Experiments App Overview

Assumptions

1. This is a sample application to purely demonstrate functionality and give the user a feel for my coding style. Security concerns are up to the user to address at a later time.

2. The master form is an experiment called [Template Experiment]. Any questions and field values form the structure of new experiments that are added. Users can choose at any time to override the standard questions.

3. At a future time, questions maybe manually changed in the JSON files.

4. Responses submitted are a snapshot of the questions asked at a point of time.
   This includes any possible answers the user was presented with.

5. Experiments are identified by a unique name.

6. The experiments data file contains experiments along with their set of questions to ask. There are to be no double ups in this file and no answers specified, even though the user could provide sample answers.

7. The responses data file contains experiments along with their set of questions and answers. There can be the same experiment appearing twice in the file because multiple users have submitted responses to the same experiment.

8. A 'Select from a list of options' is a dropdown list that only allows the user to select one option as an answer.

9. All data is to be stored in JSON data files.

Location of Data files

Doorsteps.Experiments\Doorsteps.Experiments.Api\Data

Experiments_data.json
A JSON data file containing a list of experiments and the questions they need to ask users.

Response_data.json
A JSON data file containing a list of experiments with their questions asked at a given time along with answers.

Doorsteps.Experiments\Doorsteps.Experiments.Api\Config

Spring.xml
Configuration file for JSON datasource the Web APIs will use.

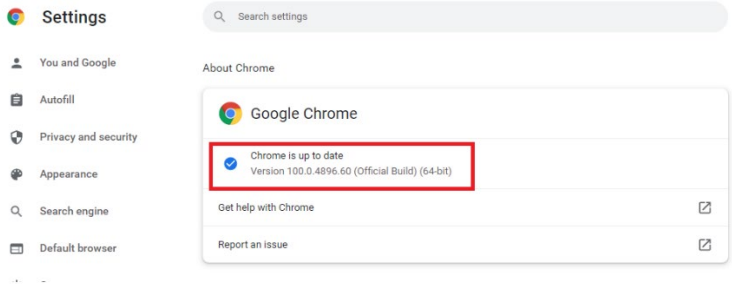
Doorsteps.Experiments\Doorsteps.Experiments.Web\Config

Spring.xml
Configuration file for RESTful web client the Web App will use.
Experiment Usage Instructions

1. On a Windows PC, download and install Visual Studio Community 2022 from https://visualstudio.microsoft.com/vs/community

2. Download and install Chrome for Windows from https://google.co.nz/chrome

3. Check out the version of Chrome that is running by opening Chrome and going to the three dots in top right hand corner of the browser > Help > About Chrome.



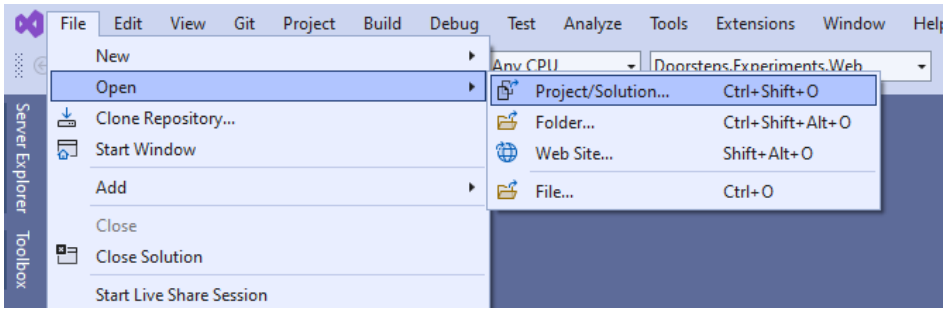
4. Navigate to https://chromedriver.chromium.org/downloads and download the version of the Chrome driver that matches your version of Chrome. i.e. chromedriver_win32.zip.

5. Check out the GitHub code for the project at
<URL>

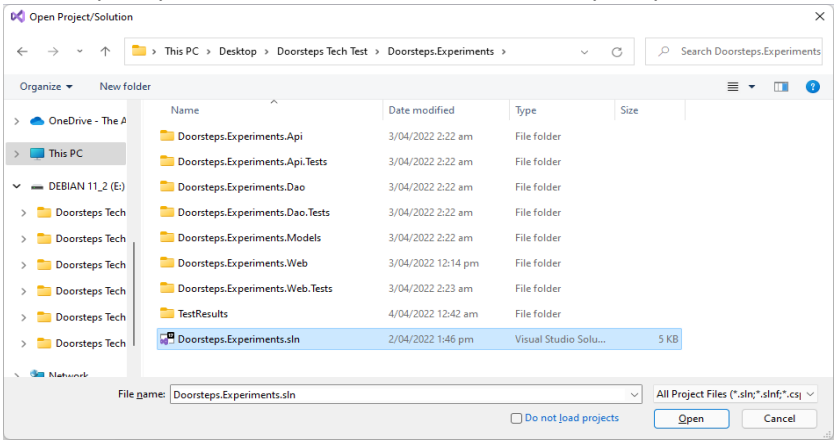
6. Unzip the code to a directory of your choosing. i.e. C:\SourceCode\Doorsteps.Experiments.

