Code Academy

Lecturer

Rokas Slaboševičius

.NET API Project

Data

Today you will learn

01 What is an API?

02 REST APi

03 Project structure

04 HTTP requests

Prerequisites

Postman (that's it)

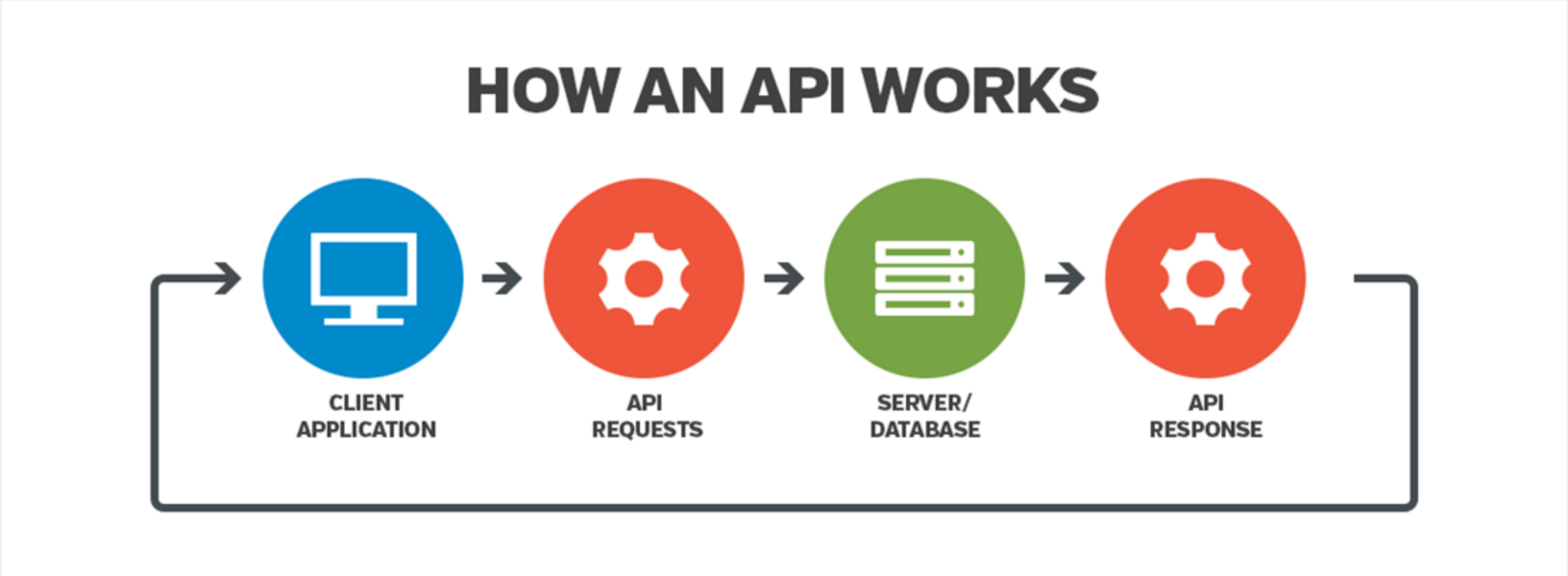
https://aws.amazon.com/what-is/api/

What is an API?

