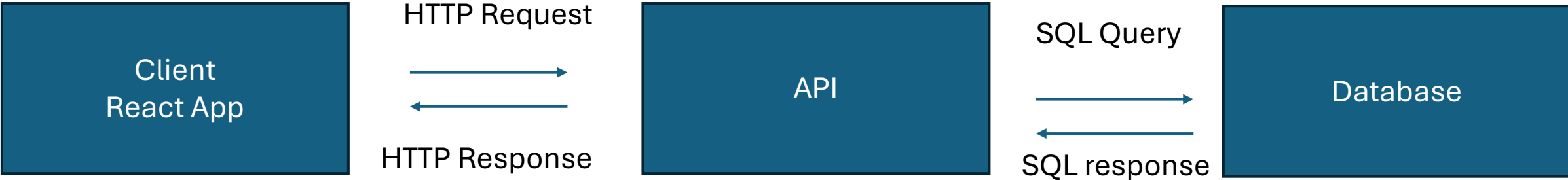


ASP.NET Web APIs



ASP.NET Core Web API

פרויקט חדש – סוג הפרוייקט:



ASP.NET Core Web API

A project template for creating a RESTful Web API using ASP.NET Core controllers or minimal APIs, with optio

C#

Linux

macOS

Windows

API

Cloud

Service

Web

Web API

Additional information

ASP.NET Core Web API

C#

Linux

macOS

Windows

API

Cloud

Service

Web

Web API

Framework [i](#)

.NET 8.0 (Long Term Support)

Authentication type [i](#)

None

☒ Configure for HTTPS [i](#)

☐ Enable container support [i](#)

Container OS [i](#)

Linux

Container build type [i](#)

Dockerfile

☒ Enable OpenAPI support [i](#)

☒ Do not use top-level statements [i](#)

☒ Use controllers [i](#)

☐ Enlist in .NET Aspire orchestration [i](#)

אם זה לא מותקן אצלכם עדיין – אז לרדת עד למטה בסוף
רשימת האפשרויות – עד שנגיע ל"התקן עוד כלים ואפשרויות"



Windows Forms Class Library

A project template for creating a class library that targets .NET Windows Forms (WinForms).

C#

Windows

Desktop

Library



Windows Forms Control Library

A project template for creating a control library that targets .NET Windows Forms (WinForms).

