

**Ministerie van Landbouw, Natuur en
Voedselkwaliteit**

**Webservice technische specificatie:
DFx (Berichtenboek)**

Versie 0.2
Assen, 11-04-2008

Opgesteld door:
A. Meinsma

Inhoudsopgave

Inhoudsopgave	2
Wijzigingshistorie	3
1 Inleiding	4
1.1 Doelgroep	4
1.2 Leeswijzer	4
2 De webservice	5
2.1 Globale werking van het systeemproces.....	5
2.2 Functionaliteit van de webservice	6
2.3 Transport level.....	6
3 Dataformaat van gegevens	8
3.1 Invoer	8
3.2 Webservice antwoord.....	9
3.2.1 Operatie geslaagd	9
3.2.2 Foutmelding	11
4 Webservice Interface specificatie	13
4.1 Bericht formaten (W3C Schema specificatie)	13
4.2 WSDL definitie van de webservice	16
5 Referenties	18

Wijzigingshistorie

Om doorgevoerde wijzigingen te kunnen verwerken is voor u van belang om exact te weten wat er per versie is aangepast en of deze aanpassing al dan niet consequenties heeft voor uw systeemontwikkeling of dat het alleen een tekstuele aanpassing betreft.

Onderstaand zijn de doorgevoerde wijzigingen vanaf versie 0.1 beschreven en is aangegeven waar deze in het document zijn aangebracht en of ze al dan niet van invloed zijn op de inhoud van het bericht.

Versie	Datum	Wijziging	Paragraaf	Impact op bericht?

1 Inleiding

1.1 Doelgroep

Dit document is bedoeld voor ontwerpers en ontwikkelaars en beschrijft de functionaliteit en gedrag van de webservice in termen van een technische specificatie.

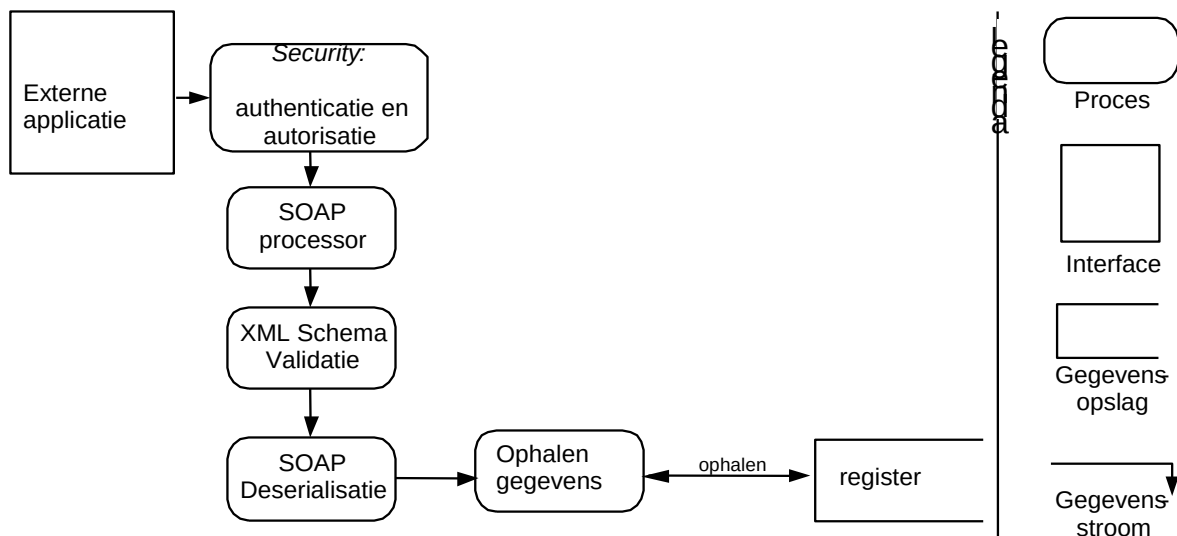
1.2 Leeswijzer

- In hoofdstuk 2 wordt het globale verwerkingsproces, de functionaliteit van de webservice en een aantal technische basiszaken beschreven.
- In hoofdstuk 3 staan de formaten van, en eisen aan de uit te wisselen gegevens.
- In hoofdstuk 4 staat de interface specificatie, en een aantal tabellen ter referentie.

