Naam: Mitchel Fleury

Datagedreven webapplicatie getest voor high performance en security

Mitchel Fleury (student)

Ontwerpdocument Inside AirBnB Paris

2024

Studentnummer: 1675059

Klas: ITA-NOTS-A-f

Course: NotS WAPP

Docent: Marcel Verheij

Datum: 07-06-2024

Versie: 1

Inhoudsopgave

[1. Inleiding 2](#_Toc168654646)

[2. Functioneel ontwerp 3](#_Toc168654647)

[3. Technisch ontwerp 4](#_Toc168654648)

[3.1 Architectuur 4](#_Toc168654649)

[3.2 Packages 4](#_Toc168654650)

[4. Performance 6](#_Toc168654651)

[4.1 Nulmeting 6](#_Toc168654652)

[4.2 Redis caching 7](#_Toc168654653)

[4.3 Memory cache 9](#_Toc168654654)

[4.4 AsNoTracking 11](#_Toc168654655)

[4.5 Indexing 12](#_Toc168654656)

[4.6 Conclusie 13](#_Toc168654657)

[5. Security 14](#_Toc168654658)

[5.1. OWASP ZAP 14](#_Toc168654659)

[5.2. Conclusie 14](#_Toc168654660)

[Literatuurlijst 15](#_Toc168654661)

[Bijlagen 16](#_Toc168654662)

[Bijlage 1: Casus Inside Airbnb 16](#_Toc168654663)

# Inleiding

In de businesscase voor Inside Airbnb zijn er een aantal uitdaging die moeten worden aangepakt. Inside Airbnb heeft data voor Airbnb listings die ze willen kunnen tonen op een kaart. Er moeten details en statistieken over deze listing kunnen worden ingezien. Daarnaast is het een uitdaging dat de statistieken alleen maar voor beheerders inzichtelijk mogen zijn. Naast deze functionaliteit moet het mogelijk kunnen zijn om te kunnen filteren op wijk, prijs en reviews.

Naast de functionaliteit is het noodzakelijk dat er is nagedacht over de security en de performance. Op deze manier blijft de applicatie veilig en hebben de gebruikers een goede ervaring wanneer ze de applicatie bezoeken.

In dit document wordt er dieper op de functionele requirements in gegaan. In het technisch ontwerp wordt de architectuur van de applicaties besproken. Vervolgens zijn er twee hoofdstukken die zich richten op het verbeteren van de performance en security.

# Functioneel ontwerp

Voordat er ontwikkeld kan worden moet er gekeken worden wat de usecases zijn (zie tabel 1). Voor deze opdracht staan alle usecases, inclusief prioriteit in de causus (**BRONVERMELDING**).

|  |  |
| --- | --- |
| Usecase | Prioriteit |
| Registreren en inloggen | Must |
| Filter op prijs | Must |
| Filter op buurt | Must |
| Filter op review | Must |
| Kaart is clickable, details rechts op pagina, maakt gebruik van Mapbox API | Must |
| Er moeten rollen toegevoegd en toegekend kunnen worden aan de geregistreerde gebruikers | Must |
| Resultaten zoals trends, totalen, gemiddelden, etc. worden weergegeven in charts, alleen te bekijken voor ADMINS. Denk daarbij aan bv. Gemiddelde beschikbaarheid per maand, gemiddelde beschikbaarheid per buurt, overzicht van gemiddelde huurprijs per buurt. Andere managementoverzichten zijn ook mogelijk, ga daarvoor op zoek naar online voorbeelden | Must |
| Layout idem als insideairbnb.com | Could |
| Locaties van zoekresultaat zichtbaar op kaar | Could |

Tabel 1: Geprioriteerde usecases

# Technisch ontwerp

## 3.1 Architectuur

Voor deze opdracht is er gekozen voor een API en een web applicatie. De web applicatie is geschreven in Next.js. Next.js is een front-end framework dat is gebouwd op React.js (Next.js, z.d.). Dit framework maakt het mogelijk om aan de usecases te voldoen. Het maakt het namelijk mogelijk om interactieve web applicaties te bouwen wat nodig is voor onder andere het filteren.

De front-end heeft data nodig om te kunnen tonen aan de eindgebruiker. Deze data wordt opgehaald vanuit de API. De API is geschreven in asp.NET op versie 8.0, dit is de laatste versie met ‘longtime support’ (.NET, z.d.). Er wordt gebruik gemaakt van Entity Framework Core om connectie te leggen met de database. De data is afkomstig van Inside Airbnb.

Om het autorisatie en authenticatie mogelijk te maken is er gebruik gemaakt van Auth0. Auth0 heeft dit al ingebakken waardoor je er zelf niet meer over na hoeft te denken (Lucas, Michiel, 2022).

