

Testplan

April 14, 2017

Versie historie

03/11/2017 Initiële versie

03/28/2017 Unit tests output toegevoegd

04/13/2017 Smoketest uitgebreid

04/14/2017 Integratietests uitgebreid

Introductie

In dit document worden de testen van dit project beschreven. Dit project calculeert het aantal inwoners voor een wijk voor het gegeven jaar. De tests die zijn uitgewerkt zijn een aantal unit tests, en een integratietest. Deze tests zijn geschreven om de juiste werking van het systeem te garanderen.

1 Testomgeving

Tijdens deze tests wordt de Wijk implementatie getest. Hiervoor wordt JUnit gebruikt.

2 Smoketest

Om de kwaliteit van het systeem te waarborgen zijn een aantal tests geschreven. Deze testen de code in een normale situatie, een situatie waarin de naam van de wijk leeg is, een situatie waarin het jaar 0 is, en een situatie waarin het jaar een negatief getal betreft. Ook wordt er gekeken of er een fout wordt gegeven als de wijknaam null is.

| Input | Expected Output | Test failed? |
|---------------------------|----------------------|--------------|
| {"Nieuwegein Zuid", 2015} | 6458 | 0 |
| {"Amsterdam Zuid", 2015} | 6457 | 0 |
| {"", 2} | 6 | 0 |
| {"", 3} | 9 | 0 |
| {"Nieuwegein Zuid", 0} | 10 | 0 |
| {"Amsterdam Zuid", 0} | 9 | 0 |
| {"Nieuwegein Zuid", -1} | 7 | 0 |
| {"Amsterdam Zuid", -1} | 6 | 0 |
| {null, 2015} | NullPointerException | 0 |
| {null, 2017} | NullPointerException | 0 |

3 Integratietest

De integratietest doet hetzelfde als de Unit tests, alleen dan van buiten. Het gebruikte SOAP endpoint voor de data hieronder is

`http://localhost:8080/InwonersFinal-1.0-SNAPSHOT/services/Wijk`

Hieronder is zijn een aantal SOAP berichten en de antwoorden beschreven:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:inw="http://inwoners">
  <soapenv:Header/>
  <soapenv:Body>
    <inw:calcInwonersVoorWijk>
      <inw:wijk>Nieuwegein Zuid</inw:wijk>
      <inw:jaar>2015</inw:jaar>
    </inw:calcInwonersVoorWijk>
  </soapenv:Body>
</soapenv:Envelope>

```

Figure 1: SOAP-UI Request 1

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <calcInwonersVoorWijkResponse xmlns="http://inwoners">
      <calcInwonersVoorWijkReturn>6458</calcInwonersVoorWijkReturn>
    </calcInwonersVoorWijkResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

Figure 2: SOAP-UI Response 1

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:inw="http://inwoners">
  <soapenv:Header/>
  <soapenv:Body>
    <inw:calcInwonersVoorWijk>
      <inw:wijk>Amsterdam Zuid</inw:wijk>
      <inw:jaar>2015</inw:jaar>
    </inw:calcInwonersVoorWijk>
  </soapenv:Body>
</soapenv:Envelope>

```

Figure 3: SOAP-UI Request 2

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <calcInwonersVoorWijkResponse xmlns="http://inwoners">
      <calcInwonersVoorWijkReturn>6457</calcInwonersVoorWijkReturn>
    </calcInwonersVoorWijkResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

Figure 4: SOAP-UI Response 2

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:inw="http://inwoners">
  <soapenv:Header/>
  <soapenv:Body>
    <inw:calcInwonersVoorWijk>
      <inw:wijk>Amsterdam Zuid</inw:wijk>
      <inw:jaar>0</inw:jaar>
    </inw:calcInwonersVoorWijk>
  </soapenv:Body>
</soapenv:Envelope>

```

Figure 5: SOAP-UI Request 3

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <calcInwonersVoorWijkResponse xmlns="http://inwoners">
      <calcInwonersVoorWijkReturn>9</calcInwonersVoorWijkReturn>
    </calcInwonersVoorWijkResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

Figure 6: SOAP-UI Response 3

4 Unit tests

Er zijn een aantal unit tests geschreven die testen of het programma de juiste output geeft en zich gedraagt zoals de bedoeling is. Het framework dat hiervoor gebruikt wordt is JUnit. Deze tests worden automatisch uitgevoerd op het moment dat het programma gecompileerd wordt. Deze kunnen ook handmatig uitgevoerd worden door het volgende commando uit te voeren:

```
mvn install -B -U
```

```

@Test
public void calcInwonersVoorWijkNormal() throws Exception {
    assertEquals(wijk.calcInwonersVoorWijk("test", 2015), 6450);
    assertEquals(wijk.calcInwonersVoorWijk("Nieuwegein Zuid", 2015), 6458);
    assertEquals(wijk.calcInwonersVoorWijk("Amsterdam Zuid", 2015), 6457);
    assertEquals(wijk.calcInwonersVoorWijk("Amsterdam Bijlmer Arena", 2015), 6464);

    assertEquals(wijk.calcInwonersVoorWijk("test", 2016), 6454);
    assertEquals(wijk.calcInwonersVoorWijk("Nieuwegein Zuid", 2016), 6461);
    assertEquals(wijk.calcInwonersVoorWijk("Amsterdam Zuid", 2016), 6461);
    assertEquals(wijk.calcInwonersVoorWijk("Amsterdam Bijlmer Arena", 2016), 6467);

    assertEquals(wijk.calcInwonersVoorWijk("test", 2017), 6457);
    assertEquals(wijk.calcInwonersVoorWijk("Nieuwegein Zuid", 2017), 6464);
    assertEquals(wijk.calcInwonersVoorWijk("Amsterdam Zuid", 2017), 6464);
    assertEquals(wijk.calcInwonersVoorWijk("Amsterdam Bijlmer Arena", 2017), 6470);
}

@Test
public void calcInwonersVoorWijkEmptyName() throws Exception {
    assertEquals(wijk.calcInwonersVoorWijk("", 0), 0);
    assertEquals(wijk.calcInwonersVoorWijk("", 1), 3);
    assertEquals(wijk.calcInwonersVoorWijk("", 2), 6);
    assertEquals(wijk.calcInwonersVoorWijk("", 3), 9);
}

@Test
public void calcInwonersVoorWijkYearZero() throws Exception {
    assertEquals(wijk.calcInwonersVoorWijk("test", 0), 2);
    assertEquals(wijk.calcInwonersVoorWijk("Nieuwegein Zuid", 0), 10);
    assertEquals(wijk.calcInwonersVoorWijk("Amsterdam Zuid", 0), 9);
    assertEquals(wijk.calcInwonersVoorWijk("Amsterdam Bijlmer Arena", 0), 16);
}

@Test
public void calcInwonersVoorWijkYearNegative() throws Exception {
    assertEquals(wijk.calcInwonersVoorWijk("test", -1), 0);
    assertEquals(wijk.calcInwonersVoorWijk("Nieuwegein Zuid", -1), 7);
    assertEquals(wijk.calcInwonersVoorWijk("Amsterdam Zuid", -1), 6);
    assertEquals(wijk.calcInwonersVoorWijk("Amsterdam Bijlmer Arena", -1), 12);
}

@Test(expected = NullPointerException.class)
public void calcInwonersVoorWijkYearNull() throws Exception {
    wijk.calcInwonersVoorWijk(null, 2017);
    wijk.calcInwonersVoorWijk(null, 2017);
    wijk.calcInwonersVoorWijk(null, 2017);
    wijk.calcInwonersVoorWijk(null, 2017);
}

```

Figure 7: Unit tests source

```
-----  
T E S T S  
-----  
Running inwoners.WijkSoapBindingImplTest  
Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.025 sec  
Results :  
Tests run: 5, Failures: 0, Errors: 0, Skipped: 0
```

Figure 8: Unit tests output