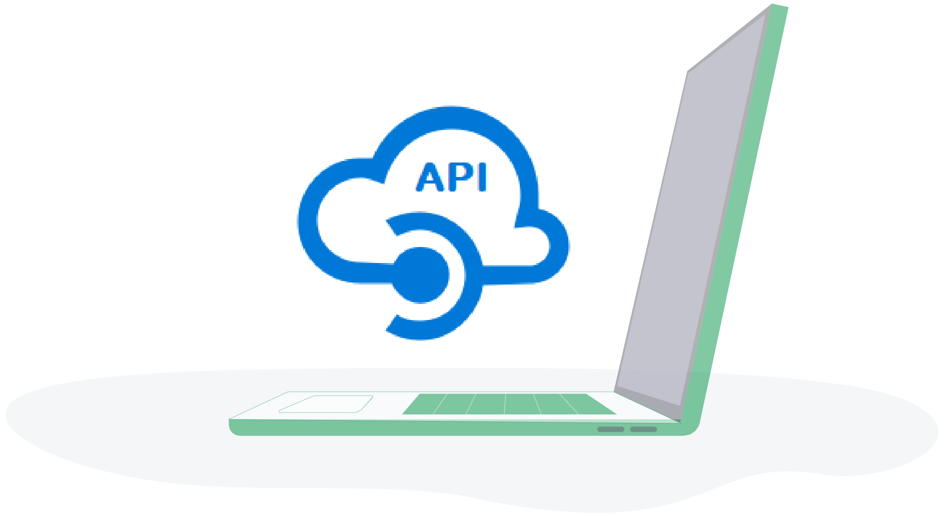
Web Application Programming Interface (API)

Tahaluf Training Center 2023

1 Overview Of Service

2 Create Service



Overview Of Service

The **services layer** is used to communicate between the Repository layer and UI. Users can call it a business or domain layer since it holds the business logic for an entity.

Service classes in the service layer are designed to do two things:

1. Query one or more Repositories.
2. Implement their own functionality, which is useful when functionality deals with more than one business object.

The Learning Hub

Create Service

Right Click on LearningHub.Infra => Add => New Folder => Service.

Right Click on LearningHub.Core => Add => New Folder => Service.

Right Click on Services in LearningHub.Core => Add => Class => ICourseService.

Right Click on Services in LearningHub.Infra => Add => Class => CourseService.

Note:

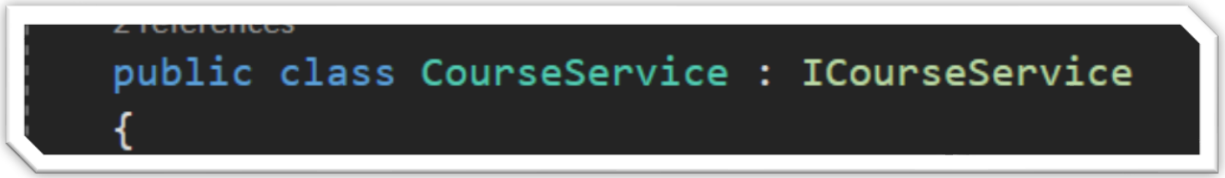
Make sure all created classes and interfaces are public.

In LearningHub.Core => Service => ICourseService add the following abstract methods:

```
List<Course> GetAllCourse();
void CreateCourse(Course course);
void DeleteCourse(int id);
public void UpdateCourse(Course course);
Course GetByCourseId(int id);
```

In LearningHub.Infra => Service => Course Service => make the class inherit the interface ICourseService:

```
public class CourseService : ICourseService
```



```
public class CourseService : ICourseService
{
```

In LearningHub.Infra => Service => Course Service add the following :

```
private readonly ICourseRepository courseRepository;

        public CourseService(ICourseRepository
courseRepository)
        {
            this.courseRepository = courseRepository;
        }
```

In LearningHub.Infra => Service => Course Service add the following :

```
public List<Course> GetAllCourse()
    {
    return courseRepository.GetAllCourse();
    }
```

In LearningHub.Infra => Service => Course Service add the following :

```
public void CreateCourse(Course course)
        {
            courseRepository.CreateCourse(course);
        }
```

In LearningHub.Infra => Service => Course Service add the following :

```
public void UpdateCourse(Course course)
{
                courseRepository.UpdateCourse(course);
               }
```

In LearningHub.Infra => Service => Course Service add the following :

```
public void DeleteCourse(int id)
        {
            courseRepository.DeleteCourse(id);
        }
```

In LearningHub.Infra => Service => Course Service add the following :

```
public Course GetByCourseId(int id)
          {
          return courseRepository.GetByCourseId(id);
          }
```

Add Services in Program

Write the following code in Configure services:

```
builder.Services.AddScoped<ICourseService, CourseService>();
```

- Right Click on Repository Folder in LearningHub.Core => Add => Class => Interface => IStudentService.

