Developer Manual

for

My Favorite Movies App

COMP. 6215 Assignment-2: Deliverable (5)

Edited by @ Emerald Chen

14 / 06 / 2019

Table of Contents

1. Introduction

This desktop application was created with C# as backend and WPF for GUI which links a MSSQL server using a self-made webapi to manipulate data with CRUD operations.

Please note that this manual only briefly introduces what the project should be built based on Windows. For more details both on Windows and Mac, please visit https://docs.jwk.nz/dotnet_core/sqlsvr_webapi/dotnet-2.1.x/01-project-setup/

2. Build a Webapi

In order to set up a Webapi successfully, installation of Visual Studio Code, Docker and Gitbash had been done beforehand.

Below are main steps in Webapi building

- Create a new project

1) Open the Gitbash terminal and navigate to your code folder typing in `git init webapi` to create a working directory with a git repository
2) Go into the folder by typing in `cd webapi`
3) Type in `dotnet --list-sdks` to check if you need to install the version that you need for your project.
4) Next type in `dotnet new global` to create a global.json file that specifies what version of dotnet core to be used for your project
5) Next create your project by typing in `dotnet new webapi -o api --no-https`
6) Then create a .gitignore file, copy the code from https://raw.githubusercontent.com/dotnet/core/master/.gitignore and paste it into .gitignore file so that you won't push all our binary and inessential files to your repository
7) Back in Gitbash terminal to start your project by typing in `dotnet run` and view your site @ http://localhost:5000/api/values

   Now we can make changes needed to connect it to a database.

Setting up the database

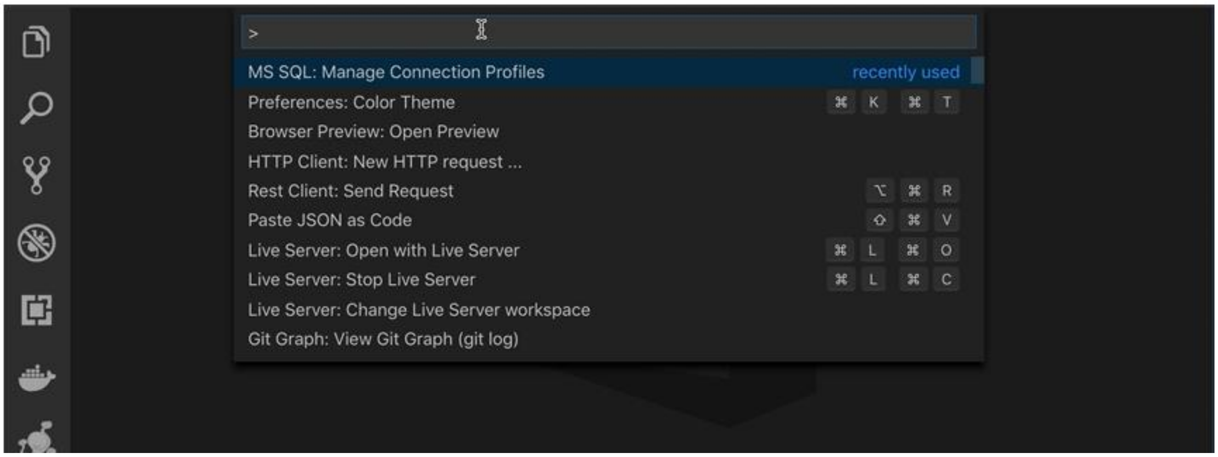
Microsoft has embraced the world of containers so we could install it as part of a docker container.

1) In docker-toolbox fit bash terminal type the following to download 2017 version of MSSQL Server
   docker pull mcr.microsoft.com/mssql/server:2017-latest-ubuntu

2) After the download is complete type in
   docker run -e 'ACCEPT_EULA=Y' -e 'yourStrongPassword' -e 'MSSQL_PID=Express' -p 1433:1433 --name webapi_mssql -d mcr.microsoft.com/mssql/server:2017-latest-ubuntu
   The above password and name are set to yourStrongPassword and webapi_mssql respectively. You should change them to your own password.
3) You can now start the database server by typing in docker start webapi_mssql

