Testplan

Kelvin Broekman

V2C

Versie 1.0

Introductie

De webservice heeft als functie het terugsturen van de standaarddeviatie uit de meegegeven getallenreeks. Er worden in totaal 5 getallen meegestuurd in de request. De webservice valideert deze getallen en maakt via een berekening een response aan die wordt teruggestuurd naar de receiver.

Testomgeving

Om de service te testen moet deze eerst met Tomcat worden opgestart. Dit gebeurt automatisch wanneer de service in Intellij wordt gestart. Nu de service is opgestart, moet er in SoapUI een request worden verstuurd.

Maak met SoapUI een verbinding naar de service. Stuur hierna een soap bericht naar de service bestaande uit een header en een body met 5 getallen.oor een nieuw project in SoapUI aan te maken en de URL van je wsdl uit de browser te gebruiken als wsdl. Nu er een verbinding ligt, wordt er op basis van input waarden een Soap enveloppe gegenereerd. Voer in de body van dit bericht 5 getallen in en stuur dit bericht naar de service. Als resultaat wordt hier een response gegeven met de standaarddeviatie van de ingevoerde getallenreeks.

Smoketest

Om de service te testen sturen we een leeg soap bericht naar de service om te kijken of de juiste output wordt teruggestuurd. In dit geval verwachten we een response terug zonder waarden. Als we een response krijgen betekent dit dat er een verbinding is met de service vanuit soapUI.

Black box test

Wanneer ik in SoapUI een request verstuur met 5 getallen, verwacht ik een response terug met de standaarddeviatie van die getallen. Voor de blackbox testen gebruik ik de volgende invoeringen in SoapUI.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Number1** | **Number2** | **Number3** | **Number4** | **Number5** | **Verwachte uitkomst** |
| 10 | 15 | 43 | 23 | 6 | 13.09 |
| 0 | 15 | 43 | 23 | 6 | Soap foutmelding |
|  |  |  |  |  | Response object |
| 1055 | 352 | 362 | 753 | 75 | 344.01 |

Unit test

Voor de Unit tests maak ik gebruik van het framework JUnit5. Dit is een framework wat al in Intellij beschikbaar is. Deze tests worden uitgevoerd voor de klassen Berekening en Deviatie.

**class** BerekeningTest {  
 @BeforeEach  
 **void** setUp() {  
 System.***out***.println(**"set up berekening test"**);  
 }  
  
 @AfterEach  
 **void** tearDown() {  
 System.***out***.println(**"after berekening test"**);  
 }  
  
 @Test  
 **void** bereken() {  
 System.***out***.println(**"berekening test"**);  
 }  
}

**class** DeviatieTest {  
 @org.junit.jupiter.api.BeforeEach  
 **void** setUp() {  
 System.***out***.println(**"set up test deviatie"**);  
 }  
  
 @org.junit.jupiter.api.AfterEach  
 **void** tearDown() {  
 System.***out***.println(**"after test deviatie"**);  
 }  
  
 @org.junit.jupiter.api.Test  
 **void** calculateDeviatie() {  
 System.***out***.println(**"deviatie test"**);  
 }  
}

Regressietest

De web service is uitgebreid getest. Er zijn hierbij geen bugs ontdekt.