7. Unzip the file chromedriver_win32.zip to the Doorsteps.Experiments.Web.Tests\Drivers subfolder. The zip file should contain a single file called chromedriver.exe.

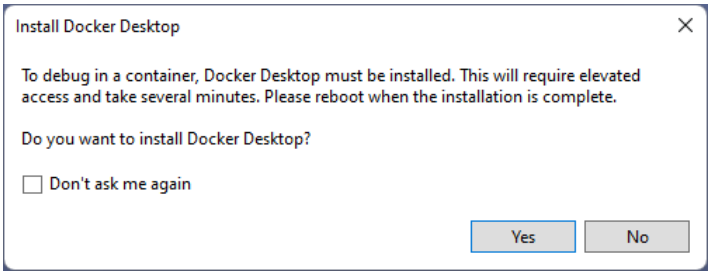
8. Open Visual Studio and go to File > Open > Project/Solution.



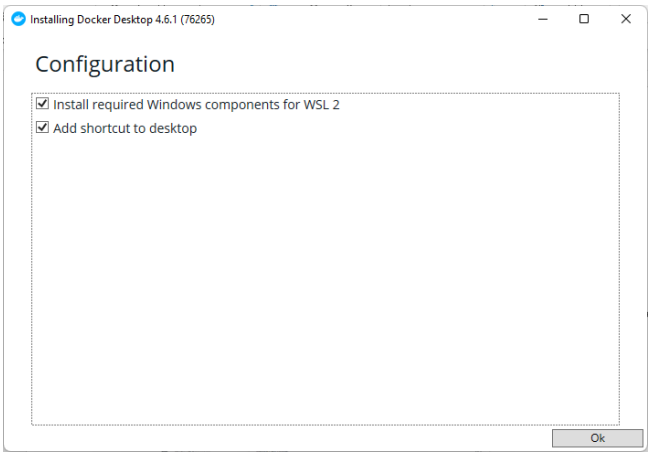
9. Navigate to the directory where the GitHub code was extracted. Click click on Doorsteps.Experiments folder. Select the Doorsteps.Experiments.sln file. Click Open.



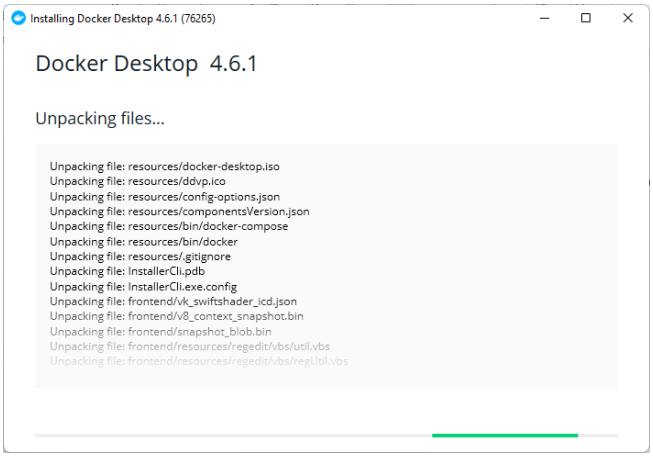
10. A box will appear regarding installing Docker Desktop. Click Yes to install.
It may take a few minutes for the installer to download and run. Please be patient.
When prompted for Administrator privileges, click Yes.



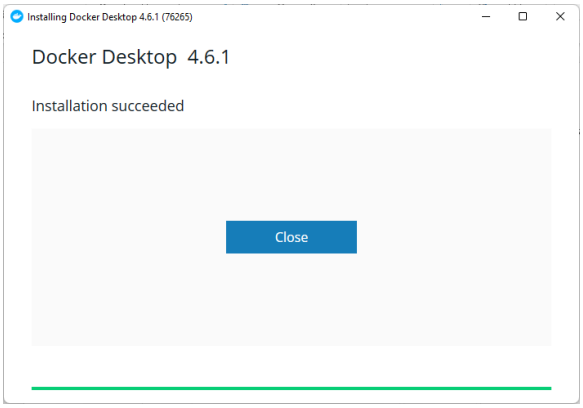
11. Make sure both components are ticked. Click Ok.