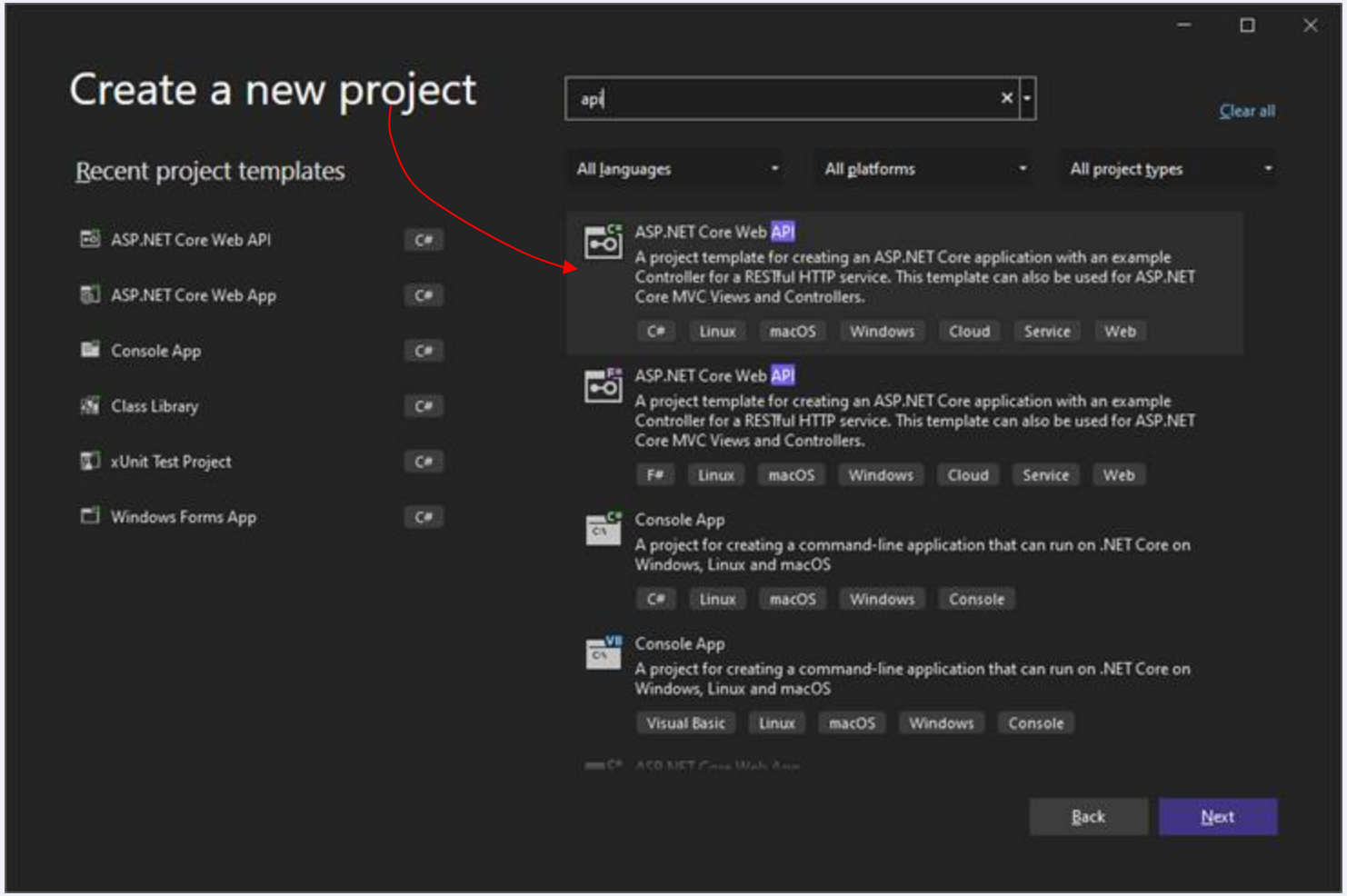
APIs are mechanisms that enable two software components to communicate with each other using a set of definitions and protocols. For example, the weather bureau's software system contains daily weather data. The weather app on your phone "talks" to this system via APIs and shows you daily weather updates on your phone.

https://aws.amazon.com/what-is/api/

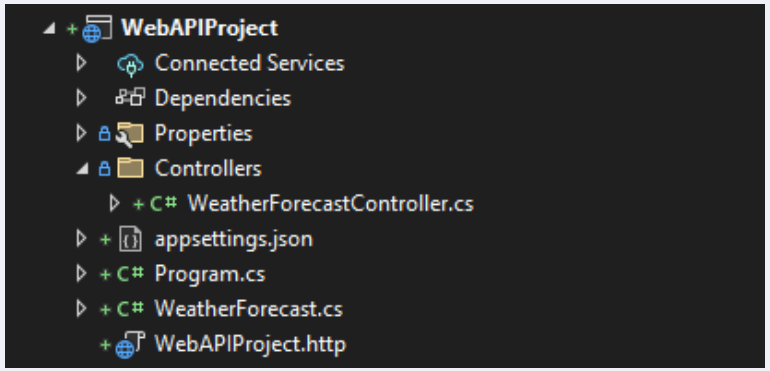
What is an API?



