

Mariusz Wieczorek

mariusz.wieczorek@kabat.pl

Streszczenie

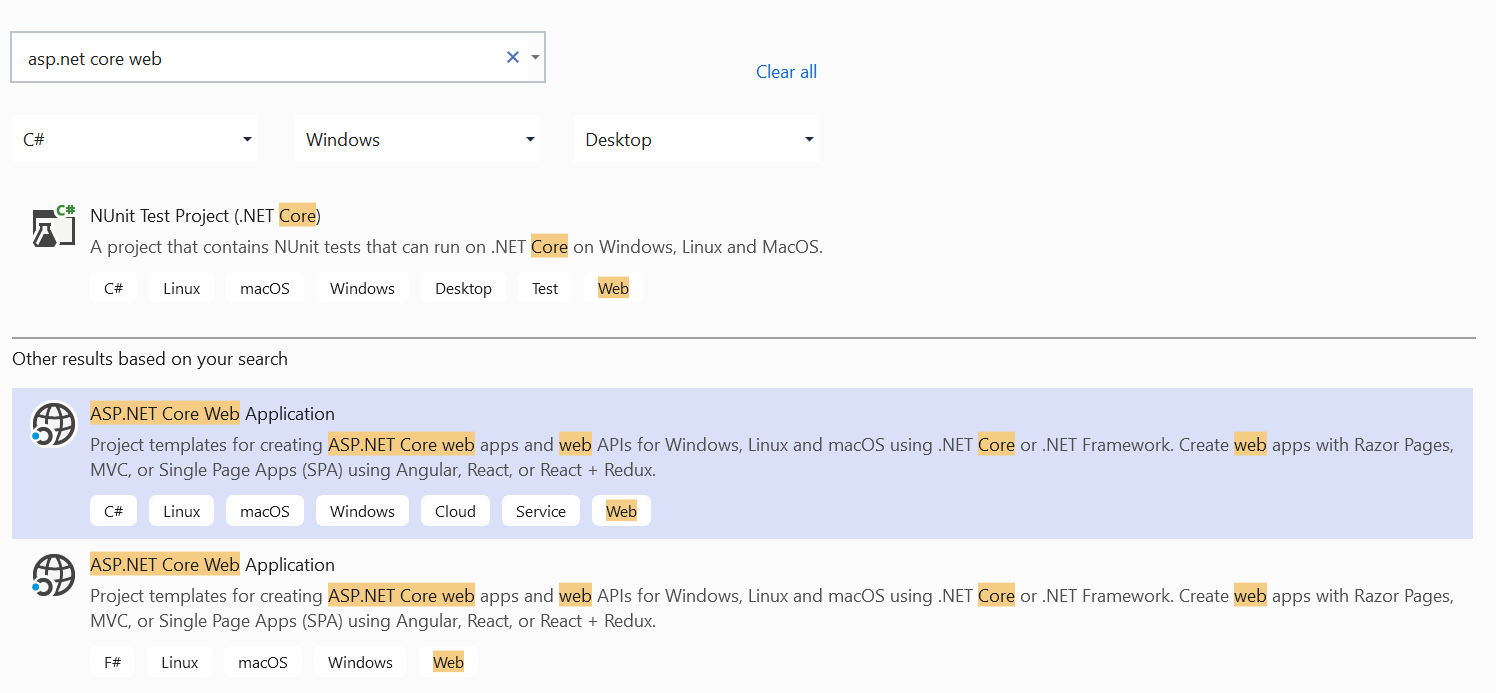
Czym jest Web API  
Czym jest REST  
Jaką aplikację stworzymy

