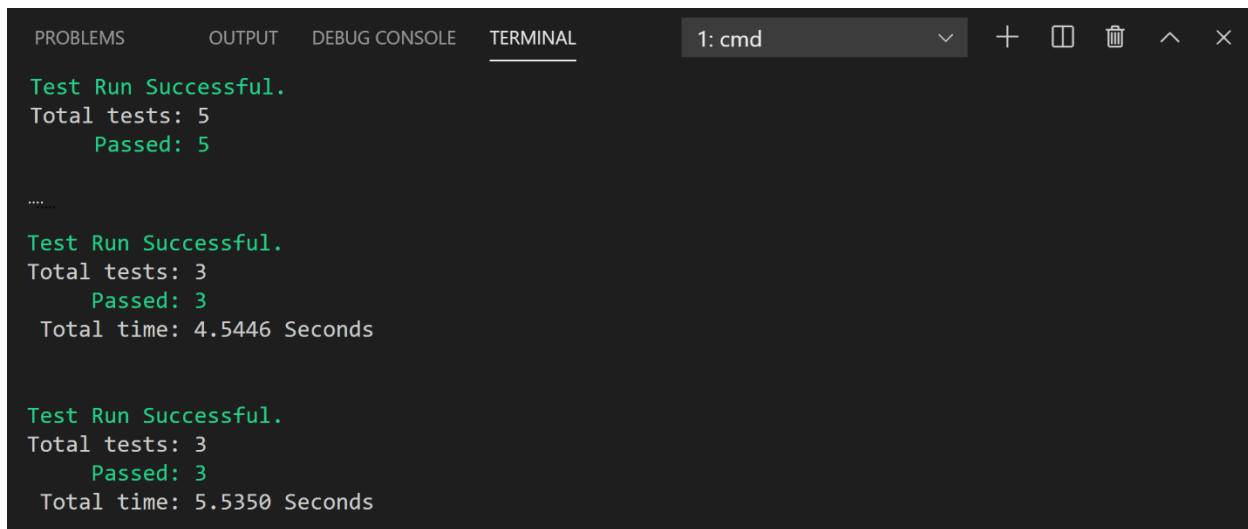To be thorough, we should run the tests.

In the terminal window, run the following command, that will find and run all tests:

```
dotnet test
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL                1: cmd

C:\Users\rickh\Source\Repos\CleanFun>dotnet test
```

You should see the following:

You will need scroll to make sure you can see it all.

So all of our tests passed.

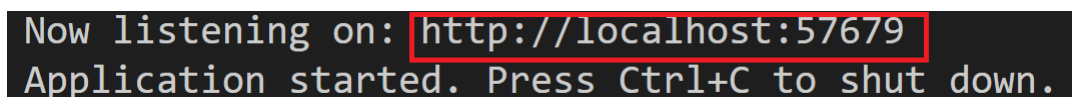Now run the web project, by typing the following into the terminal window

```
dotnet run --project
src\CleanArchitecture.Web\CleanArchitecture.Web.csproj
```

And then press enter.

After a bunch of information scrolls by, you will see a message similar to the following:

Point a browser at the http address (it may have a different port number (57679) than in the image above).

You should see the following:

To Do Items (MVC)

- Load Sample To Do Items (API call returning just a number)
- List To Do Items
- API Documentation (Swagger)

About the Author

- Blog - Ardalis.com
- Twitter - @ardalis
- GitHub

To Do Items (Razor Pages)

- Load Sample To Do Items
- List To Do Items

Now we know that it builds, runs and tests properly, we can make our adjustments.

Press ctrl-c to stop the application from running.

Now let's do a little surgery to bring the names of the solution, folders, and project names (which all start with CleanArchitecture) to line up with the name of our repository (CleanFun) – first, we will rename the projects and solution. Run the following commands in the terminal window:

```
REM Renames project, test and solution files - replace "CleanFun" with the
REM new name for your projects and solution
REM then rebuilds and runs web app
REM NOTE: no spaces! Ex. CleanFun instead of Clean Fun

rename src\CleanArchitecture.Core\CleanArchitecture.Core.csproj CleanFun.Core.csproj
rename src\CleanArchitecture.Infrastructure\CleanArchitecture.Infrastructure.csproj CleanFun.Infrastructure.csproj
rename src\CleanArchitecture.SharedKernel\CleanArchitecture.SharedKernel.csproj CleanFun.SharedKernel.csproj
rename src\CleanArchitecture.Web\CleanArchitecture.Web.csproj CleanFun.Web.csproj

rename tests\CleanArchitecture.FunctionalTests\CleanArchitecture.FunctionalTests.csproj CleanFun.FunctionalTests.csproj
rename tests\CleanArchitecture.IntegrationTests\CleanArchitecture.IntegrationTests.csproj CleanFun.IntegrationTests.csproj
```

```
rename tests\CleanArchitecture.Tests\CleanArchitecture.UnitTests.csproj CleanFun.
UnitTests.csproj

rename CleanArchitecture.sln CleanFun.sln
rename src\CleanArchitecture.Core CleanFun.Core
rename src\CleanArchitecture.Infrastructure CleanFun.Infrastructure
rename src\CleanArchitecture.SharedKernel CleanFun.SharedKernel
rename src\CleanArchitecture.Web CleanFun.Web

rename tests\CleanArchitecture.FunctionalTests CleanFun.FunctionalTests
rename tests\CleanArchitecture.IntegrationTests CleanFun.IntegrationTests
rename tests\CleanArchitecture.Tests CleanFun.Tests
```