Creating an API project

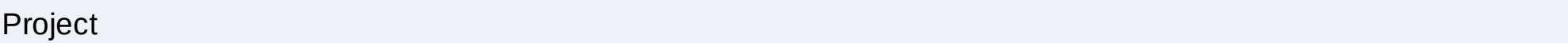


Template structure



WeatherForecastController

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

        private readonly ILogger<WeatherForecastController> _logger;

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

        [HttpGet]
        public IEnumerable<WeatherForecast> Get()
        {
            var rng = new Random();
            return Enumerable.Range(1, 5).Select(index => new WeatherForecast
            {
                Date = DateTime.Now.AddDays(index),
                TemperatureC = rng.Next(-20, 55),
                Summary = Summaries[rng.Next(Summaries.Length)]
            })
            .ToArray();
        }
    }
}
```



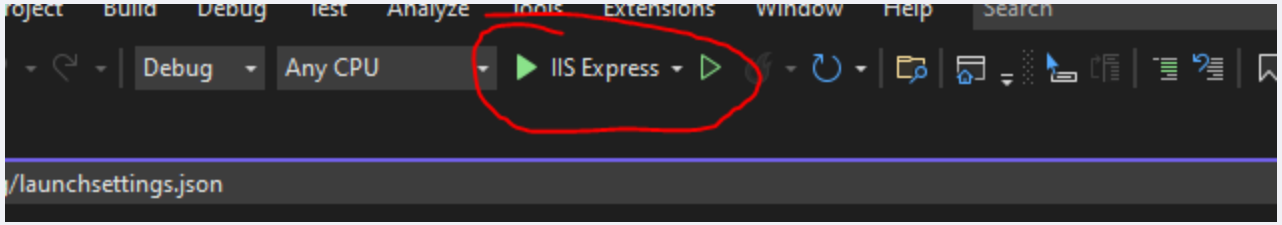
appsettings.json

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  }
}
```

