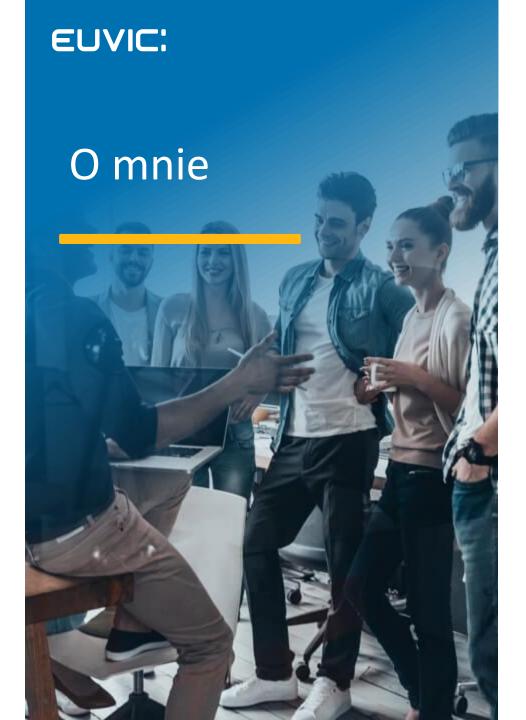
EUVIC: THE GOOD People

Final App



Kilka słów o mnie



Czy jest ktoś komu udało się zrobić ćwiczenia z identity servera ?



- CORS
- Token Silent Refresh
- Entity Framework performance
 - Generownie danych
 - Logowanie przez Serilog
 - Entity Framework Migracje
 - Deployment do Azure
 - CQRS system uprawnień



Kto z Was już natrafił na problem w którym request został zablokowany przez CORS





Czy ktoś potrafi wyjaśnić własnymi słowami dlaczego request potrafi być zablokowany przez CORS ?

Access to fetch at 'https://joke-api-strict-cors.appspot.com/r localhost/:1 andom joke' from origin 'http://localhost:3000' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource. If an opaque response serves your needs, set the request's mode to 'no-cors' to fetch the resource with CORS disabled.



cors to prostymi słowami zabezpieczenie obsługiwane przez przeglądarkę, które nie pozwoli na wykonanie

zapytania, gdy **origin** zapytania składający się z

Protokól – https

Domena – euvic.pl

Port - 443

jest inny niż oczekuje tego serwer w swojej konfiguracji CORS



A jak to wygląda w kodzie?



A co się stanie gdy wykonam request z mojego laptopa poprzez postmana.

Czy wtedy też CORS mnie zablokuje ??

Zróbmy test



Kto odpowie na pytanie dlaczego postman nie ma problemu z CORSem ?

Jaka była definicja:

"CORS to prostymi słowami

zabezpieczenie obslugiwane przez

przeglądarkę..."



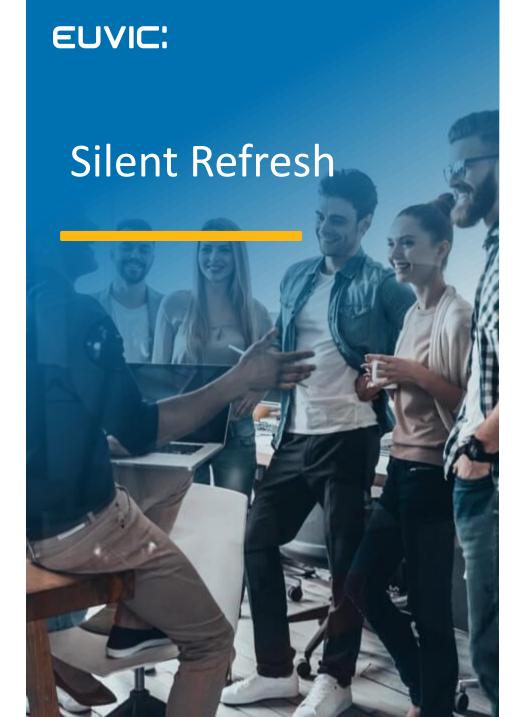
Podsumowując

- CORS to zabezpiecznie sprawdzane przez przeglądarkę
- CORS ustawiamy na protokól, domenę i port
- CORS konfigurujemy w sekcji ConfigureServices

```
services.AddCors(options =>
{
    options.AddDefaultPolicy(policy =>
        policy.WithOrigins("https://euvic.pl")
    .AllowAnyMethod()
    .AllowAnyHeader());
});
```

 Pamiętajmy o wywołaniu Middleware dla CORS w sekcji Configure

```
app.UseCors();
```



Z ostatniego szkolenia wiemy ,że token ma ważność

Gdy termin ważności minie użytkownik nie może już się tym tokenem uwierzytelnić

Kto wie jak temu zapobiec?