Here's what the explorer pane should look when you are done. Notice that the folders are now named CleanFun instead of CleanArchitecture:
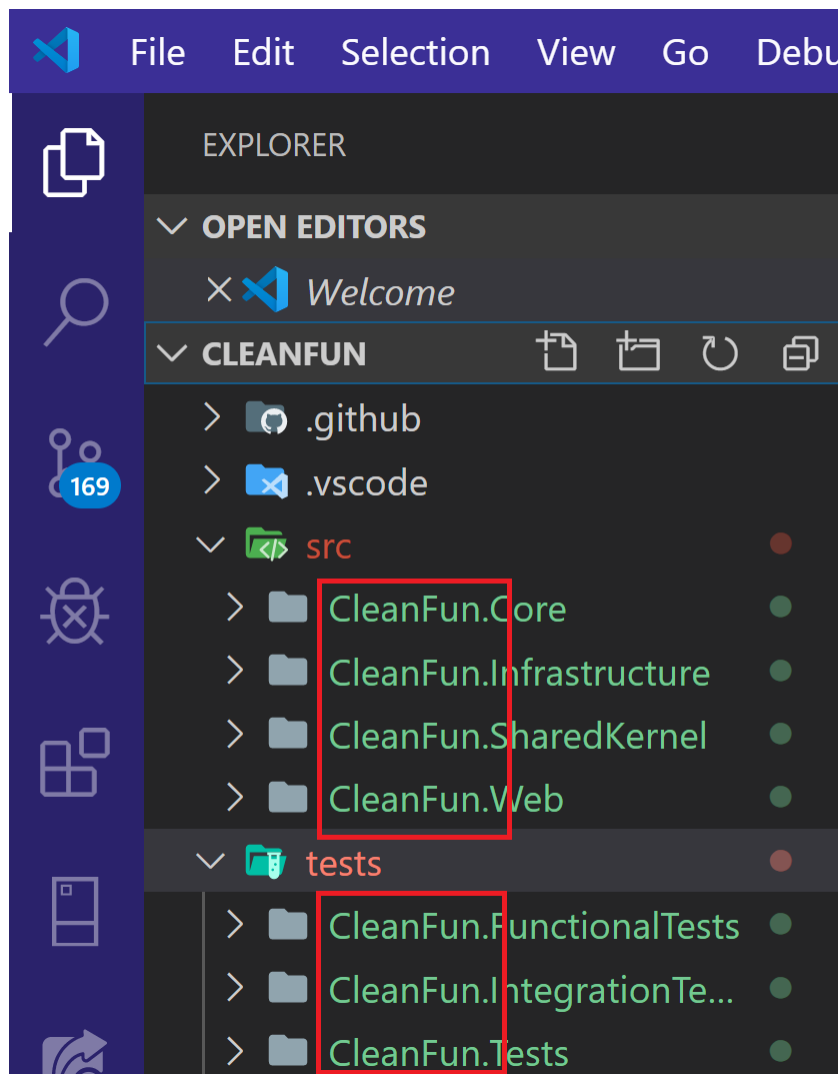
Next we want to change all the source code references from CleanArchitecture to CleanFun.

