DS3103 Webutvikling

Rolando Gonzalez
2022

.NET Web API

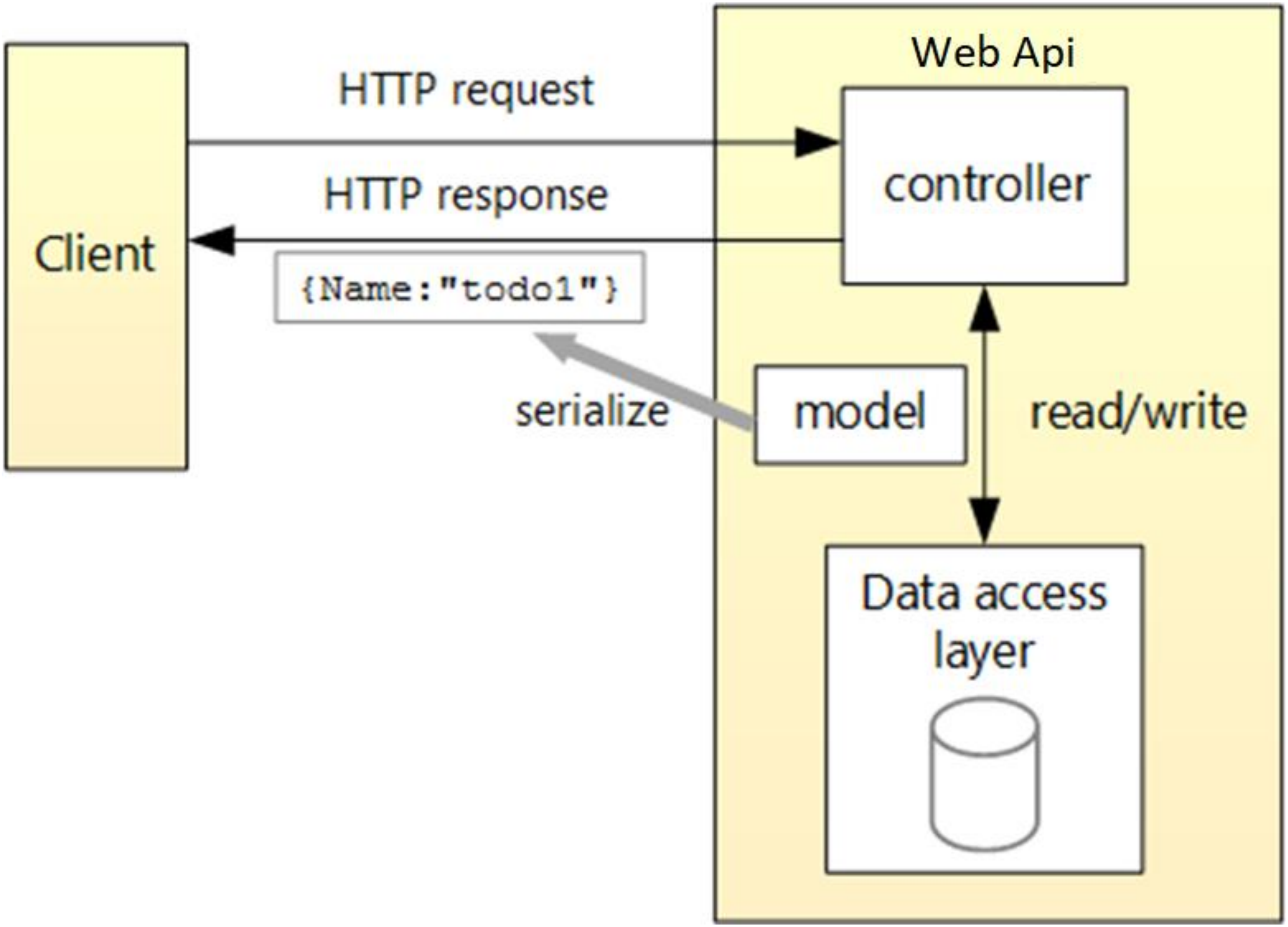
Contents

- What is a Web Api?
- The Web Api project -> Controllers
- Ajax -> Web Api methods
- Running the project
- What is a Controller?
- What is a Model?

What is Web API?