Przyład jak działa SilentRefresh



Które zapytanie wedlug was będzie najszybsze a które najwolniejsze

Zapytanie Nr 1

```
private async Task<!Enumerable<Contract.AttendeesListItem>> GetAttendeesWithProjection()
   return await _context.Attendees
      .Include(x => x.Trainings)
          .ThenInclude(x => x.Training)
      .Select(x =>
     new Contract.AttendeesListItem()
          Id = x.Id
          AllowedHours = x.AllowedHours,
          Firstname = x.Firstname,
         Lastname = x.Lastname,
          TotalHours = (double)x.Trainings.Sum(t => t.Training.Duration) / 60,
          TotalConfirmedHours = (double)x.Trainings
              .Where(x => x.StatusId == (int)TrainingAttendeeStatuses.Confirmed)
              .Sum(t => t.Training.Duration) / 60,
      .OrderBy(x => x.Lastname)
      .ThenBy(x => x.Firstname)
      .ToListAsync();
```



Zapytanie Nr 2

```
private async Task<IEnumerable<Contract.AttendeesListItem>> GetAttendeesWithoutProjection()
   var attendees = await _context.Attendees
     .Include(x => x.Trainings)
          .ThenInclude(x => x.Training)
     .ToListAsync();
   return attendees.Select(x =>
     new Contract.AttendeesListItem()
         Id = x.Id,
         AllowedHours = x.AllowedHours,
         Firstname = x.Firstname,
         Lastname = x.Lastname,
         TotalHours = (double)x.Trainings.Sum(t => t.Training.Duration) / 60,
         TotalConfirmedHours = (double)x.Trainings
              .Where(x => x.StatusId == (int)TrainingAttendeeStatuses.Confirmed)
              .Sum(t => t.Training.Duration) / 60,
     .OrderBy(x => x.Lastname)
     .ThenBy(x => x.Firstname)
     .ToList();
```







```
private async Task<IEnumerable<Contract.AttendeesListItem>> GetAttendeesWithoutProjectionAsNoTracking()
   var attendees = await _context.Attendees
      .AsNoTracking()
      .Include(x => x.Trainings)
          .ThenInclude(x => x.Training)
      .ToListAsync();
   return attendees.Select(x =>
      new Contract.AttendeesListItem()
         Id = x.Id,
         AllowedHours = x.AllowedHours,
         Firstname = x.Firstname,
         Lastname = x.Lastname,
         TotalHours = (double)x.Trainings.Sum(t => t.Training.Duration) / 60,
         TotalConfirmedHours = (double)x.Trainings
              .Where(x => x.StatusId == (int)TrainingAttendeeStatuses.Confirmed)
              .Sum(t => t.Training.Duration) / 60,
      .OrderBy(x => x.Lastname)
      .ThenBy(x => x.Firstname)
      .ToList();
```





Głosujemy na chacie

Najszybsze zapytanie to ??

Najwolniejsze zapytanie to ??

EUVIC:



EF Performance

No to sprawdzmy jakie sa wyniki nr 1 – Projekcja

```
[19:07:34 DBG] Euvic.StaffTraining.Contracts.Attendees.Queries.GetAttendeesList+Query was executed within 80 ms
[19:07:53 DBG] Euvic.StaffTraining.Contracts.Attendees.Queries.GetAttendeesList+Query was executed within 67 ms
[19:07:57 DBG] Euvic.StaffTraining.Contracts.Attendees.Queries.GetAttendeesList+Query was executed within 61 ms
[19:08:00 DBG] Euvic.StaffTraining.Contracts.Attendees.Queries.GetAttendeesList+Query was executed within 73 ms
```



nr 2 – Bez projekcji, bez AsNoTracking

```
[19:15:54 DBG] Euvic.StaffTraining.Contracts.Attendees.Queries.GetAttendeesList+Query was executed within 208 ms
19:15:58 DBG] Euvic.StaffTraining.Contracts.Attendees.Queries.GetAttendeesList+Query was executed within 222 ms
19:16:01 DBG] Euvic.StaffTraining.Contracts.Attendees.Queries.GetAttendeesList+Query was executed within 203 ms
[19:16:05 DBG] Euvic.StaffTraining.Contracts.Attendees.Queries.GetAttendeesList+Query was executed within 180 ms
```



