**Created new web.api project**

## Add nuget packages

1. Add entityframework core
2. Add entityframework core sql server

**Create model folder**

**Add model class**

add a class named Message

**Add using**

using System.ComponentModel.DataAnnotations;

**Add code**

using System.ComponentModel.DataAnnotations;

namespace WebApplication.Model

{

public class Message

{

public int MessageId { get; set; }

[Required]

public string Text { get; set; }

}

}

**Create data folder**

**Add context class**

Add a class named context

**Add using**

using Microsoft.EntityFrameworkCore;

using WebApplication.Model;

**Add code**

using Microsoft.EntityFrameworkCore;

using WebApplication.Model;

namespace WebApplication.Data

{

public class MyContext : DbContext

{

public MyContext(DbContextOptions<MyContext> options) : base(options) { }

public DbSet<Message> Messages { get; set; }

}

}

**Open startup.cs**

**Add using**

using WebApplication.Data;

using Microsoft.EntityFrameworkCore;

**Navigate to ConfigureServices()**

**Add new code after line 29**

services.AddDbContext<MyContext>(options =>

options.UseSqlServer(

Configuration.GetConnectionString("DefaultConnection")));

**Open appsettings.json**

**Add code after line 1**

Replace the value prefixed with My.

"ConnectionStrings": {

"DefaultConnection": "Server=(localdb)\\mssqllocaldb;Database=MyDatabaseName1;Trusted\_Connection=true;MultipleActiveResultSets=true"

},

**After Connection String**

Modify connection with a value appropriate for working locally or in the cloud. Replace all values prefixed with My. when using Azure, select an affordable DTU value.

* Local Machine

Server=(localdb)\\mssqllocaldb;Database=MyDatabaseName;Trusted\_Connection=True;MultipleActiveResultSets=true

* Azure SQL Databae

Server=MySqlServerName.database.windows.net;Database=MyDatabaseName;User Id=MySqlLogin;Password=MyPassword;MultipleActiveResultSets=true

**Migrate changes to the database**

1. Tools – NuGet Package Manager – Package Manager Console
2. add-migrations newPhotoModel
3. update-database

## add html

<ul id="messages">

</ul>

## Update client script to refresh

setInterval(function () {

}, 1000);

## Update client script to get server data

setInterval(function () {

$.ajax({

url: '/api/messages',

type: 'get'

}).done(function (data) {

$('#messages').html('');

for (var i in data) {

$('#messages').append('<li>' + data[i].text + '</li>');

}

});

}, 1000);

## Add html

<input id="text" />

<button id="send">Send Message</button>

## Add event handler

$('#send').click(function () {

});

## Update event handler

$('#send').click(function () {

$.ajax({

url: '/api/messages',

type: 'post',

data: JSON.stringify({ text: $('#text').val() }),

contentType: "application/json"

});

});