



“Technische afspraken”

Edibulb webservices

Technische Specificatie  
Versie 1.00

**Definitief**

Copyright © 2014 Stichting Edibulb

# Wijzigingsblad

**Datum laatste revisie:** vrijdag 21 maart 2014

**Bestandsnaam :** HvR0451v1.00 EBL Technische afspraken Webservices.doc

**Bestandsnummer :** HvR/0451v1.00

Versie	Auteur(s)	Wijzigingen
1.00	Hans van Rossen	Initiële versie.

	Datum	Functionaris	Paraaf
<b>Akkoord</b>		Hans van Rossen (Auteur)	
<b>Gecontroleerd</b>			
<b>Geaccepteerd namens Projectgroep Edibulb</b>		Henk Zwinkels (Projectleider)	

## Stichting Edibulb Confidentieel

Dit document en de intellectuele inhoud ervan zijn eigendom van Stichting Edibulb p/a te Waddinxveen, en mogen op geen enkele wijze worden vermenigvuldigd en/of kenbaar worden gemaakt aan derden, voor welk doel dan ook, zonder de schriftelijke toestemming van de eigenaresse.

# Inhoud

1.	INLEIDING	4
1.1	Doel van dit document	4
1.2	Bestemd voor...	4
1.3	Uitgangspunten	4
1.4	Afkorting en begrippen	4
2.	REFERENTIES	5
3.	SOAP OVER HTTP (WEBSERVICES)	6
3.1.1	Web service architectuur	6
3.2	Onderliggende techniek	6
3.3	Implementatiehandleiding	7
4.	FOUTAFHANDELING VIA SOAP	8
4.1	Edibulb ErrorList schema	9
4.1.1	DocumentReference	9
4.1.2	ErrorLevel	9
4.1.3	Severity	9
4.1.4	ErrorCode	10
4.1.5	ErrorLocation	10
4.1.6	DescriptionText	10
	BIJLAGE 1: VOORBEELD WSDL VOOR DE KETENPARTIJ KOPER	11

## 1. Inleiding

### 1.1 Doel van dit document

Het beschouwinggebied van dit document beperkt zich tot de conventies op gebied van adressering, envelop en netwerktoegang.

Dit document specificeert op technisch niveau de afspraken voor het uitwisselen van Edibulb XML berichten. Voor het uitwisselen van berichten zijn twee protocollen nodig: conversatieprotocol en transportprotocol.

Het conversatieprotocol is door Edibulb gestandaardiseerd en ingevuld met het Simple Object Access Protocol (SOAP) versie 1.1. Met SOAP wordt het Edibulb XML-bericht ingebed in een XML-fragment dat is opgemaakt volgens het SOAP schema. Dit beetje extra XML (ook wel de SOAP-envelop genoemd) biedt de volgende belangrijke functionaliteit:

- Wat voor type Edibulb bericht bevindt zich in de envelop;
- Voor wie (welke applicatie of welk systeem) is het bericht bedoeld;
- Welke fouten zijn opgetreden bij het uitwisselen van het bericht.

SOAP dwingt niet af welk transportprotocol en welk type netwerk infrastructuur worden gebruikt. De enige randvoorwaarde voor succesvolle communicatie is dat zowel de zender als de ontvanger gebruik maken van hetzelfde netwerk en hetzelfde transportprotocol. De meest voorkomende combinaties van netwerken en transportprotocollen zijn:

- SMTP (e-mail) over het Internet;
- Webservices over het Internet;

Dit document beschrijft de Edibulb standaarden met betrekking tot het gebruik van SOAP over HTTP (webservices).

Dit document is afgeleid van de conventies welke gelden voor het versturen van elektronische berichten in de sierteeltsector.

### 1.2 Bestemd voor...

Dit document is bestemd voor alle partijen binnen de bloembollensector die op elektronische wijze gegevens willen uitwisselen en voor organisaties die zich bezighouden met de ontwikkeling van softwaretoepassingen hiervoor.