- HTTP Service
- Logic or data accessible over HTTP
- Use programmatically (called from code)
- Accessible across the internet

(Microsoft, 2017)


Web Api
HTTP request
HTTP response
Client
controller
{Name:"todo1"}
serialize
model
read/write
Data access layer

Web API and HTTP methods

- One uses HTTP Requests with ajax to use Web API method:
  - GET (retrieve information)
  - POST (post information to save new item)
  - PUT (update item)
  - DELETE (delete an item)

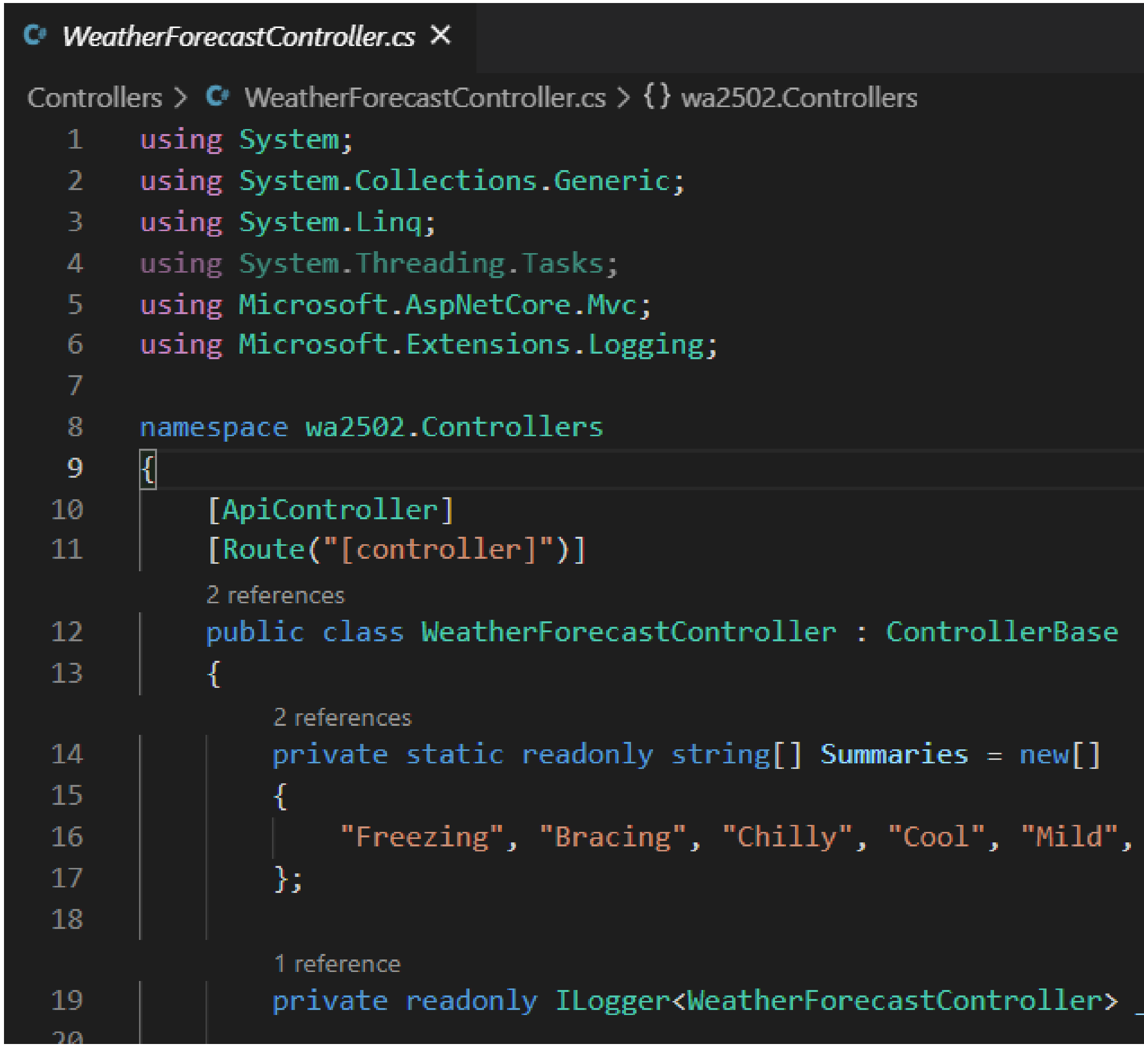
(Microsoft, 2017)

HttpStatusCode

- Web API returns status codes, for example: OK (200), Created (201), NotFound (404), Forbidden (403)
- Notice that one can control what status code to return: important to use the standards!
- For full coverage of HttpStatusCode alternatives see here:
  - https://msdn.microsoft.com/en-us/library/system.net.httpstatuscode(v=vs.118).aspx

Web API project -> Controllers

- In the Controllers folder you will find a class which is a test Web API
- The Controller will be the class that receives the Ajax calls


WeatherForecastController.cs ×
Controllers > WeatherForecastController.cs > {} wa2502.Controllers