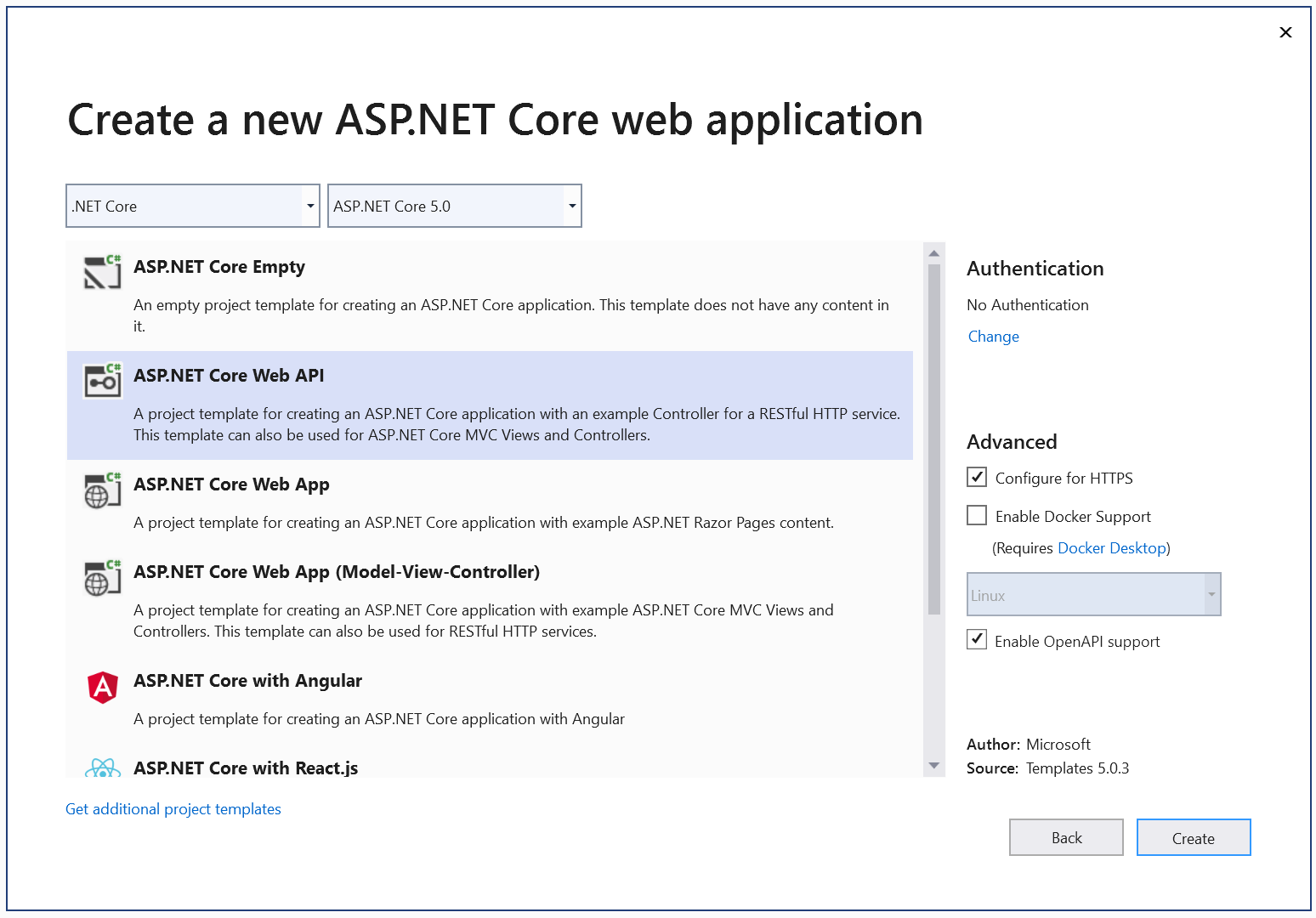
WEB API

Wstęp

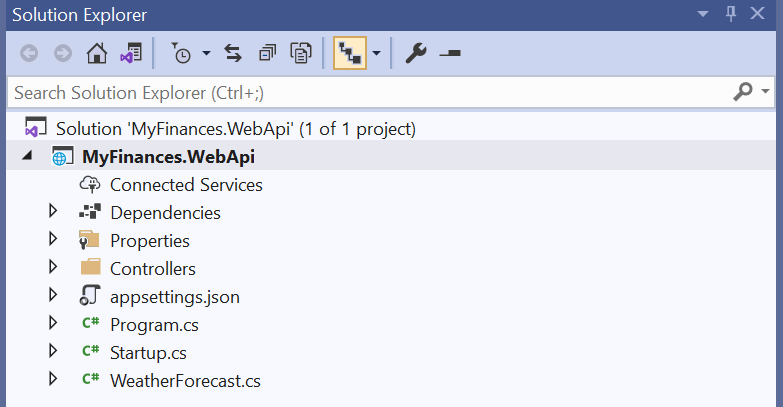
**Tworzymy nową aplikację Web API**

Wpisujemy asp.net core web i wybieramy wersję z C#





Powstał szkielet aplikacji, jest to zwykłe Web.Api, nie ma kodu odpowiedzialnego za autoryzację, a także nie mamy folderów związanych z widokami.

