Datagedreven webapplicatie getest voor high performance en security door Michelle Jolink, 07-06-2024, Course NotS-WAPP

Ontwerpdocument Inside AirBnB Parijs

2019

Inhoudsopgave

[1. Inleiding 2](#_Toc168658681)

[2. Functioneel ontwerp 2](#_Toc168658682)

[3. Technisch ontwerp 2](#_Toc168658683)

[4. Performance 4](#_Toc168658684)

[4.1 Listings 4](#_Toc168658685)

[Nulmeting listings 4](#_Toc168658686)

[AsNoTracking 5](#_Toc168658687)

[Resultaten AsNoTracking 6](#_Toc168658688)

[Redis Caching 6](#_Toc168658689)

[4.2 Statistieken admins 8](#_Toc168658690)

[Indexing 8](#_Toc168658691)

[Caching 8](#_Toc168658692)

[4.4. Conclusie 9](#_Toc168658693)

[5. Security 10](#_Toc168658694)

[5.1. OWASP ZAP 10](#_Toc168658695)

[Gevonden security issues 10](#_Toc168658696)

[X-Content-Type-Options Header Missing 10](#_Toc168658697)

[5.2 Package vulnerabilities 11](#_Toc168658698)

[5.2.1 Frontend 11](#_Toc168658699)

[5.2.2 Backend 11](#_Toc168658700)

[5.3 Conclusie 11](#_Toc168658701)