C#

Windows

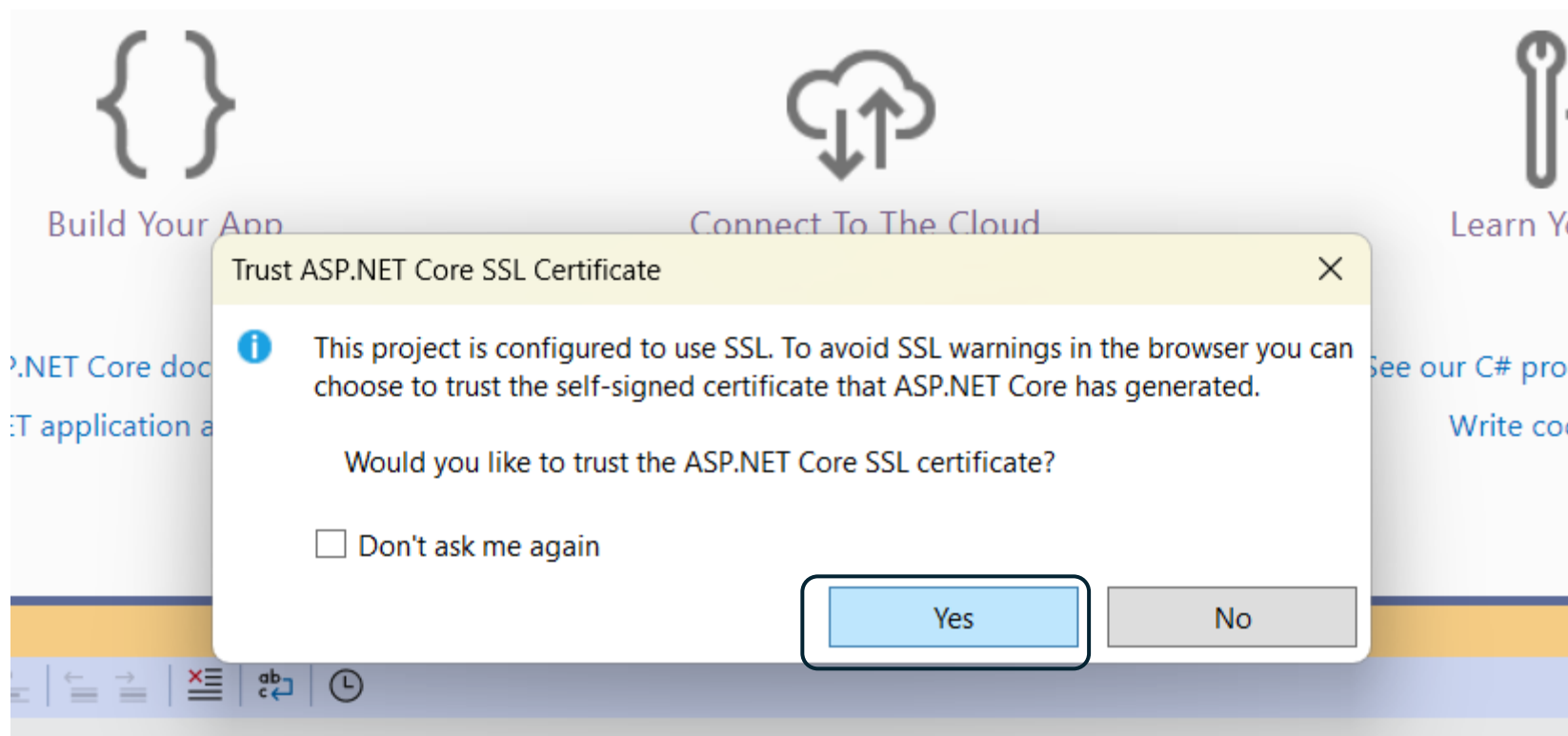
Desktop

Library

Not finding what you're looking for?

[Install more tools and features](#)

הרצה של הפרוייקט לדוגמא



Security Warning



You are about to install a certificate from a certification authority (CA) claiming to represent:

localhost

Windows cannot validate that the certificate is actually from "localhost". You should confirm its origin by contacting "localhost". The following number will assist you in this process:

Thumbprint (sha1): 41269921 4AD644F5 5C7E55BE 94E87020
19C2B89C

Warning:

If you install this root certificate, Windows will automatically trust any certificate issued by this CA. Installing a certificate with an unconfirmed thumbprint is a security risk. If you click "Yes" you acknowledge this risk.

Do you want to install this certificate?