## Packages

Voor het ontwikkelen is er gebruik gemaakt van externe packages. Deze packages beschikken over functionaliteit die gebruikt wordt in de applicaties. Er is bewust alleen voor packages gekozen waarvan de functionaliteit niet (eenvoudig) zelf gebouwd kan worden.

Packages in de API:

Afbeelding met tekst, schermopname, software, Multimediasoftware

Automatisch gegenereerde beschrijving

Packages in de web applicatie:

Afbeelding met tekst, schermopname, menu, Lettertype

Automatisch gegenereerde beschrijving

# Performance

In dit hoofdstuk ga ik in op de verschillende performance optimalisaties die zijn uitgevoerd op de Inside AirbnB webapplicatie. Eerst zal ik een nulmeting uitvoeren om de prestaties van de API te tonen voordat er verbeteringen zijn toegevoegd. Hierna zal ik proberen verbeteringen toe te voegen en dit vergelijken met de nulmeting. Ik zal na de nulmeting niet meer mijn test aanpak veranderen om er voor te zorgen dat de resultaten zo accuraat mogelijk zijn.

## 4.1 Nulmeting

Ik heb om een nulmeting te kunnen ophalen NBomber laten uitvoeren op de endpoint waar alle listing coördinaten worden opgehaald. NBomber is een load-testing framework (NBomber, z.d.). Ik heb gebruik gemaakt van een rate van 55, met een interval van 1 seconde gedurende 30 seconde zoals is te zien in figuur 1. Ik heb meerdere testen uitgevoerd op een aantal verschillende configuraties, maar uiteindelijk is dit de beste gebleken voor de nul meting. Deze configuratie zal ik niet meer aanpassen voor de volgende testen.

Afbeelding met tekst, schermopname, Lettertype, lijn

Automatisch gegenereerde beschrijving

Figuur 1: Configuratie NBomber

Afbeelding met tekst, schermopname, Lettertype, lijn

Automatisch gegenereerde beschrijving

Figuur 2: Nulmeting Nbomber summary

Afbeelding met tekst, schermopname, lijn, Perceel

Automatisch gegenereerde beschrijving

Figuur 3: Nulmeting througput & latency

In figuur 2 en 3 zijn de resultaten van de nulmeting weergegeven. In figuur 2 is te zien dat er geen fails zijn, maar in figuur 3 zie je dat na 10 seconde de applicatie de load niet meer goed aan kan en gaat de response tijd flink omhoog. Zo is de response tijd onder 7 tiende van een seconde na 10 seconde voor alle requests en zit dit al op 4 seconde na 15 seconde testen. De bedoeling is dat de applicatie verbeterd wordt waardoor de resultaten ten opzichte van deze nulmeting ook worden verbeterd.

## Redis caching

Caching is een data opslag voor eerder opgehaalde data (AWS, z.d.). Dit zorgt ervoor dat ik de data gelijk uit de opslag kan halen en kan ‘returnen’. Hiervoor hoeft de data dan, na de eerste keer, niet elke keer uit de database te komen.

Afbeelding met tekst, schermopname, Lettertype, lijn

Automatisch gegenereerde beschrijving

Figuur 4: Redis cache configuratie

Afbeelding met tekst, schermopname, software, Besturingssysteem

Automatisch gegenereerde beschrijving

Figuur 5: Eindpoint met rediscache

(De screenshots van de code zijn achteraf gemaakt waardoor er errors in de code zijn)

Om redis te gebruiken heb ik ten eerste een Redis image gepulled op docker. Hier wordt de cache in opgeslagen. Vervolgens moet er in de code ook nog wat gebeuren. Zo moet er een configuratie komen waar in dit geval ‘localhost:6379’ komt te staan (zie figuur 4), dit is het adres en poort van de Redis cache. Daarnaast is de endpoint nu ook aangepast (zie figuur 5). Hiervoor wordt er nu eerst gecontroleerd of er al een bestaande cache is, zo ja dan wordt het daar uitgehaald. Als dit nog niet het geval is wordt het zoals eerder uit de database opgehaald en vervolgens in de cache gezet.

Afbeelding met tekst, schermopname, software, Lettertype

Automatisch gegenereerde beschrijving

Figuur 6: Redis cache NBomber summary

Afbeelding met tekst, schermopname, lijn, Perceel

Automatisch gegenereerde beschrijving

Figuur 7: Redis cache througput & latency

Na het uitvoeren van de Nbomber test was ik verbaasd. Ik zag gelijk al dat er een aantal fails waren (zie figuur 6), waar dit eerst helemaal niet het geval was. Na het bekijken van de grafieken (zie figuur 7). Zag ik dat de applicatie slechter presteerde dan tijdens de nulmeting. Na 10 seconde is de response tijd al ongeveer 4 keer groter dan tijdens de nulmeting, waarna het vervolgens omhoog schier en uiteindelijk crasht.