2 De webservice

2.1 Globale werking van het systeemproces

In principe is het voor een ontwikkelaar die gebruik maakt van een webservice niet van belang om te weten hoe een webservice precies intern geïmplementeerd is. Toch is het wel van belang om het globale interne verwerkingsproces te begrijpen, omdat dit het gedrag van de webservice verklaart. Met name de volgorde waarin verschillende controles plaatsvinden geeft voor een ontwikkelaar aan in welke volgorde mogelijke fouten geconstateerd worden. Dit systeemgedrag geeft de ontwikkelaar een leidraad voor het implementeren van de communicatie in het algemeen, met name bij het afhandelen van foutmeldingen.



figuur 1: Diagram van de webservice

Zie voor een schematische weergave figuur 1. Hierin staat het systeemproces beschreven dat begint bij een externe applicatie die gegevens stuurt naar de webservice. Als eerste wordt het *security* beleid in de achtereenvolgende stappen toegepast; er wordt gecontroleerd of de gebruiker is wie hij zegt (authenticatie) en of deze gebruiker gebruik mag maken van deze dienst (autorisatie). Vervolgens wordt de inhoud van het bericht omgezet door een *SOAP processor*. Daarna wordt het SOAP-body gedeelte van het bericht (de payload) gevalideerd (*XML Schema validatie*) tegen het W3C XML Schema [3] voor berichten voor deze webservice (zie sectie 4.1 voor de definitie). Het SOAP bericht kan nu (zonder parse-fouten) omgezet worden in interne datatypes (*SOAP deserialisatie*). Daarna wordt het bericht, indien geen fouten zijn geconstateerd, aangeboden om de gegevens te verrijken met de formuliergegevens uit het register. Als er in één van de processtappen een foutsituatie optreedt wordt dit teruggemeld. Indien alle stappen met succes zijn doorlopen, krijgt de versturende partij het bericht met de verrijkte gegevens terug.

2.2 Functionaliteit van de webservice

De aangeboden functionaliteit van de webservice bestaat uit het ter beschikkingstellen van formuliergegevens.

2.3 Transport level

In deze subsectie wordt beschreven op welke (technische) manier een bericht aangeleverd kan worden aan het systeem. Dit gaat met name over de (transport level) security en de algemene inhoud van een bericht.

HTTPS

De webservice wordt NIET benaderd via http over ssl. De koppeling van de externe applicatie naar webservice gaat via een rechtstreekse en beveiligde lijn.

Basic Authentication

Een http request naar de webservice moet voorzien zijn van de http request-header behorende bij basic authentication ("*Authorization*"), zoals gespecificeerd in RFC2617 [2]. Als dit niet het geval is, of de meegestuurde credentials worden als niet geldig beschouwd (authenticatie of autorisatie mislukt), dan wordt een *http_authorized failed* response (statuscode 401) terug gestuurd. Als gebruikersnaam wordt een ABA nummer met het bijbehorende wachtwoord meegestuurd dat door DR is verstrekt. Als de authenticatie slaagt wordt de gebruiker bij het ABA nummer geautoriseerd.

Een voorbeeld (uit de RFC2617 specificatie). Als de webservice client in de gebruikersnaam "*Aladdin*" wil opsturen, en het wachtwoord "*open sesame*", dan moet het volgende http request header veld meegestuurd worden:

Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==

HTTP POST

Een bericht wordt naar de webservice verstuurd als postdata in een http post request [1]. De http postdata bestaat uit een XML bericht dat voldoet aan de SOAP specificatie [4].

SOAP

De WSDL definitie (sectie 4.2) schrijft voor hoe het SOAP bericht er uit moet zien.