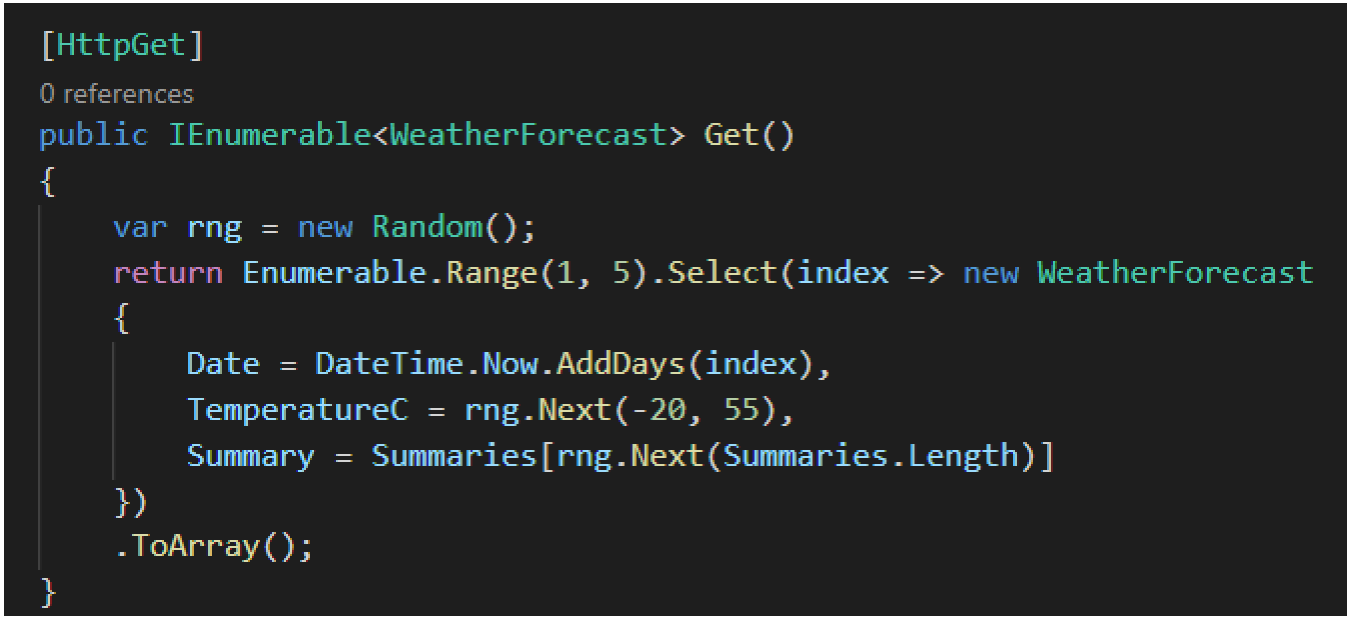
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Logging;

namespace wa2502.Controllers
{
    [ApiController]
    [Route("[controller]")]
    2 references
    public class WeatherForecastController : ControllerBase
    {
        2 references
        private static readonly string[] Summaries = new[]
        {
            "Freezing", "Bracing", "Chilly", "Cool", "Mild",
        };

        1 reference
        private readonly ILogger<WeatherForecastController> _
```

Ajax -> Web API methods

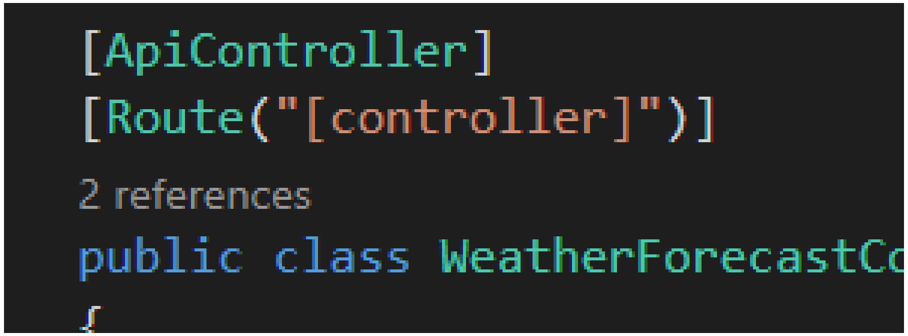
- It is the methods, in this case Get, which will be called. The url to call in this case is only the name of localhost + name of Controller:
  - localhost:5001/weatherforecast



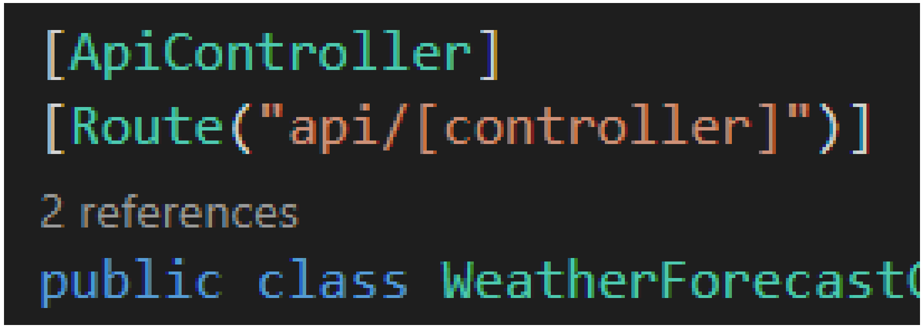
```
[HttpGet]
0 references
public IEnumerable<WeatherForecast> Get()
{
    var rng = new Random();
    return Enumerable.Range(1, 5).Select(index => new WeatherForecast
    {
        Date = DateTime.Now.AddDays(index),
        TemperatureC = rng.Next(-20, 55),
        Summary = Summaries[rng.Next(Summaries.Length)]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