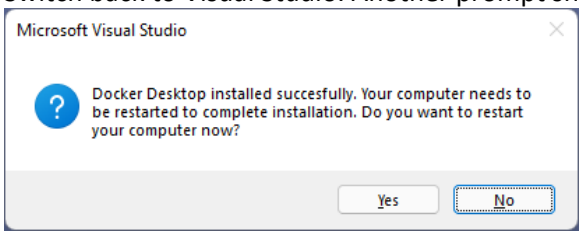
12. Wait…



13. Click Close.

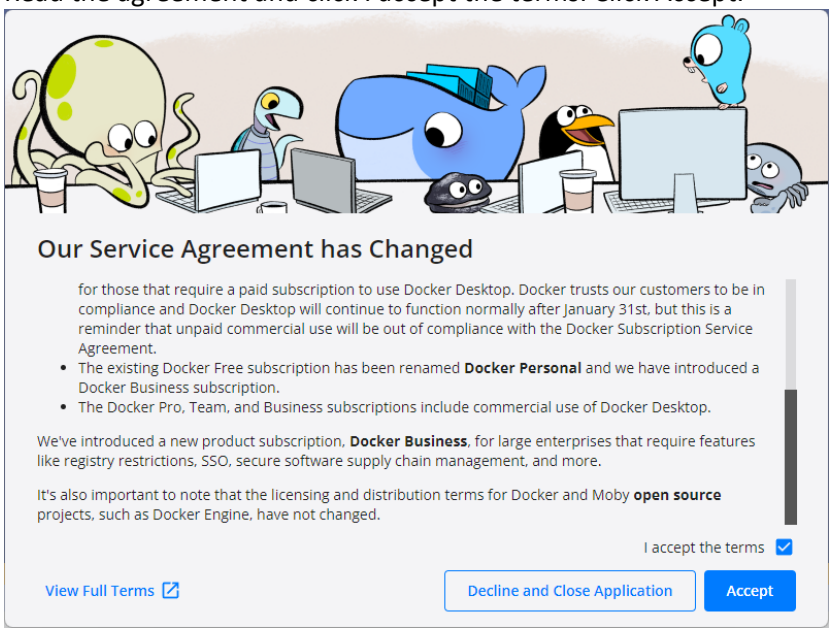


14. Switch back to Visual Studio. Another prompt shows asking to reboot. Click Yes.

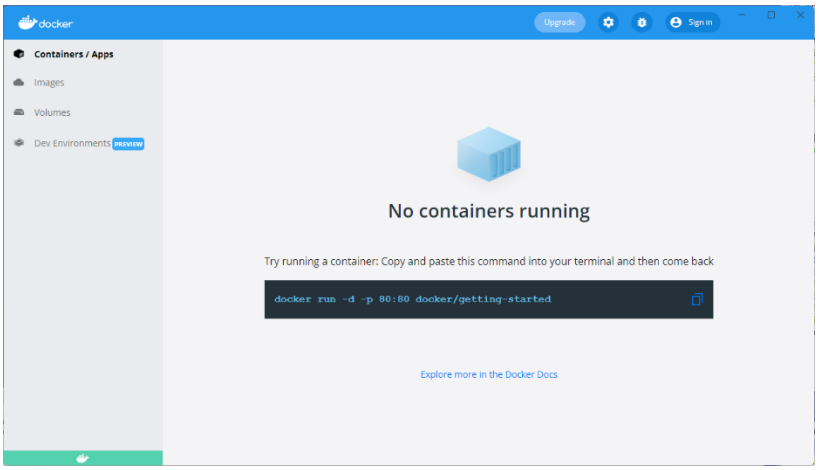


15. Navigate to folder where GitHub code was extracted from zip file. Browse to Doorsteps.Experiments. Double click on Doorsteps.Experiments.sln. Visual Studio will open.

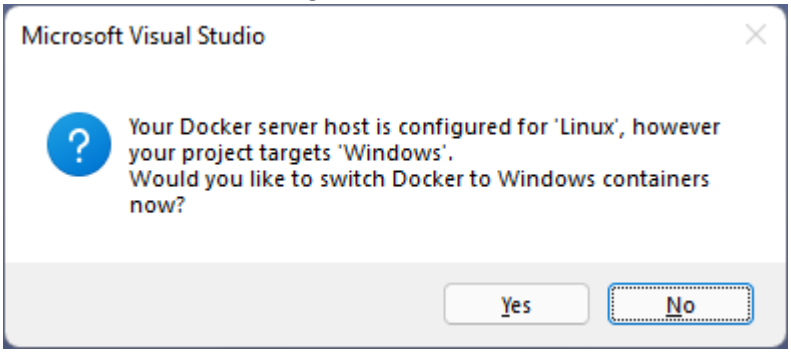
16. Read the agreement and click I accept the terms. Click Accept.