Endpoint

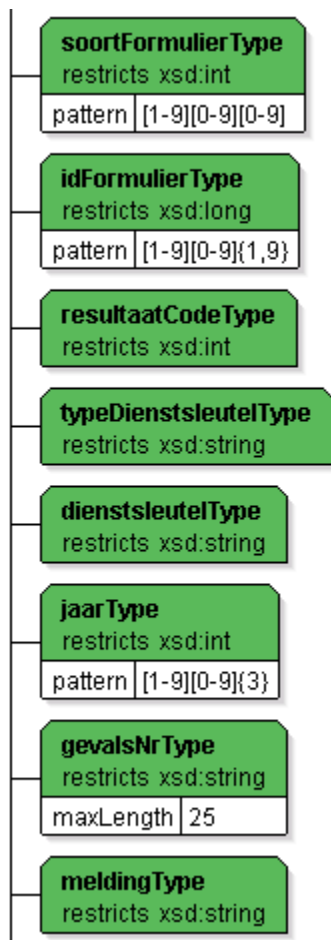
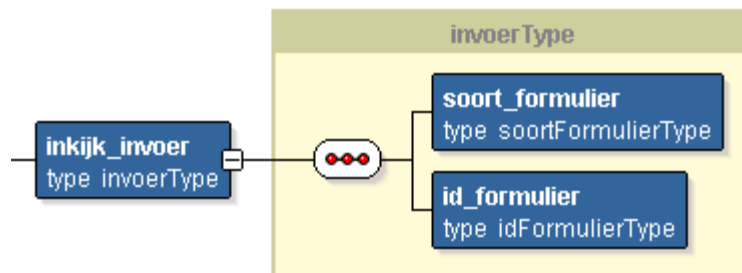
De feitelijke URL waar de webservice berichten gePOST moeten worden is nog niet bekend.

3 Dataformaat van gegevens

De webservice kan een bericht ontvangen en zal altijd een antwoord terugsturen (tenzij de server down is). In dit hoofdstuk wordt beschreven wat het formaat en de betekenis van de berichten.

3.1 Invoer

Het globale berichtformaat dat aangeleverd kan worden aan de webservice staat beschreven in onderstaande figuren.



Let op:

De toegestane waarden van alle velden zijn gedefinieerd in termen van XML Schema basistypes [3]. Hierdoor kan geen misverstand ontstaan over het formaat van datumvelden, booleans of fractionele getallen. De bij de webservice geleverde XML Schema berichtdefinitie (zoals opgenomen in sectie 4.1) is leidend voor de syntax. De schema validatie garandeert dat de verzendende partij strikt aan de vastgelegde syntax gehouden wordt.

Voorbeeld:

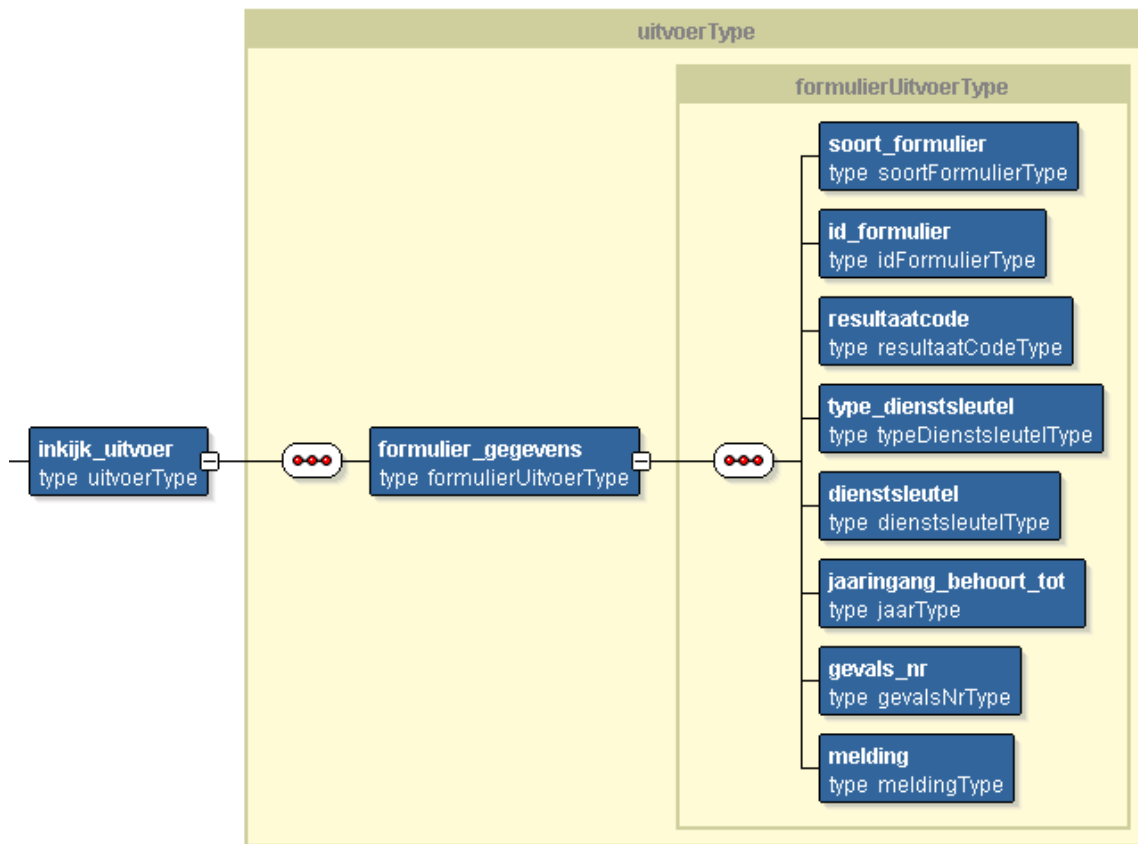
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" >
  <soapenv:Header/>
  <soapenv:Body>
    <ns:inkijk_invoer xmlns:ns="http://www.minlnv.nl/ws/dfx/1.0/">
      <soort_formulier>999</soort_formulier>
      <id_formulier>1002</id_formulier>
    </ns:inkijk_invoer>
  </soapenv:Body>
</soapenv:Envelope>
```

3.2 Webservice antwoord

Het sturen van een POST naar de webservice levert altijd een antwoord op. Dit antwoord kan twee dingen aangeven; ofwel de operatie is geslaagd, dan wel de operatie is mislukt met bijbehorende foutmelding(en). In deze sectie wordt uitgewerkt hoe deze antwoorden er uit zien.

3.2.1 Operatie geslaagd

Bij een geslaagde operatie wordt een 'normaal' SOAP response terug gestuurd. De http-statuscode zal http_ok (200) zijn, en het SOAP bericht bevat een SOAP-Body met een *inkijk_uitvoer* element.



Indien de geen resultaten oplevert of indien de zoekvraag een interne fout veroorzaakt, heeft het veld *resultaatcode* de waarde 10 en staat in het veld *melding* de bijbehorende foutmelding. In de andere gevallen bevatten de overige velden de gegevens zoals die bekend zijn binnen het register voor het desbetreffende formulier.

Voorbeeld:

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <inkijk_uitvoer xmlns="http://www.minlnv.nl/ws/dfx/1.0/">
      <formulier_gegevens>
        <soort_formulier>999</soort_formulier>
        <id_formulier>1002</id_formulier>
        <resultaatcode>0</resultaatcode>
        <type_dienstsleutel>BRS</type_dienstsleutel>
        <dienstsleutel>299571564</dienstsleutel>
        <jaaringang_behoort_tot>2006</jaaringang_behoort_tot>
        <gevals_nr xsi:nil="true"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
        <melding/>
      </formulier_gegevens>
    </inkijk_uitvoer>
  </S:Body>
</S:Envelope>
    
```

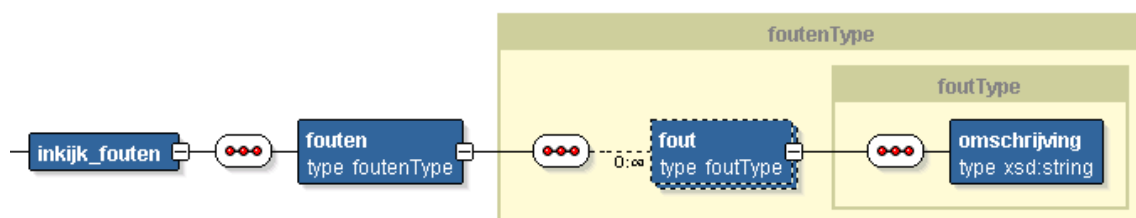
```
</S:Envelope>
```

Voorbeeld 'geen gegevens':

