Testrapport   
IAC Herkansingsopdracht 2017  
Smart Cities REST Service

“BMI Berekenen en CalorieVerbruik berekenen”

Jerre van den Berg

1649996

Contents

[Smoketest 3](#_Toc485981577)

[Functionele test 4](#_Toc485981578)

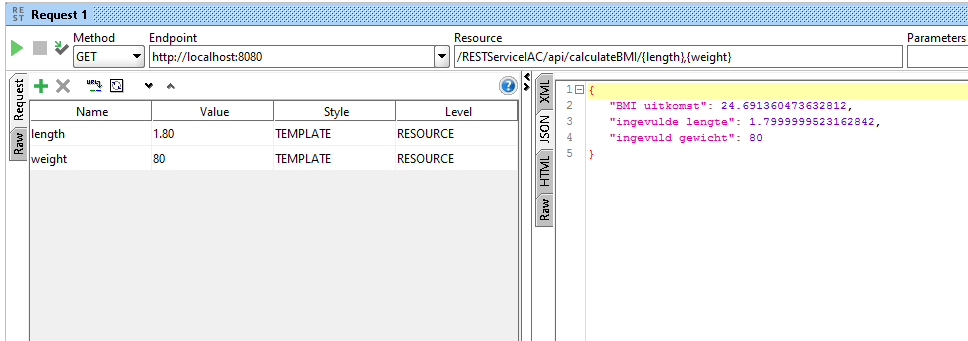
[Case 1 BMI Berekenen: Invoeren van een string 4](#_Toc485981579)

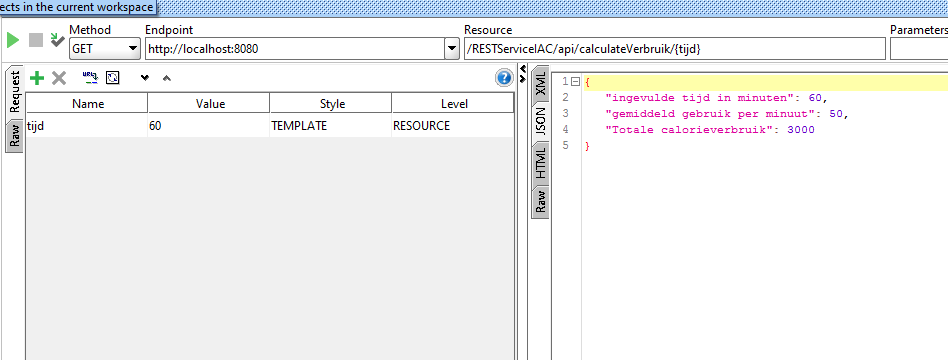
[Case 3: Invoeren van twee floats en de verwachtte reactie 5](#_Toc485981580)

[Unit Tests 6](#_Toc485981581)

# Smoketest

Om te controleren of de REST Service draait en deze ook een gewenste response teruggeeft is hiervoor een SoapUI test gedaan. Hieronder de screenshotss van de SoapUI test die teruggeeft dat de REST-service correct werkt en ook JSON teruggeeft. Beide methoden zijn getest.

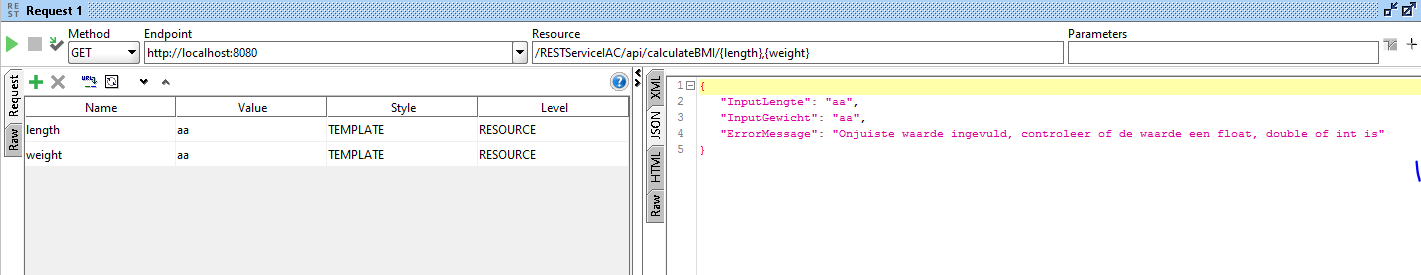


Hier de methode calculateVerbruik (die het calorie verbruik teruggeeft)  