# Inleiding

* Korte beschrijving casus

# Functioneel ontwerp

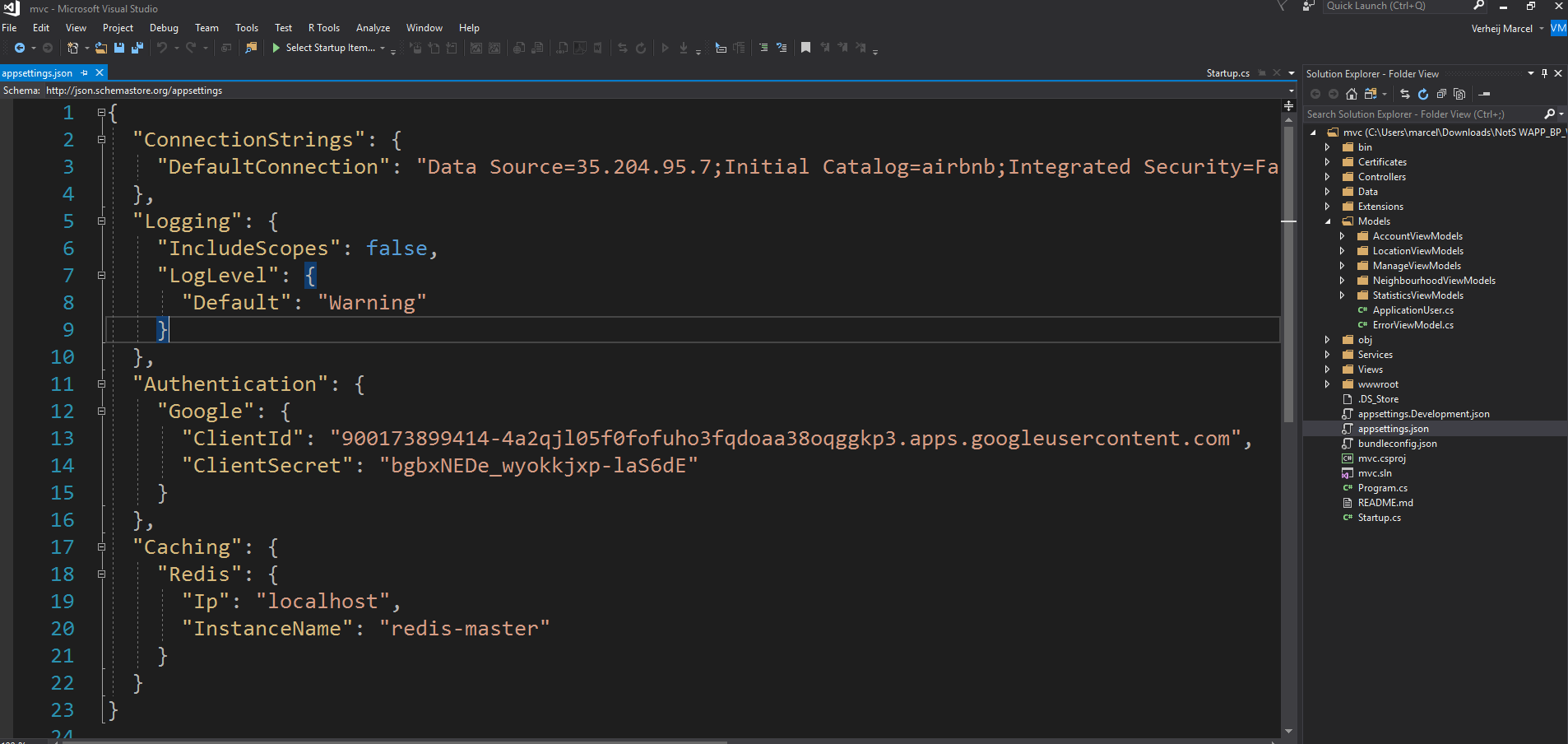
* Use Cases

|  |  |
| --- | --- |
| Use case | Prioriteit |
| Registeren 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 de mapbox API. | Must |
| Details per item waarop gefiltered is: #overnachtingen, #opbrengst in de maand. | Must |
| Er moeten rollen toegevoegd en toegekend worden aan 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 kaart. | Could |

# Technisch ontwerp

* Architectuur
* Overzicht gebruikte frameworks
* Overzicht gebruikte packages incl. versies

Configuratie externe services, etc.:



Figuur 1

# Performance

## 4.1 Listings

### Nulmeting listings

Om de verbeteringen in performance van de applicatie goed in beeld te krijgen op basis van de aanpassingen wordt er een nulmeting gedaan. Deze test zal gedaan worden met NBomber. Hierbij heb ik mijn laptop zo schoon mogelijk gehouden, hiermee bedoel ik dat ik geen andere programma’s open heb dan visual studio en mijn laptop voor de test heb herstart. Voor NBomber worden de volgende instellingen gebruikt:

Afbeelding met tekst, Lettertype, schermopname

Automatisch gegenereerde beschrijving

De nulmeting wordt gedaan op de API route die bij het opstarten van de applicatie alle 74.000+ listings ophaalt. Deze route haalt alleen de longitude, latitude en id op, maar ondanks dat is dit alsnog veeleisend.

De aangesproken endpoint voor het bepalen van de prestatie verbeteringen is:

* **GET /api/Listings/geoinfo**

Afbeelding met schermopname, Perceel, lijn, tekst

Automatisch gegenereerde beschrijving

Zoals te zien is reageert de API in het begin erg snel, maar na 15 seconden zien we dat de vertragingen beginnen op te lopen.

Afbeelding met tekst, schermopname, lijn, Lettertype

Automatisch gegenereerde beschrijving

|  |  |  |
| --- | --- | --- |
| **Minimale responsetijd** | **Maximale responsetijd** | **Gemiddelde responsetijd** |
| 0.2129 seconden | 20.0492 seconden | 9.44228 seconden |

Er falen ook geen requests, dus dat is positief. Wel duren de meeste requests langer dan 1.2 seconden. Dit willen we natuurlijk verbeteren. Naar mate de tijd en requests toenemen is de zien dat de responsetijd boven de 10 seconden komt. Dit is wel erg hoog.

### AsNoTracking

Entity Framework biedt een aantal opties voor het verbeteren van de prestaties. Een van deze opties is AsNoTracking(). Met deze optimalisatie kan er aangeven worden dat Entity Framework de resultaten van een query niet moet bijhouden. Dit betekent dat Entity Framework geen aanvullende verwerking of opslag uitvoert van de entiteiten die door de query worden geretourneerd. Wat ervoor zorgt dat er minimaal verbruik gemaakt wordt van memory. (Bron)

AsNoTracking () moet alleen maar gebruikt worden wanneer de intentie is om data op te halen en wijzigingen niet bij te houden.

