



Entity Framework Integration

1. Add to appsettings:

```
"ConnectionStrings": {  
  "FirstApiContext": "Server=  
(localdb)\\mssqllocaldb;Database=MoviesDb;Trusted_Connection=True;MultipleActiveResultS  
ets=true"  
}
```

2. Create a class that inherits from DbContext.

3. Add a constructor to this class as below

```
public FirstApiContext(DbContextOptions<FirstApiContext> options)  
    : base(options)  
{  
}
```

4. Add the following in Program.cs. Add above the statement builder.Build()

```
builder.Services.AddDbContext<FirstApiContext>(options =>  
  
options.UseSqlServer(builder.Configuration.GetConnectionString("FirstApiContext")));
```

4.1. Create Controllers by injecting FirstApiContext into the constructor

Write normal Apis.

5. Open Package Manager Console: Tools->Nuget Package Manager -> Package Manager Console

>Add-Migration

>Update-Database

If the above commands dont work then follow step 5.1

5.1. Open Developer Powershell: View -> Terminal

Install dotnet ef

>dotnet tool install --global dotnet-ef

>dotnet ef migrations add v1

>dotnet ef database update

=====

Reading material for Delegates & Lambda Functions

<https://inspireinnovativelearning.blogspot.com/2020/06/lambda-expressions-as-generic-delegates.html>

=====

ASSESSMENT LINK - TO BE COMPLETED BEFORE 2:30pm

<https://app.mymapit.in/code4/tiny/Lt1wMe>

=====

WEB API -Adding CORS Configurations

=====

STEP 1: Go to Program.cs. Just above the statement "builder.Build()", add the following

```
builder.Services.AddCors(  
    builder =>  
    {  
        builder.AddPolicy(name: "AllowOrigin", corsPolicy =>  
        {  
            corsPolicy.AllowAnyOrigin()  
                .AllowAnyHeader()  
                .AllowAnyMethod();  
        });  
    }  
);
```

STEP 2: In Program.cs, Just above the line "app.Run()", add the following

```
app.UseCors("AllowOrigin");
```

STEP 3: In the KpmgController, add the following above the class

```
[EnableCors("AllowOrigin")]
```

=====

Reading Material

<https://learn.microsoft.com/en-us/aspnet/core/security/cors?view=aspnetcore-6.0#testc6>

=====OLD ONE =====

<https://github.com/sampradan-labs/RandomNotesForPax>

<https://inspireinnovativelearning.blogspot.com/2022/09/approach-for-students-learners-of-new.html>

Reading Material

1. Class Inheritance

https://www.w3schools.com/cs/cs_inheritance.php

2. Interfaces

https://www.w3schools.com/cs/cs_interface.php

3. Abstract Class

https://www.w3schools.com/cs/cs_abstract.php

4. Virtual keyword with override

https://www.w3schools.com/cs/cs_polymorphism.php

=====

Tip for interviews / mcqs:

Overloading is known as Compile-Time polymorphism as even before you run your application, the compiler knows how many overloads for your method exist.

Overriding is known as Run-Time Polymorphism as only at run time we can know if the base class or derived class method will execute.

=====

The four pillars of OOPs

1. Inheritance
2. Encapsulation: Using public, private, protected, internal keywords so that you can share classes with other projects or not.
3. Abstraction: Using abstract keyword, using Interfaces
4. Polymorphism: Overriding methods