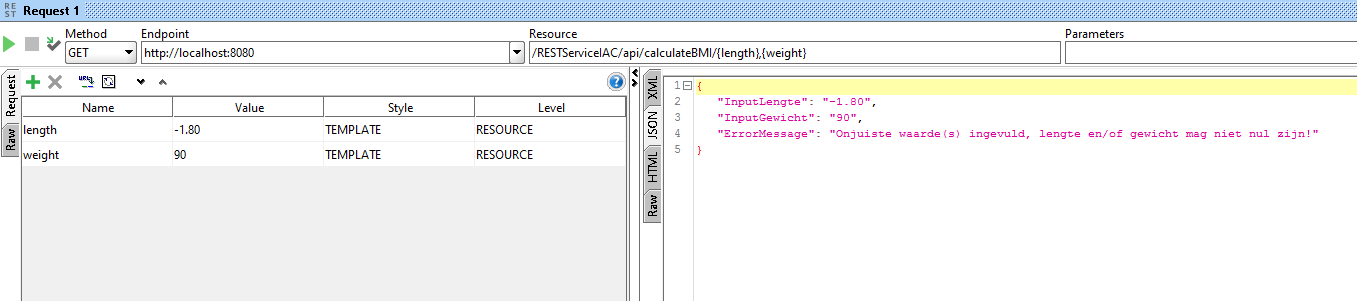
# Functionele test

In dit hoofdstuk wordt getest of de REST Service ook daadwerkelijk de juiste data teruggeeft op basis van de berekening. Hier zijn ook JUnit tests voor geschreven. Tevens is er gekeken of er error messages worden afgevangen. Er worden een aantal test scenarios doorgelopen, die gecontroleerd worden met behulp van SoapUI.

Case 1 BMI Berekenen: Invoeren van een string

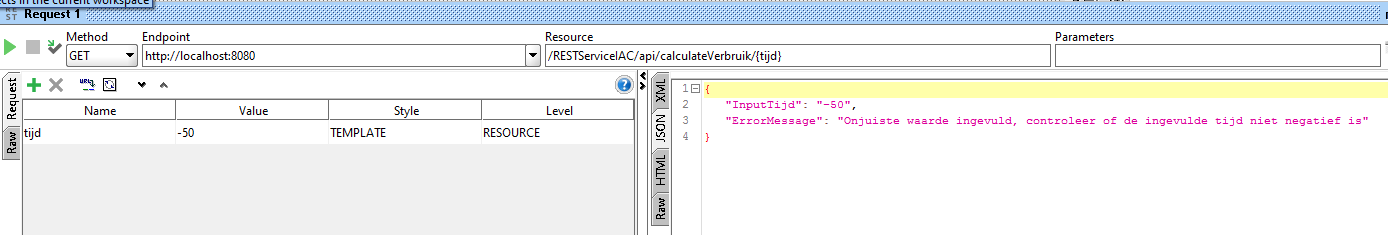


Zoals te zien is, bij het invoeren van een string wordt er in JSON een error message gegeneerd.  
  
Case 2: BMI Berekenen: Invoeren van een negatief getal



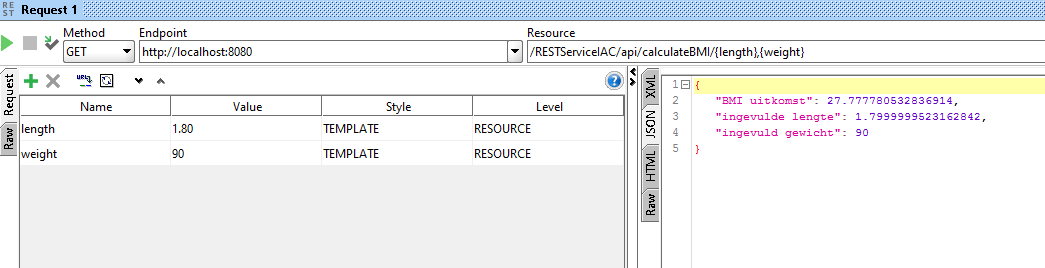
Zoals te zien, wordt het invoeren van een negatieve waarde afgevangen met een errormessage in JSON.