Route

- The Route definition indicates how to access the Controller
- “[controller]” means “name of controller”.
- You can define it as you want, for example adding “api” in front



```
[ApiController]
[Route("[controller]")]
2 references
public class WeatherForecastCo
{
```



```
[ApiController]
[Route("api/[controller]")]
2 references
public class WeatherForecastC
```

Running the project

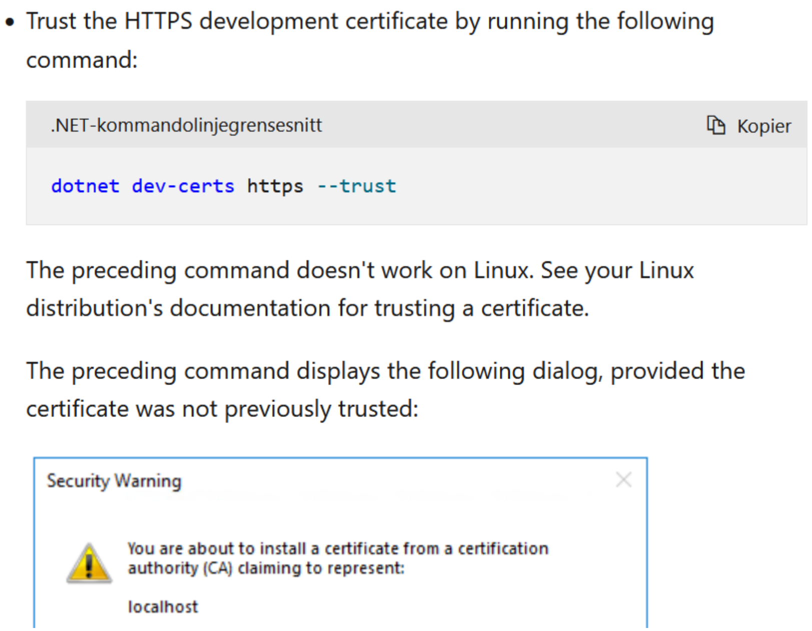
Running the project

- The two next slides show two alternative ways of running the project so you can test in the browser:
    1. Through commando line (terminal / command prompt)
    2. Through in-built run function in VSC. This option may for different reasons work for all

Sikkerhetssertifikat

- Det kan være at man må kjøre dev-certs-kommandoen for at løsningene vi lager skal kjøre på maskinen.
- Dette står nevnt i Microsoft-veiledningen:

https://docs.microsoft.com/nb-no/aspnet/core/tutorials/first-web-api?view=aspnetcore-6.0&tabs=visual-studio-code



- Trust the HTTPS development certificate by running the following command:

```
dotnet dev-certs https --trust
```

The preceding command doesn't work on Linux. See your Linux distribution's documentation for trusting a certificate.

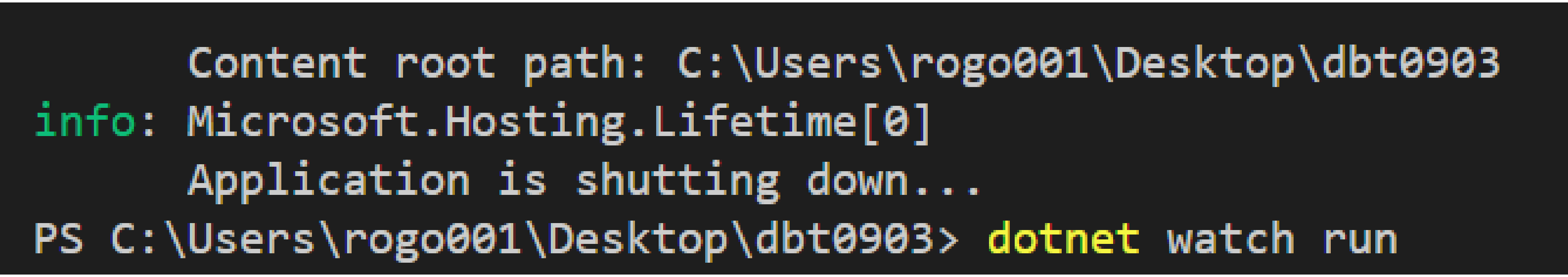
The preceding command displays the following dialog, provided the certificate was not previously trusted:

