*Getting Started Developer Guide  
Ver. 1.0  
Date: 05/15/2019  
Author: Raffi Tchakmakjian*

Getting Started Developer Guide

Objective of this document:

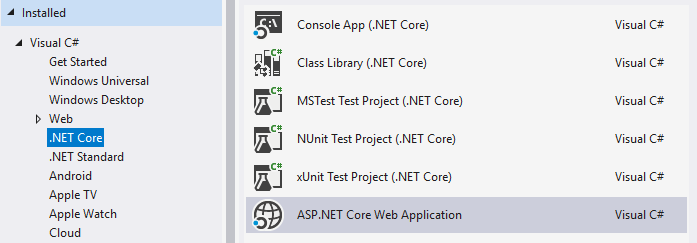
The objective of this document is to familiarize the developer with the tools and development environment setup required to run, debug, and develop the KLoveCompanyCRUD application.

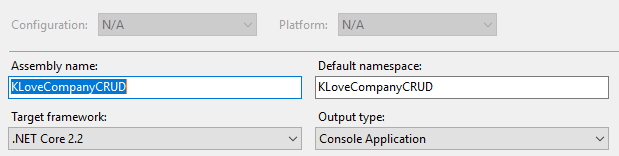
### Introduction

The KLoveCompanyCRUD application was developed using Visual Studio 2017 Community Edition.



The project is a .NET Core 2.2 web app:



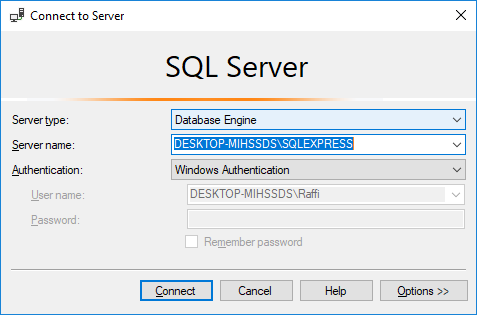
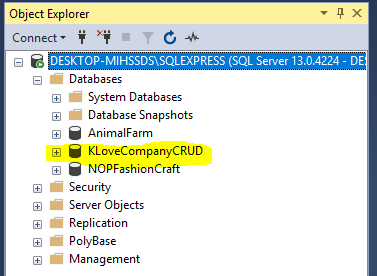


### My development environment

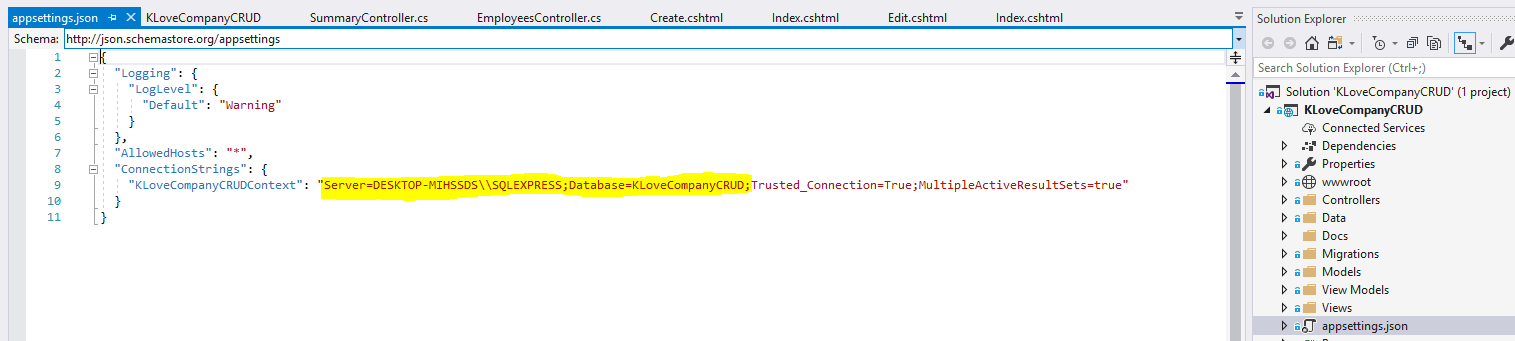
I used a code-first approach with the database and entities and created my first CRUD objects using this guide:

<https://docs.microsoft.com/en-us/aspnet/core/tutorials/razor-pages/model?view=aspnetcore-2.2&tabs=visual-studio>

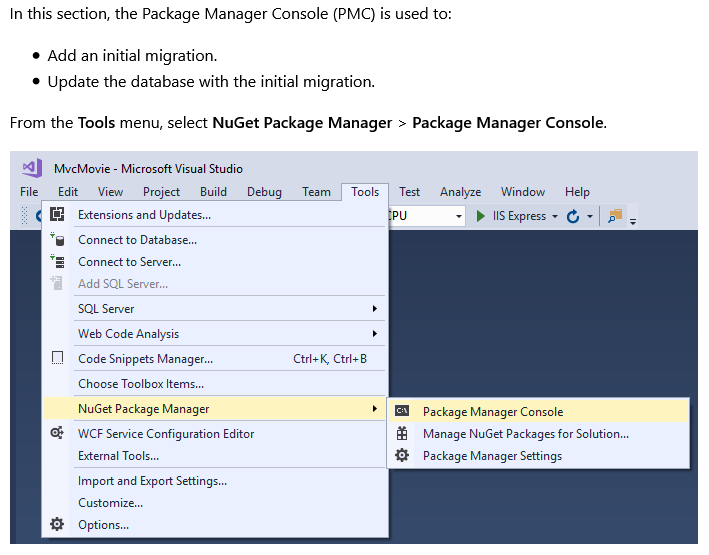
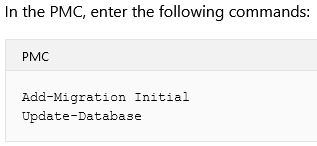
Initially, I created an empty local database called KLoveCompanyCRUD on my locally installed SQLExpress server/instance.