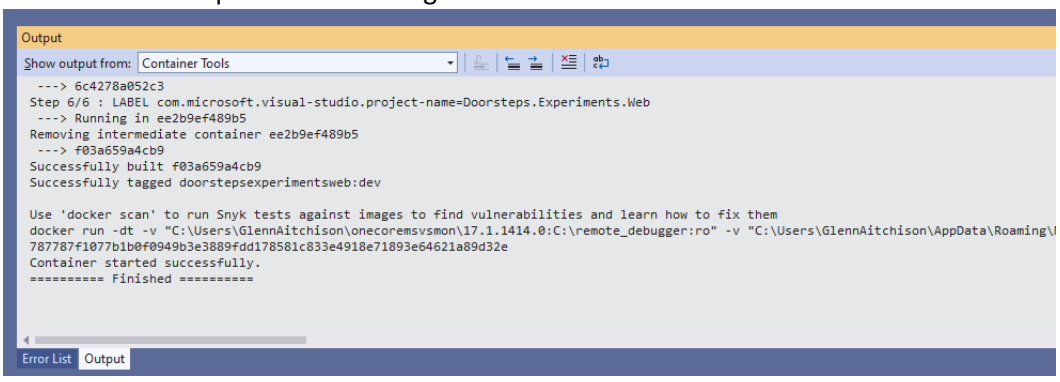
17. A Docker window will appear and after a minute or show will show the following – Minimize the window.



18. Switch back to Visual Studio – you will now see the following prompt – Click Yes to switch to using Windows Containers.



19. Wait while the required docker images are downloaded.



20. Navigate to Doorsteps.Experiments.Web\Config in Visual Studio. Double click on Spring.xml.

    Change <your ip> with your computer's ip address on the network. i.e. 192.168.0.1.

    Save the file.

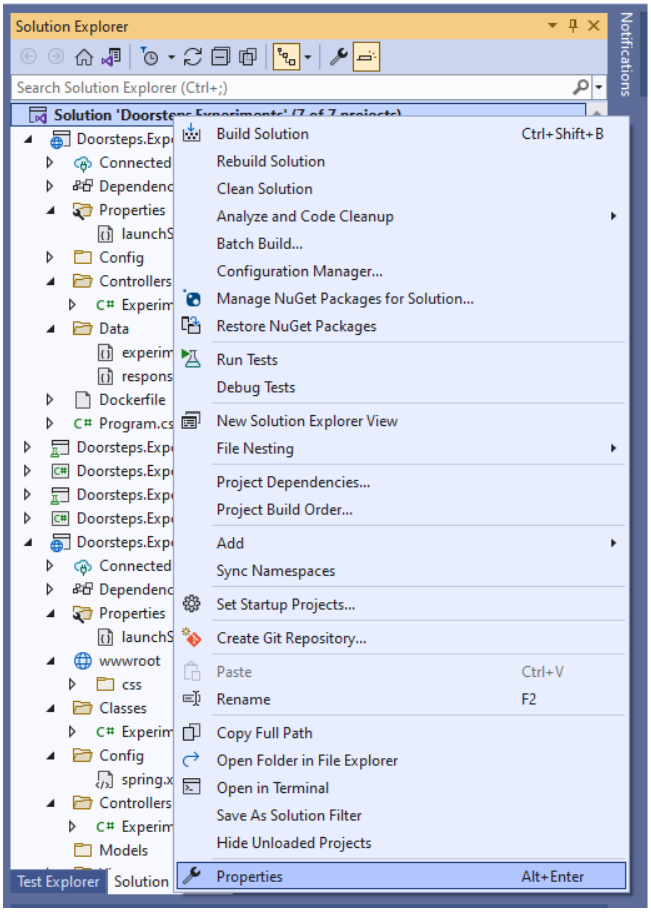
```
<?xml version="1.0" encoding="utf-8" ?>
<objects xmlns="http://www.springframework.net">

    <object name="myExperimentsApi" type="RestSharp.RestClient, RestSharp">
```