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

localhost

dotnet watch run

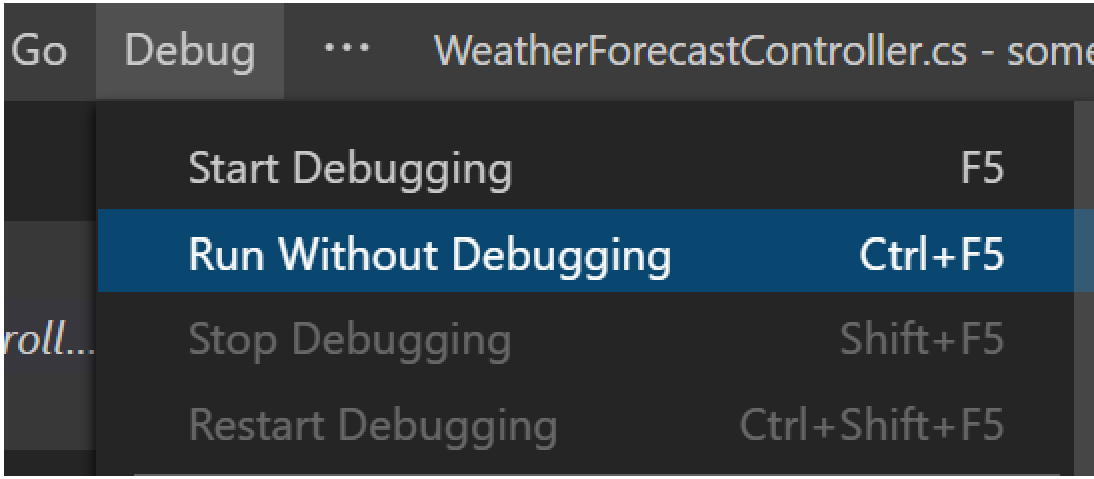
- dotnet watch run is a command which creates a test web page for your application
- In addition it is like live reload

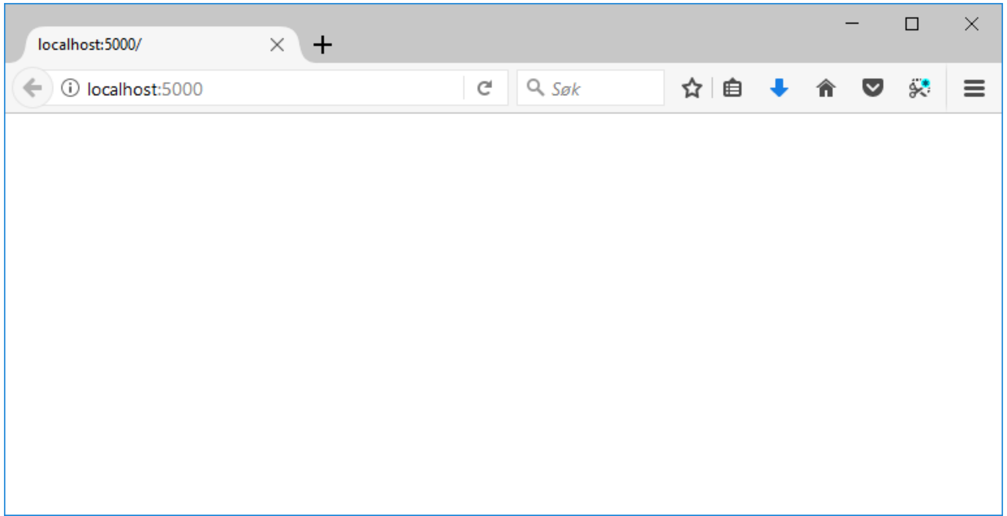


```
      Content root path: C:\Users\rogo001\Desktop\dbt0903
info: Microsoft.Hosting.Lifetime[0]
      Application is shutting down...
PS C:\Users\rogo001\Desktop\dbt0903> dotnet watch run
```