Populating the Database with some data

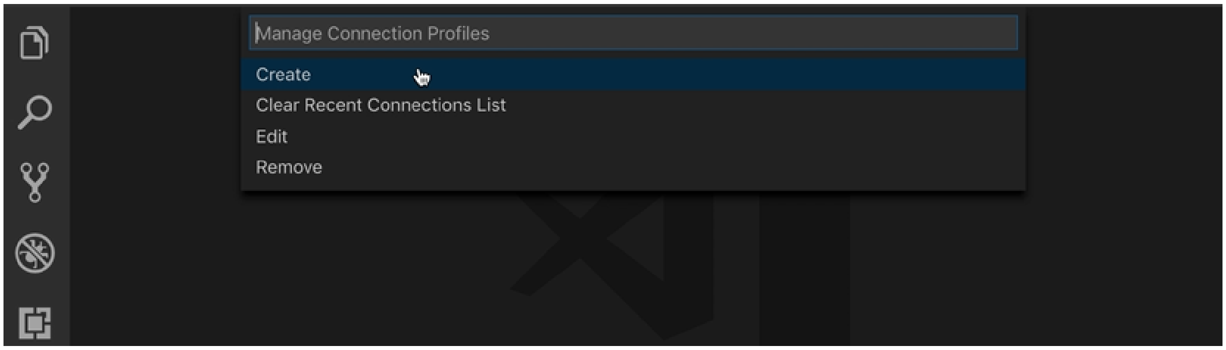
1) Type the following command in the terminal to install the extension if you use VS Code code --install-extension ms-mssql.mssql
2) Connect VS Code to your database:

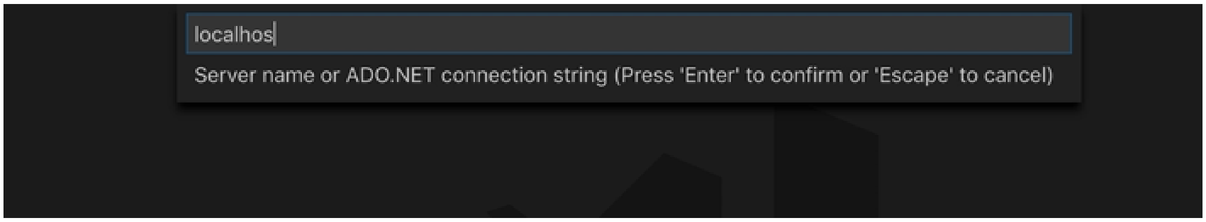


MS SQL: Manage Connection Profiles

Preferences: Color Theme ⌘ K ⌘ T
Browser Preview: Open Preview
HTTP Client: New HTTP request ...
Rest Client: Send Request ⌥ ⌘ R

Live Server: Open with Live Server ⌘ L ⌘ O
Live Server: Stop Live Server ⌘ L ⌘ C
Live Server: Change Live Server workspace
Git Graph: View Git Graph (git log)

