

# Interdata web service manual

VWE ontzorgt de voertuigbranche



**VWE**  
Automotive



<b>Introduction .....</b>	<b>3</b>
<b>1. Environments .....</b>	<b>4</b>
1.1. Test/ acceptance environment .....	4
1.2. Production environment.....	4
1.3. Methods .....	4
1.3.1. DataRequest.....	4
1.3.2. StandardDataRequest .....	5
<b>2. Technical details .....</b>	<b>6</b>
2.1. Protocols used .....	6
2.2. Messaging security .....	6
2.3. Encoding .....	6
2.4. Input and output.....	6
<b>3. Messages DataRequest .....</b>	<b>7</b>
3.1. Request message.....	7
3.2. Reply message .....	7
3.3. Error messages .....	8
3.3.1. XML element errors .....	8
3.3.2. Data errors .....	8
<b>4. Messages StandardDataRequest .....</b>	<b>10</b>
4.1. Request message.....	10
4.1.1. Request message with end-user authentication.....	10
4.2. Reply message .....	11
4.3. Error messages .....	11
4.3.1. XML element errors .....	11
4.3.2. Data errors .....	11
<b>5. Data sections .....</b>	<b>13</b>
5.1. Overview data sections.....	13
5.2. Overview XSD schemas .....	14
<b>6. Support .....</b>	<b>15</b>
<b>7. Appendix 1 PHP example standaard request.....</b>	<b>16</b>
<b>8. Appendix 1 PHP example maatwerk/tailormade request..</b>	<b>18</b>



## Introduction

---

Interdata is the generic application site for searching vehicle information based on predetermined criteria. This generic site can provide information from all data sources available at VWE, mostly based on criteria entered by an user. The chosen solution is a web service, which is a global platform-independent standard.

Document like this can always be improved. If you have any questions, suggestions, or comments, please mail them to [info@vwe.nl](mailto:info@vwe.nl).



# 1. Environments

---

## 1.1. Test/ acceptance environment

For the purpose of testing and the acceptance of applications that communicate with Interdata, a test / acceptance environment is available.

Location: <https://acceptatie-interdata.vwe.nl/>

WSDL location: <https://acceptatie-interdata.vwe.nl/DataAanvraag.asmx?WSDL>

**IMPORTANT!** Access to this area will be provided for a limited period when purchasing a new type of message. For each message type, VWE provides a balance with a number of credits in the acceptance environment.

If after the initial acceptance period, access to the acceptance environment is required again, this can be requested.

**IMPORTANT!** The test / acceptance environment of VWE is not being updated continuously with the latest updates from our data providers. It is therefore quite possible that in particular for the newer registrations, no result is obtained in our acceptance environment.

## 1.2. Production environment

The production environment is intended for use by applications accepted for production purposes. The transactions that pass through this production URLs are also charged.

Location: <https://interdata.vwe.nl/>

WSDL location: <https://interdata.vwe.nl/DataAanvraag.asmx?WSDL>

## 1.3. Methods

Interdata has 2 methods for searching the vehicle information:

- DataRequest with parameter *XMLBody*  
<https://interdata.vwe.nl/DataAanvraag.asmx?op=DataRequest>
- StandardDataRequest with parameter *requestXml*  
<https://interdata.vwe.nl/DataAanvraag.asmx?op=standaardDataRequest>

### 1.3.1. DataRequest

Interdata previously offered only one method, namely DataRequest. Through this method, dozens of messages have been, and are being delivered.

Through this platform, both standard messages (once created and then used by multiple clients), and customised messages are delivered.

For a description of operational aspects, see Chapter 3.



### 1.3.2. StandardDataRequest

Since the number of data sources with vehicle information that VWE offers is increasing, as are the number of customers and available messages, VWE has developed a second version of Interdata. In this version VWE splits the data sources into data sections. These data sections can then be linked to a message. Due to the standardization of data sections, the documentation of the elements is more complete and accurate. The StandardDataRequest messages generally begin with the name “SB” (for example: SB-RDWA-ATL-MIL).

For a description of operational aspects, see Chapter 4.