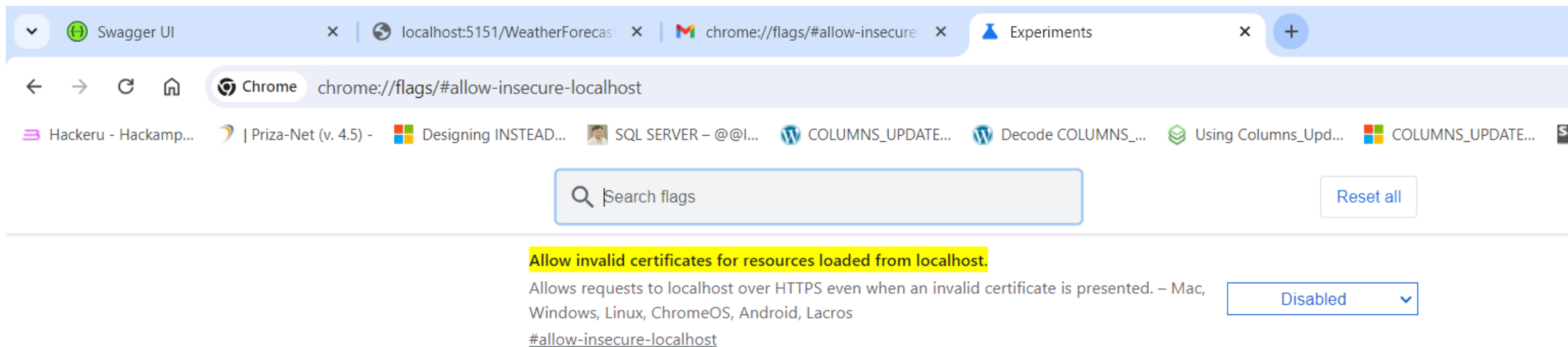
Yes

No

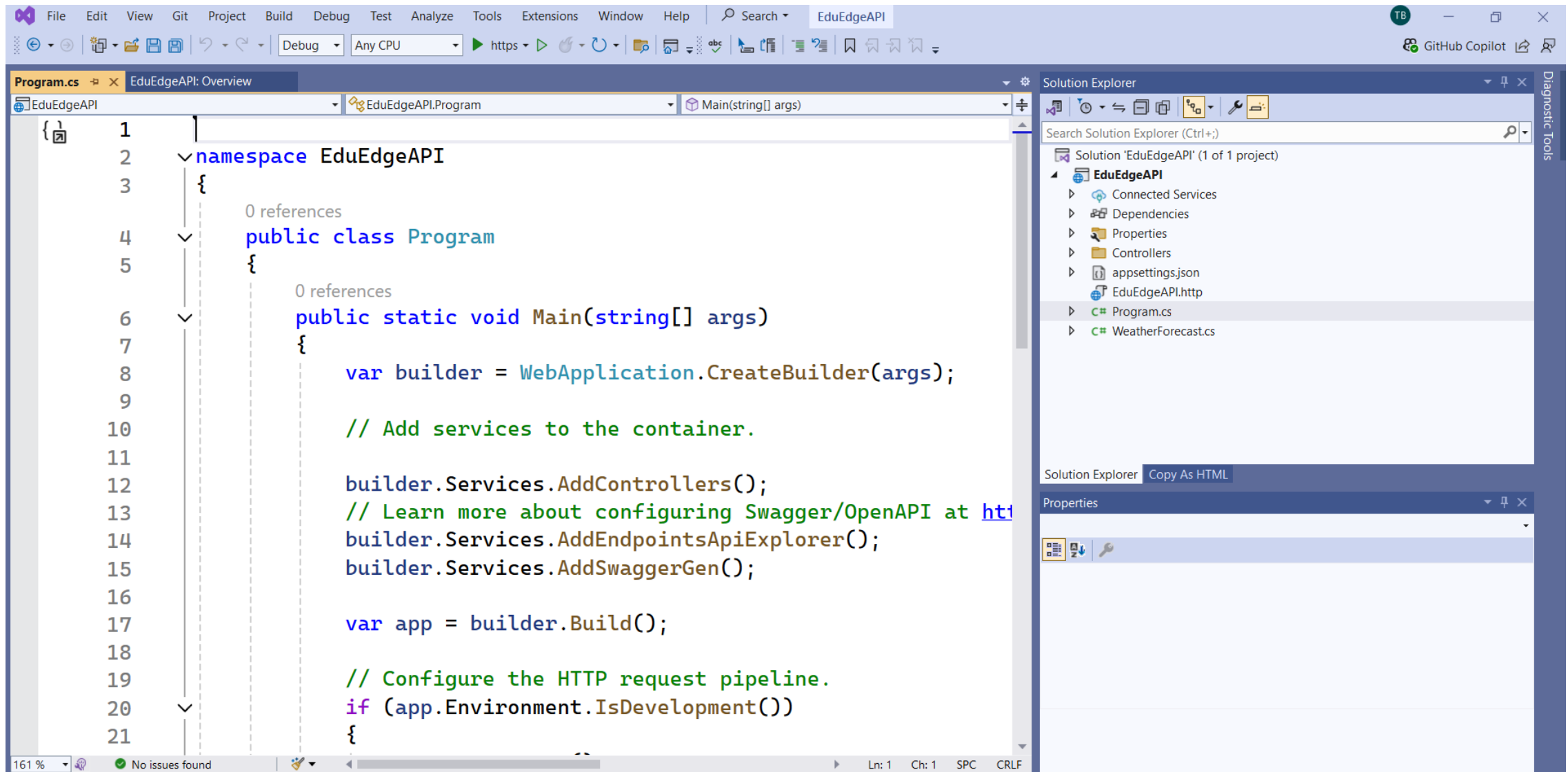
אם הדפדפן לא בוטח בתעודת ה ssl

נכנס להגדרות של הדפדפן בכתובת הבאה:

chrome://flags/#allow-insecure-localhost



הקובץ הראשי - Program.cs



The screenshot displays the Visual Studio IDE with the **Program.cs** file open in the **EduEdgeAPI** project. The code is as follows:

```
1 namespace EduEdgeAPI
2 {
3     0 references
4     public class Program
5     {
6         0 references
7         public static void Main(string[] args)
8         {
9             var builder = WebApplication.CreateBuilder(args);
10
11             // Add services to the container.
12
13             builder.Services.AddControllers();
14             // Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swash
15             builder.Services.AddEndpointsApiExplorer();
16             builder.Services.AddSwaggerGen();
17
18             var app = builder.Build();
19
20             // Configure the HTTP request pipeline.
21             if (app.Environment.IsDevelopment())
22             {
23             }
```

The **Solution Explorer** on the right shows the project structure for **EduEdgeAPI**, including **Program.cs** and **WeatherForecast.cs**. The **Properties** window is also visible at the bottom right.