- Right Click on Repository Folder in LearningHub.Infra => Add => Class => StudentService.

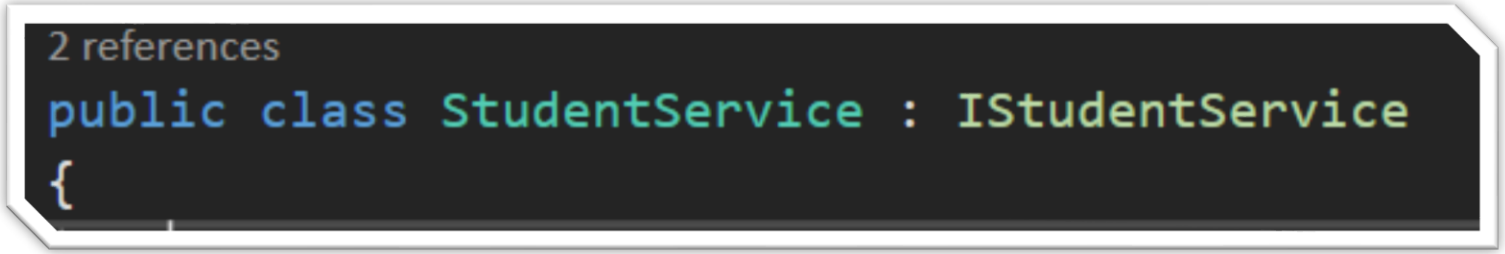
Note:
Make sure all created classes and interfaces are public.

In LearningHub.Core => Service => IStudentService add the following abstract methods:

```
List<Student> GetAllStudent();
void CreateStudent(Student Student);
void UpdateStudent(Student Student);
void DeleteStudent(int id);
Student GetStudentById(int id);
```

In LearningHub.Infra => Service => StudentService => make the class inherit the interface IStudentService:

```
public class StudentService : IStudentService
```



```
2 references
public class StudentService : IStudentService
{
```

In LearningHub.Infra => Service => StudentService add the following :

```
 private readonly IStudentRepository
_studentRepository;

        public StudentService(IStudentRepository
studentRepository)
        {
            _studentRepository = studentRepository;
        }
```

In LearningHub.Infra => Service => StudentService add the following :

```
public List<Student> GetAllStudent()
           {
           return _studentRepository.GetAllStudent();
           }
```

In LearningHub.Infra => Service => StudentService add the following :

```
public void CreateStudent(Student Student)
          {
           _studentRepository.CreateStudent(Student);
          }
```

In LearningHub.Infra => Service => StudentService add the following :

```
public void UpdateStudent(Student Student)
          {
          _studentRepository.UpdateStudent(Student);
          }
```

In LearningHub.Infra => Service => StudentService add the following :

```
public void DeleteStudent(int id)
          {
           _studentRepository.DeleteStudent(id);
          }
```

In LearningHub.Infra => Service => StudentService add the following :

```
 public Student GetStudentById(int id)
           {
           return 
_studentRepository.GetStudentById(id);
           }
```

Add Services in Program

Write the following code in Configure services:

```
builder.Services.AddScoped<IStudentService, StudentService>();
```

- Right Click on Repository Folder in LearningHub.Core => Add => Class => Interface => IStudentCourseService.
- Right Click on Repository Folder in LearningHub.Infra => Add => Class => StudentCourseService.