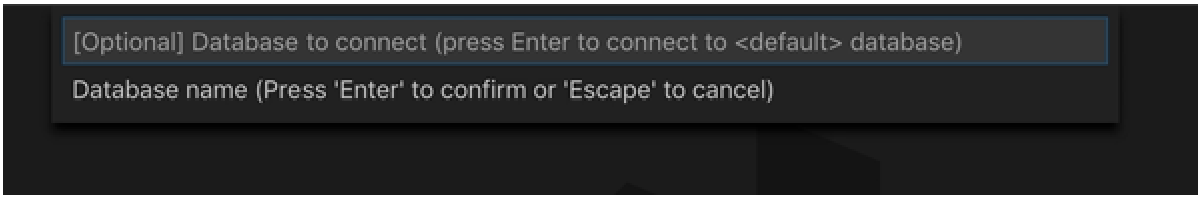




localhos

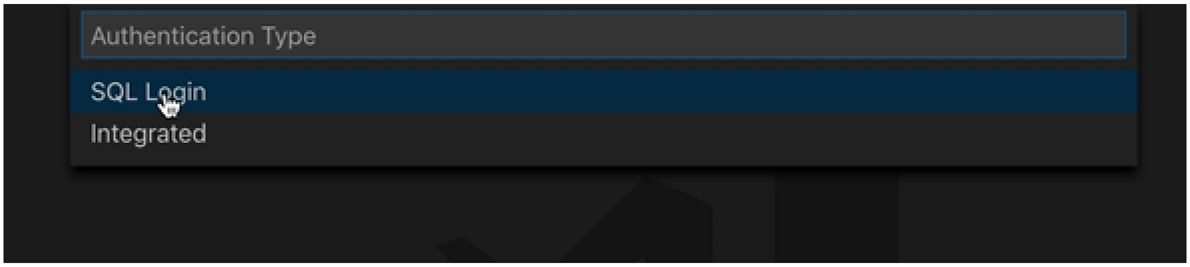
Server name or ADO.NET connection string (Press 'Enter' to confirm or 'Escape' to cancel)

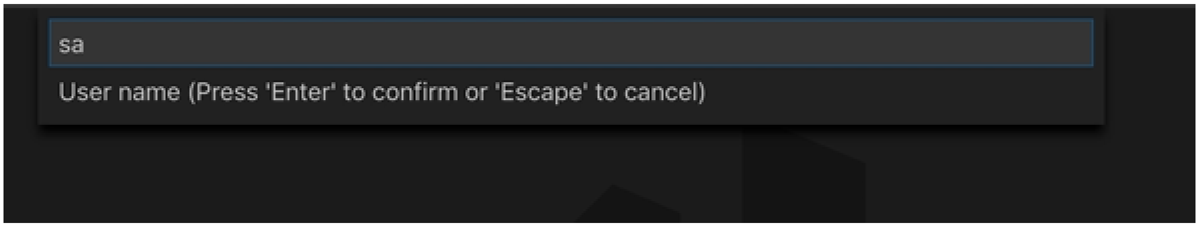
On Windows, instead of typing in localhost, you will need to type in the VM's ip address, which is: 192.168.99.100



[Optional] Database to connect (press Enter to connect to <default> database)

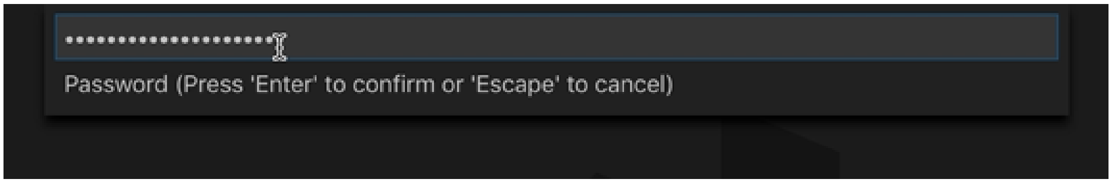
Database name (Press 'Enter' to confirm or 'Escape' to cancel)





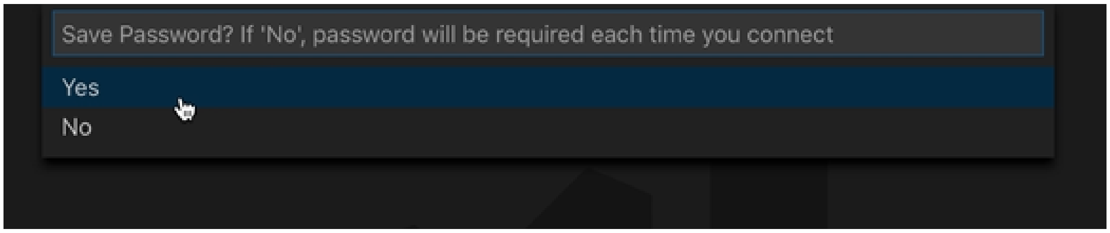
sa

User name (Press 'Enter' to confirm or 'Escape' to cancel)

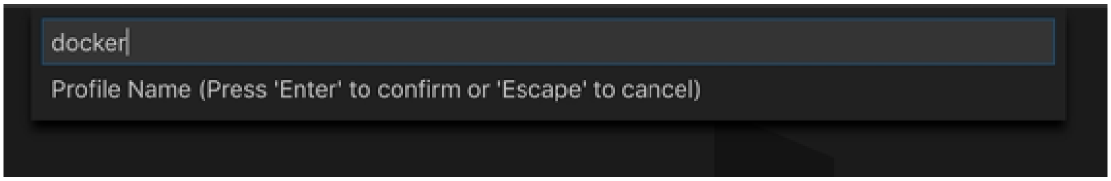


••••••••••••••••••••

Password (Press 'Enter' to confirm or 'Escape' to cancel)



Save Password? If 'No', password will be required each time you connect



docker

Profile Name (Press 'Enter' to confirm or 'Escape' to cancel)

Once you have the connection setup, go to the sql files and right click and select Execute Query

NOTE:

The above steps assume your database is still running in the docker container.

