



# Testplan template Integration and Communication CODE TCIF-V2IAC1-15

## HBO-ICT SIE Jaar 2 2016-2017

**Cursuseigenaar:** jeroen.weber@hu.nl

**Auteurs**

Alex Jongman  
Jeroen Weber

**Review**

BAR:  
Student

**Datum**

28 – 01 – 2017

**Versie 0.1**

© Hogeschool Utrecht, Utrecht, 2015

## Inhoudsopgave

Inhoudsopgave .....	2
1. Versie .....	3
2. Introductie.....	3
3. Testomgeving.....	3
4. Smoketest .....	3
5. Testcases black box test (integratietest of functionele test).....	3
6. Unit Tests .....	7

## 1. Versie

- Neem een versie-historie op. Verwijs naar het gebruikte testplan versie in je testrapport.

## 2. Introductie

- Ik ga de service DroneServices testen. Hiermee kan je informatie van een drone opslaan, lezen, bewerken en verwijderen. Dit word op het moment in het geheugen opgeslagen, maar kan uitgebreid worden met een .csv file of een database erachter.
- Als tools maak ik gebruik van JUnit 4.1 en SOAPUI. Via JUnit test ik iedere afzonderlijke functie, en met SOAPUI test ik of alle functies werken, en achter elkaar zouden kunnen werken.

## 3. Testomgeving

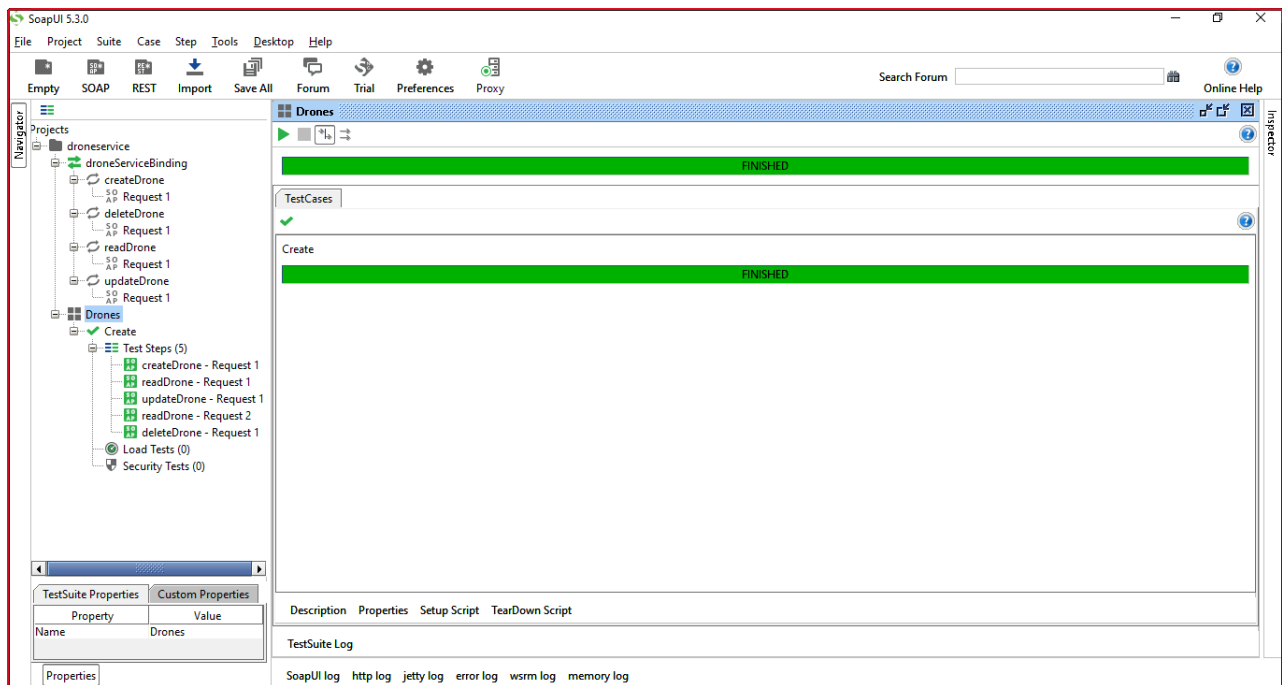
- Via JUnit ga ik de volgende services testen:
  - o createDrone
  - o readDrone
  - o updateDrone
  - o deleteDroneDit wordt dan getest of het individueel (inclusief create bij alle 4) werkt.
- Via SOAPUI ga ik dezelfde services testen, alleen dan op integratie gebied. Werken de services als ik ze achter elkaar uitvoer?

## 4. Smoketest

- Als smoketest heb ik ten eerste gekeken of ik op localhost het pad <http://localhost:8080/ws-producer/> en <http://localhost:8080/ws-producer/soap/> kon vinden. Deze kon ik vinden, al doet hij af en toe wat buggy ☺
- Ten tweede heb ik via SOAPUI gekeken of het programma werkte en of basic invoer werkte, dat kon.
- Tot slot heb ik gekeken of er een drone met hetzelfde ID kon worden gemaakt, en dit kan niet. Al krijg je daar geen custom error voor.