The simplest way to do this is click on the Search icon in VSCode (circled in red on the left, below), and replace CleanArchitecture with CleanFun, and press the "replace all" button (circled in red below).
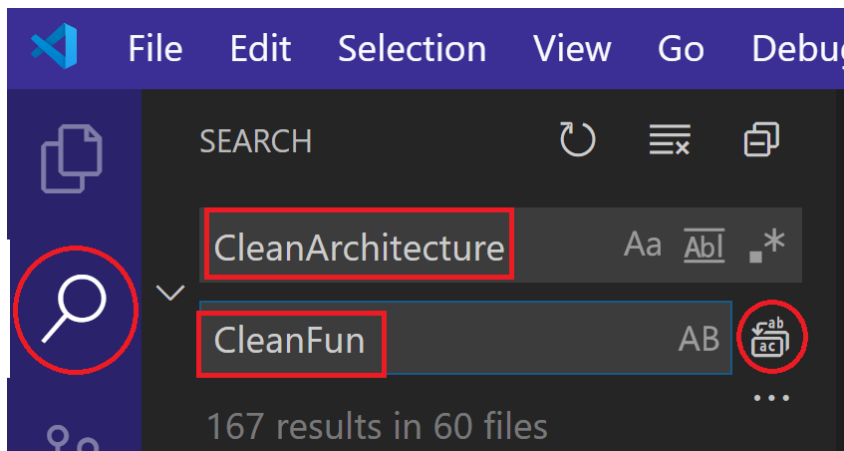
You will see a dialog that asks if you want to replace with CleanFun, choose the Replace button.
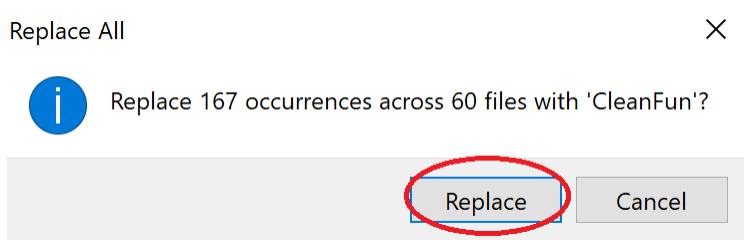
Now lets run the web application.

Go back to the terminal window, run the web application by executing the following command. Note that since we have changed the name of the web application folder and project from CleanArchitecture to CleanFun, we have to change the dotnet run command (changes highlighted in yellow)

```
dotnet run --project src\CleanFun.Web\CleanFun.Web.csproj
```
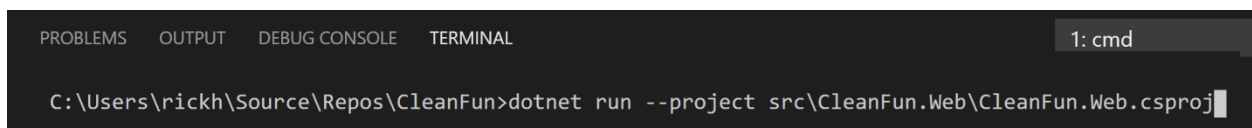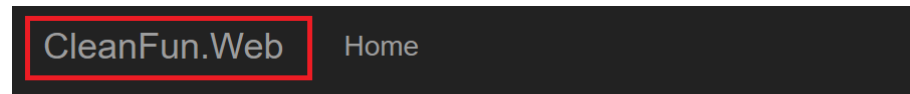

You will see a message similar to the following:

```
Now listening on: http://localhost:57679
Application started. Press Ctrl+C to shut down.
```

Point a browser at the http address (it may be a different port than in the image above).

You should see the following (notice that application now shows CleanFun.Web in the header instead of CleanArchitecture.Web:
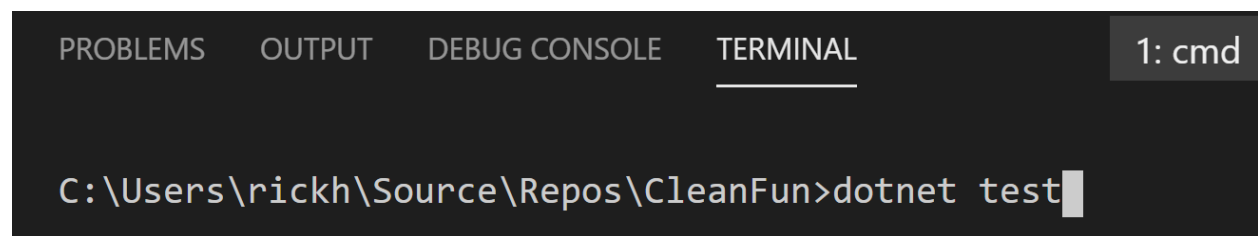
CleanFun.Web   Home

To Do Items (MVC)

- Load Sample To Do Items (API call returning just a number)
- List To Do Items
- API Documentation (Swagger)

Go back to VSCode's terminal window and press ctrl-c to stop the application.

