

# Testplan Integration and Communication



Naam: **Philippe Haffmans**  
Studentnummer: **1683290**  
Klas: **SIE-V2C**  
Vak: **TCIF-V2IAC1-15**  
Docent: **Jeroen Weber**  
Datum: **25-03-17**

Document versie: **1.0**

# Inhoudsopgave

Onderdeel	Pagina
1. Introductie	3
2. Testomgeving	4
3. Smoketest	5
4. Testcases black box test (integratietest)	6
5. Unit tests	7
6. Regressietests	7

# 1. Introductie

## Introductie

In dit testverslag ga ik het proces beschrijven dat ik afgegaan ben om mijn applicatie te testen. Dit is de eerste versie van mijn project en testverslag. De opdracht die ik gekozen heb is *Geef het aantal bewoners in een bepaald jaar in een wijk*.

## Beschrijving werking systeem en stappen om tot realisatie te komen

De eerste stap die ik heb genomen om mijn REST applicatie te laten werken is het opzetten van mijn project structuur. Ik ben begonnen met het opzetten van mijn werkomgeving. Als eerst heb ik Tomcat & Maven geïnstalleerd en geconfigureerd waarna ik een nieuw project aangemaakt heb in mijn IntelliJ IDEA. Hierna heb ik Maven- en web support toegekend aan het project en Jersey toegevoegd met behulp van een pom.xml. Vervolgens heb ik een main bestand aangemaakt die ik kan deployen met Tomcat. Om de applicatie te bouwen ben ik begonnen met het bouwen van de Inwoners klasse. Daarin staat een statische lijst met data. Ook zit er een SearchInwoners functie in die op naam en jaar kan zoeken en op basis daarvan een hoeveelheid inwoners terug kan sturen. Daarnaast zijn er getNames en getYear functies in deze klasse. Hiermee controleer ik of het ingevoerde jaartal en buurtnaam aanwezig zijn in de lijst.

De BuurtService klasse zelf krijgt via de URL een buurtnaam en jaartal mee die in binnen het request gebruikt kunnen worden om het juiste JSON object terug te sturen. Er wordt een result variabele aangemaakt waarin het jaartal komt. Uit de boolean containsname en containsyear komt een true of false waaruit geconcludeerd wordt of het ingevoerde jaartal en buurtnaam in de lijst voorkomt. Vervolgens wordt er in een if else statement gecontroleerd of er een correct resultaat terug gegeven kan worden en als dat niet zo is wordt er een foutmelding terug gestuurd.

## Testsoorten

De Rest service ga ik een smoketest, een black box test (functionele test), een unit test en eventuele regressietest uitvoeren. Met de smoketest ga ik de algemene werking van de service controleren. Vervolgens ga ik met de black box test de verschillende operaties beschrijven en de test cases testen. In de unit test ga ik nog een keer controleren of de tests. Tot slot ga ik in de regressietest de eventuele bugs beschrijven die ik tegen ben gekomen en hoe ik deze vervolgens opgelost heb.

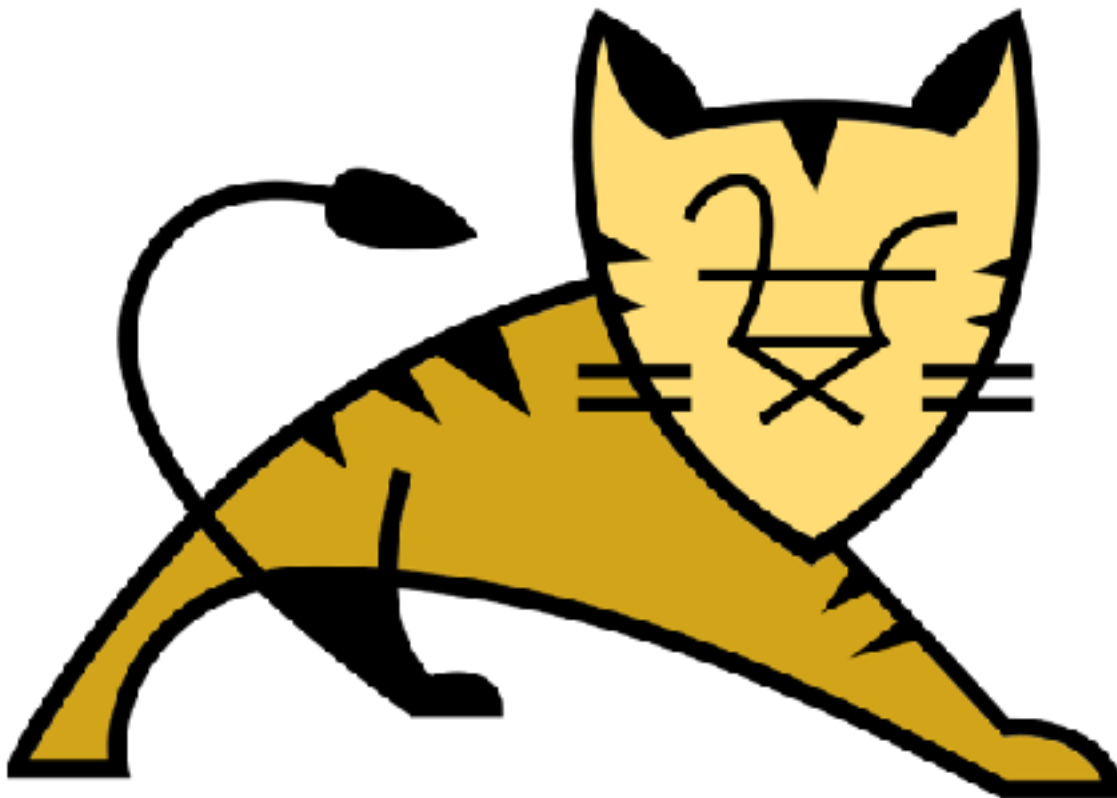
De resultaten van de diverse tests gaat terug te lezen zijn in het testrapport.