## Memory cache

Nadat de teleurstellende resultaten van Redis heb ik geprobeerd om gebruik te maken van Memory cache.



Figuur 8: Configuratie memory cache

Afbeelding met tekst, schermopname, software, Lettertype

Automatisch gegenereerde beschrijving

Figuur 9: Endpoint met memory cache

De endpoint (zie figuur 9) is amper verandert ten opzichte van Redis. Er zijn minimale veranderingen en het idee is het zelfde. De configuratie (zie figuur 8) is wel versimpeld. Hier hoeft niks geconfigureerd te worden, maar hoeft de memory cache alleen maar gebruikt te worden.

Afbeelding met tekst, software, nummer, Lettertype

Automatisch gegenereerde beschrijving

Figuur 10: Memory cache Nbomber summary

Afbeelding met tekst, schermopname, lijn, Perceel

Automatisch gegenereerde beschrijving

Figuur 11: Memory cache througput & latency

De memory cache is een wereld van verschil ten opzichte van de Redis cache. Het viel meteen op dat alle resultaten onder de 800ms zitten bij de summary (zie figuur 10). Dit is een flinke verbetering ten opzichte van de nulmeting. En na het kijken van de grafieken in figuur 11 ziet het er een stik beter uit. De response tijd is op 10 seconde al met een halve seconde verbeterd en blijft gedurende de test onder de halve seconde waar ik al vrij te vrede mee ben.

## AsNoTracking

Nadat de caching de prestaties hebben verbeterd ben ik gaan kijken naar andere mogelijkheden om de applicatie nog beter te laten presteren. Ik heb geprobeerd om AsNoTracking te gebruiken bij het uitvoeren van de queries. Dit zorgt ervoor dat veranderingen aan de data niet worden bijgehouden door efcore wat geheugen en ‘processing power’ bespaard (Sahoo, Sayta Prakash, 2023).

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

Figuur 12: Query met AsNoTracking

AsNoTracking is een simple toevoeging aan de query zoals is te zien in figuur 12.

Afbeelding met tekst, schermopname, Lettertype, nummer

Automatisch gegenereerde beschrijving

Figuur 13: AsNoTracking Nbomber summary

Afbeelding met tekst, schermopname, lijn, Perceel

Automatisch gegenereerde beschrijving

Figuur 14: AsNoTracking througput & latency

De resultaten lijken iets constanter te zijn dan eerder al is de piek performance wel iets minder. (zie figuur 14).

## Indexing

Voor het ophalen van details van een listing wordt er gezocht op het ‘id’ van een listing. Om dit sneller te laten verlopen heb ik indexing toegepast op de kolom ‘id’ in de listing tabel. Indexing zorgt er voor dat queries sneller verlopen (W3Schools, z.d.).



Figuur 15: Listings details zonder indexing



Figuur 16: Listing details met indexing

Om te testen of de indexing verschil maakt voor het ophalen van de listing details heb ik het uitgetest door via de front-end requests te doen. Ik heb hiervoor de netwerk tab in de browser gebruikt om het verschil te kunnen inzien (zie figuur 10 en 11). Zoals te zien is, is de response tijd ongeveer gehalveerd.

## Conclusie

Ik heb een aantal technieken toegepast om de prestaties van de applicatie te verbeteren. Deze waren niet allemaal even succesvol. Er is gebruik gemaakt van NBomber en de netwerk tab in de browser om de prestatie verschillen te kunnen meten.

Ten eerste is er natuurlijk een nulmeting gehouden, zodat er aan het eind een goede vergelijking gemaakt kan worden. Tijdens de nulmeting zat de response tijd onder de 700ms na 10 seconde, maar daarna schoot het omhoog.

Er is caching toegepast. Dit werd eerst gedaan doormiddel van Redis, maar dit bleek de prestaties alleen maar te verslechteren. Er is vervolgens Memory cache toegepast. Dit maakte een groot verschil. Toen zat de response tijd onder de 150ms na 10 seconde testen en ging het langzaam iets omhoog tot aan het einde van de test.

Hoewel ik al erg tevreden was met het resultaat van de cache heb ik AsNoTracking toegepast op de query. Dit zorgde er voor dat de data van de queries niet worden ‘getracked’ waardoor de prestaties wat constanter bleven in plaats van dat dit langzaam oploopt. De absolute performance werd hierdoor wel iets benadeeld, maar als de applicatie veel requests binnen krijgt verbeterd dit wel de algehele performance.

Ten slotte heb ik nog indexing toegepast. Dit heeft geen effect gehad op het ophalen van alle listings en valt dus niet met de nulmeting zoals hierboven beschreven te vergelijken. Hiervoor is een nieuwe nulmeting uitgevoerd op de desbetreffende fetch in de front-end. Het indexen maakte weldegelijk verschil, zo werd de response tijd ongeveer gehalveerd.