To be thorough, we should run the tests again

In the terminal window, run the following command, that will find and run all tests:

```
dotnet test
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL                    1: cmd


C:\Users\rickh\Source\Repos\CleanFun>dotnet test
```

You should see the following:

```
Test Run Successful.
Total tests: 5
     Passed: 5

.....

Test Run Successful.
Total tests: 3
     Passed: 3
 Total time: 4.5446 Seconds


Test Run Successful.
Total tests: 3
     Passed: 3
 Total time: 5.5350 Seconds
```
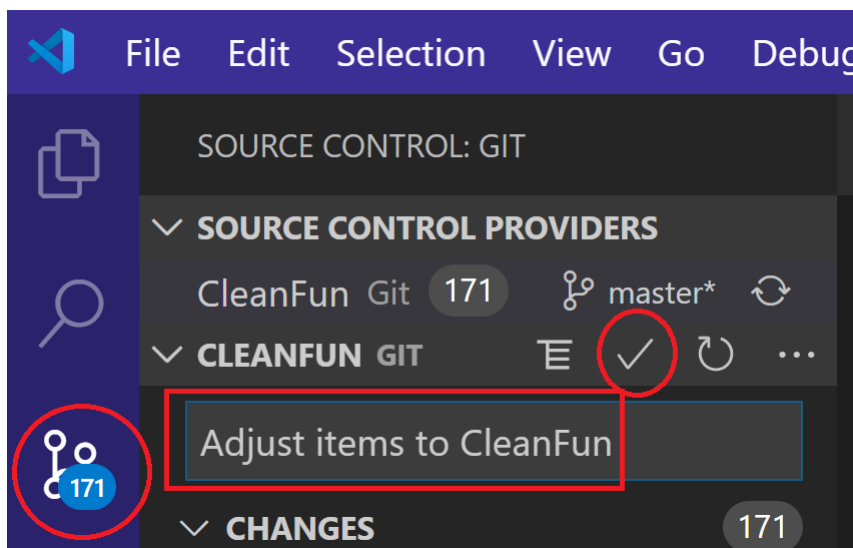
You will need scroll to make sure you can see it all.

So all of our tests passed.

Finally, we need to check in the changes to our repository.

To do this, click on the Source Control pane, enter a commit message of "Adjust items to CleanFun", and press the commit button (the check mark circled in red, below):


Because we changed the name of files and folders, you will see a dialog that asks if you want to automatically stage all your changes and commit them directly, click Yes:
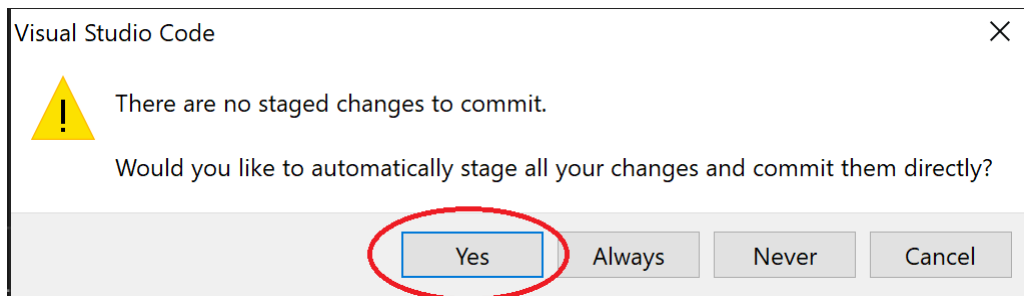
You have committed the changes to the local repository, so now you need to push the changes to github. To do this, you need to click the "synchronize changes" button, circled in red, below:
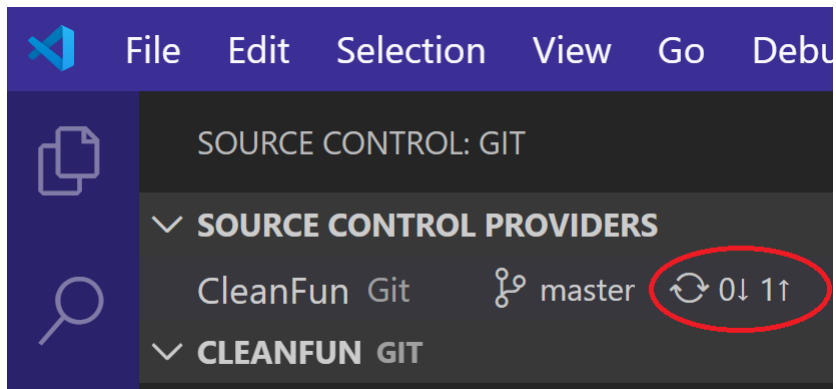
VS Code will ask if you want to synchronize changes, click Ok:
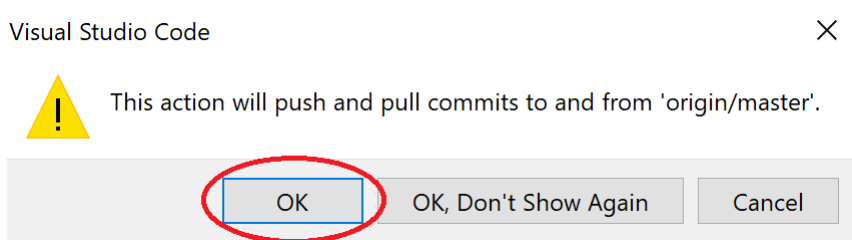
You are all set now – coding can commence!