So, if you have any errors, check that first by typing in `docker ps -a` into the terminal

- Setup EF Core (Entity Framework)

This project we use **Data First** approach

1) Type in `dotnet restore` to install the nuget packages.
2) Make some changes in the **api.csproj** file and **startup.cs** file. Follow the instruction in the section of SETUP EF Core(Entity Framework) @ https://docs.jwk.nz/dotnet_core/sqlsvr_webapi/dotnet-2.1.x/01-project-setup/

- Create a CRUD API that is connected MSSQL
  1) Create models
  2) Create a DBContext
  3) Create controllers

Please find more details @ https://docs.jwk.nz/dotnet_core/sqlsvr_webapi/dotnet-2.1.x/02-create-api/

- Test the API
  1) You could perform the test using VS Code by following the steps @ https://docs.jwk.nz/dotnet_core/sqlsvr_webapi/dotnet-2.1.x/04-testing-the-api/
  2) The other way of testing is use Postman @ https://www.getpostman.com/

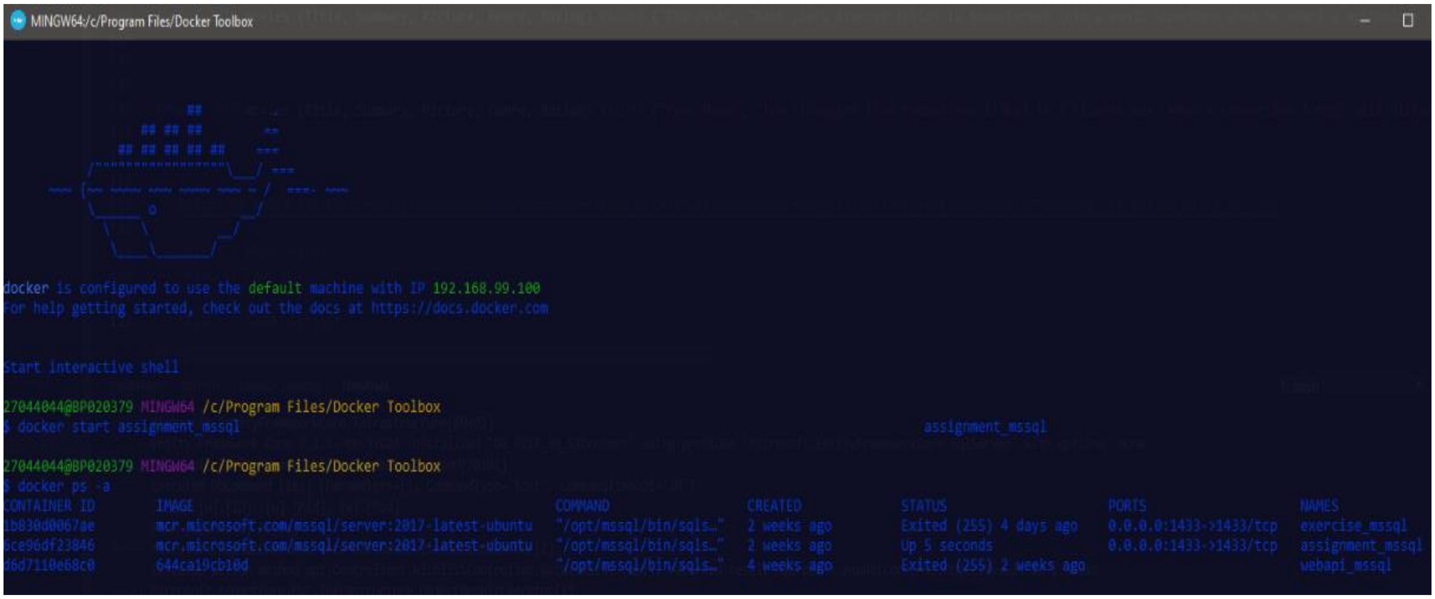
3. Set up GUI

The frond end of this app is imitative of code from https://github.com/to-jk11/fleetcarswpf

You could use any other GUI language that you prefer.