Running the project

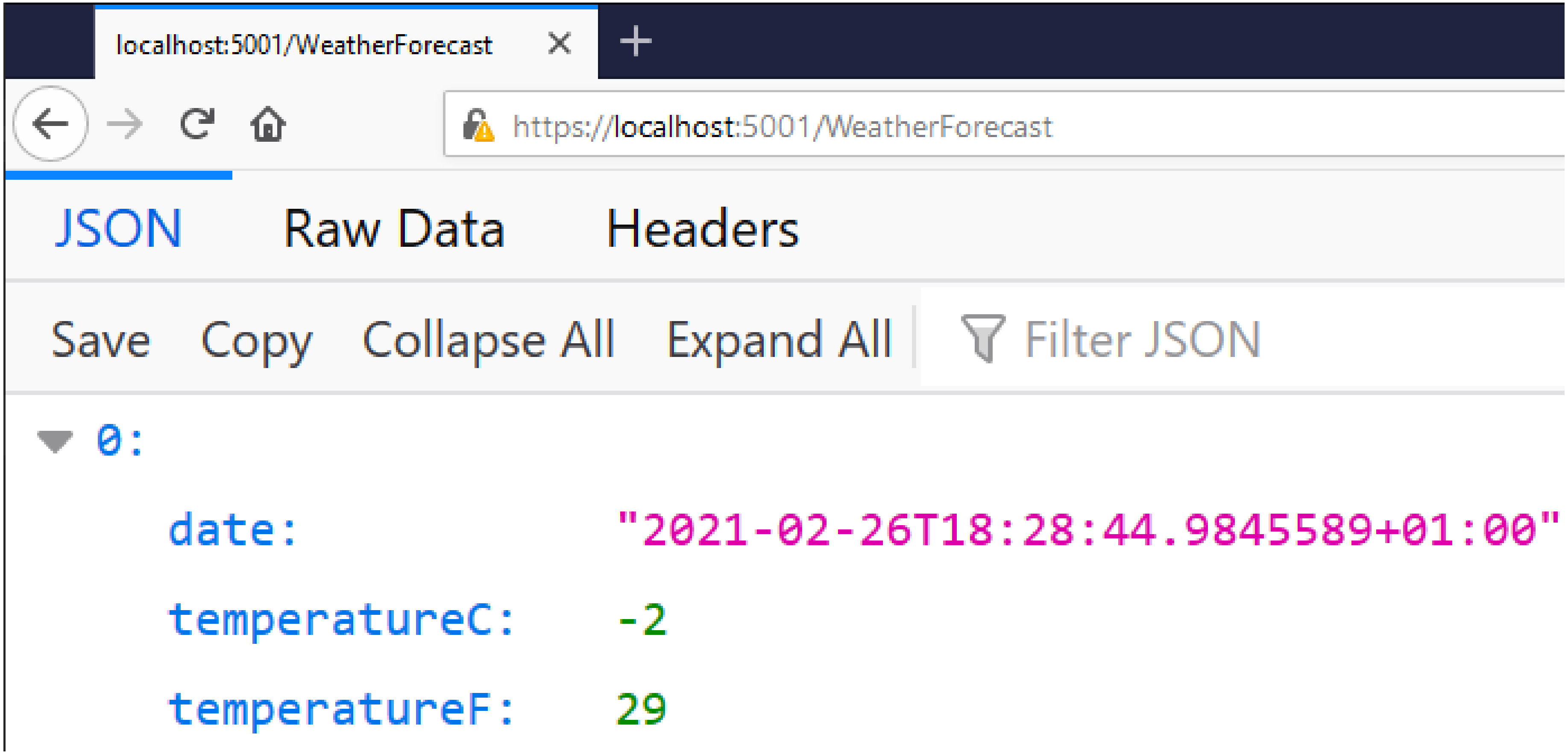
- If you go into Debug -> Run Without Debugging, you will start the Web API project
- Usually you will use localhost:5000 or localhost:5001 in the browser


Go
Debug
...
WeatherForecastController.cs - some
roll...
Start Debugging F5
Run Without Debugging Ctrl+F5
Stop Debugging Shift+F5
Restart Debugging Ctrl+Shift+F5



Running the project

- In the browser you can run the GET method in the web api by using url in browser