nr 3 – Bez projekcji, z AsNoTracking

19:20:44 DBG] Euvic.StaffTraining.Contracts.Attendees.Queries.GetAttendeesList+Query was executed within 119 ms 19:20:48 DBG] Euvic.StaffTraining.Contracts.Attendees.Queries.GetAttendeesList+Query was executed within 176 ms 19:20:53 DBG] Euvic.StaffTraining.Contracts.Attendees.Queries.GetAttendeesList+Query was executed within 133 ms 19:20:58 DBG] Euvic.StaffTraining.Contracts.Attendees.Queries.GetAttendeesList+Query was executed within 132 ms

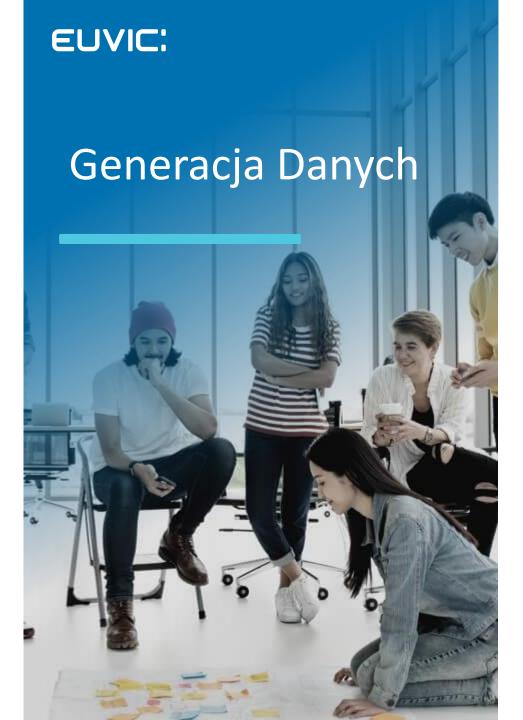




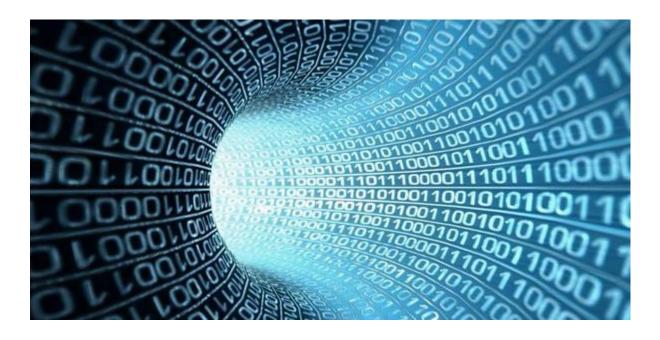
```
private async Task<IEnumerable<Contract.AttendeesListItem>> GetAttendeesWithProjection()
   return await _context.Attendees
      .Include(x => x.Trainings)
          .ThenInclude(x => x.Training)
      .Select(x =>
     new Contract.AttendeesListItem()
         Id = x.Id
          AllowedHours = x.AllowedHours,
          Firstname = x.Firstname,
         Lastname = x.Lastname,
         TotalHours = (double)x.Trainings.Sum(t => t.Training.Duration) / 60,
         TotalConfirmedHours = (double)x.Trainings
              .Where(x => x.StatusId == (int)TrainingAttendeeStatuses.Confirmed)
              .Sum(t => t.Training.Duration) / 60,
      .OrderBy(x => x.Lastname)
      .ThenBy(x => x.Firstname)
      .ToListAsync();
```

Pamiętajcie aby tam gdzie się da korzystać w projekcji bo to w 90% przypadków da zawsze najlepszą wydajność





Zeby móc przetestować wydajność zapytań trzeba miec odpowiednią ilość danych





Nawet najgorsze zapytanie wykona się błyskawicznie jeżeli w tabelce mamy 100 wierszy

Gdy widzimy skomplikowane zapytania do bazy miejsca warto zapytać ile potencjalnie wierszy oczekują w tej tabelce

Czasem może to być kilka tysięcy wiec nawet cieżkie zapytanie wykona się szybko, a może klient powie ze będzie tam 500k co 6 miesiecy



Dla miejsc które, wymagają szczególnej uwagi warto wygenerować ilość danych adekwatną to potencjalnej ilości na produkcji i zobaczyć jak sobie te zapytanie radzi.