Al met al is de prestatie van de applicatie flink verbeterd ten opzichte van de nulmeting(en) en ben ik tevreden met hoe deze nu presteert.

# 5. Security

In dit hoofdstuk wordt uitleg gegeven op de security issues die zijn gevonden met OWASP ZAP.

## 5.1. OWASP ZAP

De onderstaand waarschuwingen (zie figuur 17) zijn gevonden na het uitvoeren van OWASP ZAP. Deze ga ik oplossen.

Afbeelding met tekst, Lettertype, schermopname

Automatisch gegenereerde beschrijving

Figuur 17: Waarschuwingen van OWASP ZAP

Na het bekijken van de details van de waarschuwingen heb ik de onderstaande code toegepast (zie figuur 18).

Afbeelding met tekst, schermopname, software, Multimediasoftware

Automatisch gegenereerde beschrijving

Figuur 18: Response headers

Ik heb er voor gekozen om de aanbevelingen van OWASP ZAP op te zoeken en toe te passen. Voor de cache control heb ik er voor gekozen om no-cache te gebruiken. Ik heb hiervoor gekozen, omdat er anders data wordt opgeslagen in de browser. Hoewel dit in principe niet erg is, omdat het geen sensitive data is of dat er toch niks wordt aangepast, heb ik er toch voor gekozen om dit niet te doen omdat alle listings misschien te veel geheugen in beslag nemen in de browser.

Daarnaast heb ik de Strict-Transport-Security waarschuwing niet kunnen oplossen in localhost. In de front-end kreeg ik een SSL error. Deze waarschuwing kan worden aangepakt door ‘UseHsts’(Convesio, 2024) te gebruiken. Dit forceert een SSL (White, Daniel A., 2018). Het probleem is wel dat dit alleen in productie gebruikt kan worden en niet op localhost (Microsoft, z.d.).

## 5.2. Conclusie

Na de aanpassingen zijn alle waarschuwingen, op de strict-transport-security header op localhost na, die OWASP ZAP aangaf in figuur 17 verholpen (zie figuur 19).

Afbeelding met tekst, schermopname, Lettertype, Merk

Automatisch gegenereerde beschrijving

Figuur 19: Resultaat OWASP ZAP na aanpassingen

# Literatuurlijst

* .NET (z.d.). Download .NET. Geraadpleegd op 6 juni 2024, van <https://dotnet.microsoft.com/en-us/download/dotnet>
* AWS (z.d.). Caching overview. Geraadpleegd op 6 juni 2024, van <https://aws.amazon.com/caching/>
* Convesio (2024, 21 januari). Understanding the HSTS Error: Causes, Solutions, and Prevention. Geraadpleegd op 7 juni 2024, van <https://convesio.com/knowledgebase/article/understanding-the-hsts-error-causes-solutions-and-prevention>
* Lucas, Michiel (2022, 2 februari). Auth0: het platform voor betere beveiliging? Geraadpleegd op 7 juni 2024, van <https://www.makeitfly.group/nl/blog/auth0-het-platform-voor-betere-beveiliging>
* Microsoft (z.d.). Enforce HTTPS in ASP.NET Core. Geraadpleegd op 7 juni 2024, van <https://learn.microsoft.com/en-us/aspnet/core/security/enforcing-ssl?view=aspnetcore-8.0&tabs=visual-studio%2Clinux-ubuntu>
* NBomber (z.d.). NBomber Docs. Geraadpleegd op 6 juni 2024, van <https://nbomber.com/docs/getting-started/overview/>
* Next.js (z.d.). About React and Next.js. Geraadpleegd op 6 juni 2024, van <https://nextjs.org/learn/react-foundations/what-is-react-and-nextjs>
* Sahoo, Sayta Prakash (2023, 30 april). What is the use of AsNoTracking() in Entity Framework? Geraadpleegd op 6 juni 2024, van <https://medium.com/@satyaprakashsahoo.blog/what-is-the-use-of-asnotracking-in-entity-framework-cb6c453d1f1>
* W3Schools (z.d.). SQL CREATE INDEX Statement. Geraadpleegd op 7 juni 2024, van <https://www.w3schools.com/sql/sql_create_index.asp>
* White, Daniel A. (2018, 29 juni). What is the difference between app.UseHsts() and app.UseExceptionHandler()? Geraadpleegd op 7 juni 2024, van <https://stackoverflow.com/questions/51093139/what-is-the-difference-between-app-usehsts-and-app-useexceptionhandler>

# Bijlagen

## Bijlage 1: Casus Inside Airbnb

[Casus InsideAirbnb-NoTS-Azure.pdf](Casus%20InsideAirbnb-NoTS-Azure.pdf)