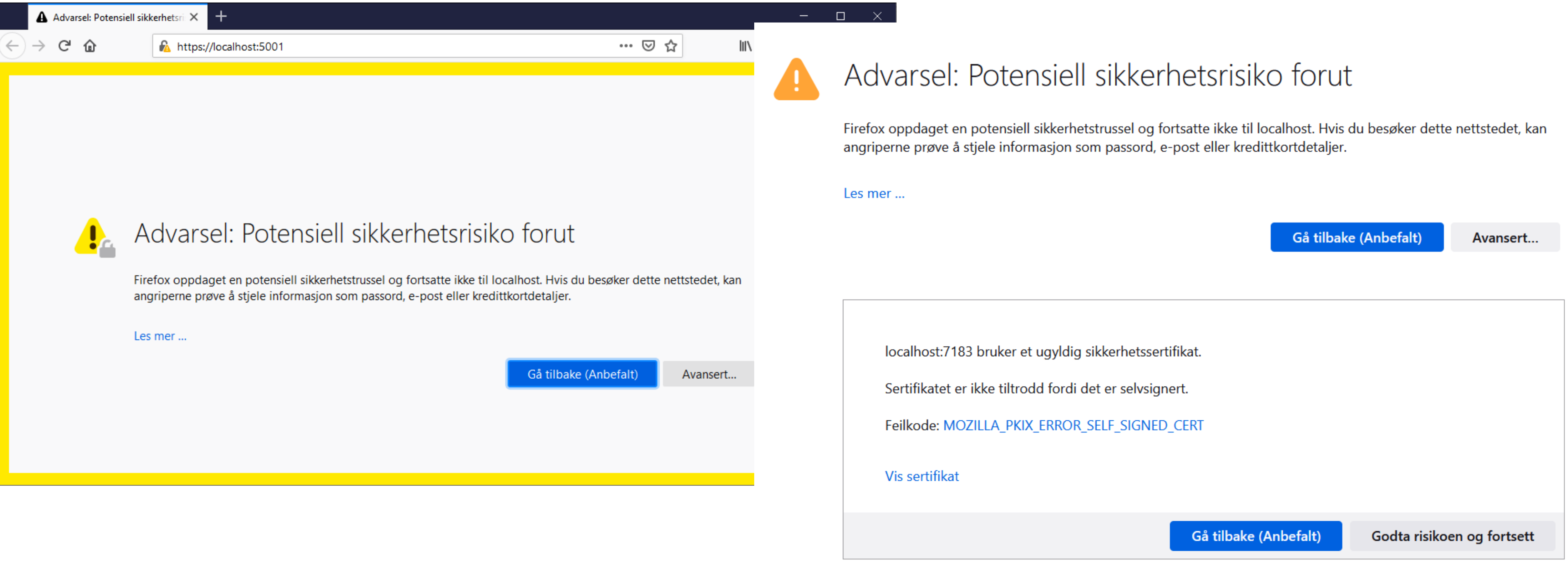


https://localhost:5001/WeatherForecast

```
0:
    date:           "2021-02-26T18:28:44.9845589+01:00"
    temperatureC:   -2
    temperatureF:   29
```

Error Message

- Første gangen man prøver å kjøre løsningen kan man få advarsel. Velg Godta Risikoen Og Fortsett (det er din egen løsning som kjøres).



Advarsel: Potensiell sikkerhetsrisiko forut

Firefox oppdaget en potensiell sikkerhetstrussel og fortsatte ikke til localhost. Hvis du besøker dette nettstedet, kan angriperne prøve å stjele informasjon som passord, e-post eller kredittkortdetaljer.

Advarsel: Potensiell sikkerhetsrisiko forut

Firefox oppdaget en potensiell sikkerhetstrussel og fortsatte ikke til localhost. Hvis du besøker dette nettstedet, kan angriperne prøve å stjele informasjon som passord, e-post eller kredittkortdetaljer.

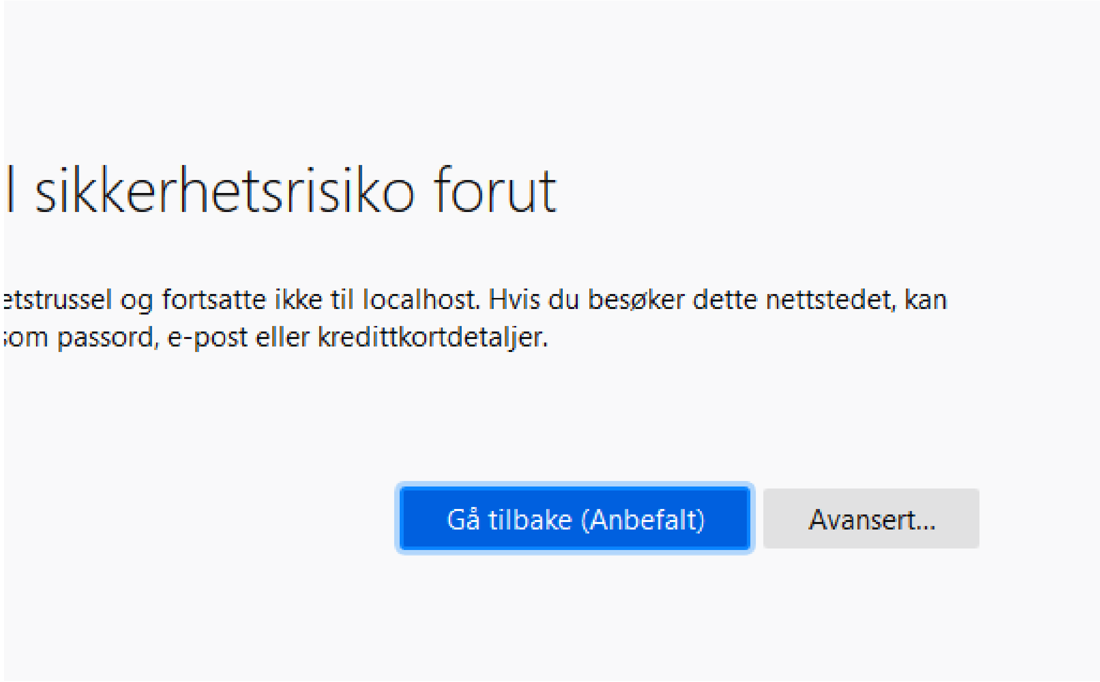
localhost:7183 bruker et ugyldig sikkerhetssertifikat.

Sertifikatet er ikke tiltrodd fordi det er selvsignert.

Feilkode: MOZILLA_PKIX_ERROR_SELF_SIGNED_CERT

Error Message

- If you get an error message like shown on previous slide go into Advanced -> Accept Risk; it is safe to use localhost, but you still get an error message



l sikkerhetsrisiko forut

etstrussel og fortsatte ikke til localhost. Hvis du besøker dette nettstedet, kan
;om passord, e-post eller kredittkortdetaljer.