### 1.3 Uitgangspunten

Dit document is gebaseerd op:

- Florecom CIL0001 Webservices versie 0.2 van juli 2009

### 1.4 Afkortingen en begrippen

Afkorting / begrip	Omschrijving
XML	eXtended Markup Language
SOAP	Simple Object Access Protocol

## 2. Referenties

De volgende referenties zijn van belang als basis voor de technische afspraken:

1. Technische afspraken Florinet.com
2. W3C, 2009. Simple Object Access Protocol (SOAP) 1.1, 8-5-2000, [http://www.w3.org/TR/2000/NOTE-SOAP-20000508/], accessed 2-9-2009
3. RFC821: Simple Mail Transfer Protocol.
4. RFC822: Standard for the Format of Internet text Messages.
5. RFC959: File Transfer Protocol (FTP)
6. RFC1939: Post Office Protocol, Version 3 (POP3).
7. RFC2060: Internet Message Access Protocol, Version 4 (IMAP4).
8. RFC2045: MIME Part 1: Format of Internet Message Bodies
9. RFC2046: MIME Part 2: Media Types
10. RFC2047: MIME Part 3: Message Header Extensions for Non-ASCII Text
11. RFC2048: MIME Part 4: Registration Procedures
12. RFC2049: MIME Part 5: Conformance Criteria and Examples
13. RFC1767: MIME Encapsulation of EDI Objects.
14. RFC1847: Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted
15. RFC1939: Post Office Protocol, Version 3 (POP3).
16. RFC1952: GZIP file format specification version 4.3.
17. RFC2045: MIME Part 1: Format of Internet Message Bodies
18. RFC2046: MIME Part 2: Media Types
19. RFC2047: MIME Part 3: Message Header Extensions for Non-ASCII Text
20. RFC2048: MIME Part 4: Registration Procedures
21. RFC2049: MIME Part 5: Conformance Criteria and Examples
22. RFC2060: Internet Message Access Protocol, Version 4 (IMAP4).
23. RFC2157: Mapping between X.400 and RFC-822/MIME Message Bodies.
24. RFC2183: Communicating Presentation Information in Internet Messages: The Content-Disposition Header Field.
25. RFC2279: UTF-8 encoding
26. RFC2781: UTF-16 encoding
27. RFC3023: XML Media Types.
28. RFC3902: The "application/soap+xml" media type

Bovengenoemde referenties zijn te vinden op de site van het Internet Mail Consortium <http://www.imc.org/> of de site van IETF <http://www.ietf.org/>.

### 3. SOAP over HTTP (webservices)

Webservices is een standaard van het World Wide Web Consortium (W3C). Deze standaard is bedoeld als oplossing om de interoperabiliteit te verbeteren tussen de grote diversiteit aan applicaties die draaien op verschillende soorten hardware en besturingssystemen.

Een webservice is een applicatie die net als een website op het Internet luistert tot er iemand een verzoek doet met behulp van http. In het http-verzoek is een XML-bericht opgenomen en de webservice heeft als taak om dit bericht te verwerken en op zijn beurt te beantwoorden met een XML-bericht.

Naast de eis van het gebruik van SOAP voor het opmaken van de XML-berichten, bieden webservices een standaard voor het kenbaar maken van de geboden functionaliteit aan de handelsrelaties met behulp van zogenaamde Web Service Description Language (WSDL) documenten. Deze WSDL's bevatten alle technische specificaties (syntax en structuur) om systemen op technisch niveau met elkaar te laten praten.

Of systemen elkaar daadwerkelijk begrijpen hangt af van de afspraken die op een hoger niveau in de Edibulb standaarden zijn gemaakt, zoals de bericht invulinstructies.

#### 3.1 Webservice architectuur

De Edibulb webservices bieden functionaliteit die nodig is voor het uitwisselen van informatie tussen ketenpartijen. Edibulb publiceert standaard WSDL's per schakel. Dit heeft een aantal voordelen, waaronder:

- Minder overhead doordat in de WSDL alleen de Edibulb berichtstandaarden (XML Schema's) worden opgenomen die relevant zijn voor die schakel;
- Eenvoudige authenticatie doordat er per schakel een eigen URL kan worden gebruikt waar alleen de webservice operaties voor die schakel beschikbaar worden gesteld.