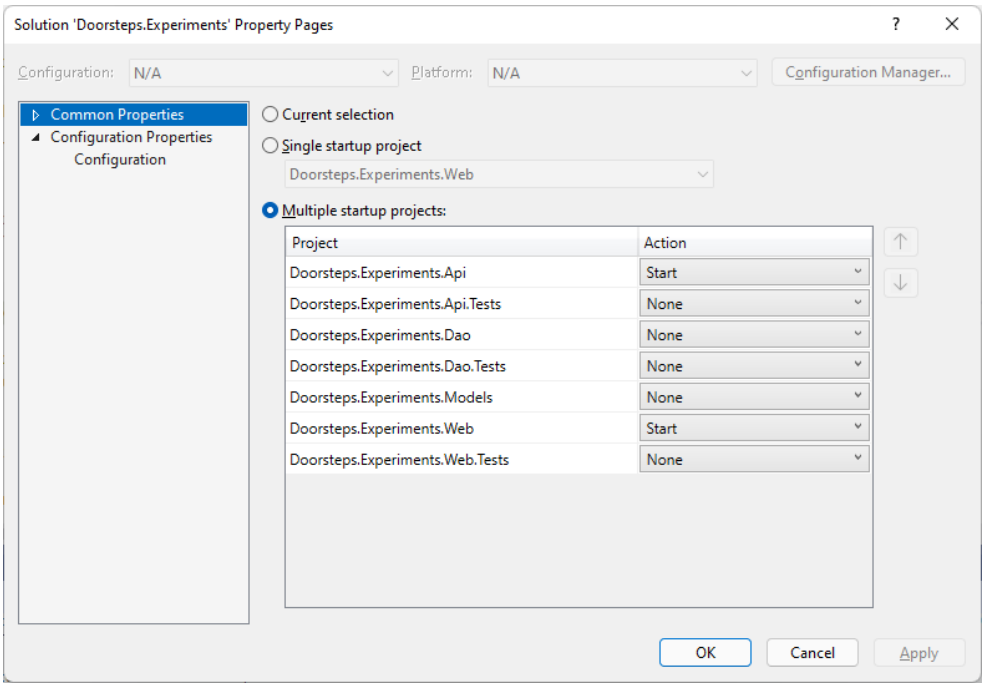
```
        <constructor-arg value="http://<your ip>:44312/api/experiments"/>
    </object>

</objects>
```

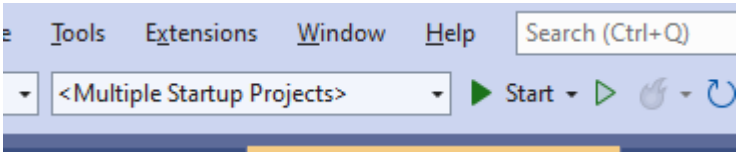
21. Right click on the Solution in Visual Studio and Properties.



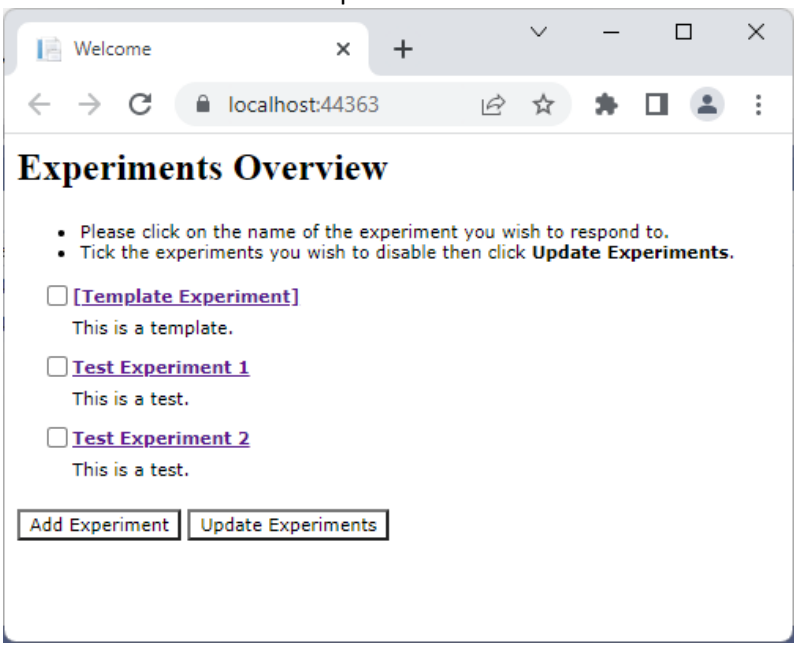
22. Select Start for both Doorsteps.Experiments.Api and Doorstep.Experiments.Web projects. Click OK.