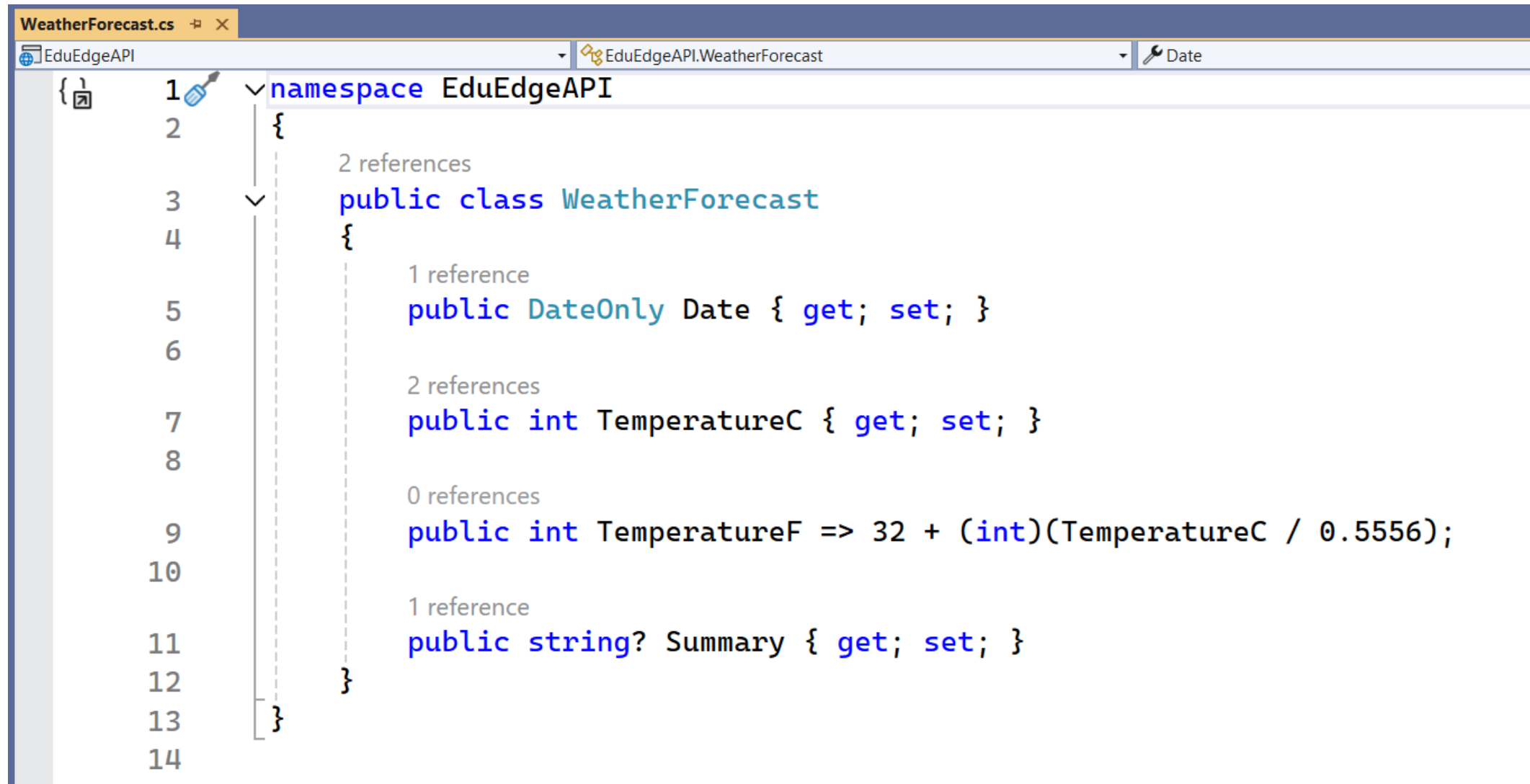
קובץ הגדרות

```
appsettings.json
Schema: https://json.schemastore.org/appsettings.json
1 {
2   "Logging": {
3     "LogLevel": {
4       "Default": "Information",
5       "Microsoft.AspNetCore": "Warning"
6     }
7   },
8   "AllowedHosts": "*"
9 }
10
```

הגדרות ללוגים

הגדרות לכתובות IP

מחלקת מודל (דאטה-קלאס) לפרוייקט לדוגמא:



```
1 namespace EduEdgeAPI
2 {
3     2 references
4     public class WeatherForecast
5     {
6         1 reference
7         public DateOnly Date { get; set; }
8
9         2 references
10        public int TemperatureC { get; set; }
11
12        0 references
13        public int TemperatureF => 32 + (int)(TemperatureC / 0.5556);
14
15        1 reference
16        public string? Summary { get; set; }
17    }
18 }
```

הקובץ החשוב ביותר הוא ה Controller

```
namespace EduEdgeAPI.Controllers
{
    [ApiController]
    [Route("[controller]")]
    3 references
    public class WeatherForecastController : ControllerBase
    {
        private static readonly string[] Summaries = new[]
        {
            "Freezing", "Bracing", "Chilly", "Cool", "Mild", "Warm", "Balmy", "Hot"
        };

        private readonly ILogger<WeatherForecastController> _logger;

        0 references
        public WeatherForecastController(ILogger<WeatherForecastController> logger)
        {
            _logger = logger;
        }
    }
}
```

הקובץ החשוב ביותר הוא ה Controller

```
[HttpGet(Name = "GetWeatherForecast")]
```