De WSDL kan gezien worden als een volledige communicatiestandaard tussen zender en ontvanger. Niet alleen specificeert de WSDL welke Edibulb XML berichten gebruikt moeten worden, maar ook welke berichten de partij ondersteunt en via welke URL deze beschikbaar worden gesteld. De WSDL is een XML document dat per e-mail gestuurd kan worden aan een nieuwe zakenrelatie, maar het kan ook gepubliceerd worden op publieke bibliotheken (bijvoorbeeld UDDI) of op de eigen website.

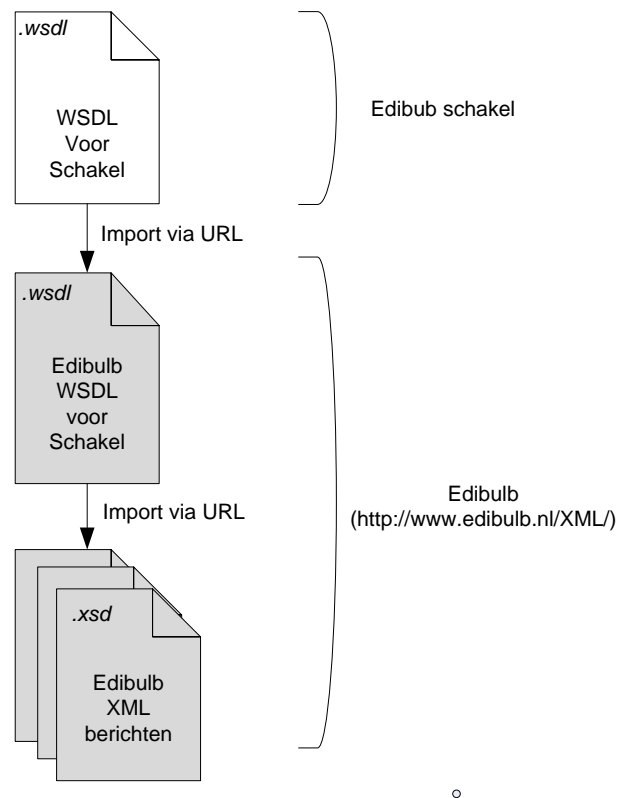
Met gestandaardiseerde WSDL's wordt het mogelijk om meerdere schakels met elkaar te verbinden op een uniforme manier.

#### 3.2 Onderliggende techniek

De onderliggende technieken van webservices zijn SOAP (versie 1.1) en HTTP.

Figuur 1 toont de architectuur van de WSDL's. De handelspartij ontwerpt zijn eigen WSDL per schakel in de keten (bijvoorbeeld een koper). De handelspartij gebruikt hiervoor een template van Edibulb. Deze templates zijn beschikbaar via de URL: <http://www.edibulb.nl/XML/services/Examples/>.

De WSDL van de handelspartij maakt gebruik van verwijzingen met behulp van een URL naar de generieke WSDL van Edibulb voor de betreffende schakel. Hierin zijn alle relevante Edibulb XML schema's opgenomen uit de Edibulb berichtenset.



**Figuur 1: WSDL architectuur van overerving**

### 3.3 Implementatiehandleiding

De WSDL's zijn voor een groot deel gestandaardiseerd. De enige wijziging die een software leverancier dient uit te voeren in de voorbeeld WSDL's is:

1. Het invullen van de eigen webservice URL.

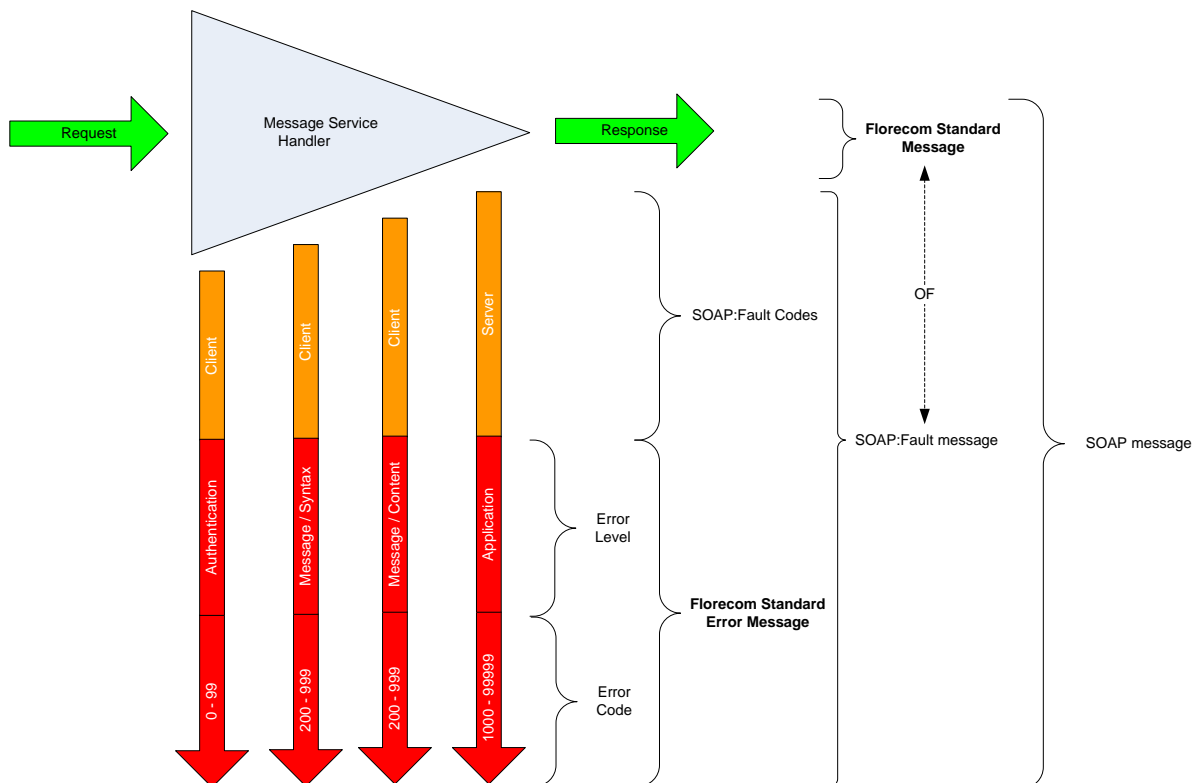
Daarna is de WSDL klaar om gedistribueerd te worden aan de gebruikers van de applicatie. Bijlage 1 toont een voorbeeld webservice voor een koper.

De webservice is een applicatie die functioneert als interface tussen de handelssystemen van ketenschakels.

## 4. Foutafhandeling via SOAP

De fouten die kunnen optreden op transportniveau zijn al gestandaardiseerd door het betreffende transportprotocol. Zo kennen http en smtp een groot aantal statuscodes om de transportfouten te beschrijven. Fouten op alle hogere niveaus (conversatie, bericht, proces en keten) worden allemaal verzameld en teruggemeld op het niveau van SOAP. SOAP kent hiervoor het SOAP:Fault element.

Het Fault-element van SOAP dient om de cliënt op de hoogte te stellen van eventuele fouten die zijn ontstaan tijdens het verwerken van het binnengekomen bericht. Onderstaand figuur beschrijft een generieke Message Service Handler zoals bijvoorbeeld een webservice. Het request bericht wordt verwerkt door de service en bij geen problemen (happy flow) wordt er een response bericht gestuurd. Eventuele fouten worden via het SOAP:Fault element teruggemeld. Binnen het Fault-element bevindt zich een detail-element dat gevuld dient te worden volgens het errorList-schema. Deze component maakt het mogelijk om fouten te onderscheiden in error levels, ieder met een eigen verzameling aan standaard error codes.



