# .NET Web API - Code First Approach

## Step 1: Install Required Packages

Run the following command in Package Manager Console (PMC):  
Install-Package Microsoft.EntityFrameworkCore.SqlServer  
Install-Package Microsoft.EntityFrameworkCore.Tools  
  
Or using .NET CLI:  
dotnet add package Microsoft.EntityFrameworkCore.SqlServer  
dotnet add package Microsoft.EntityFrameworkCore.Tools

## Step 2: Create the Database Context

Create a new file `ApplicationDbContext.cs` and add the following code:

using Microsoft.EntityFrameworkCore;  
  
public class ApplicationDbContext : DbContext  
{  
 public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options) : base(options) { }  
  
 public DbSet<Student> Students { get; set; }  
}

## Step 3: Configure Connection String (`appsettings.json`)

Modify `appsettings.json` and add the following connection string:

"ConnectionStrings": {  
 "DefaultConnection": "Server=(localdb)\mssqllocaldb;Database=MyDatabase;Trusted\_Connection=True;MultipleActiveResultSets=true"  
}

## Step 4: Register Database Context in `Program.cs`

using Microsoft.EntityFrameworkCore;  
  
var builder = WebApplication.CreateBuilder(args);  
  
builder.Services.AddDbContext<ApplicationDbContext>(options =>  
 options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConnection")));  
  
builder.Services.AddControllers();  
var app = builder.Build();  
  
app.UseAuthorization();  
app.MapControllers();  
app.Run();

## Step 5: Create Model Class

public class Student  
{  
 public int Id { get; set; }  
 public string Name { get; set; }  
 public int Age { get; set; }  
}

## Step 6: Generate Migrations & Database

Run the following commands to create and update the database:

dotnet ef migrations add InitialCreate  
dotnet ef database update

## Step 7: Verify Database

Open SQL Server Management Studio (SSMS) and run the following query:

SELECT \* FROM Students;

## Step 8: CRUD Operations in Controller

Create `StudentController.cs` with the following content:

using Microsoft.AspNetCore.Mvc;  
using System.Collections.Generic;  
using System.Linq;  
  
[Route("api/[controller]")]  
[ApiController]  
public class StudentController : ControllerBase  
{  
 private readonly ApplicationDbContext \_context;  
  
 public StudentController(ApplicationDbContext context)  
 {  
 \_context = context;  
 }  
  
 [HttpGet]  
 public ActionResult<List<Student>> GetStudents()  
 {  
 return \_context.Students.ToList();  
 }  
  
 [HttpPost]  
 public ActionResult<Student> AddStudent(Student student)  
 {  
 \_context.Students.Add(student);  
 \_context.SaveChanges();  
 return Ok(student);  
 }  
}

## Step 9: Run & Test

Start the application (`Ctrl + F5`) and test with Swagger or Postman:  
- `GET /api/student` → Fetch students.  
- `POST /api/student` → Add a student.

## Common Entity Framework Core Commands

Some useful EF Core commands:

|  |  |
| --- | --- |
| Command | Description |
| dotnet ef migrations add <MigrationName> | Creates a new migration file. |
| dotnet ef database update | Applies migrations to the database. |
| dotnet ef database drop | Deletes the database. |
| dotnet ef migrations remove | Removes the last migration. |
| dotnet ef migrations list | Lists all migrations. |