

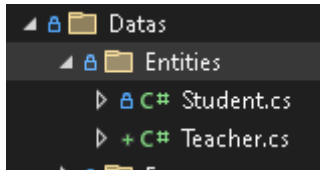
Backend

- [Bevezetés a kliens- és szerveroldali webalkalmazások készítésébe](#)

Tanár adatok elérése a meglévő projektben

Tanár model

A Datas/Entities mappába létrehozuk az új modelt, a Teacher osztályt!



- [What Is GUID In C# \(c-sharpcorner.com\)](#)
- [Working With Guid in C# - Code Maze \(code-maze.com\)](#)

```
public class Teacher
{
    public Teacher()
    {
        Id = Guid.NewGuid();
        FirstName = string.Empty;
        LastName = string.Empty;
        BirthsDay = new DateTime();
        IsWoomen = false;
        IsHeadTeacher = false;
    }

    public Teacher(string firstName, string lastName, DateTime birthsDay, bool isWoomen, bool isHeadTeacher)
    {
        Id = Guid.NewGuid();
        FirstName = firstName;
        LastName = lastName;
        BirthsDay = birthsDay;
        IsWoomen = isWoomen;
        IsHeadTeacher = isHeadTeacher;
    }

    public Teacher(Guid id, string firstName, string lastName, DateTime birthsDay, bool isWoomen, bool
isHeadTeacher)
    {
        Id = id;
        FirstName = firstName;
        LastName = lastName;
        BirthsDay = birthsDay;
        IsWoomen = isWoomen;
        IsHeadTeacher = isHeadTeacher;
    }

    public Guid Id { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public DateTime BirthsDay { get; set; }
    public bool IsWoomen { get; set; }
    public bool IsHeadTeacher { get; set; }

    public override string ToString()
    {
        return $"{LastName} {FirstName}";
    }
}
```

Kréta adatbázis kontextus

A KretaContext adatmodel leképező osztályban felvesszük a tanár adatokat tartalmazó leképező halmazt.

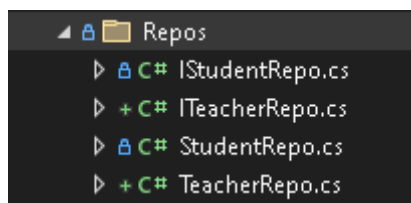
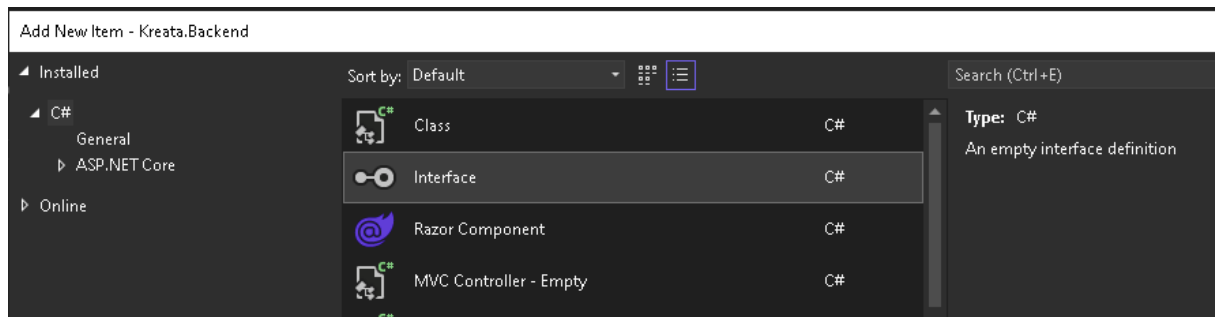
- [DbContext in Entity Framework Core \(entityframeworktutorial.net\)](http://entityframeworktutorial.net)
- [In-memory Database Provider - EF Core | Microsoft Learn](https://learn.microsoft.com/en-us/ef/core/in-memory/)
- [Using MySQL With Entity Framework \(c-sharpcorner.com\)](http://c-sharpcorner.com)

```
public class KretaContext : DbContext
{
    public DbSet<Student> Students { get; set; }
    public DbSet<Teacher> Teachers { get; set; }
}
```

Repository réteg

- [interface – C# referencia - C# | Microsoft Learn](https://learn.microsoft.com/en-us/dotnet/csharp/interfacs/)
- [Interfészek | C# Tutorial.hu \(csharptutorial.hu\)](http://csharptutorial.hu)
- [C# Interface \(w3schools.com\)](http://w3schools.com)

Létrehozzuk a ITeacherRepo interfacet és a TeacherRepo osztályt!