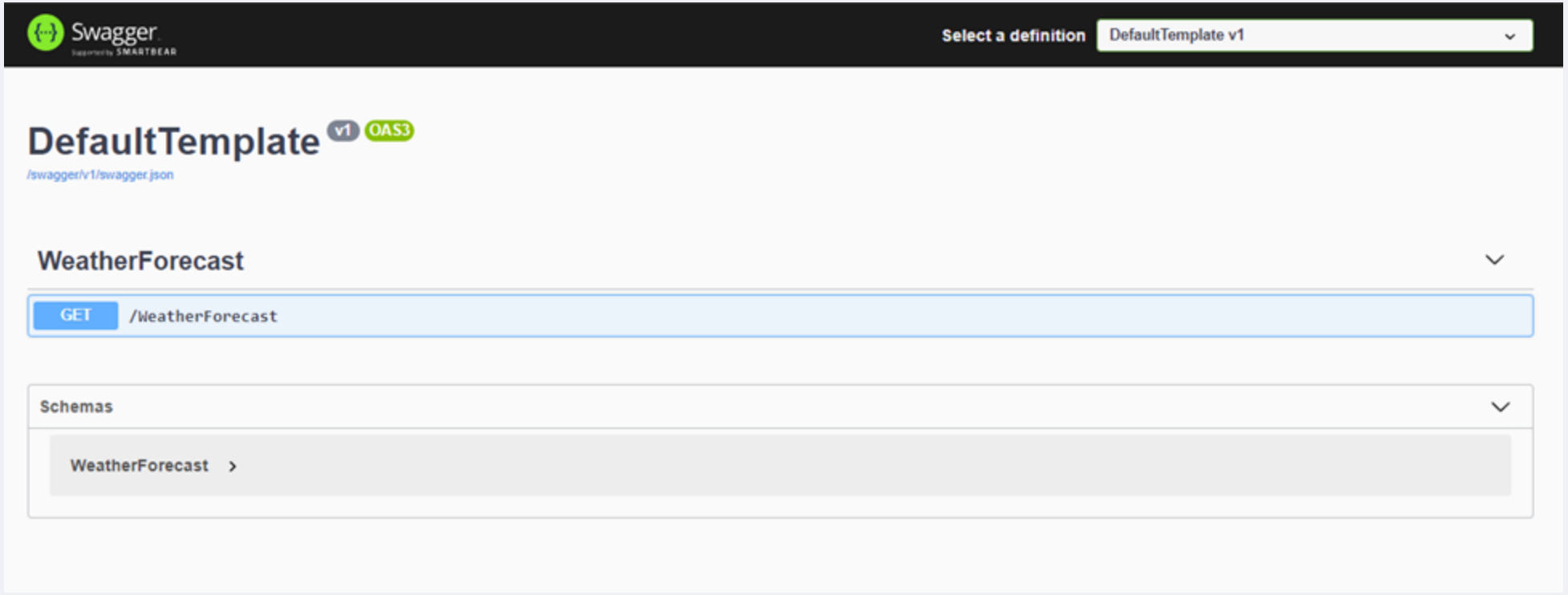
Startup.cs
(don't be scared)

```
0 references
public class Program
{
    0 references
    public static void Main(string[] args)
    {
        var builder = WebApplication.CreateBuilder(args);

        // Add services to the container.

        builder.Services.AddControllers();
        // Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle
        builder.Services.AddEndpointsApiExplorer();
        builder.Services.AddSwaggerGen();

        var app = builder.Build();

        // Configure the HTTP request pipeline.
        if (app.Environment.IsDevelopment())
        {
            app.UseSwagger();
            app.UseSwaggerUI();
        }

        app.UseHttpsRedirection();

        app.UseAuthorization();


        app.MapControllers();

        app.Run();
    }
}
```