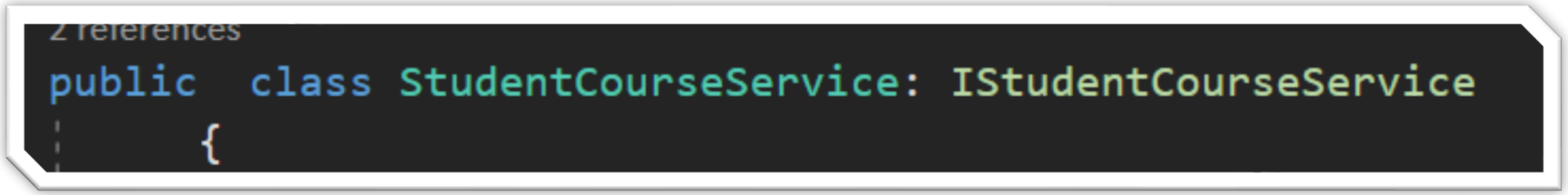
Note:
Make sure all created classes and interfaces are public.

In LearningHub.Core => Service => IStudentCourseService add the following abstract methods:

```
 List<Stdcourse> GetAllStudentCourse();
        void CreateStudentCourse(Stdcourse
studentCourse);
        void DeleteStudentCourse(int id);
        void UpdateStudentCourse(Stdcourse
studentCourse);
        Stdcourse GetStudentCourseById(int id);
```

In LearningHub.Infra => Service => StudentCourseService => make the class inherit the interface IStudentCourseService:

```
public  class StudentCourseService: IStudentCourseService
```



```
2 references
public  class StudentCourseService: IStudentCourseService
        {
```

In LearningHub.Infra => Service => StudentCourseService add the following :

```
 private readonly IStudentCourseRepository
_studentCourseRepository;

       public
StudentCourseService(IStudentCourseRepository
studentCourseRepository)
           {
           _studentCourseRepository =
studentCourseRepository;
           }
```

In LearningHub.Infra => Service => StudentCourseService add the following :

```
 public void CreateStudentCourse(Stdcourse
studentCourse)
            {

_studentCourseRepository.CreateStudentCourse(studentC
ourse);
            }
```

In LearningHub.Infra => Service => StudentCourseService add the following :

```
 public void DeleteStudentCourse(int id)
           {

_studentCourseRepository.DeleteStudentCourse(id);
           }
```

In LearningHub.Infra => Service => StudentCourseService add the following :

```
 public List<Stdcourse> GetAllStudentCourse()
            {
            return
_studentCourseRepository.GetAllStudentCourse();
            }
```

In LearningHub.Infra => Service => StudentCourseService add the following :

```
 public List<Stdcourse> GetAllStudentCourse()
            {
            return 
_studentCourseRepository.GetAllStudentCourse();
            }
```

In LearningHub.Infra => Service => StudentCourseService add the following :

```
 public void UpdateStudentCourse(Stdcourse
studentCourse)
            {

_studentCourseRepository.UpdateStudentCourse(studentCou
rse);
            }
```

In LearningHub.Infra => Service => StudentCourseService add the following :

```
 public Stdcourse GetStudentCourseById(int id)
           {
          return
_studentCourseRepository.GetStudentCourseById(id);
           }
```

Add Services in Program

Write the following code in Configure services:

```
builder.Services.AddScoped<IStudentCourseService, StudentCourseService>();
```

Exercise

- ✓ Create a function to display FirstName and LastName from table student.
- ✓ Create a function to display students by firstName.
- ✓ Create a function to display students by BirthOfDate.
- ✓ Create a function to display a student by BirthOfDate interval.
- ✓ Create a function to display the student name with the highest 3 marks

In LearningHub.Core => Service => IStudentService add the following :

```
List<Student> GetStudentByFName(string name);
        List<Student> GetStudentFNameAndLName();
        List<Student> GetStudentByBirthdate(DateTime
Birth_Date);
        List<Student> GetStudentBetweenDate(DateTime
DateFrom, DateTime DateTo);
        List<Student> GetStudentsWithHighestMarks(int
numOfStudent);
```

In LearningHub.Infra => Service => StudentService add the following :

```
 public List<Student> GetStudentByFName(string name)
            {
           return
_studentRepository.GetStudentByFName(name);
            }
```

In LearningHub.Infra => Service => StudentService add the following :

```
 public List<Student> GetStudentFNameAndLName()
           {
           return 
_studentRepository.GetStudentFNameAndLName();
           }
```