## Case 3: CalculateVerbruik – Negatieve tijd invullen



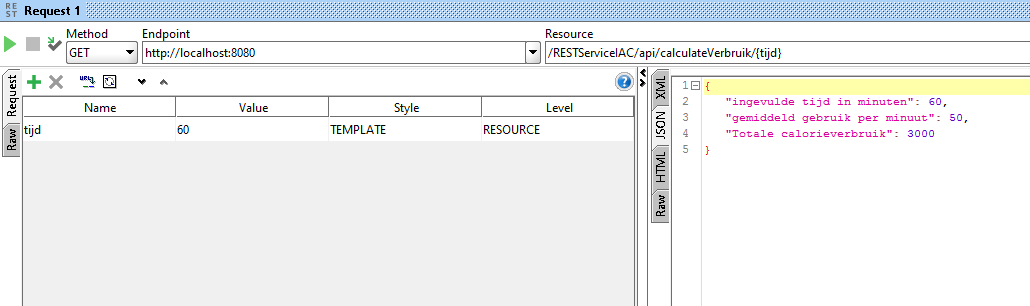
Bij het invoeren van een negatieve tijd wordt er in JSON een error message getoond.

## Case 4 : BMI Berekenen – Juiste berekening



De BMI berekening is als volgt: Gewicht / (Lengte \* Lengte).  
Bij de ingevuld waardes zou dan de BMI 27,7 zijn. Zoals te zien is klopt dit.

## Case 5: CalculateVerbruik – Juiste berekening



Hier is te zien dat de berekening van het calorieverbruik is opgesteld als zijnde:  
Totale calorieverbruik = Aantal minuten \* Gebruik per minuut  
60x50 = 3000 dus de functionele test is geslaagd.

# Unit Tests

**package** test;

**import** **static** org.junit.Assert.\*;

**import** jerrevandenberg.main.CalorieVerbruik;

**import** junit.framework.Assert;

**import** org.junit.Test;

**public** **class** TestCalorieVerbruik {

@Test

**public** **void** testGetTijd() {

CalorieVerbruik cv = **new** CalorieVerbruik(60);

Assert.*assertEquals*(cv.getTijdInMinuten(), 60);

}

@Test

**public** **void** testSetTijd(){

CalorieVerbruik cv = **new** CalorieVerbruik(60);

cv.setTijdInMinuten(40);

Assert.*assertEquals*(cv.getTijdInMinuten(), 40);

}

@Test

**public** **void** testBerekenVerbruik(){

CalorieVerbruik cv = **new** CalorieVerbruik(60);

**int** totaalVerbruik = cv.getVerbruikteCalorieen();

**int** verwachtteVerbruik = 60 \* 50;

Assert.*assertEquals*(totaalVerbruik, verwachtteVerbruik);

}

}

De test van de klasse ‘CalorieVerbruik’

**package** test;

**import** org.junit.BeforeClass;

**import** org.junit.Test;

**import** org.junit.Assert;

**import** jerrevandenberg.main.BMICalculation;

**public** **class** TestBMICalculation {

@SuppressWarnings("deprecation")

@Test

**public** **void** testGetLength(){

BMICalculation bmic = **new** BMICalculation(1.70f, 80f);

**float** testLength = bmic.getLength();

Assert.*assertEquals*(testLength, 1.70f, 0.00001f);

}

@Test

**public** **void** testSetLength(){

BMICalculation bmic = **new** BMICalculation(1.70f, 80f);

bmic.setLength(1.80f);

Assert.*assertEquals*(bmic.getLength(), 1.80f, 0.00001f);

}

@Test

**public** **void** testCalculateBMI(){

BMICalculation bmic = **new** BMICalculation(1.70f, 80f);

**float** calculatedBMI = bmic.getBMI();

**float** expectedBMI = 80f / (1.70f \* 1.70f);

Assert.*assertEquals*(calculatedBMI, expectedBMI, 0.00001f);

}

}

Dit is de test van de Java klasse ‘BMICalculcation’

De tests runnen succesful in zowel maven (mvn test) als JUnit debug (Eclipse).