Jak to zrobić?

https://github.com/bchavez/Bogus
https://github.com/borisdj/EFCore.Bul
kExtensions



Dotnet core nie posiada wbudowanego providera do logowania poza przekierowaniem logów do konsoli

Czyli bez dodatkowej biblioteki nie jesteśmy w stanie logować np. do pliku

 SERILOG to dość prosta w konfiguracji biblioteka oraz ma ogromną ilość providerów



Ćwiczenie konfiguracja Seriloga

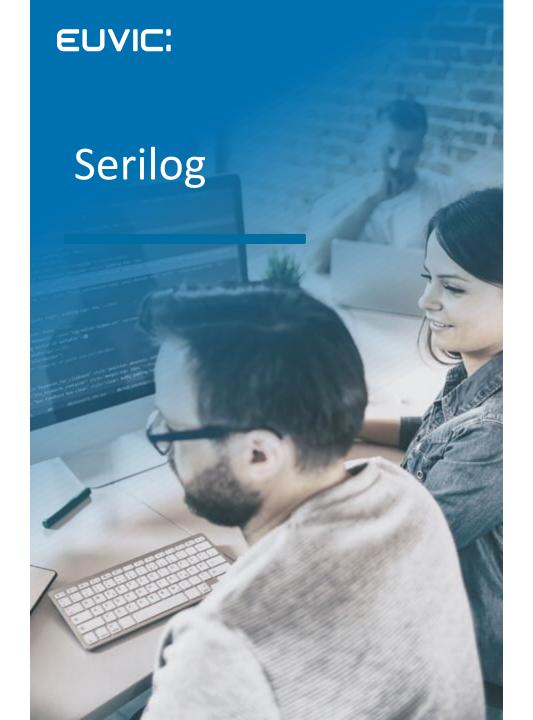


Czy ktoś może wyjaśnić dlaczego górny log ma parametery na biało a dolny ma pokolorowane?

[08:31:17 INF] Searching trainings that contain 'aaa'. Take page number 23 [08:31:17 WRN] Searching trainings that contain 'aaa'. Take page number 23

```
EUVIC:
Serilog
```

```
public IActionResult GetTrainings(string name, int page)
{
    _logger.LogInformation($"Searching trainings that contain '{name}'. Take page number {page}");
    _logger.LogWarning("Searching trainings that contain '{name}'. Take page number {page}", name, page);
    return Ok();
```



Proszę aby wszyscy teraz specjalnie zepsuli jsona w appsettings i uruchomili aplikację



A teraz poprawcie appsettings aby były poprawne i wrzucie

throw new System.Exception("Error !");

na koniec

public void ConfigureServices(IServiceCollection services)



Zmiana poziomu widoczności logów

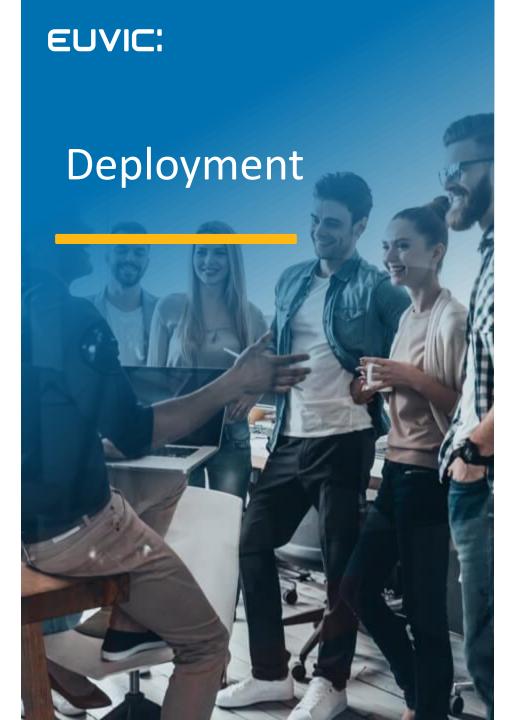


```
"Serilog": {
   "Using": [
        "Serilog.Sinks.Console",
        "Serilog.Sinks.ApplicationInsights"
],
   "MinimumLevel": {
        "Default": "Debug",
        "Override": {
             "Microsoft": "Warning",
              "Microsoft.AspNetCore": "Warning",
              "Microsoft.Hosting.Lifetime": "Information",
              "System": "Warning"
        }
},
```

Najważniejsza zasada przy override to nadpisujemy tylko poziom wyżej niż jest default. Serilog nie akceptuje override na niższy poziom niż jest w default.