In LearningHub.Infra => Service => StudentService add the following :

```
 public List<Student> GetStudentByBirthdate(DateTime
Birth_Date)
            {
            return
_studentRepository.GetStudentByBirthdate(Birth_Date);
            }
```

The Learning Hub

In LearningHub.Infra => Service => StudentService add the following :

```
 public List<Student> GetStudentBetweenDate(DateTime
DateFrom, DateTime DateTo)
            {
            return
_studentRepository.GetStudentBetweenDate(DateFrom,
DateTo);
            }
```

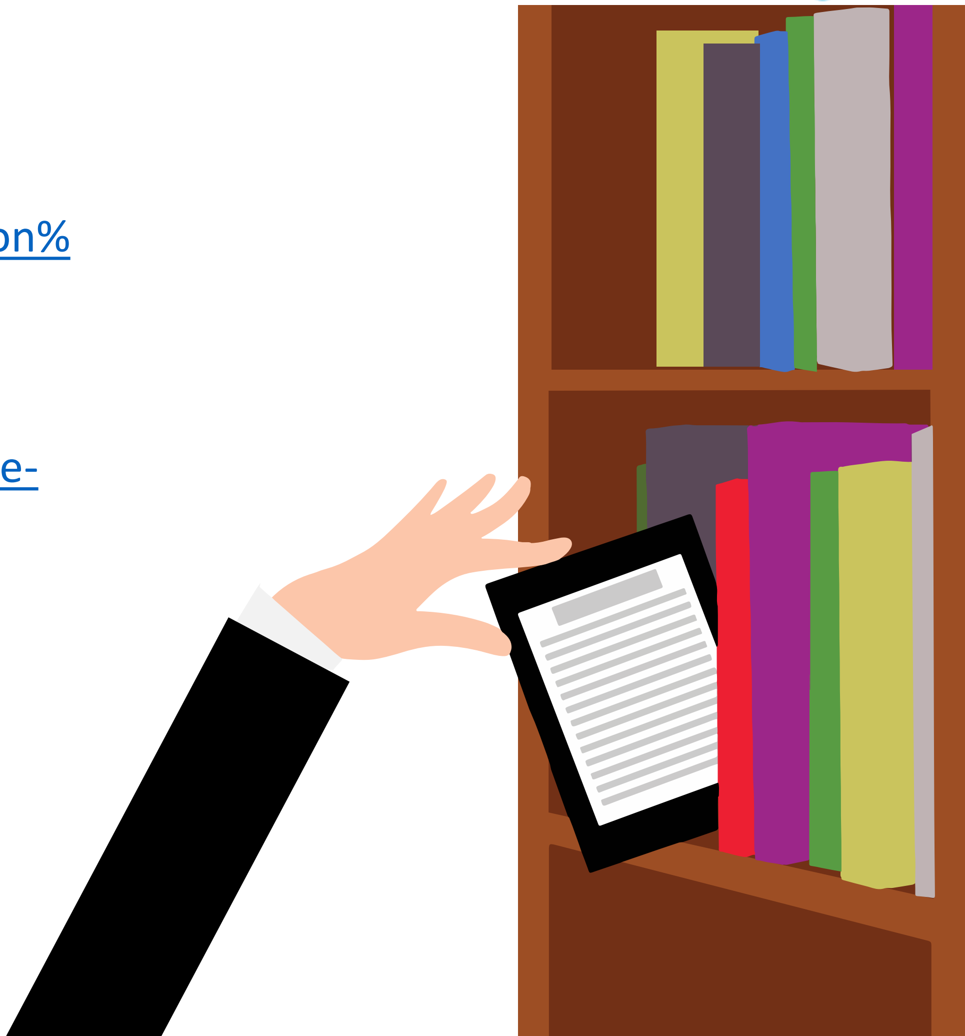
In LearningHub.Infra => Service => StudentService add the following :

```
 public List<Student> GetStudentsWithHighestMarks(int
numOfStudent)
            {
            return
_studentRepository.GetStudentsWithHighestMarks(numOfStu
dent);
            }
```

References

[1]. https://www.codeguru.com/csharp/understanding-onion-architecture/#:~:text=Onion%20Architecture%20is%20based%20on,on%20the%20actual%20domain%20models

[2]. https://docs.microsoft.com/en-us/dotnet/api/microsoft.entityframeworkcore.dbcontext?view=efcore-5.0



The Learning Hub