**Step 1:**

public class Employee

{

public int Id { get; set; }

public string Name { get; set; }

public string Batch { get; set; }

public int Marks { get; set; }

}

**Step 2:**

public class StudentDbContext : DbContext

{

public StudentDbContext(DbContextOptions<StudentDbContext> options)

: base(options) { }

public DbSet<Employee> Students { get; set; }

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

modelBuilder.Entity<Employee>().

HasData(new Employee

{

Id = 1,

Name = "Ajay",

Batch = "B001",

Marks = 90

}, new Employee

{

Id = 2,

Name = "Deepak",

Batch = "B002",

Marks = 98

}

);

}

}

**Step 3:**

Microsoft.EntityFrameworkCore

Microsoft.EntityFrameworkCore.SqlServer

Microsoft.EntityFrameworkCore.Tools

**Step 4:**

"ConnectionStrings": {

"StudentDBContext": "server=LAPTOP-53S2KQS8\\SqlExpress;database=BookStore;integrated security=true",

"AllowedHosts": "\*"

}

**Step 5:**

**In Startup.cs file, add**

services.AddDbContext<StudentDbContext>(op => op.UseSqlServer(Configuration["ConnectionStrings:StudentDbContext"]));

**With EFCore you do not need to "enable" migrations - they are always-enabled. Just add new migration with Add-Migration.**

**Step 6:**

Add-Migration name

It will build project

**Step 7:**

Update-database

Seed some test data {Optional}

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

modelBuilder.Entity<Author>().HasData(new Author

{

AuthorId = Guid.NewGuid(),

FirstName = "Bob",

LastName = "Ross",

Genre = "Drama"

}, new Author

{

AuthorId = Guid.NewGuid(),

FirstName = "David",

LastName = "Miller",

Genre = "Fantasy"

});

}

}

After that ,

Add-Migration WebApi.Models.LibraryContextSeedData

Update-database

<https://www.c-sharpcorner.com/article/create-restful-api-using-asp-net-core-with-entity-framework-corecreate-restful-a/>

For Swagger

install-package Swashbuckle.AspNetCore

<https://code-maze.com/net-core-web-api-ef-core-code-first/>

<https://www.youtube.com/watch?v=CoQoWrO7Qhk>

ConfigureServices ()

{

services.AddSwaggerGen();

}

Configure()

{

app.UseSwagger();

// Enable middleware to serve swagger-ui (HTML, JS, CSS, etc.),

// specifying the Swagger JSON endpoint.

app.UseSwaggerUI(c =>

{

c.SwaggerEndpoint("/swagger/v1/swagger.json", "My API V1");

});