## 5. Testcases black box test (integratietest of functionele test)

- Voor deze test heb ik SOAPUI gebruikt. Ik heb eerst 4 testpaden gemaakt en daarna deze achter elkaar laten uitvoeren. Deze tests zijn gelukt, zoals te zien is in de screenshot hieronder.



De individuele testpaden zagen er als volgt uit:

### createDrone Request

Request:

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:cre="http://nl.hu.fnt/droneservice/createrequest">
  <soapenv:Header/>
  <soapenv:Body>
    <cre:createRequest>
      <cre:DroneID>12345</cre:DroneID>
      <cre:MotorFrame>MotorFrame 0.1</cre:MotorFrame>
      <cre:Motor>Motor 8000</cre:Motor>
      <cre:Transmitter>Transmitter 9000</cre:Transmitter>
      <cre:Receiver>Receiver 7000</cre:Receiver>
      <cre:Propellers>Propeller Series 5300</cre:Propellers>
      <cre:NumberOfPropellers>38</cre:NumberOfPropellers>
      <cre:Batteries>Wes-034</cre:Batteries>
    </cre:createRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <droneResponse xmlns="http://nl.hu.fnt/droneservice/response"
xmlns:ns2="http://nl.hu.fnt/droneservice/fault">
```

```

xmlns:ns3="http://nl.hu.fnt/droneservice/readrequest"
xmlns:ns4="http://nl.hu.fnt/droneservice/deletererequest"
xmlns:ns5="http://nl.hu.fnt/droneservice/updaterequest"
xmlns:ns6="http://nl.hu.fnt/droneservice/createrequest">
    <message>De drone is met succes geregistreerd</message>
</droneResponse>
</S:Body>
</S:Envelope>

```

## readDrone Request

### Request:

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:read="http://nl.hu.fnt/droneservice/readrequest">
    <soapenv:Header/>
    <soapenv:Body>
        <read:readRequest>
            <read:DroneID>12345</read:DroneID>
        </read:readRequest>
    </soapenv:Body>
</soapenv:Envelope>

```

### Response:

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <droneResponse xmlns="http://nl.hu.fnt/droneservice/response"
xmlns:ns2="http://nl.hu.fnt/droneservice/fault"
xmlns:ns3="http://nl.hu.fnt/droneservice/readrequest"
xmlns:ns4="http://nl.hu.fnt/droneservice/deletererequest"
xmlns:ns5="http://nl.hu.fnt/droneservice/updaterequest"
xmlns:ns6="http://nl.hu.fnt/droneservice/createrequest">
            <message>Drone{droneID=12345, motorFrame='MotorFrame 0.1',
motor='Motor 8000', transmitter='Transmitter 9000', receiver='Receiver
7000', propellers='Propeller Series 5300', numberOfPropellers=38,
batteries='Wes-034'}</message>
        </droneResponse>
    </S:Body>
</S:Envelope>

```

## updateDrone Request

### Request:

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:upd="http://nl.hu.fnt/droneservice/updaterequest">
    <soapenv:Header/>
    <soapenv:Body>
        <upd:updateRequest>
            <upd:DroneID>12345</upd:DroneID>
            <upd:MotorFrame>MotorFrame 0.1</upd:MotorFrame>
            <upd:Motor>Motor 8000</upd:Motor>

```

```

    <upd:Transmitter>Transmitter 9000</upd:Transmitter>
    <upd:Receiver>Receiver 7000</upd:Receiver>
    <upd:Propellers>Propeller Series 5300</upd:Propellers>
    <upd:NumberOfPropellers>38</upd:NumberOfPropellers>
    <upd:Batteries>Pim Razor 38000x</upd:Batteries>
  </upd:updateRequest>
</soapenv:Body>
</soapenv:Envelope>

```

#### Response:

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <droneResponse xmlns="http://nl.hu.fnt/droneservice/response"
      xmlns:ns2="http://nl.hu.fnt/droneservice/fault"
      xmlns:ns3="http://nl.hu.fnt/droneservice/readrequest"
      xmlns:ns4="http://nl.hu.fnt/droneservice/deleterrequest"
      xmlns:ns5="http://nl.hu.fnt/droneservice/updaterequest"
      xmlns:ns6="http://nl.hu.fnt/droneservice/createrequest">
      <message>Succes, nieuwe waardes van je drone:
      Drone{droneID=12345, motorFrame='MotorFrame 0.1', motor='Motor 8000',
      transmitter='Transmitter 9000', receiver='Receiver 7000',
      propellers='Propeller Series 5300', numberOfPropellers=38,
      batteries='Pim Razor 38000x'}</message>
    </droneResponse>
  </S:Body>
</S:Envelope>

```

### deleteDrone Request

#### Request:

```

<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:del="http://nl.hu.fnt/droneservice/deleterrequest">
  <soapenv:Header/>
  <soapenv:Body>
    <del:deleteRequest>
      <del:DroneID>12345</del:DroneID>
    </del:deleteRequest>
  </soapenv:Body>
</soapenv:Envelope>

```

#### Response:

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <droneResponse xmlns="http://nl.hu.fnt/droneservice/response"
      xmlns:ns2="http://nl.hu.fnt/droneservice/fault"
      xmlns:ns3="http://nl.hu.fnt/droneservice/readrequest"
      xmlns:ns4="http://nl.hu.fnt/droneservice/deleterrequest"
      xmlns:ns5="http://nl.hu.fnt/droneservice/updaterequest"
      xmlns:ns6="http://nl.hu.fnt/droneservice/createrequest">

```

```

        <message>Drone{droneID=12345, motorFrame='MotorFrame 0.1',
motor='Motor 8000', transmitter='Transmitter 9000', receiver='Receiver
7000', propellers='Propeller Series 5300', numberOfPropellers=38,
batteries='Pim Razor 38000x'} has been removed.</message>
    </droneResponse>
</S:Body>
</S:Envelope>

```

## 6. Unit Tests

- Ik heb voor de Unit Tests JUnit 4.1 gebruikt.
- Al heb ik ze vormgegeven als unit tests, onderwater zijn 3 van de 4 tests eigenlijk meer integratie testen. Dit komt omdat Read, Update en Delete nu niet kunnen zonder dat er eerst een create is. Ik heb dus voor die drie methoden een Create methode vanuit de implementatie klasse aangeroepen.

Code:

```

package nl.hu.iac.wsproducer;

import nl.hu.iac.wsinterface.*;

import java.math.BigInteger;

import static org.junit.Assert.assertEquals;
import static org.junit.Assert.assertTrue;

/**
 * Created by Tegrohir on 12-3-2017.
 */
public class DroneServiceImplTest {
    @org.junit.Test
    public void createDrone() throws Exception {
        System.out.println("\n\nCreate Drone test");
        CreateRequest request = new CreateRequest();
        createTestRequest(request);
        DroneServiceImpl impl = new DroneServiceImpl();
        boolean thrown = false;

        try {
            DroneResponse response = impl.createDrone(request);
            System.out.println(response.getMessage());
        } catch (Exception e) {
            System.out.println("Test Create eerst joh, stom programma
:(  ");
            thrown = true;
        }
        try {
            DroneResponse response2 = impl.createDrone(request);
            System.out.println(response2.getMessage());
        } catch (Exception e) {

```

```

        System.out.println("Exception gevangen");
        thrown = true;
    }
    assertTrue(thrown);
}

@org.junit.Test
public void readDrone() throws Exception {
    System.out.println("\n\nRead Drone test");
    DroneServiceImpl impl = new DroneServiceImpl();
    Drone drone = createTestDrone();
    ReadRequest request = new ReadRequest();
    request.setDroneID(drone.getDroneID());

    DroneResponse response = impl.readDrone(request);
    assertEquals(drone.toString(), response.getMessage());
}

@org.junit.Test
public void updateDrone() throws Exception {
    System.out.println("\n\nUpdate Drone test");
    DroneServiceImpl impl = new DroneServiceImpl();
    Drone drone = createTestDrone();
    UpdateRequest request = new UpdateRequest();
    createTestRequest(request);
    System.out.println(request.getPropellers());
    request.setDroneID(drone.getDroneID());

    DroneResponse response = impl.updateDrone(request);
    assertEquals("Succes, nieuwe waarden van je drone: " +
drone.toString(), response.getMessage());
}

@org.junit.Test
public void deleteDrone() throws Exception {
    System.out.println("\n\nDelete Drone test");
    DroneServiceImpl impl = new DroneServiceImpl();
    Drone drone = createTestDrone();
    DeleteRequest request = new DeleteRequest();
    request.setDroneID(drone.getDroneID());

    DroneResponse response = impl.deleteDrone(request);
    assertEquals(drone.toString() + " has been removed.",
response.getMessage());
}

private Drone createTestDrone() {
    CreateRequest request = new CreateRequest();
    createTestRequest(request);
    DroneServiceImpl impl = new DroneServiceImpl();
    try{
        impl.createDrone(request);
    }
}

```



```

    } catch (Exception e) {
        System.out.println("Drone already exists in Impl class");
    }
    Drone drone = new Drone(request);
    System.out.println(drone.toString());

    return drone;
}

private IRequest createTestRequest(IRequest request) {

    request.setDroneID(new BigInteger("12345"));
    request.setMotorFrame("MotorFrame 0.1");
    request.setMotor("Motor 8000");
    request.setTransmitter("Transmitter 9000");
    request.setReceiver("Receiver 7000");
    request.setPropellers("Propeller Series 5300");
    request.setNumberOfPropellers(new BigInteger("38"));
    request.setBatteries("Wes-034");
    return request;
}
}

```