Uvod

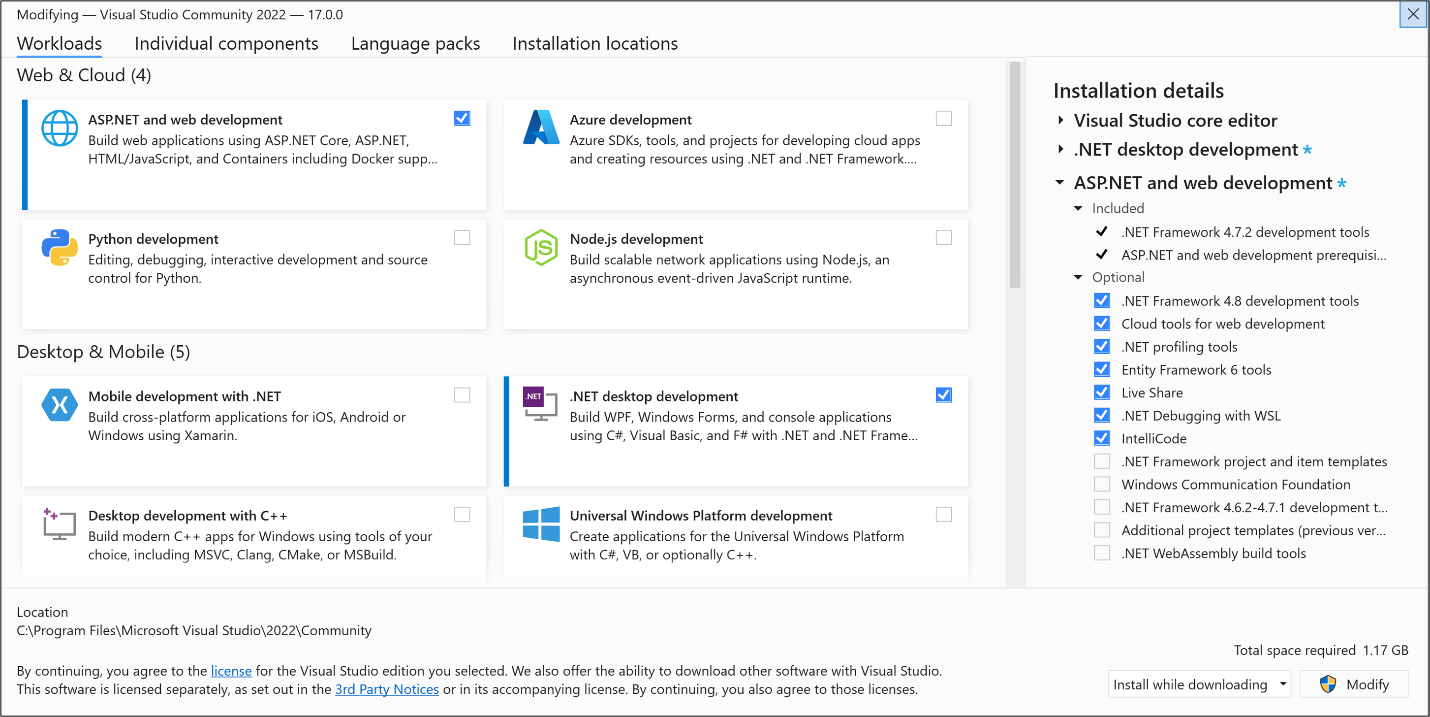
Sta je .NET?

Okruženje za razvoj sofvera, razvijano od strane Microsoft-a za Windows platforme, iOS , Android OS. Uključuje veliku biblioteku klasa (Framework Class Library). Microsoft je sa razvojem .NET-a počeo ranih 1990-tih, pod nazivom Next Generation Windows Services. Početkom 2000-tih prva beta verzija .NET 1.0 je objavljena, a u avgustu 2000. u saradnji sa Intel-om i HP-om, Microsoft je počeo sa standardizacijom Common Language Infrastrukture (CLI), koja će omogućiti izvršavanje različitih programskih jezika na različitim arhitekturama-platformama.

Instalacija

* Instalacija Visual Studio 2022

Prvo je potrebno skinuti VS 2022 sa sledeceg linka <https://visualstudio.microsoft.com/vs/> I izabrati sledece Workload-e



