|  |
| --- |
| AIRBNB London\_  NotS-WAPP |

Kevin Slingerland (1664295)

22 juni 2023

INHOUDSOPGAVE

[1 Software Architectuur 4](#_Toc138789881)

[1.1 Functioneel Ontwerp 4](#_Toc138789882)

[1.1.1 Use Cases 4](#_Toc138789883)

[1.2 Technisch Ontwerp 5](#_Toc138789884)

[1.3 Dependancy’s 6](#_Toc138789885)

[1.3.1 Frameworks 6](#_Toc138789886)

[1.3.2 Packages / Libs 6](#_Toc138789887)

[1.4 Configuration 7](#_Toc138789888)

[2 Preformance 8](#_Toc138789889)

[2.1 Async 8](#_Toc138789890)

[2.2 AsNoTracking 8](#_Toc138789891)

[2.3 FusionCache 9](#_Toc138789892)

[2.3.1 Distributed Cache 9](#_Toc138789893)

[2.4 Map Feature Layer 10](#_Toc138789894)

[3 Security 11](#_Toc138789895)

[3.1 OWASP ZAP 11](#_Toc138789896)

[3.1.1 CORS 11](#_Toc138789897)

[3.1.2 RESULTAAT 12](#_Toc138789898)

[3.2 SECURITY HEADERS 13](#_Toc138789899)

INLEIDING

dit document worden de ontwikkelingen, bevindingen, prestaties en veiligheid van de app behandeld. Tijdens het ontwikkelproces heb ik gebruik gemaakt van .NET-technologieën, waarbij ASP.NET 6.0 als backend werd gebruikt. Het integreren van Redis heeft geleid tot aanzienlijke verbeteringen in de prestaties van de API- Requests

In overleg met de docent ben ik op onderzoek uitgegaan naar caching met behulp van fusion cache hier zitten een aantal fail-safes en connfiguratie opties in die het waard maken om naar te kijken.

Tot slot behandelt dit document de veiligheidsaspecten die zijn toegepast binnen de webapplicatie. Er zal een initiële stand worden geschetst, waarmee het eindresultaat zal worden vergeleken.

# Software Architectuur

## Functioneel Ontwerp

### Use Cases

AirBnbInsights kent 2 soorten actoren, bezoekers en administrators. De bezoeker zal binnen het systeem basisinzichten willen krijgen zoals de verschillende “listings” en de bijbeoordende details. Een administrator zal binnen het systeen overzichten willen krijgen over aantallen en andere raportcijfers.

#### Als bezoeker wil ik een overzicht krijgen van de listings

Een bezoeker kan door middel van een kaart inzicht verkrijgen in de locaties van Airbnb-woningen in london.

#### Als bezoeker wil ik details van een listing kunnen inzien zodat ik meer informatie kan inzien

Een bezoeker kan door een woning te selecteren de gegevens van de woning inzien.

#### Als bezoeker wil ik listings kunnen filteren om mijn zoekselecte te vereenvoudigen

Een bezoeker heeft de mogelijkheid filters in te geven. Aan de hand van de toegepaste filters wordt de weergave van het aantal woningen aangepast op de kaart.

#### Als administrator wil ik een overzicht hebben van verschillende trends, totalen & gemiddelden om meer inzicht te krijgen.

Een administratieve bezoeker navigeert naar de inlog gelegenheid. Na het succesvol ingeven van de login gegevens navigeert de bezoeker door naar het rapportage overzicht om de cijfers in te zien.

## Technisch Ontwerp

## Dependancy’s

Een korte beschrijving van alle gebruikte packages inclusief de gebruikte versie, dit zijn alle dependancy’s voor het gehele beroepsproduct.