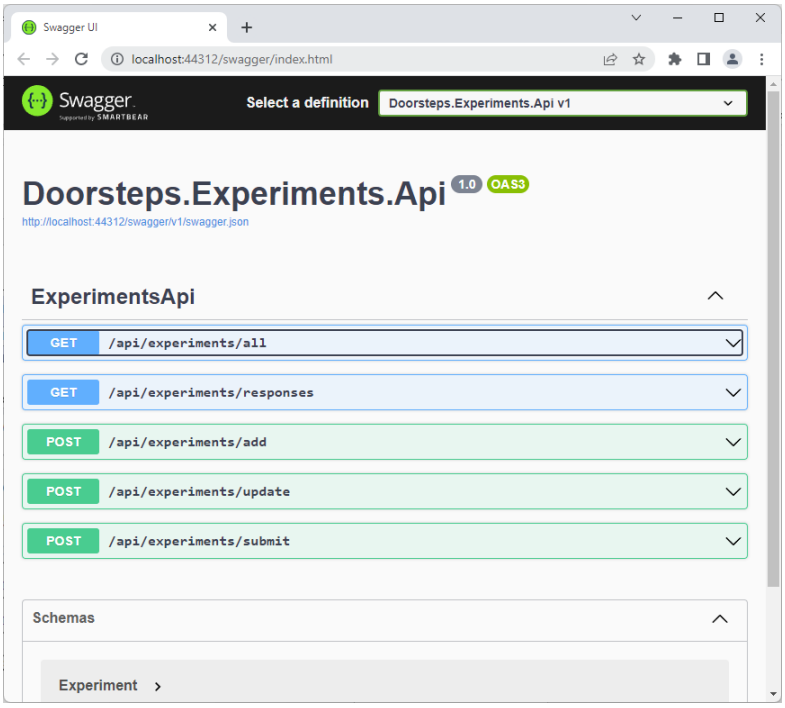
23. On the visual studio toolbar you will see Multiple Startup Projects are selected. Click the Start button next to it.



24. Two browser windows will open -



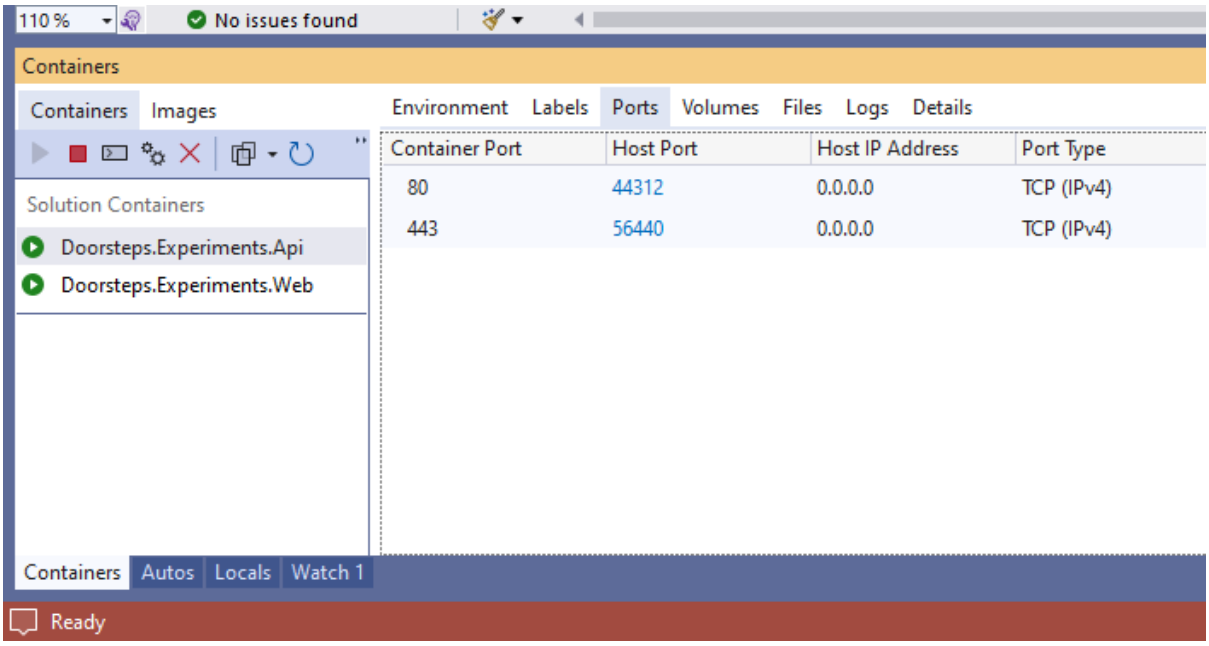
The web application where you can interact with the Experiments application.