**Aangepaste code:**

**Afbeelding met tekst, schermopname, software, Lettertype

Automatisch gegenereerde beschrijving**

Zoals te zien is heb ik de AsNoTracking heel gemakkelijk toe kunnen voegen aan de pipeline. Laten we kijken wat dit voor de performance heeft gedaan:

### Resultaten AsNoTracking

Afbeelding met schermopname, tekst, lijn, Perceel

Automatisch gegenereerde beschrijving

Zoals we kunnen zien is de piek ongeveer 5 seconden verbeterd, want nu loopt de piek pas op na 20 seconden.

Afbeelding met tekst, schermopname, lijn, Lettertype

Automatisch gegenereerde beschrijving

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Minimale responsetijd** | **Maximale responsetijd** | **Gemiddelde responsetijd** |
| ***Nulmeting*** | 0.21290 seconden | 20.0492 seconden | 9.44228 seconden |
| ***Na asNoTracking*** | 0.20309 seconden | 16.8268 seconden | 7.34436 seconden |
| ***Verbetering*** | 0,00981 seconden | 3,2224 seconden | 2,09792 seconden |

Ook zien we hier dat een deel van de requests die eerst in het gebied > 1.2s zaten nu in het gebied tussen 800ms en 1.2s zitten. De minimale responsetijd is minimaal afgenomen, maar bij de maximale en gemiddelde responsetijd is al zeker verbetering te zien. De maximale responsetijd is van 20.04 naar 16.8 gegaan en de gemiddelde responsetijd van 9.4s naar 7.3s. Al een stuk beter dus! Toch streven we naar meer!

### Redis Caching

//// TODO

## 4.2 Statistieken admins

Een endpoint om admin statistieken data op te halen voert een JOIN uit en is erg langzaam bij aanroepen. Hiervoor wil ik ook de performance verbeteren. Ik gebruik dezelfde NBomber instellingen als bij nulmeting listings.

### Indexing

### Caching

## 4.4. Conclusie

**TODO**

Als de hoogste aantal requests per seconde gepakt worden bestaat er een verbetering van 2122.22%. De grootste verbetering ontstaat door gebruik te maken van In-Memory caching, wat de latency omdat de applicatie niks van de database hoeft op te halen.

# 5. Security

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

## 5.1. OWASP ZAP

Om security issues op te sporen zal ik met ZAP een scan uitvoeren op mijn API: https://localhost:7049/

Om de scan uit te voeren gebruik ik 1 specifieke route, omdat je met ZAP niet de base url mee kang even end at ZAP zelf de endpoints vind. Ik heb gecontroleerd of bij verschillende routes verschillende waarschuwingen egtoond warden, maar dit was niet het geval. Daarom ga ik de scan uitvoeren op ***/Listings/geoinfo.***

### Gevonden security issues

Afbeelding met tekst, schermopname, Lettertype, nummer

Automatisch gegenereerde beschrijving

### X-Content-Type-Options Header Missing

Deze header zorgt ervoor dat de browser niet gaat proberen te raden wat de MIME type is van een bestand dat wordt opgehaald vanuit de server. Dit heb ik opgelost met de onderstaande code waarbij ik de header ‘*X-Content-Type-Options*’. Zie figuur…

Afbeelding met tekst, schermopname, Lettertype, lijn

Automatisch gegenereerde beschrijving

Strict-Transport-Security Header Not Set

Deze header geeft aan dat het ook mogelijk is dat de gebruiker van het HTTP protocol gebruikt kan maken. Dit is natuurlijk niet altijd veilig. Daarom gaan we dit oplossen. Dit ga ik doen door bij hetzelfde codeblock als hierboven nog een header toe te voegen, genaamd ‘*Strict-Traponsport-Security*’

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

## 5.2 Package vulnerabilities

Om de beveiliging zo goed mogelijk te maken, controleert Snyk de dependencies op kwetsbaarheden in zowel de front- als backend van de applicatie.