I then configured the appsettings.json file in the VS project with the values for my local SQLExpress instance.

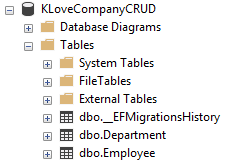


I made a couple of migrations as described in the aspnet core tutorials I linked above and created and updated the database using the following commands:

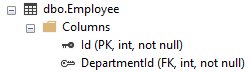
 

You’ll likely only have to run Update-Database if you create a database with the same name since the migration elements have already been added to the project (ensure the appsettings.json string is configured correctly to connect to the local db).

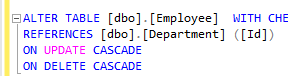
Once Update-Database excecutes successfully, your database should look like this:



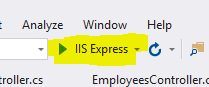
The Employee table DepartmentId column has a foreign key relationship to the Database table Id column:



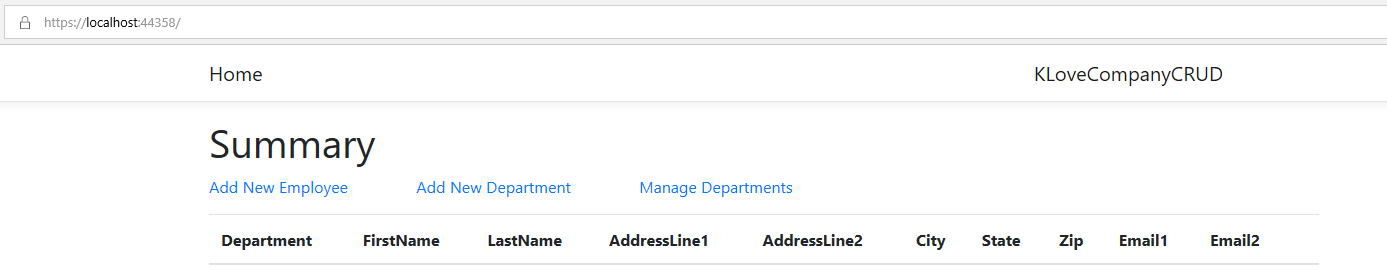
There are also a cascase on delete and cascade on update relationship from the Deparmtnet table to the Employee table.



Once the database and tables have been successfully configured by excecuting Update-Database command in the Package Manager Console in VS, you should be able to launch/debug the application by clicking on the start button (highlighted for emphasis). Here we can see the app will run on a virtualized IIS Express server instance on my local system:



The application should launch your web browser and if everything is working correctly you should be presented with this screen:



### Basic smoke test

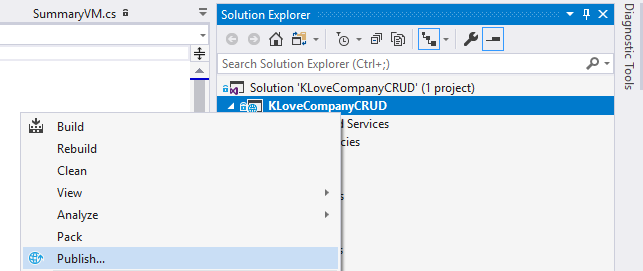
From the summary screen, to ensure the app is working correctly, click on Add New Department, add a department, then back from the Summary screen, click on Add New Employee and add an employee. Once the employee is successfully created, edit the employee and confirm the edit works. Then delete the employee.

### Publishing the application

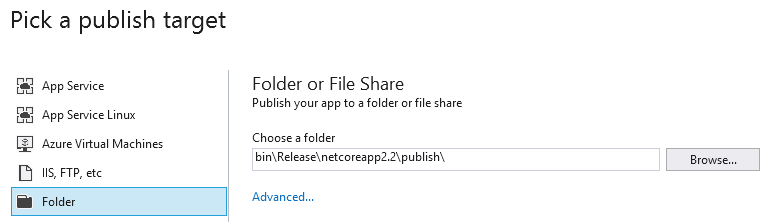
To publish the application to your local IIS instance or to a server, switch the project to release build config:



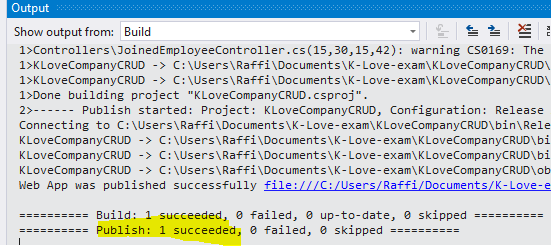
right click the project, left click Publish:



Select Folder for target and note the publish location and click Publish button:



Confirm the publish was successful by observing the output window:



At this point, you will either have to have IIS installed on your system or on whatever server you’re targeting. Once you copy the publish files to the target computer and remote into the server (or if you’re on youre own computer and have IIS), launch the IIS manager by typing inetmgr in the start search bar and hit enter.

Once IIS manager launches, you will have to create an application an place it in an application pool.

TODO: since I don’t have the means to describe this full at the moment, I’ve included this guide.

<https://docs.microsoft.com/en-us/aspnet/web-forms/overview/deployment/visual-studio-web-deployment/deploying-to-iis>

The TODO is here to update this doc in the future with screenshots of deploying the app to IIS on a Windows Server and assigning it to an application pool and configuring the URL and port the app will be exposed on.