0 references

```
public IEnumerable<WeatherForecast> Get()
```

```
{
```

```
    return Enumerable.Range(1, 5).Select(index => new WeatherForecast
```

```
{
```

```
        Date = DateOnly.FromDateTime(DateTime.Now.AddDays(index)),
```

```
        TemperatureC = Random.Shared.Next(-20, 55),
```

```
        Summary = Summaries[Random.Shared.Next(Summaries.Length)]
```

```
    })
```

```
    .ToArray();
```

```
}
```

```
}
```

```
}
```

בקיצור – לולאה שרצה 5 פעמים
יוצרת מערך של אובייקטים

זאת תהיה התגובה של ה Controller הזה
לבקשת HTTP GET

The Controllers Base URL

הכתובת הראשית של הקונטרולר
בכתובת הזאת כל הבקשות יתחילו
המילה [controller] בתוך המחרוזת

מייצגת את שם הקונטרולר עצמו

```
namespace EduEdgeAPI.Controllers
{
    [ApiController]
    [Route("[controller]")]
    3 references
    public class WeatherForecastController : ControllerBase
    {
        private static readonly string[] Summaries =
        [
            "Freezing", "Bracing", "Chilly", "Cool", "Mild",
        ];
    }
}
```

```
[ApiController]
[Route("/api/v1/[controller]")]
```

המחלקה יורשת מ ControllerBase

namespace EduEdgeAPI.Controllers

{

[ApiController]

[Route("[controller]")]

3 references

public class WeatherForecastController : ControllerBase

{

private static readonly string[] Summaries =

[

"Freezing", "Bracing", "Chilly", "Cool", "Mild",

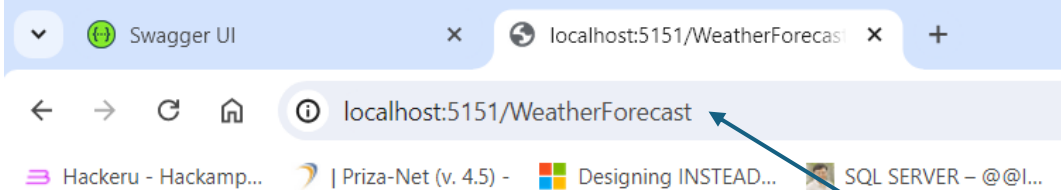
];

יורשת מ ControllerBase



[ApiController]

[Route("/api/v1/[controller]")]