Controller

Controller

- **C**ontrollers: Classes that handle browser requests. They retrieve model data and call view templates that return a response. In an MVC app, the view only displays information; the controller handles and responds to user input and interaction. For example, the controller handles route data and query-string values, and passes these values to the model. (Anderson, 2017)

Controller

- Metoder eksekveres når routingen har funnet riktig Controller
- Controllers benyttes for å gjøre logiske operasjoner på data fra Model og kaller videre på et View
- Controllere håndterer brukerinteraksjon på en nettside

Model

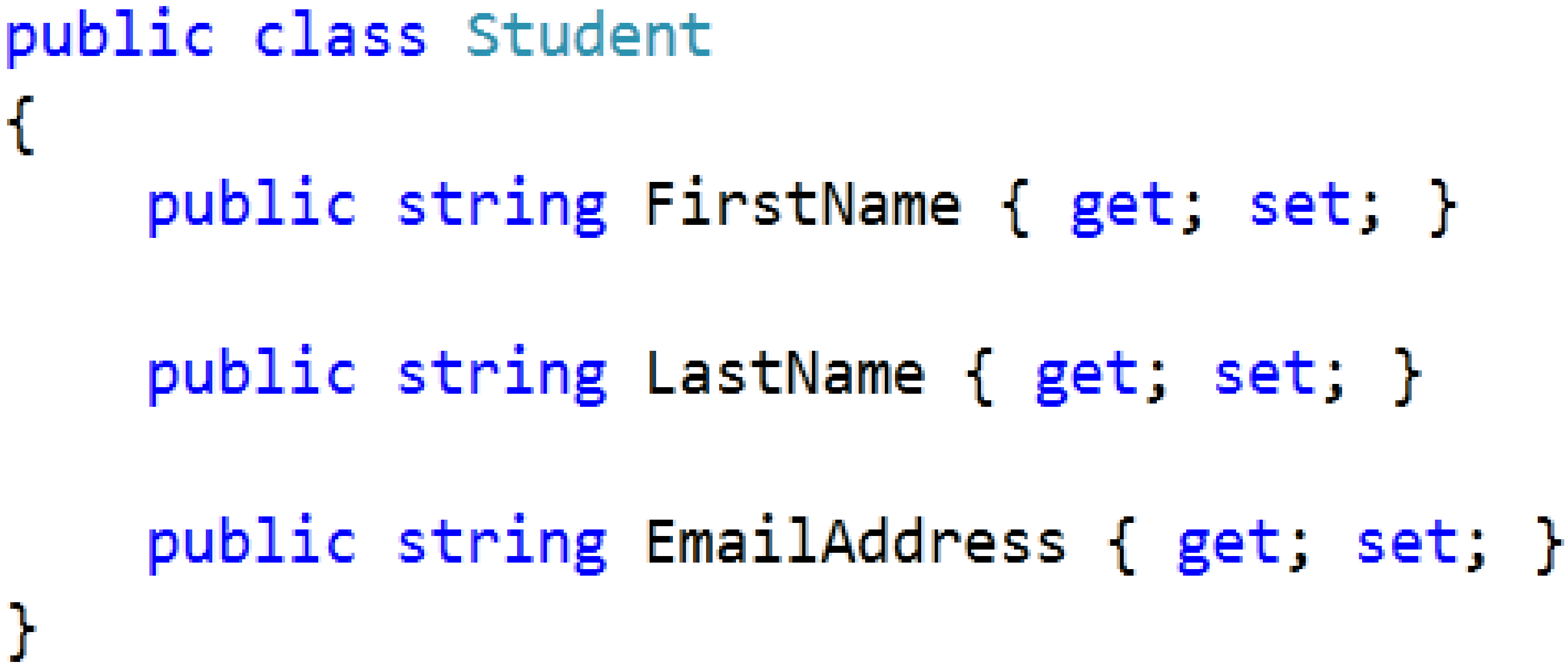
Model

- **M**odels: Classes that represent the data of the app. The model classes use validation logic to enforce business rules for that data. Typically, model objects retrieve and store model state in a database. (Anderson, 2017)

Model

- Model er klasser/objekter som implementerer logikk for applikasjonens domene
- Domeneklasser er klasser som har med en applikasjons hoveddata å gjøre. For eksempel så kan domeneklasser for en webshop være Kunde, Vare, Salg osv. Dette i motsetning til andre klasser som man kan betegne som støtteklasser

Model



```
public class Student
{
    public string FirstName { get; set; }

    public string LastName { get; set; }

    public string EmailAddress { get; set; }
}
```

Referance list

- Dan Roth (2017): Building Web APIs with ASP.NET Core 2.0, Microsoft, https://www.youtube.com/watch?v=aIkpVzqLuhA
- Microsoft (2021), Controllers, https://learn.microsoft.com/nb-no/aspnet/core/tutorials/first-web-api?view=aspnetcore-7.0&tabs=visual-studio-code, [last accessed 25.10.22]