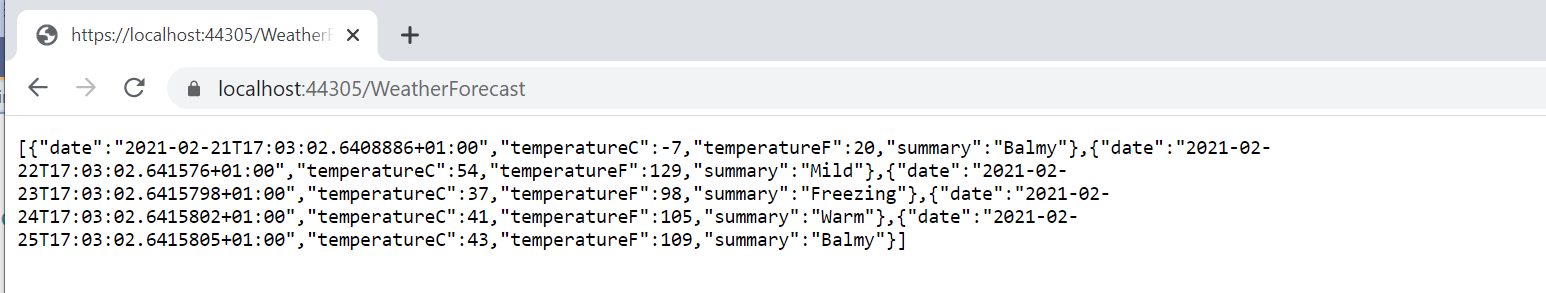


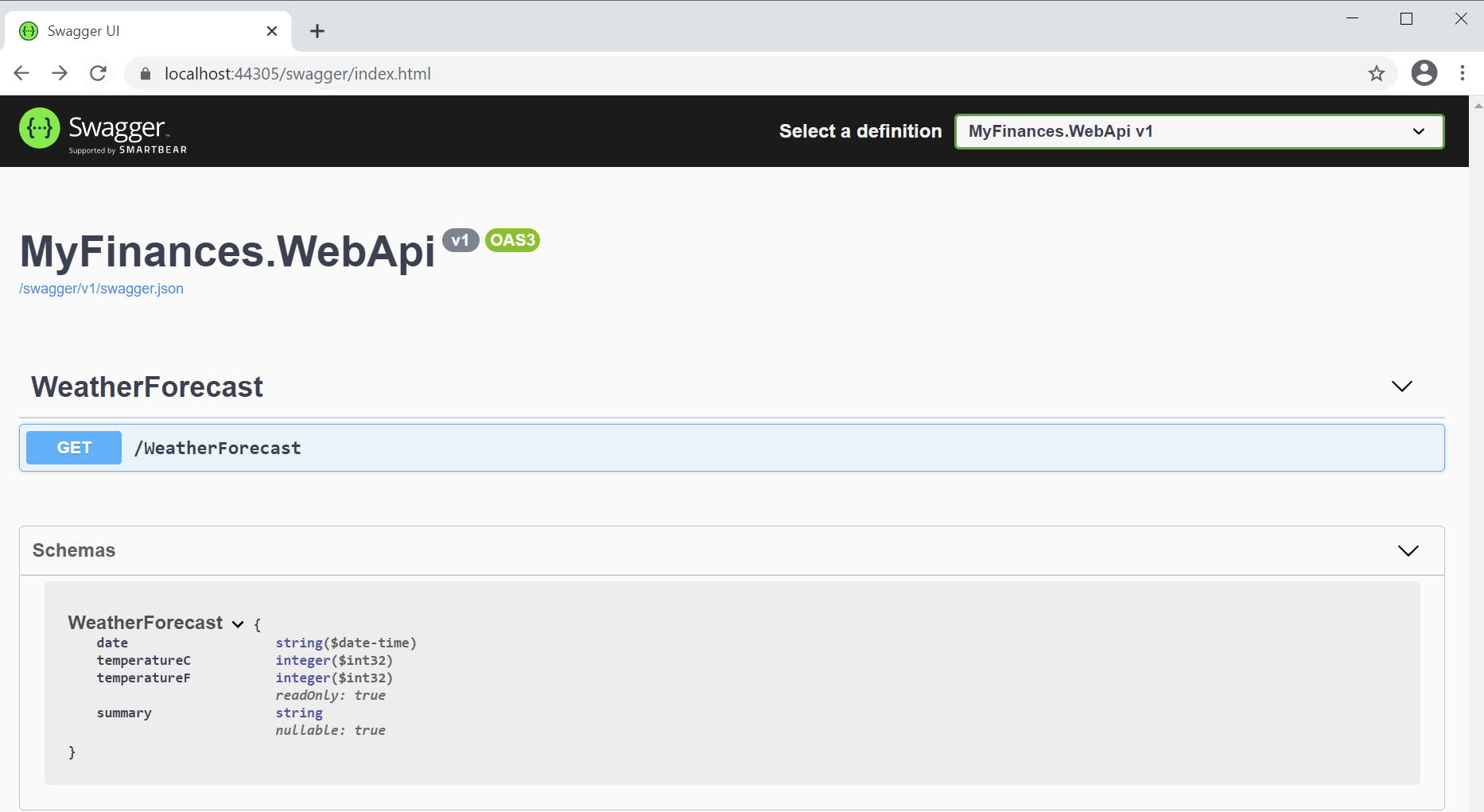
W WeatherForecast.cs mamy testowe modele, którą za chwilę usuniemy.  
W folderze Controllers mamy przykładowy kontroler