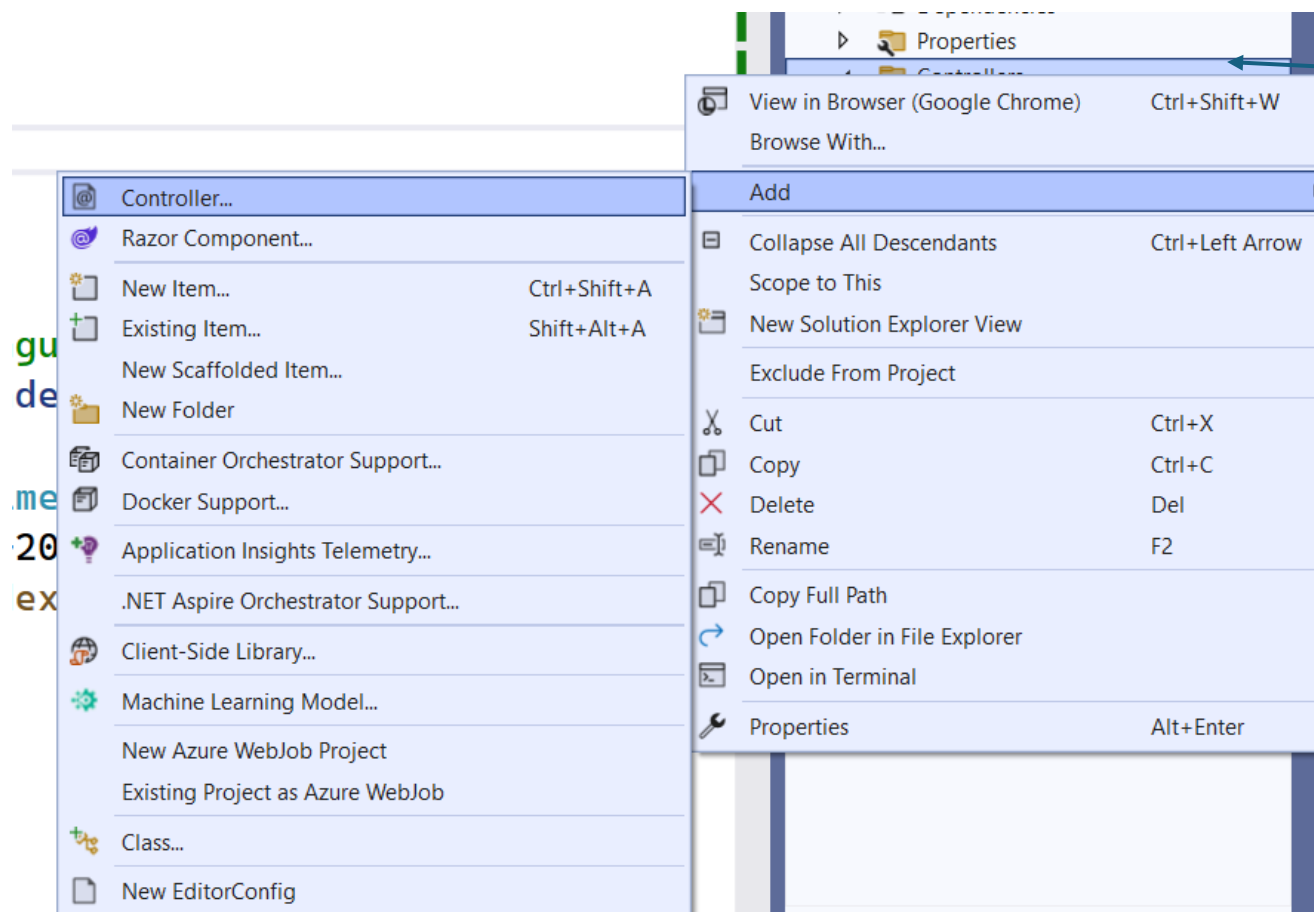


הכתובת של הקונטרולר וסוג הבקשה היא בקשת HTTP GET

```
[
  {
    "date": "2024-07-11",
    "temperatureC": -10,
    "temperatureF": 15,
    "summary": "Chilly"
  },
  {
    "date": "2024-07-12",
    "temperatureC": 32,
    "temperatureF": 89,
    "summary": "Sweltering"
  },
  {
    "date": "2024-07-13",
    "temperatureC": 3,
    "temperatureF": 37,
    "summary": "Bracing"
  },
  {
    "date": "2024-07-14",
    "temperatureC": 7,
    "temperatureF": 44,
    "summary": "Freezing"
  },
  {
    "date": "2024-07-15",
    "temperatureC": -18,
    "temperatureF": 0,
    "summary": "Freezing"
  }
]
```

```
1 // 20240710162135
2 // http://localhost:5151/WeatherForecast
3
```

יצירת Controller משלנו:

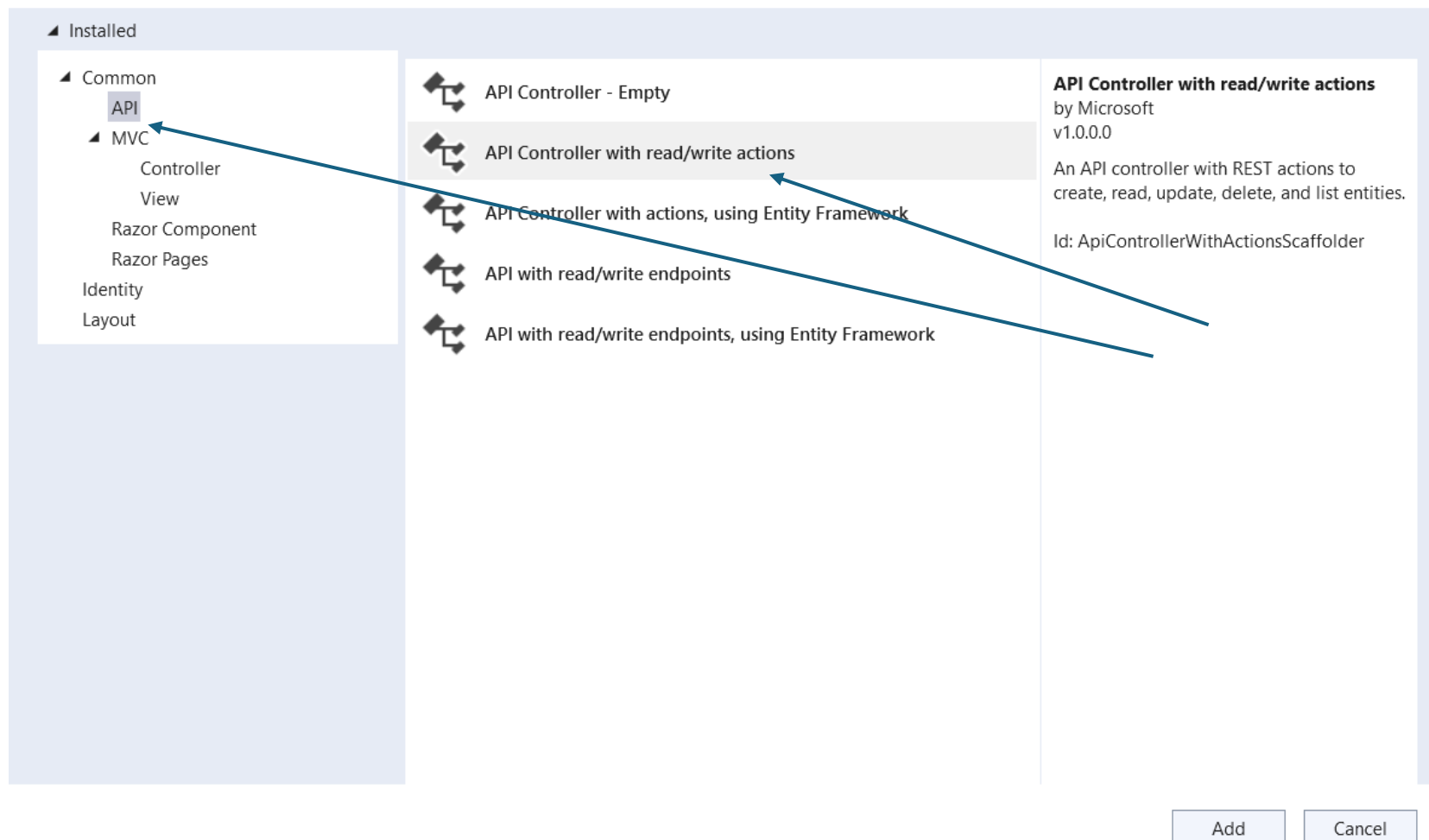


שימו לב

שאתם עושים קליק ימני
על התיקיה של הקונטרולרים
ולא על הפרוייקט

יצירת Controller משלנו:

Add New Scaffolded Item



שימו לב לסמן פה
API
אנחנו לא צריכים View

אם לא נסמן זה ייצור
Controller
ולא BaseController

בעמוד הבא ישאלו
לשם הקובץ – נקרא לו
CustomersController

זה ה- Controller שנוצר CustomersController

```
namespace EduEdgeAPI.Controllers  
{
```

```
    [Route("api/[controller]")]  
    [ApiController]
```

0 references

```
    public class CustomersController : ControllerBase  
    {
```

```
        // GET: api/<CustomersController>  
        [HttpGet]
```

0 references

```
        public IEnumerable<string> Get()  
        {
```

```
            return new string[] { "value1", "value2" };  
        }
```

```
        // GET api/<CustomersController>/5  
        [HttpGet("{id}")]
```

0 references

```
        public string Get(int id)  
        {  
            return "value";  
        }
```

העמסת מתודות
ויש פה
Path variable

זה ה- Controller שנוצר CustomersController

```
// POST api/<CustomersController>  
[HttpPost]
```

0 references

```
public void Post([FromBody] string value)  
{  
}
```

פשוט בקשת פוסט
הנתיב זה הנתיב של הקונטרולר

```
// PUT api/<CustomersController>/5  
[HttpPut("{id}")]
```

0 references

```
public void Put(int id, [FromBody] string value)  
{  
}
```

בקשת PUT
עם Path variable

וערך של ה Body

```
// DELETE api/<CustomersController>/5  
[HttpDelete("{id}")]
```

0 references

```
public void Delete(int id)  
{  
}
```

בקשת DELETE
עם Path variable

כמובן בלי ערך

```
}
```

IActionResult

החזרת סטטוס בתגובה לבקשת Http

```
[HttpPost]
public IActionResult Post(Person p)
{
    people.Add(p);
    p.Id = Guid.NewGuid();
    return Created();
}
```