* Instalacija Visual Studio Code-a

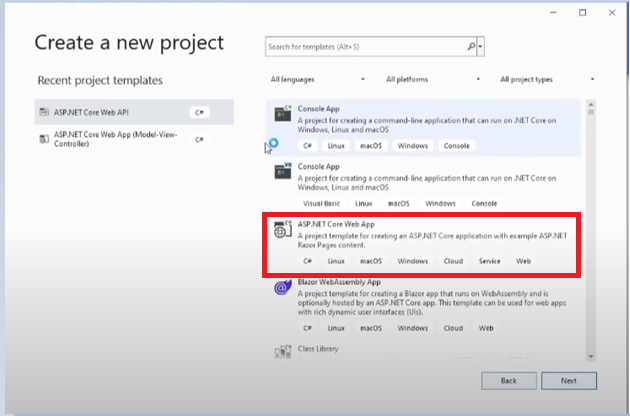
<https://code.visualstudio.com>

* Instalacija SQL Server Express

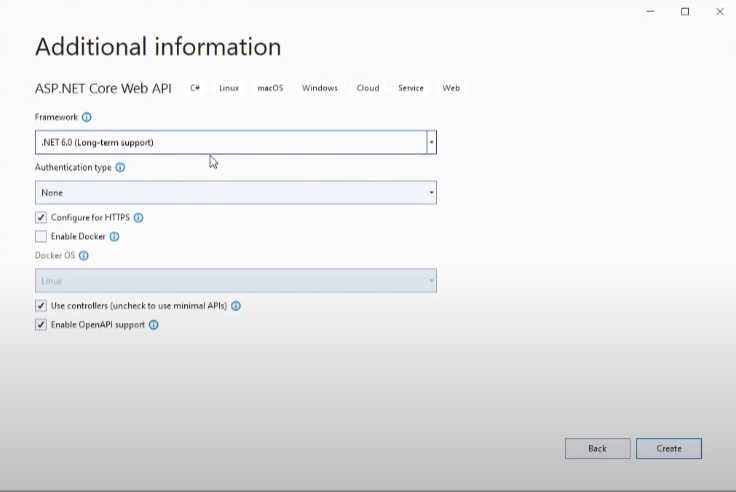
<https://www.microsoft.com/en-us/sql-server/sql-server-downloads>

Kreiranje projekta

Klikom na Create new project zapocinjete proces kreiranja projekta.Prvo ce vam se pojaviti prozor koji vam nudi razlicite vrste templejtova, za nas projekat biramo ASP.NET Core Web API



Klikom na dugme Next otvara vam se prozor u kojem vam je ponudjeno ime projekta I lokacija.Kada ponovo kliknete na next izacice vam sledeci prozor.



Izaberite ista podesavanja kao sa slike I kliknite Create.

Ako ste ispratili sve korake onda ste uspesno kreirali svoj prvi projekat ;D

Kreiranje modalnih klasa

U okviru Solution Explorer-a unutar trenutnog projekta kreiramo novi folder u koji cemo smestati modalne klase. Zatim kreiramo obicnu klasu:

public class Inspection

{

public int Id { get; set; }

[StringLength(20)]

public string Status { get; set; } = string.Empty;

[StringLength(200)]

public string Comments { get; set; } =string .Empty;

public int InspectionTypeId { get; set; }

}

Dodavanje zavisnosti(Dependencies)

Klikom na Tools=>NuGet Package Menager=>Manage NuGet Packages otvarate prozor za Instalaciju zavisnosti.Za nas projekat je potrebno instalirati sledece zavisnosti:

- Microsoft.AspNet.WebApi.Cors

- Microsoft.EntityFrameworkCore

- Microsoft.EntityFrameworkCore.Design

- Microsoft.EntityFrameworkCore.SqlServer

- Microsoft.EntityFrameworkCore.Tools

- Microsoft.VisualStudio.Web.CodeGeneration.Design

Cross-Origin Resource Sharing (CORS)

Stvara vezu izmedju frontend-a I backend-a

U .Net veziji 6 veza se ostvaruje na sledeci nacin:

-Otvoriti Program.cs klasu I nalepiti sledeci kod

// Enable CORS

var myAllowSpecificOrigins = "\_myAllowSpecificOrigins";

builder.Services.AddCors(options =>

{

options.AddPolicy(name: myAllowSpecificOrigins,

builder =>

{

builder.WithOrigins("http://localhost:4200")

.AllowAnyMethod()

.AllowAnyHeader();

});

});

app.UseCors(myAllowSpecificOrigins);

Kreiranje migracija za bazu podataka

-prvo kreiramo folder Data I u njemu kreiramo klasu DataContext.cs sa sledecim kodom

public class DataContext:DbContext

{

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

public DbSet<Inspection> Inspections { get; set; }

public DbSet<InspectionType> InspectionTypes { get; set; }

public DbSet<Status> Statuses { get; set; }

}

-Otvorite ponovo Program.cs I dodajte sledeci kod:

builder.Services.AddDbContext<DataContext>(options =>

{

options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConnection"));

});

-Sada je u appsetings.json potrebno dodati informaciju za DefaultConnection

"ConnectionStrings": {

"DefaultConnection": "server=@X;database=@Y;trusted\_connection=true"

}

Umesto @X stavite naziv servera a umesto @Y stavite zeljeno ime baze.

Sada nam jos samo ostaje da napravimo migraciju.Otvorite Tools=>NuGet Package Menager=>Package Menager Console

U konzoli unesite sledece komande:

-dotnet ef migrations add @NazivMigracije

-dotnet ef database update

Kontroleri

Kontroleri su neophodni za bilo kakvu direktnu komunikaciju sa Frontend-om, odnosno za primanje GET I POST zahteva od naseg Frontend-a.

Kontroleri se kreiraju desnim klikom na folder “Controllers” i biranjem opcije “Add controler”.

Koristićemo template API Controller with actions,using Entity Framework koji će se generisati od strane Dependency-a CodeGenerator.

Kreirane controller-e možemo testirati pomoći swagger-a gde testiramo sve moguće zahteve kao što su POST,GET,PUT,DELETE...