## 2. Technical details

---

### 2.1. Protocols used

The protocols used are as follows:

- TCP/IP for network communication.
- HTTP for application communication.
- SOAP for wrapping the method calls.

### 2.2. Messaging security

The traffic is over SSL with the server identifying itself using a server certificate.

### 2.3. Encoding

VWE uses UTF-8 as encoding.

### 2.4. Input and output

For both the input and the output the HTTP protocol is applied. A POST or a GET can be used with different content. The possible combinations of methods and content are as follows:

- HTTP POST with SOAP packed request message as content (recommended)
- HTTP POST with the request message as content
- HTTP GET with the request message in the query string

In all cases a reply message will be returned containing the response. The details of the content of the POST and GET can be found by browsing to the Interdata URL.

An example code for PHP "as-is" is included in Appendix 1.



## 3. Messages DataRequest

---

### 3.1. Request message

An Interdata request message always consists of a root node and two mandatory XML blocks. The root node is always: <message>. The two mandatory XML blocks within the message are: <authenticatie> and <parameters>. Below is an example:

```
<bericht>
  <authenticatie>
    <naam>XXXX</naam>
    <wachtwoord>XXXX</wachtwoord>
    <berichtsoort>XXXX</berichtsoort>
    <referentie>XXXX</referentie>
  </authenticatie>
  <parameters>
    <kenteken>AABB11</kenteken>
  </parameters>
</bericht>
```

The authentication block should always contain the elements <naam>, <wachtwoord> and <berichtsoort>. The contents of the parameter block will vary per message.

**TIP!** An Interdata customer can include its own reference in the <referentie> element. This is not mandatory. The content of this field is returned in the reply message.

### 3.2. Reply message

An Interdata reply message will echo the blocks <authenticatie> and <parameters> from the request message. The root node is always: <antwoordbericht>. The authentication block contains a number of additional elements that are the result of the request. Also, an Interdata reference will be provided. After these two blocks the requested response follows. Below is an example:

```
<antwoordbericht>
  <authenticatie>
    <naam>XXXX</naam>
    <wachtwoord>XXXX</wachtwoord>
    <berichtsoort>XXXX</berichtsoort>
    <referentie>XXXX</referentie>
    <resultaat>00</resultaat>
    <Interdata-referentie>999999</Interdata-referentie>
  </authenticatie>
  <parameters>
    <kenteken>AABB11</kenteken>
  </parameters>
  <voertuig>
    <merk>XXXXXXXXXXXX</merk>
    <model>XXXXXXXXXXXX</model>
    <bouwjaar>9999</bouwjaar>
  </voertuig>
</antwoordbericht>
```

*Authentication and parameters are fixed blocks. The vehicle node is illustrative: the node name and contents may vary by message type.*

In the element <resultaat> a numeric code is included indicating the status of the request. If an error has occurred a brief description of the error is provided in the <reden> element. There is not necessarily always a reason; with a good result there will be no <reden> element present. The possible values for the element <resultaat> can be found in section 3.3.2.



### 3.3. Error messages

Error messages can occur in 2 forms:

- XML element errors, such as errors in name elements
- Data errors such as invalid login names, incorrect filling parameters, etc.

#### 3.3.1. XML element errors

If any of the three mandatory elements of the message is not present, an error message will be returned. Such message looks like this when there is no authentication block present:

```
<fout-echo>  
  <melding>Required XML tag &lt;/bericht/authenticatie&gt; is  
missing</melding>  
</fout-echo>
```

#### 3.3.2. Data errors

If factual errors are detected then in the element <resultaat> a code will be returned, in some cases supplemented by the element <reden>. The element <reden> is not filled if the result is OK.

By default, the following codes can be returned:

Code	Reason
00	OK, no errors occurred
10	Format message is not correct
30	No message in data account
80	No vehicle data with that input
90	Name and password is incorrect
91	Unknown message type
92	Required XML tag ..... is missing
93	Invalid or unknown namespace defined
94	Message is not active
95	Balance is not sufficient
99	Application or unknown error





An example is the following reply message:

```
<antwoordbericht>
  <authenticatie>
    <naam>XXXX</naam>
    <wachtwoord>XXXX</wachtwoord>
    <berichtsoort>XXXX</berichtsoort>
    <referentie>XXXX</referentie>
    <resultaat>90</resultaat>
    <reden>Naam wachtwoord is onjuist</reden>
    <Interdata-referentie>999999</Interdata-referentie>
  </authenticatie>
  <parameters>
    <kenteken>AABB11</kenteken>
  </parameters>
</antwoordbericht>
```

Besides the standard result codes it is also possible create a series of additional result codes specifically per message. Codes are usually in the 50 or 80 series.



