**Entity Framework Setup**

**How to setup Ef Core in Program.cs to get connected and all works**

First open program.cs ang type this then you can modify depends on your own setup and connection name

using Asp.NetCore\_MVC\_Practice.EntityDbData;

using Microsoft.EntityFrameworkCore;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllersWithViews();//ito yung connection string at connection ng Ef Core

builder.Services.AddDbContext<ApplicationDbContext>(options=> options.UseSqlServer(builder.Configuration.GetConnectionString("DbConnection")));// ito yung name ng connection string

var app = builder.Build();

// Configure the HTTP request pipeline.

if (!app.Environment.IsDevelopment())

{

app.UseExceptionHandler("/Home/Error");

// The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.

app.UseHsts();

}

app.UseHttpsRedirection();

app.UseRouting();

app.UseAuthorization();

app.MapStaticAssets();

app.MapControllerRoute(

name: "default",

pattern: "{controller=Home}/{action=Index}/{id?}")

.WithStaticAssets();

app.Run();

1. Install Entity Framework Core & Entity Framework Sqlserver in ManageNuget Package
2. Create a Model class and name what ever you want [key] represent reminder to id this is Id [required ] is reminder to this is data of user input

using System.ComponentModel.DataAnnotations;

namespace Asp.NetCore\_MVC\_Practice.Models

{

public class DbTable

{

[Key]

public int Id { get; set; }

[Required]

public string Name { get; set; }

public int age { get; set; }

}

}

1. Create a folder outside after that inside of folder create a class then write this

using Asp.NetCore\_MVC\_Practice.Models;

using Microsoft.EntityFrameworkCore;

using Microsoft.EntityFrameworkCore.Diagnostics;

namespace Asp.NetCore\_MVC\_Practice.EntityDbData

{

public class ApplicationDbContext : DbContext

{

public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options) : base (options)

{

}

public DbSet<DbTable> StudentInfo { get; set; }//ito yung table sa Model yung variable name dito ito yung magiging table name

}

}

1. Command to Execute this table

First make sure nan aka install ito install moto sa **Package manager console** - Install-Package Microsoft.EntityFrameworkCore.Tools

Second after ma install yan type mo tong command nato – add migration

Add-migration means execute the class into SqlServer

Third after ma migrate type mo ito – update-database

update-database means yung table nayun after ma migrate kapag nag type ka nito automatic ma crecreate na yung table

Note: kapag gagawa another table ganyan lang rin bali gagawa kalang new Models class

Bali ang pinaka core logic dito kapag mag add ka ng table add-migration muna bago update-database ganyan lang rin kapag may changes ka sa column or kapag nag add ka ng column

Bonus tips

Paano mag add ng data sa database without textbox or ui gamit ef core ito yung code

using Asp.NetCore\_MVC\_Practice.Models;

using Microsoft.EntityFrameworkCore;

using Microsoft.EntityFrameworkCore.Diagnostics;

namespace Asp.NetCore\_MVC\_Practice.EntityDbData

{

public class ApplicationDbContext : DbContext

{

public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options) : base (options)

{

}

public DbSet<DbTable> StudentInfo { get; set; }

public DbSet<Login> Account { get; set; }

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

modelBuilder.Entity<DbTable>().HasData(

new DbTable { Id = 1, Name = "Joshua", age = 19, Lastname = "Escarez" },

new DbTable { Id = 2, Name = "Josh", age = 20, Lastname = "Manalo" },

new DbTable { Id = 3, Name = "Leodevier", age = 19, Lastname = "Semilia" },

new DbTable { Id = 4, Name = "U Morales", age = 30, Lastname = "Gaspado" }

);

}

}

}