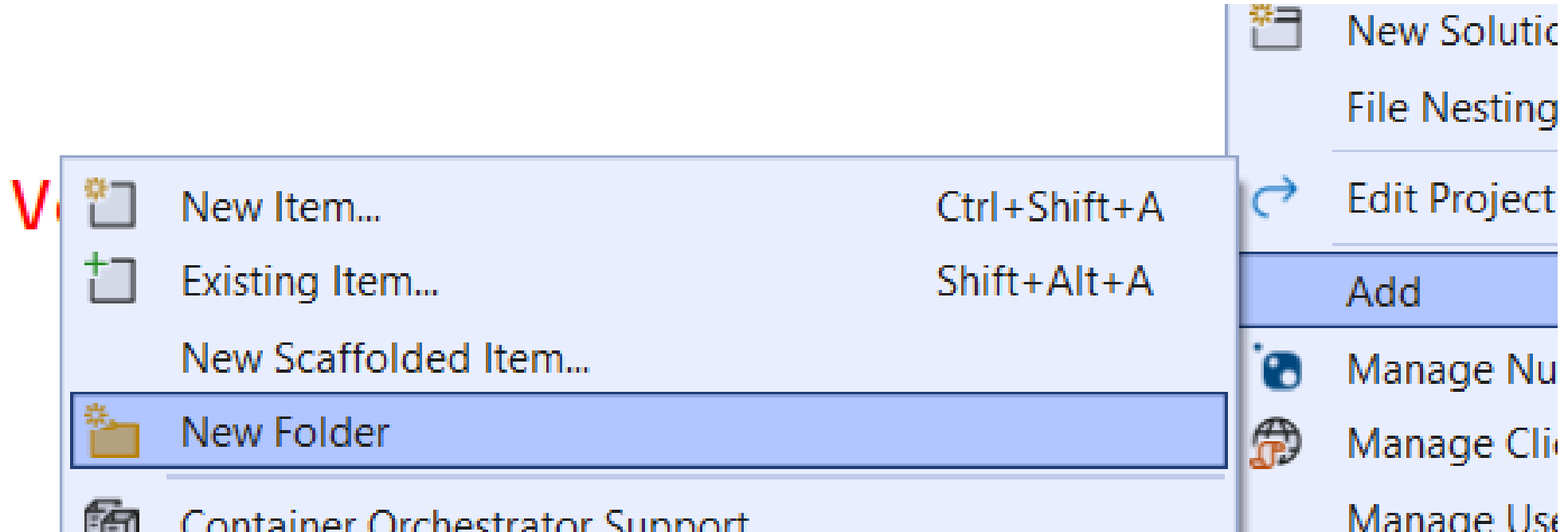
עוד אפשרויות

Nội Gộ ð Bắ ð Rê ð u ê ş ę Cs ê ắ ę ð ê Ô l Nội Cộ ð ę ę ð ę ş ę ắ ę ð u ş Cộ ð ê _ . _ ð ê s ê î ş ę ð ê cộ ð ÿ

ניצור מודל ללקוחות בתוך תיקית Models (ניצור תיקיה ומודל)

1

שם התיקיה - Models



בתוך התיקיה ניצור מחלקה בשם Customers

המחלקה Customer

```
public class Customer()
{
    5 references
    public string? Id { get; set; }
    3 references
    public string Name { get; set; } = string.Empty; //=""

    0 references
    public Customer(string name, string? id = null) : this()
    {
        Id = id ?? new Guid().ToString();
        Name = name;
    }

    0 references
    public Customer(Customer customer) : this()
    {
        ArgumentNullException.ThrowIfNull(customer);
        Id = customer.Id ?? Guid.NewGuid().ToString();
        Name = customer.Name;
    }
}
```

הקוד בקונטרולר – בקשות GET/GET by ID

```
public class CustomersController : ControllerBase
{
    private static List<Customer> Customers = [];
    // GET: api/<CustomersController>
    [HttpGet]
    public ActionResult Get() => Ok(Customers);
```

משתנה סטטי – חובה כי אחרת הערך ידרס
בכל בקשה
מאותחל Controller

```
[HttpGet("{id}")]
public IActionResult Get(string id)
{
    var c = Customers.SingleOrDefault(c => c.Id == id);
    if (c == null)
    {
        return NotFound();
    }
    return Ok(c);
}
```

הטיפול המוחזר
ActionResult
מאפשר לנו להחזיר תשובה
סמנטית מה API
במילים אחרות לשלוח
בסטטוס

הקוד בקונטרולר – בקשת PUT

```
// PUT api/<CustomersController>/5
[HttpPut("{id}")]
public IActionResult Put(string id, [FromBody] Customer customer)
{
    var current = (from c in Customers
                    where c.Id == id
                    select c).SingleOrDefault();

    if (current == null)
    {
        return NotFound();
    }

    current.Name = customer.Name;
    //copy all other props... keep the id intact.

    //we mutated the object in the list, so no further action is needed
    return Ok(current);
}
```


שיעורי בית

צרו פרוייקט חדש מסוג ASP.NET Core Web API

צרו תיקיה לModels שלכם

צרו מחלקת מודל חדשה – Movie

תכונות:

Id, title, description

צרו Controller חדש בשם MovieController

ממשו את הפעולות בController על בסיס רשימה כמו שעשינו בכיתה.

*ממשו את הפעולות בController על בסיס מסד נתונים SQL

*ממשו את הפעולות בController על בסיס מסד נתונים MongoDB

אפשר להתייחס לפעולות בController – Actions כמתודה שמופעלת ברגע שהמשתמש שולח בקשה בדומה למתודה שהופעלה בלחיצת כפתור בWPF

(לגבי Mongo – נפתור את זה יחד בשיעור הבא – כדי שנוכל להמשיך משם את הפרוייקט – אבל כדאי ומומלץ לנסות לפתור בעצמכם וכמובן שמוזמנים לשאול שאלות בקבוצה. עבודה נעימה.