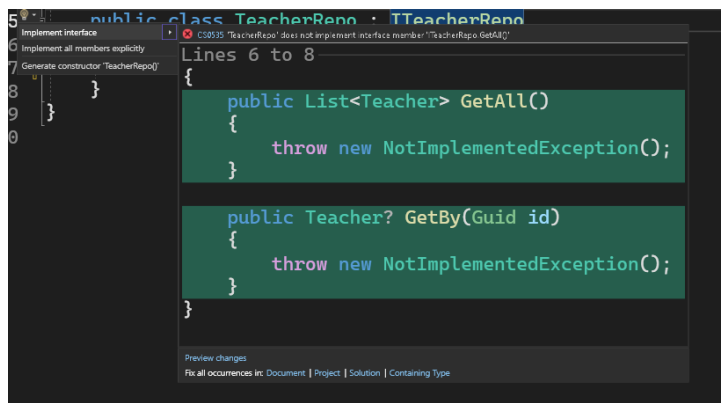


Az ITeacher interface-ben deklaráljuk, hogy a TeacherRepo milyen interfaceket biztosít a kívüllág felé. Azaz megadjuk a külső objektumok milyen metódusokon keresztül tudnak kommunikálni a repository réteggel.

```
public interface ITeacherRepo
{
    Task<List<Teacher>> GetAll();
    Task<Teacher?> GetBy(Guid id);
}
```

A TeacherRepo osztály ettől az interfác-tól öröklődik! Implementáljuk az interfacet!

```
public class TeacherRepo : ITeacherRepo
{
}
```



Implementálás után:

```
public class TeacherRepo : ITeacherRepo
{
    public async Task<List<Teacher>> GetAll()
    {
        throw new NotImplementedException();
    }

    public async Task<Teacher?> GetBy(Guid id)
    {
        throw new NotImplementedException();
    }
}
```

A repository a KretaContext által kezelt Set-eken keresztül éri el az adatbázist, ezért injektálunk (dependency injection) ebbe az osztályba egy KretaContext osztályt!

```
private readonly KretaContext _dbContext;

public TeacherRepo(KretaContext dbContext)
{
    _dbContext = dbContext;
}
```

A GetAll metódus az összes tanár adatát adja vissza:

```
public async Task<List<Teacher>> GetAll()
{
    return await _dbContext.Teachers.ToListAsync();
}
```

A GetBy metódus az id-vel azonos tanár adatait adja vissza, vagy null értékkel tér vissza, ha nem talál.

```
public async Task<Teacher?> GetBy(Guid id)
{
    return await _dbContext.Teachers.FirstOrDefaultAsync(s => s.Id == id);
}
```

Dependency Injection

A KretaBackendExtensions fájlba a ConfigureRepo metódusba lehetővé tesszük a TeacherRepo osztály injektálását:

```
public static void ConfigureRepos(this IServiceCollection services)
{
    services.AddScoped<IStudentRepo, StudentRepo>();
    services.AddScoped<ITeacherRepo, TeacherRepo>();
}
```

Controller elkészítése

Vizsgáljuk meg a StudentControllert:

Annotációkat használunk, hogy bizonyos kódokat ne keljen megírni!

- [ASP.NET Core webes API-vezérlők - Training | Microsoft Learn](#)
- [Webes API létrehozása ASP.NET Core vezérlőkkel - Training | Microsoft Learn](#)
- [2.1 Az API-okról röviden - YouTube](#)
- [Web API Tutorials .Net Core 7 C# - YouTube](#)

```
[ApiController]
[Route("api/[controller]")]
public class StudentController : ControllerBase
{
    private IStudentRepo _studentRepo;

    public StudentController(IStudentRepo studentRepo)
    {
        _studentRepo = studentRepo;
    }

    [HttpGet]
```

```

public async Task<IActionResult> SelectAllRecordToListAsync()
{
    List<Student>? users = new();

    if (_studentRepo != null)
    {
        users = await _studentRepo.GetAll();
        return Ok(users);
    }
    return BadRequest("Az adatok elérhetetlenek!");
}

[HttpGet("{id}")]
public async Task<IActionResult> GetBy(Guid id)
{
    Student? entity = new();
    if (_studentRepo is not null)
    {
        entity = await _studentRepo.GetBy(id);
        if (entity != null)
            return Ok(entity);
    }
    return BadRequest("Az adatok elérhetetlenek!");
}
}