## 4. Messages StandardDataRequest

---

Conceptually, the method Standard Data Request is similar to Data Request. The method has one parameter of the data type String.

### 4.1. Request message

An Interdata request message always consists of a root node and two mandatory XML blocks. The root node is always: <bericht>. The two mandatory XML blocks within the message are: <authenticatie> and <parameters>. Below is an example:

```
<bericht>
  <authenticatie>
    <naam>XXXX</naam>
    <wachtwoord>XXXX</wachtwoord>
    <berichtsoort>XXXX</berichtsoort>
    <referentie>XXXX</referentie>
  </authenticatie>
  <parameters>
    <kenteken>AABB11</kenteken>
  </parameters>
</bericht>
```

The authentication block should always contain the elements <naam>, <wachtwoord> and <berichtsoort>. The contents of the parameter block will vary per message.

**TIP!** An Interdata customer can include its own reference in the <referentie> element, which is not mandatory. The content of this field is returned in the reply message.

#### 4.1.1. Request message with end-user authentication

Within the StandardDataRequest it is also possible to add an end-user. The advantage is that the customer can add a third party customer who can use the message and will receive the invoice for usage. The third party customer needs to be or become a customer of VWE. See below the example how to add the end-user XML part. This is the part between the words **OPTIONAL**.

```
<bericht>
  <authenticatie>
    <naam>XXXX</naam>
    <wachtwoord>XXXX</wachtwoord>
    <berichtsoort>XXXX</berichtsoort>
    <referentie>XXXX</referentie>
  </authenticatie>
  (OPTIONAL)
  <eindgebruiker>
    <klantnummer>XXXXXX</klantnummer>
    <inlognaam>XXXXXX</inlognaam>
    <wachtwoord>XXXXXX</wachtwoord>
  </eindgebruiker>
  (OPTIONAL)
  <parameters>
    <kenteken>AABB11</kenteken>
  </parameters>
</bericht>
```



## 4.2. Reply message

An Interdata reply message will echo the request message in the element Request. The root node is always: <antwoordbericht>.

The result block contains a number of additional elements that are the result of the request. Also, an Interdata reference will be provided. After these two blocks the requested response follows. Below is an example:

```
<antwoordbericht xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <aanvraag>&lt;bericht&gt;&lt;authenticatie&gt;&lt;naam&gt;XXXX&lt;/naam&
  &gt;&lt;wachtwoord&gt;XXXX&lt;/wachtwoord&gt;&lt;berichtsoort&gt;XXXX&lt;/beric
  htsoort&gt;&lt;/authenticatie&gt;&lt;parameters&gt;&lt;kenteken&gt;AABB11&lt
  ;/kenteken&gt;&lt;/parameters&gt;&lt;/bericht&gt;</aanvraag>
  <resultaat>
    <code>00</code>
    <omschrijving>Ok</omschrijving>
    <interdataReferentie>9156856</interdataReferentie>
  </resultaat>
  <rubrieken>
    <rdwInfoBasic xsi:type="rdw:rdwInfoBasicTypeV1"
    xmlns:rdw="http://www.xmlmode.nl/interdata/rdw">
      <rdw:kenteken>AABB11</rdw:kenteken>
      <rdw:merk code="SEAP">seat</rdw:merk>
      <rdw:handelsbenaming>ALTEA; 1.9 TDI</rdw:handelsbenaming>
    </rdwInfoBasic>
  </rubrieken>
</antwoordbericht>
```

*Request and Result are fixed blocks. After the node sections 1 or more data items are added. See Chapter 5 for more information on the items.*

In the element <resultaat> a numeric code is included that indicates the status of the request. If an error has occurred a brief description of the error is provided in the <reden> element. There is not necessarily always a reason; with a good result there will be no <reden> element. The possible values for the element <resultaat> can be found in section 4.4.2.