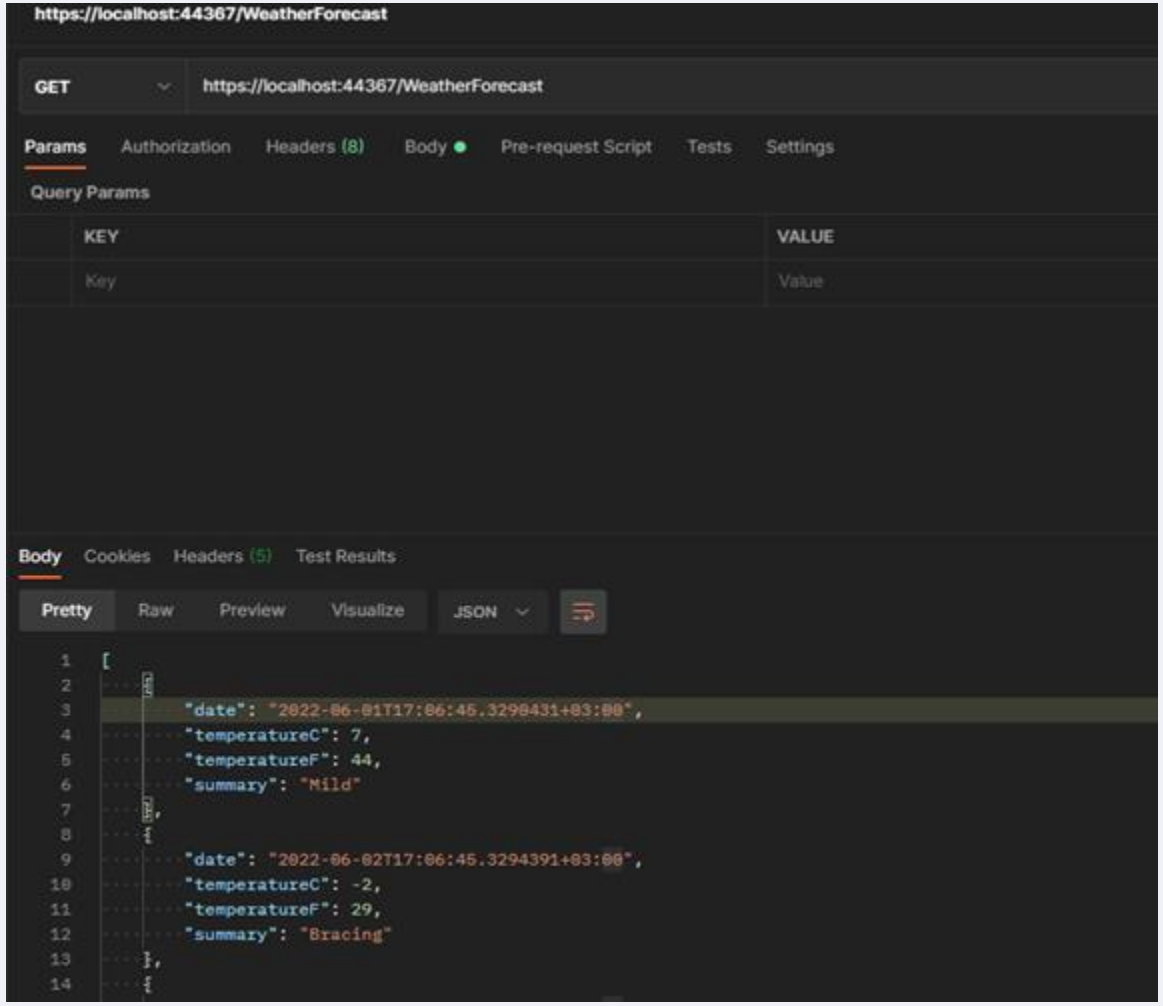
Starting a project



Swashbuckle nuget gives us the following image



We can see the same with Postman



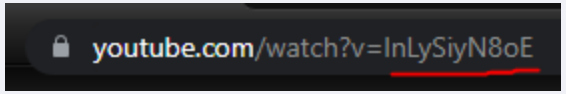
What is a REST service?

Terms:

1. Client - a client is a person/application using our API. The client sends an **HTTP** request to retrieve, save or update information.
2. A resource is information that the API can provide to a customer, Facebook resources would be customers, photos, etc.
3. Server - the server is the place where the API calls to get/update resources for the client.

More on the theoretical side: https://www.redhat.com/en/topics/api/what-is-a-rest-api

What are HTTP requests?



Basic HTTP requests:

1. GET - For receiving data from the server, to refine the request it can "bring" a parameter with it via **url**
2. POST - For saving new data on the server. The biggest difference is that the data is carried in the body and with SSL protections this is encrypted, so the information is much more securely transported.
3. PUT - For updating data. Technologically, its structure is the same as POST, but according to REST principles, a PUT request must be **idemptotent** - this means that if you send a request more than once, the result must not change after the first time.
4. Delete - For deleting an entry, body can be used but is not recommended.

Each HTTP request also carries with it a Header section

More on the theoretical side: https://www.redhat.com/en/topics/api/what-is-a-rest-api

What is a Controller?

Controller is a class where we describe the **endpoints** to which **HTTP** requests will come.

E.g. WeatherForecast Get() controller'is with endpoint 'https://localhost:44367/WeatherForecast'

Questions

.NET API Project

Task 1

- Create a simple web page that will send GET PUT POST DELETE request objects to the new Controller you created
- The controller will have a List of the objects you have created and will operate according to the type of the received request