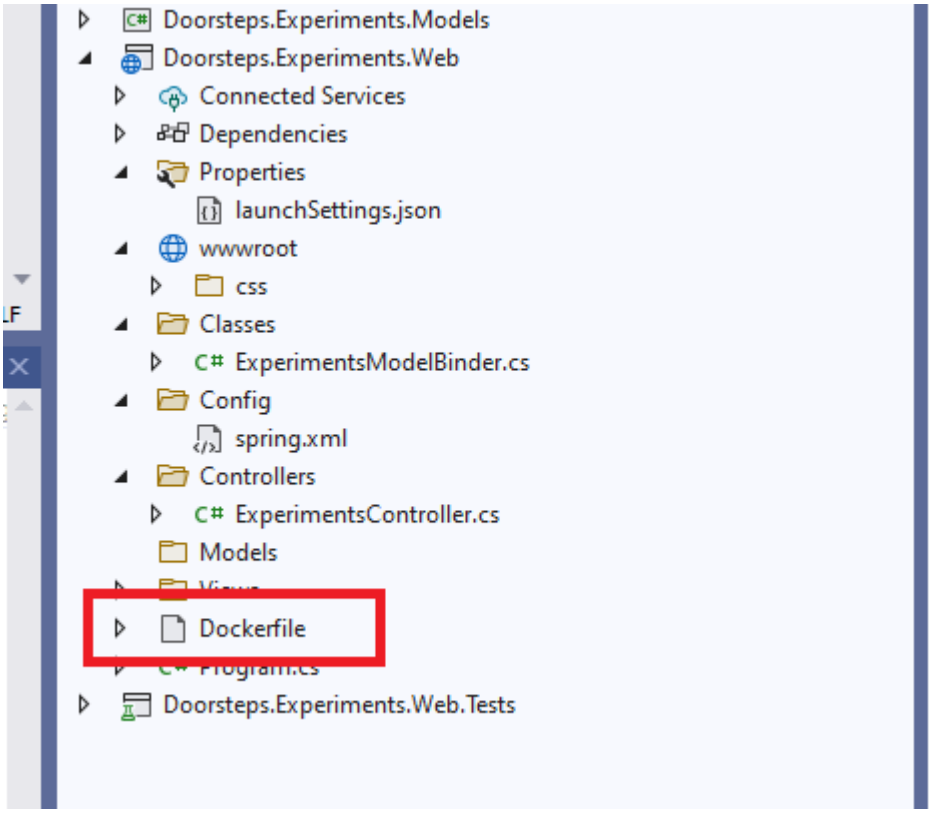
A front-end called Swagger to interface directly with the Experiment APIs.
It has been enabled to allow you tryout the APIs that have been built in real time yourself.

You will also notice Visual Studio automatically launches Docker and runs the API and web application within them. Looking at the bottom part of the Visual Studio window when the Start buttons been pressed shows –

As you can see there are two windows containers running on ports 80,443 which is mapped to 44312 and 56440 on the host.

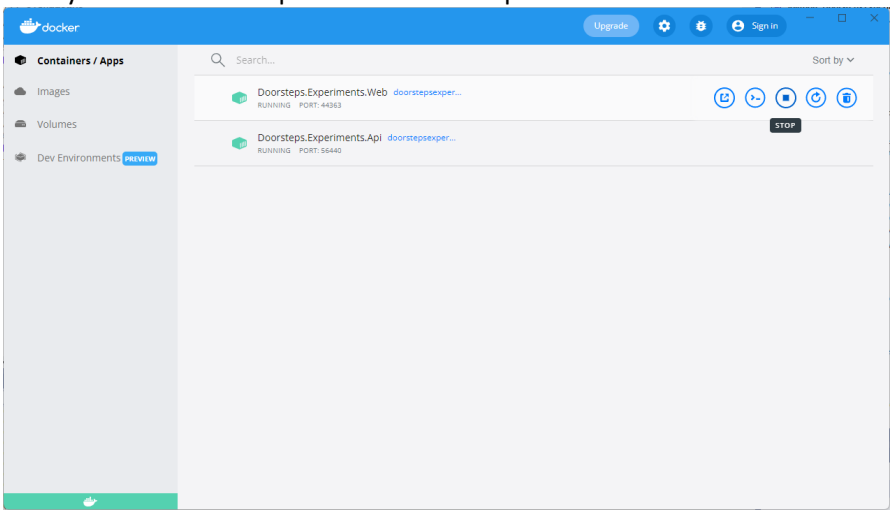


Also note the containers contain a Dockerfile (which Visual Studio created for me) – phew! This file is used to build the docker container before it is run.