Een voorbeeld van een SOAP:Fault bericht dat gebruik maakt van het component ErrorList:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>Fout tijdens verwerken van het bericht.</faultstring>
      <detail>
        <fsm:ErrorList xmlns:fsm="http://www.edibulb.nl/XML/Order">
          <fsm:Error>
            <fsm:DocumentReference>
              <fsm:IssuerAssignedID schemeID="IRN">8713782589954001</fsm:IssuerAssignedID>
            </fsm:DocumentReference>
          </fsm:Error>
        </fsm:ErrorList>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



```

</fsm:DocumentReference>
<fsm:ErrorLevel>message</fsm:ErrorLevel>
<fsm:Severity>error</fsm:Severity>
<fsm:ErrorCode>205</fsm:ErrorCode>
<fsm:ErrorLocation>/order/buyer/dt:partyDetails/ebcc:typeCode</fsm:ErrorLocation>
<fsm:DescriptionText>
  Het name attribuut van het CodeType component verwijst naar een onbekende codelijst.
</fsm:DescriptionText>
</fsm:Error>
</fsm:ErrorList>
</detail>
</SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

## 4.1 Edibulb ErrorList schema

### 4.1.1 DocumentReference

Het element DocumentReference verwijst naar de regel (line item) uit het bronbericht waar de fout op van toepassing is. Dit is noodzakelijk indien er meerdere items (bijvoorbeeld afbeeldingen of orders) worden aangeboden in hetzelfde bericht.

### 4.1.2 ErrorLevel

Een waarde uit de onderstaande codelijst die het groepsniveau van de melding bepaalt.

ErrorLevel	ErrorCode van	ErrorCode tot	Omschrijving
authentication	0	99	Meldingen met betrekking tot het verkrijgen van toegang tot de berichtverwerkende applicatie van de ontvanger.
message	100	199	Foutmeldingen met betrekking tot het niet nakomen van de XML standaard (well-formedness en schema validatie).
message	200	999	Foutmeldingen met betrekking tot het niet nakomen van de generieke Edibulb bericht- en processpecificaties. Bijvoorbeeld onjuist gebruikte coderingen (bijvoorbeeld: artikelcodes en conditiecodes).
application	1000	999.999.999	Applicatie specifieke foutmeldingen. Deze foutmeldingen worden niet gestandaardiseerd door Edibulb, maar worden gespecificeerd en uitgegevens door de eigenaar van de berichtverwerkende applicatie.

### 4.1.3 Severity

Een waarde uit de onderstaande codelijst die de aard van de melding bepaalt.

ErrorLevel	Omschrijving
warning	Het bericht is (deels) verwerkt, maar er is een voor de zender relevante waarschuwing die betrekking heeft op de verwerking van het bericht. Een

	vervolgactie is niet noodzakelijk voor de verdere afhandeling van dit bericht,
error	Het bericht is niet verwerkt en een vervolgactie is noodzakelijk voor de juiste afhandeling van het bericht.

#### 4.1.4 ErrorCode

Het doel van error code is om de cliënt programmatuur de mogelijk te geven de fout te herstellen en op een geautomatiseerde manier te verwerken. Het bereik van de codes is afhankelijk van de gebruikte ErrorLevel (zie paragraaf 3.1.2).

ErrorLevel	Error Code	Omschrijving
authentication	<b>200</b>	De toegang tot het bericht verwerkende systeem van de ontvanger is geweigerd.
authentication	<b>201</b>	De toegang tot de opgegeven soapAction is geweigerd door de ontvanger.
authentication	<b>202</b>	De toegang tot functionaliteit uit een onderliggende applicatie is geweigerd door de ontvanger.
message	<b>300</b>	Het bericht is niet well-formed en voldoet niet aan de XML specificatie van het W3C.
message	<b>301</b>	Het bericht is niet valide volgens het Edibulb berichtschemata.
message	<b>400</b>	Het listID attribuut van het CodeType component verwijst naar een onbekende codelijst.
message	<b>401</b>	Het listAgencyID attribuut van het CodeType component verwijst naar een onbekende partij.
message	<b>402</b>	Het listAgencyName attribuut van het CodeType component verwijst naar een onbekende partij.
message	<b>403</b>	Het listName attribuut van het CodeType component verwijst naar een onbekende lijst.
message	<b>404</b>	Het listVersion attribuut van het CodeType component verwijst naar een onbekende codelijst versie.
message	<b>405</b>	Het name attribuut van het CodeType component verwijst naar een onbekende codelijst.
message	<b>406</b>	Het languageCode attribuut van het CodeType component verwijst naar een onbekende taal van de codelijst.
message	<b>407</b>	Het listURI attribuut van het CodeType component verwijst naar een onbekende codelijst.
message	<b>408</b>	Het listScheme attribuut van het CodeType component verwijst naar een onbekende codelijst.
message	<b>409</b>	De waarde van het CodeType component komt niet voor in de codelijst.
message	<b>420</b>	Het schemeURI attribuut van het IDType component bevat een voor de ontvanger onbekende referentie.
message	<b>421</b>	Het schemeName attribuut van het IDType component bevat een onjuiste waarde. Deze waarde is niet gedefinieerd in de Edibulb referetietechniek.
message	<b>422</b>	Het SchemeDataURI attribuut van het IDType component bevat een voor de ontvanger onbekende referentie.
message	<b>423</b>	De waarde van het IDType component verwijst naar een onbekend document. De relatie kan niet gelegd worden, omdat het document waarnaar verwezen wordt niet bekend is bij de ontvanger van het bericht.

#### 4.1.5 ErrorLocation

Dit element bevat de XPATH locatie van het element in het bericht dat de fout heeft veroorzaakt, bijvoorbeeld: */order/buyer/dt:partyDetails/ebcc:typeCode*

#### 4.1.6 DescriptionText

Een omschrijving van de fout die de verzender van het bericht helpt om het probleem te herstellen.

## Bijlage 1: Voorbeeld WSDL voor de ketenpartij Koper