## 2. Testomgeving

Om de applicatie te testen heb ik gebruik gemaakt van een lokale Tomcat server die de Applicatie draait op poort **localhost:8000/**. Vanuit SoapUI kan ik vervolgens diverse requests afroepen op **localhost:8080/aantalinwoners/{buurtnaam}/{jaartal}**.



# SoapUI



### 3. Smoketest

Om de smoketest uit te voeren ga ik verschillende requests met diverse waarden afvuren op de applicatie om te controleren of hij de juiste waarden ook daadwerkelijk terug geeft. Deze test ga ik uitvoeren in SoapUI.

## 4. Testcases black box test (integratietest)

### Tool

De tool die ik gekozen heb om mijn requests te versturen en een response terug te ontvangen is de SoapUI. Dit is een tool die eenvoudig te gebruiken is waarin ik alle testgevallen in één keer kan testen en ook gelijk kan controleren of het verwachte resultaat ook daadwerkelijk terug gegeven wordt.

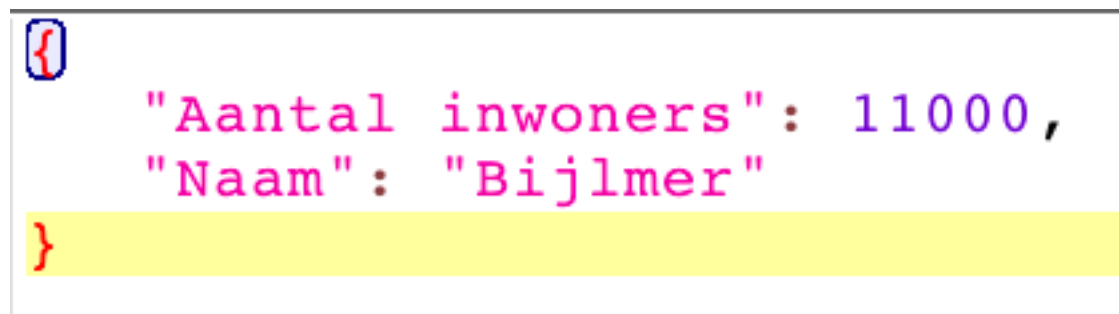
### Testcase 1

In dit testgeval voer ik de voer ik de onderstaande waarden in. De wijk is de Bijlmer en het jaartal is 2015. Als het goed is moet er 11000 terug gegeven worden.

#### Request URL

localhost:8000/aantalinwoners/Bijlmer/2015

#### Verwachtte Response



```
{  
  "Aantal inwoners": 11000,  
  "Naam": "Bijlmer"  
}
```

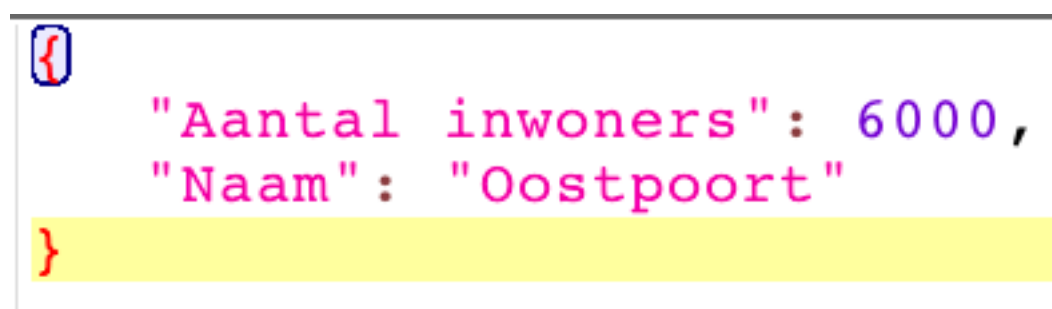
### Testcase 2

In dit testgeval voer ik de voer ik de onderstaande waarden in. De wijk is de Oostpoort en het jaartal is 2014. Als het goed is moet er 6000 terug gegeven worden.

#### Request URL

localhost:8000/aantalinwoners/Oostpoort/2014

#### Verwachtte Response



```
{  
  "Aantal inwoners": 6000,  
  "Naam": "Oostpoort"  
}
```

**Testcase 3**

In dit testgeval voer ik een naam en een naam in dat niet bestaat. In dat geval zal de service een foutmelding terug moeten geven.

Request URL

localhost:8000/aantalinwoners/Oostpoorts/2014

Verwachte Response

Naam wordt niet herkend!

**Testcase 4**

In dit testgeval voer ik een naam en een jaartal in dat niet bestaat. In dat geval zal de service een foutmelding terug moeten geven.

Request URL

localhost:8000/aantalinwoners/Bijlmer/2018

Verwachte Response

Jaar wordt niet herkend!

## 5. Unit tests

Helaas is het mij niet gelukt om een framework te implementeren waardoor ik de unit test niet heb kunnen doen.

## 6. Regressietests

Tijdens het testen van de Service ben ik een foutmeldingen tegen gekomen waarop ik de applicatie moest corrigeren.