### Frameworks

* React 18.2.0
* .NET 6.0LTS

### Packages / Libs

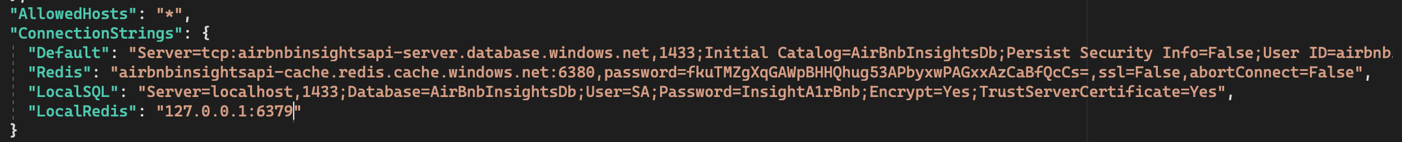
#### Front-End

* React-Router: ^6.13.0
* Mapbox-GL: ^2.15.0
* React-Charts: ^3.0.0-beta.54
* @mui ^5.10.17
* Axios ^1.4.0
* React-query ^3.39.3

#### Back-End

* Microsoft.EntityFrameworkCore 7.0.7
* Microsoft.EntityFrameworkCore.Design 7.0.7
* Microsoft.EntityFrameworkCore.SqlServer 6.0.18
* Microsoft.EntityFrameworkCore.Tools 7.0.7
* Microsoft.Extentions.Caching.StackExchangeRedis 7.0.7
* Newtonsoft.Json 13.0.3
* Swashbuckle.AspNetCore 6.5.0
* ZiggyCreatures.FusionCache 0.21.0
* ZiggyCreatures.FusionCache.Serialization.NewtonsoftJson 0.21.0

## Configuration



Figuur : Service Configuratie

A screen shot of a computer program

Description automatically generated with low confidence

Figuur : Cache Configuratie

# Preformance

## Async

Het gebruik van asynchrone functionaliteit in een API biedt aanzienlijke prestatieverbeteringen. Asynchrone API's kunnen meerdere aanvragen gelijktijdig verwerken, waardoor wachttijden worden verminderd. Ze maken efficiënter gebruik van systeembronnen en verbeteren de algehele reactievermogen en schaalbaarheid. Dit resulteert in een betere gebruikerservaring.

## AsNoTracking

Er zijn verschillende opties beschikbaar in Entity Framework om de prestaties te verbeteren, waarvan één optie AsNoTracking() is. Deze optimalisatie stelt je in staat om aan te geven dat Entity Framework de resultaten van een query niet hoeft bij te houden. Met andere woorden, Entity Framework voert geen extra verwerking of opslag uit van de entiteiten die door de query worden geretourneerd, waardoor het geheugengebruik tot een minimum wordt beperkt. (Bron)

Het gebruik van AsNoTracking() is aan te raden wanneer je alleen gegevens wilophalen en geen wijzigingen wilt bijhouden.

## Map Feature Layer

Ik heb ervoor gekozen om de datapunten toe te voegen als feature map, er is een mogelijkheid om alle punten toe te voegen als datapoints maar hierdoor rerenderen alle punten wanneer de map beweegt. Hierom heb ik ervoor gekozen om de datapunten toe te voegen als featues.

## FusionCache

Voor de caching oplossing heb ik gebruikt gemaakt van FusionCache van ZiggyCreatures, deze package wordt officieel uitgelicht in de demo applicaties van Microsoft en is een gemakkelijkere manier om een caching oplossing te implementeren.

A picture containing text, screenshot, line, diagram

Description automatically generated

Figuur Chaching Bennifits

“F**usionCache is an easy to use, fast and robust cache with advanced resiliency features and an optional distributed 2nd layer.”**

A picture containing text, diagram, screenshot, font

Description automatically generated

Figuur FusionCache Architecture

FusionCache ondersteund gemakkelijke configuratie en fallback opties, ook ondersteunt het een distributed caching layer zoals Redis

### Distributed Cache

Voor de AirBnbInsights applicatie heb ik gekozen voor een relatief lage retention time in de cache dit zodat de werking van de cache goed zichtbaar was. In productie situaties zou dit opgevoerd kunnen worden voor bepaalde endpoints zoals de algemene listing & dashboard endpoints.

## Preformance Metingen

Voor mijn preformance testen heb ik gekozen om K6 te gebruiken, dit is een populaire makkelijk te gebruiken preformance test tool, voor de uitgvoerde testen heb ik voor een simpel senatio gekozen deze is bijgevoegd in mijn repo en test voornamelijk de response tijden. Binnen k6 is zeer uitgebreide configuratie mogelijk zoals threshholds, automatisch stoppen, stages, continous load ect.