```
<?xml version="1.0"?>
```

```
<!--
```

```
=====
Example WSDL for Edibulb rol Koper
=====
```

```
Please follow the instructions in this file to implement your own web service.
```

```
Schema agency:  EDIBULB
```

```
Schema version:  0.1
```

```
Schema date:    6 april 2011
```

```
Author:        H. van Rossen
```

```
Copyright (C) EDIBULB (2011). All Rights Reserved.
```

```
==>
```

```
<definitions
```

```
  xmlns:tns="http://webservice.edibulb.nl/XML/Koper"
```

```
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
```

```
  xmlns:defs="http://webservice.edibulb.nl/XML/PortDefinitions "
```

```
  xmlns="http://schemas.xmlsoap.org/wsdl/"
```

```
  name="CommercialBuyer"
```

```
  targetNamespace="http://webservice.edibulb.nl/XML/Koper">
```

```
<!--IMPORT: Import of Edibulb's message and port definitions. This import is mandatory i.e. do not alter!-->
```

```
<import namespace="http://webservice.edibulb.nl/XML/PortDefinitions"
```

```
  location="http://www.edibulb.nl/XML/services/koper.wsdl"/>
```

```
<!-- STEP 1: Remove the SOAP operations that you did not implement-->
```

```
<binding name="Koper" type="defs:KoperPoorten">
```

```
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
```

```
  <operation name="PutOrder">
```

```
    <soap:operation soapAction="http://webservice.edibulb.nl/XML/Koper/PutOrder" style="document"/>
```

```
    <input>
```

```
      <soap:body use="literal"/>
```

```
    </input>
```

```
    <output>
```

```
      <soap:body use="literal"/>
```

```
    </output>
```

```
  </operation>
```

```
</binding>
```

```
<!-- STEP 2: Change [MYWEBSERVICEURL] to the URL of your webservice-->
```

```
<service name="CommercialBuyer">
```

```
  <documentation>Edibulb Koper interface</documentation>
```

```
  <port name="KoperPort" binding="tns:Koper">
```

```
    <soap:address location="http://MYWEBSERVICEURL"/>
```

```
  </port>
```

```
</service>
```

```
</definitions>
```