### 5.2.1 Frontend

Om Snyk in mijn frontend applicatie te runnen, heb ik gebruik gemaakt van de extensie binnen Visual Studio Code. <https://docs.snyk.io/integrate-with-snyk/use-snyk-in-your-ide/visual-studio-code-extension>

Zoals hieronder te zien in figuur … heeft snyk in totaal 4 vulnerabilities gevonden, waarvan 1 wordt aangemerkt als een hoge vulnerability, de *nth-check* en 3 als medium, de *inflight*, *postcss* en *serialize-javascript*. Ook is te zien dat ze allemaal vanauit 1 dependency komen, namelijk *react scripts versie 5.0.1.*

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving



Om deze vulnerabilities op te lossen, moeten de nth-check, postcss en serialize-javascript dependencies geupgrade worden naar een hogere versie. Dit heb ik gedaan door de volgende code in de package.json:  
Afbeelding met tekst, Lettertype, schermopname, handschrift

Automatisch gegenereerde beschrijving

Voor de warning van inflight is helaas nog geen oplossing, zoals snyk zelf aangeeft:

Afbeelding met tekst, Lettertype, schermopname, wit

Automatisch gegenereerde beschrijving

Na alle dependencies warnings opgelost te hebben die mogelijk waren, heb ik een tweede scan uitgevoerd. Dit zijn dit de overgebleven vulnerabilities:

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

### 5.2.2 Backend

Om Snyk in mijn backend applicatie te runnen, heb ik gebruik gemaakt van de extensie binnen Visual Studio. <https://docs.snyk.io/integrate-with-snyk/use-snyk-in-your-ide/visual-studio-extension>

Na de eerste scan te hebben uitgevoerd zijn er 3 vulnerabilities gevonden, waarvan 1 met een hoge severity, 1 medium en 1 low, zie figuur hieronder.

Afbeelding met tekst, schermopname, software, Lettertype

Automatisch gegenereerde beschrijving

Als je naar heb tabje Installed en dan show unly vulnerable aanklikt zien we ook precies die 3 dependencies:

Afbeelding met tekst, schermopname, software, Multimediasoftware

Automatisch gegenereerde beschrijving

Ik heb dit opgelost door de volgende packages te upgraden

***System.Text.RegularExpressions*** van versie *4.3.0* naar *4.3.1*

Hierdoor is de issue van System.Text.RegularExpressions opgelost.

***Azure.Identity*** van versie *1.10.3* naar *1.11.3*

Hierdoor zijn de issues van Azure.Identity en Microsoft.Identity.Client verholpen.

Na deze aanpassingen heb ik de 2e scan uitgevoerd en bevinden zich geen zwakheden meer in de packages van de backend:



## 5.3 Conclusie

In dit security hoofdstuk hebben we verschillende stappen ondernomen om de veiligheid van de applicatie te waarborgen. Door een ZAP-scan uit te voeren op de API, specifiek op de route /Listings/geoinfo, zijn beveiligingsproblemen geïdentificeerd. Dit waren het ontbreken van de X-Content-Type-Options en Strict-Transport-Security headers. Deze problemen zijn succesvol opgelost door de juiste headers toe te voegen aan de serverconfiguratie.

Daarnaast heb ik met behulp van Snyk de kwetsbaarheden in de dependencies van zowel de frontend als backend van de applicatie opgespoord en aangepakt. In de frontend applicatie zijn vier kwetsbaarheden gevonden en drie ervan konden worden opgelost door dependencies te upgraden. In de backend applicatie zijn drie kwetsbaarheden geïdentificeerd en allemaal opgelost door de packages te updaten.

Al met al kunnen we concluderen dat de applicatie nu aanzienlijk verbeterd is vergeleken met de applicatie voor deze verbeteringen.