Use MSSql in a docker container

Install and configure Docker

If you don't already have Docker installed, you'll need to download and install it.

https://docs.docker.com/docker-for-mac/install/

Next step, you'll need to increase Docker's available memory to 4GB or more.

Docker -> Preferences
Increase Memory to at least 4GB
Click Apply & Restart

Get the Docker image

Open a Terminal window, and download the latest SQL Server for Linux Docker image.

```
User:~$ docker pull microsoft/mssql-server-linux:2017-latest
```

Now, launch an instance of the Docker image.

```
User:~$  docker run -d --name SQLServer -e 'ACCEPT_EULA=Y' -e
'SA_PASSWORD=p@ssw0rd' -e 'MSSQL_PID=Developer' -p 1433:1433 microsoft/
mssql-server-linux:2017-latest
```

(Choose your own password, and use the same one in the appsettings.json later)

You only need to do this once, from now one restart the server with

```
User:~$ docker restart SQLServer
```

If you forgot how you named your docker image, you can run

```
User:~$ docker ps -a
```

To list all docker processes which are installed

You should now have SQL Server running on your Mac, ready for action!

Configure the visual studio project

In visual studio, appsettings.json

```
{
  "ConnectionStrings": {
    "RecipeContext": "Server=127.0.0.1,1433;
User Id=sa;Password=p@ssw0rd;Database=Recipes"
    //"RecipeContext": "Server=.\
\sqlexpress;Database=Recipes;Trusted_Connection=True"
  },
  "Logging": {
    "LogLevel": {
      "Default": "Warning"
    }
  },
  "AllowedHosts": "*"
}
```

User Secrets

Setting user secrets on macOS (or linux) is best done using the command line. Make sure you are inside the folder with the ProjectName.csproj file (it contains a <UserSecretsId> key which is needed for this to work) and issue the following command to set the key:

```
dotnet user-secrets set "Tokens:Key" "secretthatislongenough"
```

You only need to do this once.
Make sure the secret is a long enough string, or you'll get weird errors.