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <inkijk_uitvoer xmlns="http://www.minlnv.nl/ws/dfx/1.0/">
      <formulier_gegevens>
        <soort_formulier>999</soort_formulier>
        <id_formulier>1002</id_formulier>
        <resultaatcode>10</resultaatcode>
        <type_dienstsleutel xsi:nil="true"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
        <dienstsleutel xsi:nil="true"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
        <jaaringang_behoort_tot>0</jaaringang_behoort_tot>
        <gevals_nr xsi:nil="true"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
        <melding>De zoekvraag op formuliersoort i.c.m. formuliernummer levert geen
          resultaten op
        </melding>
      </formulier_gegevens>
    </inkijk_uitvoer>
  </S:Body>
</S:Envelope>
```

3.2.2 Foutmelding

Een XML Schema validatie fout wordt teruggekoppeld via een *SOAPFault*. Als het een fout betreft die opgetreden is in de processtappen XML Schema validatie of inhoudelijke controle (uit figuur 1), dan zal het *detail* gedeelte van de SOAPFault een *inkijk_fouten* element met uitleg over de foutmeldingen bevatten. In andere gevallen kan het detail gedeelte ontbreken. Er kunnen meerdere fouten opgetreden zijn. De http statuscode zal mogelijkterwijs INTERNAL_SERVER_ERROR (code 500) zijn.



Hier onder volgt een voorbeeld van een SOAP Fault (in de http response body) zoals deze gegenereerd kan worden door de webservice. Conform de SOAP specificatie geeft de faultCode in de SOAPFault aan of het een fout op de server (*env:Server*) of aan de client kant is (*env:Client*).

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:Fault xmlns:ns2="http://schemas.xmlsoap.org/soap/envelope/">
      <faultcode>ns2:Server</faultcode>
      <faultstring>Het bericht voldoet niet aan de specificaties</faultstring>
      <detail>
        <inkijk_fouten:inkijk_fouten
          xmlns:inkijk_fouten="http://www.minlnv.nl/ws/dfx/1.0/">
          <fouten xmlns="http://www.minlnv.nl/ws/dfx/1.0">
            <fout>
              <omschrijving>
                cvc-datatype-valid.1.2.1: '2a' is not a valid value for
                'integer'.
              </omschrijving>
            </fout>
            <fout>
              <omschrijving>
                cvc-type.3.1.3: The value '2a' of element 'ns:id_formulier'
                is not valid.
              </omschrijving>
            </fout>
          </fouten>
        </inkijk_fouten:inkijk_fouten>
      </detail>
    </ns2:Fault>
  </S:Body>
</S:Envelope>
```

4 Webservice Interface specificatie

De webservice is te gebruiken volgens het WSDL document in sectie 4.2. Het specifieke endpoint ontbreekt nog in de WSDL specificatie en zal op een later tijdstip aangeleverd worden. De berichten die (in de SOAP Body en SOAP Header) uitgewisseld zullen worden zijn gespecificeerd in het WSDL document die de definitie weer importeert uit het XML Schema in sectie 4.1.

4.1 Bericht formaten (W3C Schema specificatie)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.minlnv.nl/ws/dfx/1.0/"
  targetNamespace="http://www.minlnv.nl/ws/dfx/1.0/">
  <xsd:annotation>
    <xsd:documentation>Het schema voor koppeling met DfX.</xsd:documentation>
  </xsd:annotation>

  <xsd:element name="inkijk_invoer" type="invoerType"/>
  <xsd:element name="inkijk_uitvoer" type="uitvoerType"/>

  <xsd:element name="inkijk_fouten">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="fouten" type="foutenType"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:simpleType name="soortFormulierType">
    <xsd:restriction base="xsd:int">
      <xsd:pattern value="[1-9][0-9][0-9]" />
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="idFormulierType">
    <xsd:restriction base="xsd:long">
      <xsd:pattern value="[1-9][0-9]{1,9}" />
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="resultaatCodeType">
    <xsd:restriction base="xsd:int"/>
  </xsd:simpleType>
  <xsd:simpleType name="typeDienstsleutelType">
    <xsd:restriction base="xsd:string"/>
  </xsd:simpleType>
```