```

Készítsük el a Teacher controllert:

```

[ApiController]
[Route("api/[controller]")]
public class TeacherController : ControllerBase
{
    private ITeacherRepo _teacherRepo;

    public TeacherController(ITeacherRepo teacherRepo)
    {
        _teacherRepo = teacherRepo;
    }

    [HttpGet("{id}")]
    public async Task<IActionResult> GetBy(Guid id)
    {
        Teacher? entity = new();
        if (_teacherRepo is not null)
        {
            entity = await _teacherRepo.GetBy(id);
            if (entity != null)
                return Ok(entity);
        }
        return BadRequest("Az adatok elérhetetlenek!");
    }

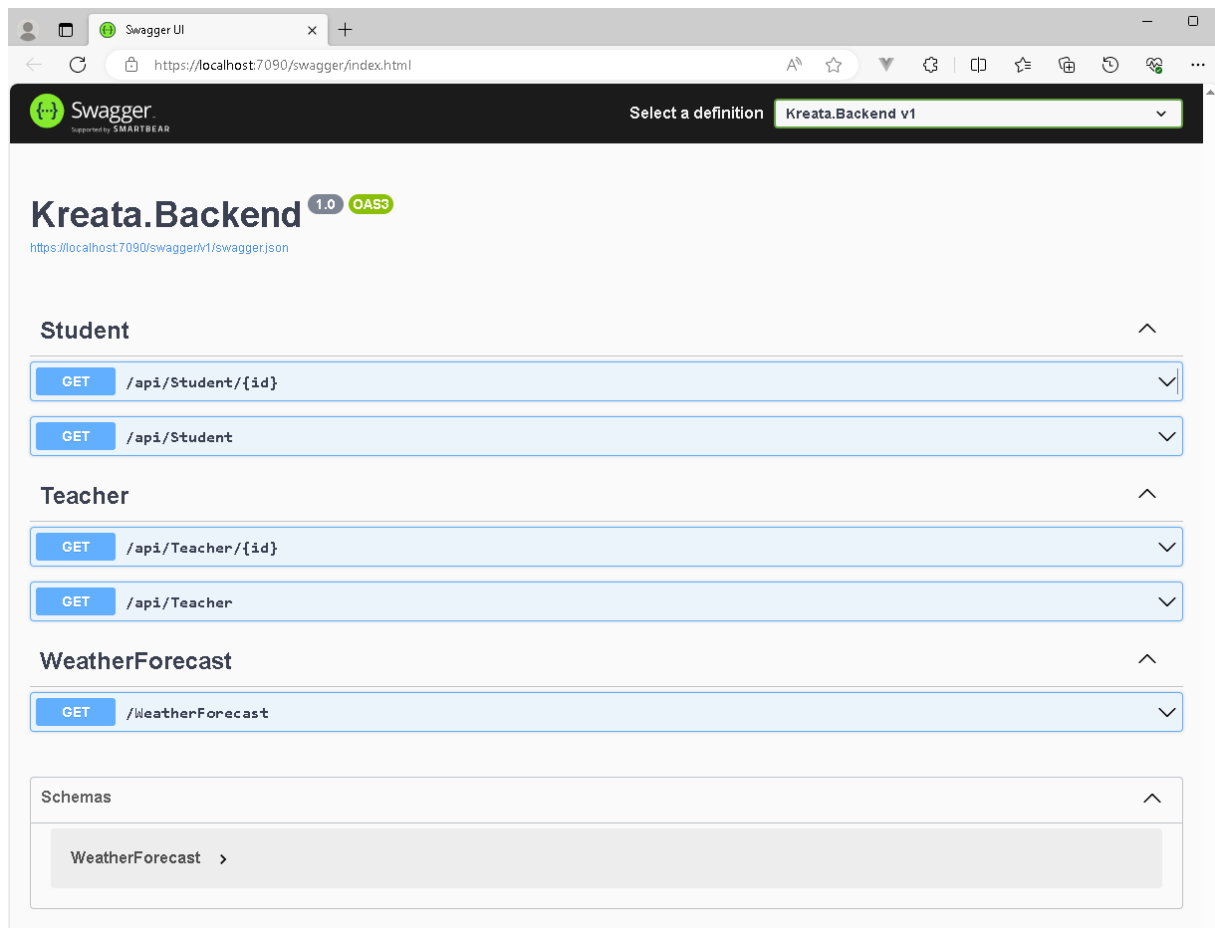
    [HttpGet]
    public async Task<IActionResult> SelectAllRecordToListAsync()
    {
        List<Teacher>? users = new();

        if (_teacherRepo != null)
        {
            users = await _teacherRepo.GetAll();
            return Ok(users);
        }
        return BadRequest("Az adatok elérhetetlenek!");
    }
}

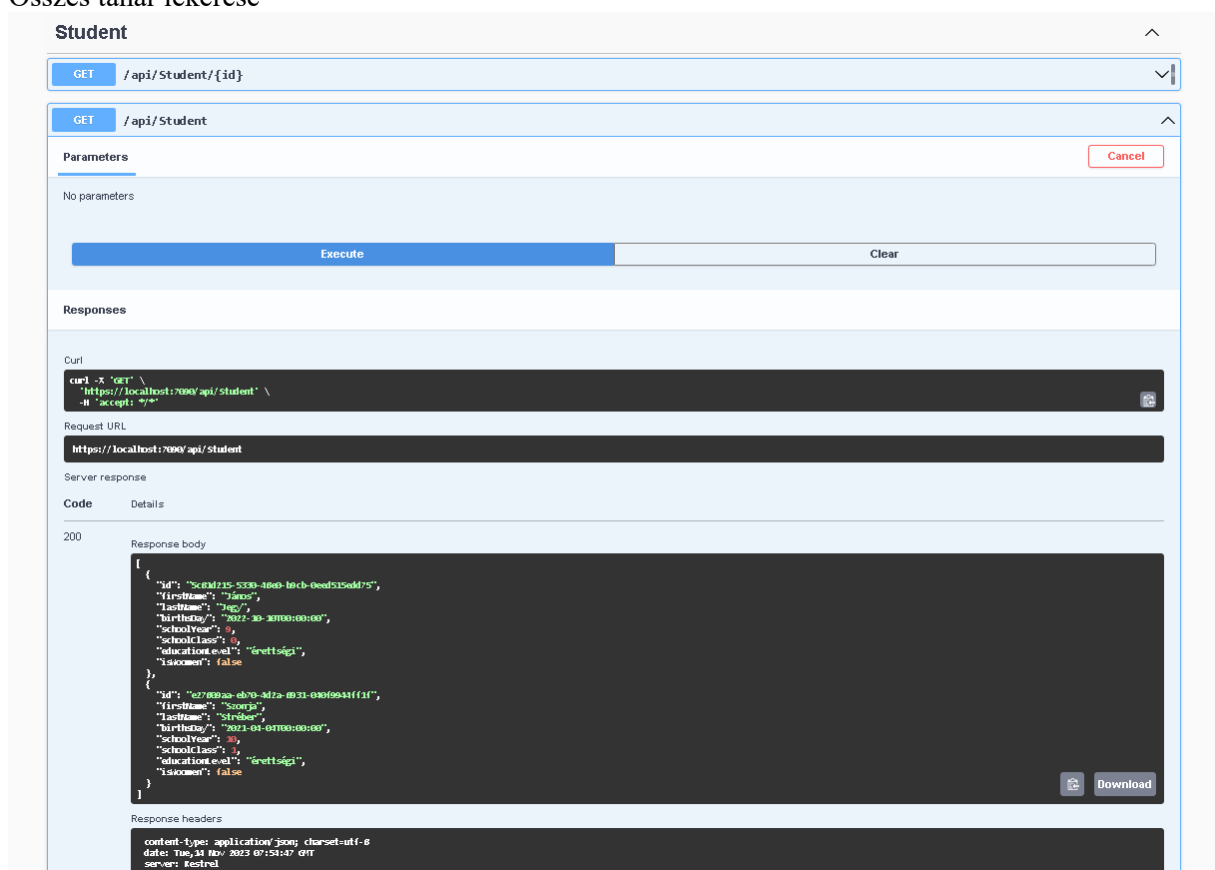
```

Tesztelés

Backend nyitó képernyő:



Összes tanár lekérése



Az id megadása után egy adott tanár adatainak lekérdezése:

GET

/api/Student/{id}

Parameters

Cancel

Name	Description
id <small>* required</small> string(\$uuid) (auth)	e27809aa-e670-4d2a-8931-040f9944f1f1

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  https://localhost:7090/api/Student/e27809aa-e670-4d2a-8931-040f9944f1f1 \
  -H 'accept: */*'

```

Request URL

https://localhost:7090/api/Student/e27809aa-e670-4d2a-8931-040f9944f1f1

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "id": "e27809aa-e670-4d2a-8931-040f9944f1f1", "firstname": "Sörpö", "lastname": "Ströber", "birthday": "2021-04-07T00:00:00", "schoolyear": 40, "schoolclass": 1, "education.level": "főretöltés", "isowner": false }</pre></div><div><div>Download</div></div></div> <div><div>Response headers</div><div><pre>content-type: application/json; charset=utf-8 date: Tue, 14 Nov 2023 07:55:39 GMT server: Restful </pre></div></div>

Responses

Code	Description	Links
200	Success	No links