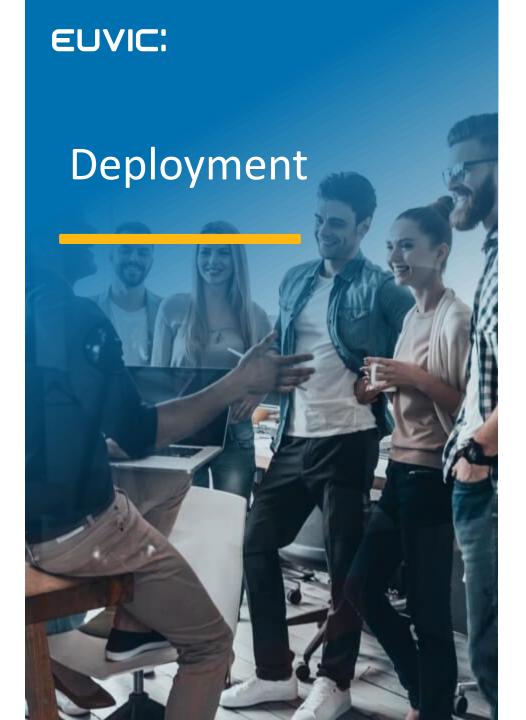


Logowanie do appinsights



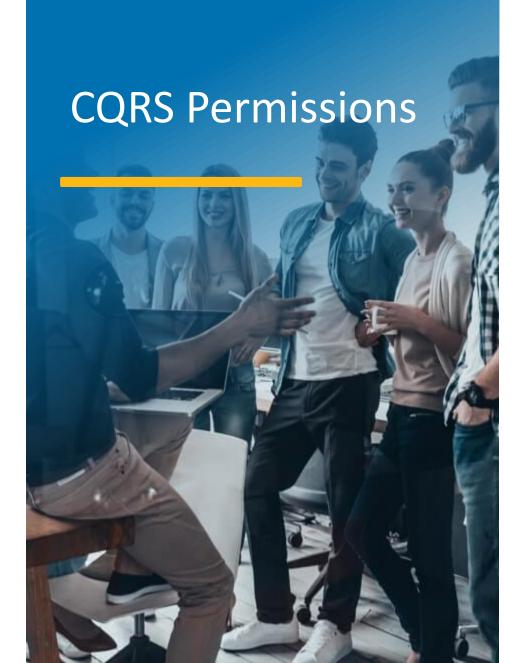
O czym musimy pamiętać przygotowując projekt do deploymentu

- Przygotować appsettings dla każdego środowiska
- Na każdym środowisku oznaczyć zmienną środowiskową "ASPNETCORE_ENVIRONMENT" jakie to środowisko
- Upewnić się że mamy wyłączone automigracje podczas uruchamiania aplikacji na środowisku
- Wygenerować migracje jako skrypt i wykonać je na bazie



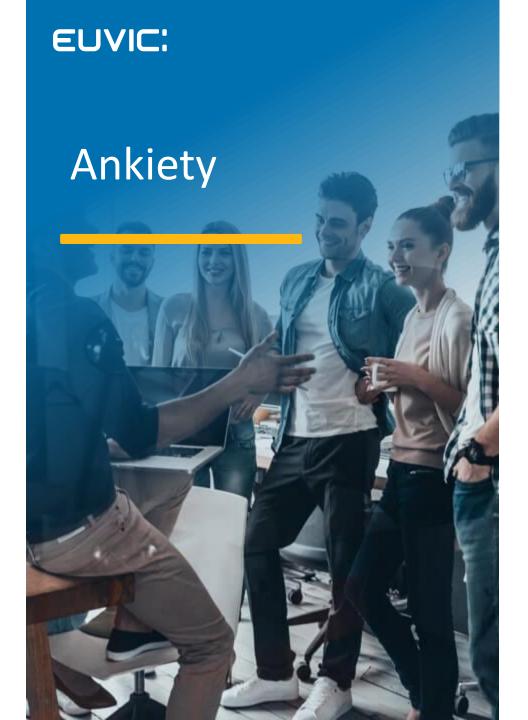
Jak zrobić deployment w Visual Studio?





BONUS

Przykładowy system uprawnień oparty na CQRS



Proszę wszystkich aby każdy teraz wypełnił ankietę

I pamiętajcie że najgorsza ocena jeżeli jest zgodna z prawdą jest lepsza niż brak oceny

Dziękuję bardzo ©



www.euvic.com