```
<xsd:simpleType name="dienstsleutelType">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name="jaarType">
  <xsd:restriction base="xsd:int">
    <xsd:pattern value="[1-9][0-9]{3}" />
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="gevalsNrType">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="25"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="meldingType">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:complexType name="formulierUitvoerType">
  <xsd:sequence>
    <xsd:element name="soort_formulier" type="soortFormulierType"
      nillable="false"/>
    <xsd:element name="id_formulier" type="idFormulierType" nillable="false"/>
    <xsd:element name="resultaatcode" type="resultaatCodeType" nillable="false"/>
    <xsd:element name="type_dienstsleutel"
      type="typeDienstsleutelType" nillable="true"/>
    <xsd:element name="dienstsleutel" type="dienstsleutelType"
      nillable="true" />
    <xsd:element name="jaaringang_behoort_tot" type="jaarType"
      nillable="false" />
    <xsd:element name="gevals_nr" type="gevalsNrType" nillable="true"/>
    <xsd:element name="melding" type="meldingType" nillable="true"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="uitvoerType">
  <xsd:sequence>
    <xsd:element name="formulier_gegevens" type="formulierUitvoerType"
      maxOccurs="1" minOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="invoerType">
  <xsd:sequence>
    <xsd:element name="soort_formulier" type="soortFormulierType"
      nillable="false" maxOccurs="1" />
    <xsd:element name="id_formulier" type="idFormulierType"
      nillable="false" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>
```

Webservice technische specificatie: DFx (Berichtenboek)

```
<xsd:complexType name="foutenType">
  <xsd:sequence>
    <xsd:element name="fout" type="foutType" minOccurs="0"
      maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="foutType">
  <xsd:sequence>
    <xsd:element name="omschrijving" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>

</xsd:schema>
```

4.2 WSDL definitie van de webservice

```
<?xml version = '1.0' encoding = 'UTF-8' standalone="yes"?>
<definitions targetNamespace="http://www.minlnv.nl/ws/dfx/1.0/"
    xmlns:tns="http://www.minlnv.nl/ws/dfx/1.0/"
    xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:wssu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:dfxws="http://www.minlnv.nl/ws/dfx/1.0/">
    <types>
        <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
            <xsd:import namespace="http://www.minlnv.nl/ws/dfx/1.0/"
                schemaLocation="dfx.xsd"/>
        </xsd:schema>
    </types>
    <message name="inkijk_uitvoer">
        <part name="inkijk_uitvoer" element="dfxws:inkijk_uitvoer"/>
    </message>
    <message name="inkijk_fouten">
        <part name="reason" element="dfxws:inkijk_fouten"/>
    </message>
    <message name="inkijk_invoer">
        <part name="inkijk_invoer" element="dfxws:inkijk_invoer"/>
    </message>
    <portType name="DFxService">
        <operation name="inkijk_invoer">
            <input message="tns:inkijk_invoer"/>
            <output message="tns:inkijk_uitvoer"/>
            <fault message="tns:inkijk_fouten" name="inkijk_fouten"/>
        </operation>
    </portType>
    <!-- concrete part -->
    <binding name="DFxPortBinding" type="tns:DFxService">
        <soap:binding style="document"
            transport="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="inkijk_invoer">
            <soap:operation soapAction="http://www.minlnv.nl/ws/dfx/1.0/getFormulierGegevens"/>
            <input>
                <soap:body use="literal" parts="inkijk_invoer"/>
            </input>
            <output>
                <soap:body use="literal" parts="inkijk_uitvoer"/>
            </output>
        </operation>
    </binding>
</definitions>
```



```
</output>
<fault name="inkijk_fouten">
  <soap:fault use="literal" name="inkijk_fouten"/>
</fault>
</operation>
</binding>
<service name="DFxService">
  <port name="DFxService" binding="tns:DFxPortBinding">
    <soap:address location="REPLACE_WITH_ACTUAL_URL"/>
  </port>
</service>
</definitions>
```

5 Referenties

- [1] RFC2616 *"Hypertext Transfer Protocol -- HTTP/1.1"*,
 <http://www.w3.org/Protocols/rfc2616/rfc2616.html>
- [2] RFC2617 *"HTTP Authentication: Basic and Digest Access Authentication"*,
 <ftp://ftp.isi.edu/in-notes/rfc2617.txt>
- [3] SCHEMA *"W3C XML Schema specification"*
 <http://www.w3.org/XML/Schema>
- [4] SOAP *"SOAP Version 1.2 Part 1: Messaging Framework"*,
 <http://www.w3.org/TR/soap12-part1/>
- [5] SSL3.0 *"SSL 3.0 Specification"*, <http://wp.netscape.com/eng/ssl3/>