4. Run the application

Make sure the container and webapi are both running when you start the app.

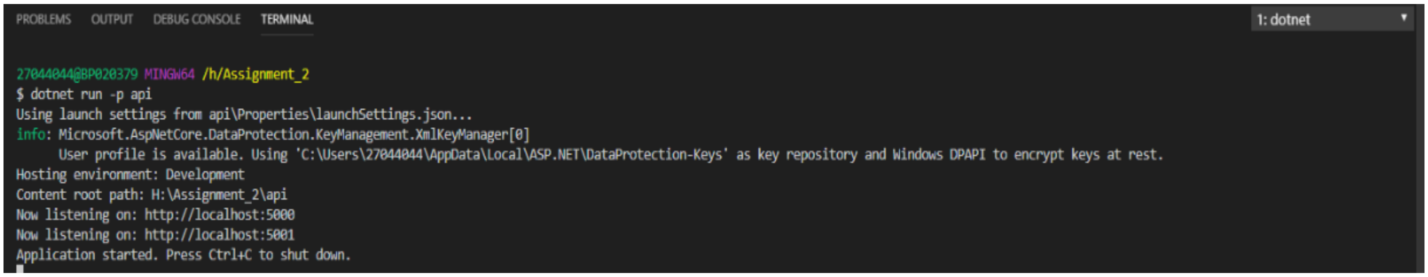


```
docker is configured to use the default machine with IP 192.168.99.100
For help getting started, check out the docs at https://docs.docker.com


Start interactive shell

27044044@BP020379 MINGW64 /c/Program Files/Docker Toolbox
$ docker start assignment_mssql
assignment_mssql

27044044@BP020379 MINGW64 /c/Program Files/Docker Toolbox
$ docker ps -a
CONTAINER ID        IMAGE                                         COMMAND                  CREATED             STATUS                       PORTS                    NAMES
1b830d0067ae        mcr.microsoft.com/mssql/server:2017-latest-ubuntu   "/opt/mssql/bin/sqls…"   2 weeks ago         Exited (255) 4 days ago      0.0.0.0:1433->1433/tcp   exercise_mssql
6ce96df23846        mcr.microsoft.com/mssql/server:2017-latest-ubuntu   "/opt/mssql/bin/sqls…"   2 weeks ago         Up 5 seconds                 0.0.0.0:1433->1433/tcp   assignment_mssql
d6d7110e68c0        644ca19cb10d                                  "/opt/mssql/bin/sqls…"   4 weeks ago         Exited (255) 2 weeks ago                              webapi_mssql
```



```
27044044@BP020379 MINGW64 /h/Assignment_2
$ dotnet run -p api
Using launch settings from api\Properties\launchSettings.json...
info: Microsoft.AspNetCore.DataProtection.KeyManagement.XmlKeyManager[0]
      User profile is available. Using 'C:\Users\27044044\AppData\Local\ASP.NET\DataProtection-Keys' as key repository and Windows DPAPI to encrypt keys at rest.
Hosting environment: Development
Content root path: H:\Assignment_2\api
Now listening on: http://localhost:5000
Now listening on: http://localhost:5001
Application started. Press Ctrl+C to shut down.
```

As the database is originally from

https://gist.github.com/to-jk11/7ad9d85fe7dd86b59f8e555fc44483cf

You could only log in with either of the followings unless any of it is updated after login

1. Uname: SRogers

   Password: America1

2. Uname: TStark

   Password: Iron1

In addition, there are only four movies in the given database as below.

1 Avengers: End Game
2 The Godfather
3 The Mask
4 Your Name

5. Contribute to this project

You are very welcome if you want to contribute to this project.

Please find the whole folder of this project @ https://github.com/ToiOhomaiBCS/comp6215-ro-2019-s1-assignment-02-EmeraldNZ

For the guide to contributing to a Github project, please find the links as below

https://gist.github.com/MarcDiethelm/7303312

https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/