Ik heb in mijn tests gekozen om de impact van de caching oplossing te testen, dit omdat dit voor mij nieuw is en ik beniewd ben in hoevere dit impact heeft.

A screenshot of a computer program

Description automatically generated with medium confidence

Figuur K6 Config

k6 vus 1500 duration 2.30m 80% < 1000ms

GET <HTTPS://localhost:7158/api/listings/:listingid>

REDIS CACHE, ASYNC, ASNOTRACKING

A screenshot of a computer

Description automatically generated with medium confidence

Figuur Load Test - Redis Cache

MEMORY CACHE, ASYNC, ASNOTRACKING

A screenshot of a computer

Description automatically generated with medium confidence

Figuur Load Test - Memory Cache

NO CACHE, ASYNC, ASNOTRACKING

A picture containing text, screenshot

Description automatically generated

Figuur Load Test - No Cache

Opvallend is hier dat je op het eerste ook zou zeggen dat de no-cashing stragegie beter werkt maar wanneer er goed gekeken word kan je zien dat eigenlijk alle requests timed-out zijn, de api is in dit geval zo druk bezig met de database bevragen dat de vraag teveel word.

# Security

Om security binnen de app te kunnen testen maak ik gebruik van 2 tools dit zijn OWASP ZAP & Security headers. Deze tools geven een duidelijk overzicht van een aantal belangrijke zaken binnen de applicatie en zijn eerder behandeld in de lessen.

## OWASP ZAP

OWASP ZAP (Zed Attack Proxy) is een open-source tool voor het testen van de beveiliging van webapplicaties. Het helpt bij het identificeren van kwetsbaarheden door het simuleren van aanvallen en het genereren van gedetailleerde rapporten.

A picture containing text, screenshot, font, number

Description automatically generated

Figuur OWASP ZAP 1e Run

CSP: Wildecard Directive  
Deze waarschuwing verwijst opnieuw naar het gebruik van "unsafe-inline". Het wijst erop dat zowel script-src als style-src het [onveilig] type toestaan.

o CSP: script-src en style-src unsafe-inline  
Zoals eerder vermeld, is het vrijwel onvermijdelijk om "unsafe-inline" te vermijden bij het gebruik van de React JS-stack.

o TimeStamp Disclosure – Unix  
Dit is een handmatige validatie die ZAP aan de gebruiker presenteert. Deze waarschuwing vraagt om te controleren of de datums die worden meegegeven in de header geen patroon hebben dat kan worden onderschept.

o Information Disclosure  
Deze waarschuwing wijst de gebruiker erop dat contentnamen discreet moeten worden gemaakt om minder informatieprijs te geven over de code.

o Modern Web Application  
Deze waarschuwing geeft aan dat ZAP detecteert dat het een moderne webapplicatie is, gezien het gebruik van <script>-tags in plaats van <img>-tags in het HTML-bestand.

## SECURITY HEADERS

Security Headers is een website waar je een webURL invoert en vervolgens een rapport ontvangt. Dit rapport beoordeelt de robuustheid van de webpagina op een alfanumerieke schaal, waarbij A+ de hoogste score is en F aangeeft dat er aanzienlijke verbeteringen nodig zijn.

Toen ik de applicatie voor de eerste keer door het Security Headers kwam hier de score “Not bad..” uit.

A screenshot of a computer

Description automatically generated with low confidence

Figuur Security Headers - 1e Run

Om verbetering aan te brengn in deze situatie heb ik een klasgenoot en nodige documentatie geradpleegd en hier is de volgende configuratie uit komen rollen.

A screen shot of a computer

Description automatically generated with medium confidence

Figuur WebServer Config

A screenshot of a computer

Description automatically generated with low confidence

Figuur Security Headers - 2e Run

Na configuratie van een aantal instellingen van de web-server is het resultaat al een stuk beter, het ontbrekende deel hier voor een A+ is de content-security-policy maar dit is inarent aan een react architectuur.

