Asp.Net Core Project -> Api -> No Authentication

Nugget

* Microsoft.EntityFrameworkCore
* Microsoft.EntityFrameworkCore.SqlServer
* Microsoft.EntityFrameworkCore.Tools
* Swashbuckle

Add models ->

public class Student

{

[Key]

public int Id { get; set; }

public string Name { get; set; }

public string Department { get; set; }

}

Add DbContext

public class ApiContext :DbContext

{

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

{

}

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

}

Appsettings.json

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft": "Warning",

"Microsoft.Hosting.Lifetime": "Information"

}

},

"AllowedHosts": "\*",

"ConnectionStrings": {

"ApiDb": "Data Source= DESKTOP-8KOARQS;Initial Catalog=ApiDB;Integrated Security=True"

}

}

Startup.cs

public void ConfigureServices(IServiceCollection services)

{

services.AddMvc();

services.AddDbContext<ApiContext>(options => options.UseSqlServer(Configuration.GetConnectionString("ApiDb")));

services.AddControllers();

//Register Sqagger

services.AddSwaggerGen();

services.AddSwaggerGen(c =>

{

c.SwaggerDoc("v1", new OpenApiInfo { Title = "My API", Version = "v1" });

});

}

app.UseSwagger();

app.UseSwaggerUI(x =>

{

x.SwaggerEndpoint("/swagger/v1/swagger.json", "My Api V1");

});

Controller

[Route("api/[controller]")]

[ApiController]

public class StudentsController : ControllerBase

{

private readonly ApiContext \_db;

public StudentsController(ApiContext db)

{

\_db = db;

}

// GET: api/<StudentsController>

[HttpGet]

public IEnumerable<Student> Get()

{

return \_db.Students;

}

// GET api/<StudentsController>/5

[HttpGet("{id}",Name ="Get")]

public Student Get(int id)

{

return \_db.Students.SingleOrDefault(x => x.Id == id);

}

// POST api/<StudentsController>

[HttpPost]

public void Post([FromBody] Student student)

{

\_db.Students.Add(student);

\_db.SaveChanges();

}

// PUT api/<StudentsController>/5

[HttpPut("{id}")]

public void Put(int id, [FromBody] Student student)

{

\_db.Students.Update(student);

\_db.SaveChanges();

}

// DELETE api/<StudentsController>/5

[HttpDelete("{id}")]

public void Delete(int id)

{

var item = \_db.Students.FirstOrDefault(x => x.Id == id);

if (item != null)

{

\_db.Students.Remove(item);

\_db.SaveChanges();

}

}

}