|  |
| --- |
| namespace MyFinances.WebApi.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();  }  }  } |

Klasa kontrolera musi myć oznaczona atrybutem ApiController  
i musi dziedziczyć po klasie abstrakcyjnej ControllerBase  
Następnie mamy ścieżkę za pomocą której możemy się dostać do tego kontrolera, w nawiasach ma słowo controller zamiast niego podamy nazwę naszego kontrolera w tym przypadku WeatherForecast.  
Następnie mamy pole \_logger, które zostaje wstrzyknięte za pomocą DI do konstruktora.  
Następnie mamy metodę Get

Gdy uruchomimy aplikację



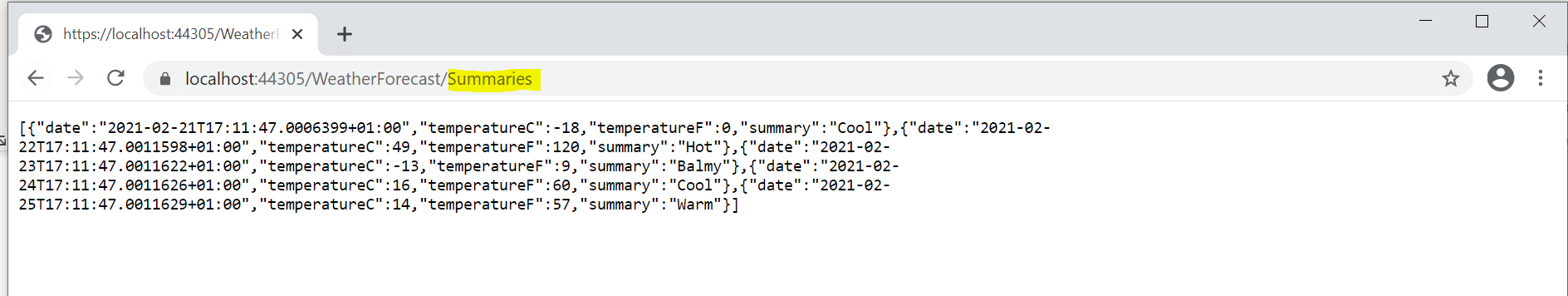


Nazwa metody Get() nie ma znaczenia, gdy ją zmienimy na inną np. GerSummaries() to rezultat po odpaleniu przeglądarki będzie taki sam.

Jeżeli daną akcję chcemy wywoływać pod inną ścieżką, musimy do akcji dodać atrybut Route.  
Jak w przykładzie poniżej

|  |
| --- |
| [HttpGet]  [Route("Summaries")]  public IEnumerable<WeatherForecast> Get()  { |

Wtedy nasz wynik jest dostępny pod inną ścieżką



Na koniec usuwamy przykładowe pliki