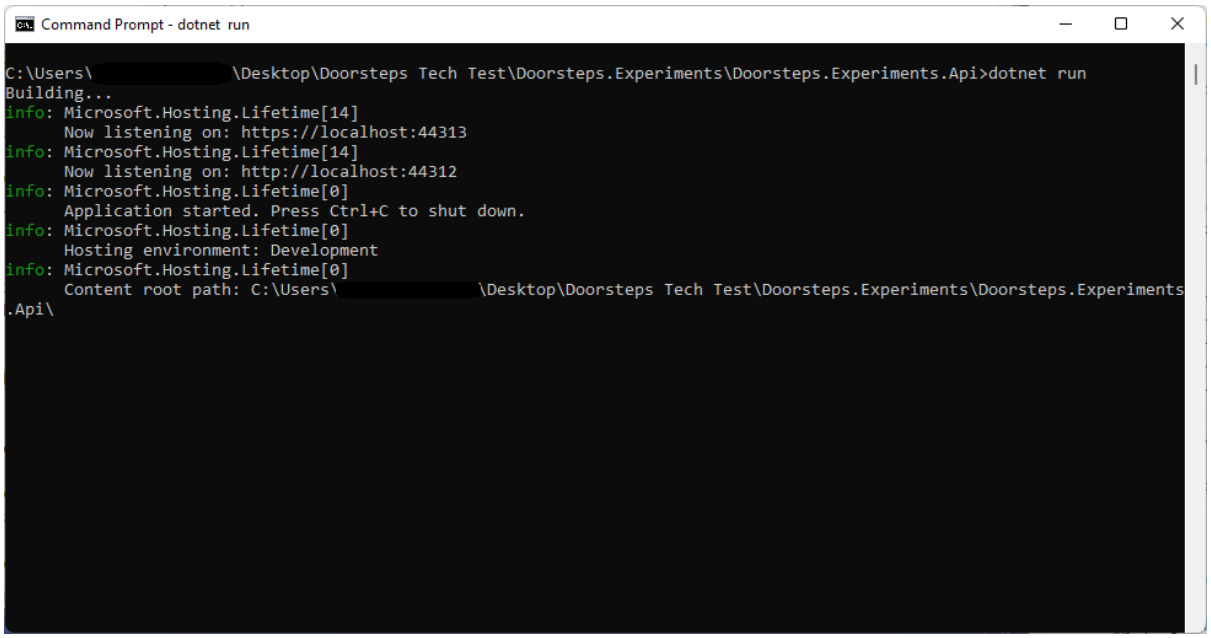
If you wish to run the applications without Docker you can run them in two command windows on your computer. First open a command prompt and change directory to your Doorsteps.Experiments\Doorsteps.Experiments.Api folder.

Next you will need to open Docker Desktop from the Start Menu.

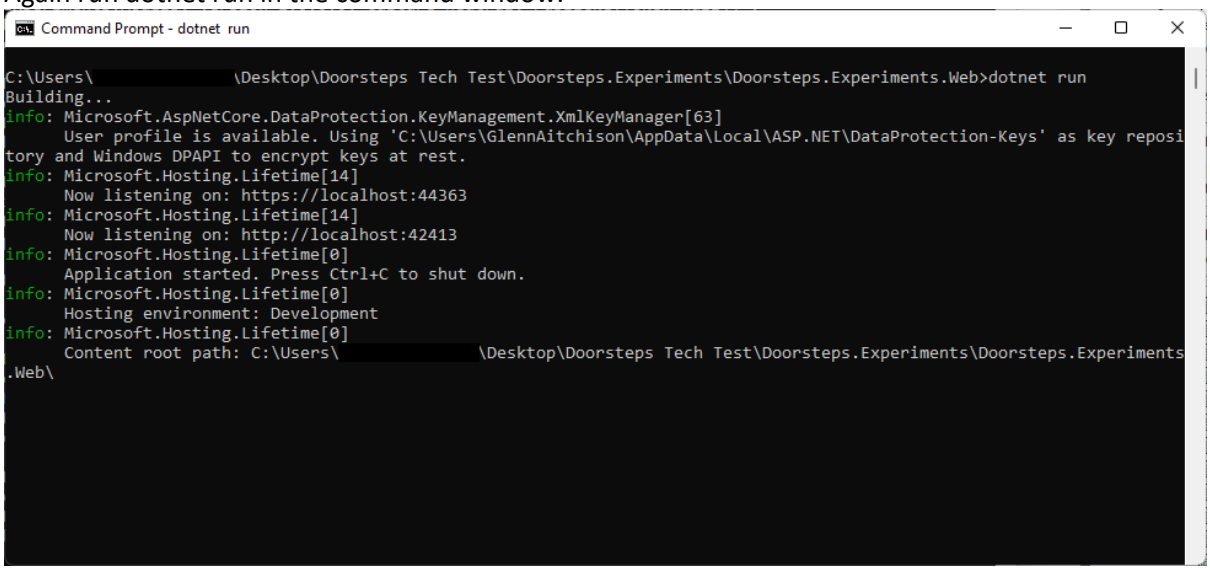


Move your mouse over each container and click the Delete button.

In the command prompt, type dotnet run.
You will see the Web APIs running in a container and is listening on port 44313 and 44312.



Open another command prompt and change directory to
your Doorsteps.Experiments\Doorsteps.Experiments.Web folder.
Again run dotnet run in the command window.



Open a web browser and navigate to https://localhost:44363 and the website opens –