## 4.3. Error messages

Error messages can occur in 2 forms:

- XML element errors, such as errors in name elements
- Data errors such as invalid login names, incorrect filling parameters, etc.

### 4.3.1. XML element errors

If any of the three mandatory elements of the message is not present, an error message will be returned. Such message looks like this when there is no authentication block present:

```
<fout-echo>
  <melding>Required XML tag &lt;/bericht/authenticatie&gt; is
  missing</melding>
</fout-echo>
```

### 4.3.2. Data errors

If factual errors are detected then in the element <resultaat> a code will be returned, in some cases supplemented by the element <reden>. The element <reden> is not filled if the result is OK.



By default, the following codes are possible:

Code	Reason
00	OK, no errors occurred
10	Format message is not correct
30	No message in data account
80	No vehicle data with that input
90	Name and password is incorrect
91	Unknown message type
92	Required XML tag ..... is missing
93	Invalid or unknown namespace defined
94	Message is not active
95	Balance is not sufficient
99	Application or unknown error



## 5. Data sections

VWE has defined the following data sections. For each section documentation is composed in the form of an XSD. These XSDs are available on request, but are also available on the developer site on: <http://www.vwe.nl>.

### 5.1. Overview data sections

VWE offers the following data sections:

Section code	Source	Section description
rdwInfoBasic	RDW	RDW and BKR registration data
rdwInfoAdvanced	RDW	RDW and BKR registration data with additional fields
rdwInfoFull	RDW	Basic BKR registration data, all fields
milieuInfoBasic	RDW	Environmental information from RDW CVO's
rdwHistInfoBasic	VWE	Information about the number of owners that have had a registration.
rdwHistInfoAdvanced	VWE	Extensive information on the number of owners, the number of days held and the type of owner
atIMmtInfo	Autotelex	AutoTelex versions connected to a registration
atITechInfoBasic	Autotelex	Technical vehicle information from AutoTelex
atITechInfoAdvanced	Autotelex	Technical vehicle information from AutoTelex, all fields
atIPriceInfoBasic	Autotelex	Price details from AutoTelex based on registration and possibly AutoTelex version ID
atIOptieFabriek	Autotelex	Factory options from AutoTelex based on registration and possibly AutoTelex version ID
atIOptiePakket	Autotelex	Package options from AutoTelex based on registration and possibly AutoTelex version ID
atIOptieStandaard	Autotelex	Standard equipment from AutoTelex based on registration and possibly AutoTelex version ID
atITaxatieInfoBasic	Autotelex	Offline database appraisals from AutoTelex. Contains the minimum residual values of a vehicle
atITaxatieInfoAdvanced	Autotelex	Offline database appraisals from AutoTelex with additional information on list prices.
atITaxatieOnline	Autotelex	Includes AutoTelex residual values based on specified mileage and any other correction factors
atdKoppel	Autodisk	Autodisk versions connected to a registration
wegenbelasting	VWE	Road tax rates for the registration



## 5.2. Overview XSD schemas

There are 3 generic XSD schemas:

XSD schema	Details
RequestStandaardberichten.xsd	XSD of request message
ResponseDataTypes.xsd	documentation on first part of reply message
SharedVehicleDataTypes	Contains generic vehicle types

For each section there is a specific XSD.

Data section	XSD schema
rdwInfoBasic	rdwDataTypes.xsd
rdwInfoAdvanced	rdwDataTypes.xsd
rdwInfoFull	rdwDataTypes.xsd
milieuInfoBasic	milieuDataTypes.xsd
rdwHistInfoBasic	rdwDataTypes.xsd
rdwHistInfoAdvanced	rdwDataTypes.xsd
atIMmtInfo	AtIDataTypes.xsd
atITechInfoBasic	AtIDataTypes.xsd
atITechInfoAdvanced	AtIDataTypes.xsd
atIPriceInfoBasic	AtIDataTypes.xsd
atIOptieFabriek	AtIDataTypes.xsd
atIOptiePakket	AtIDataTypes.xsd
atIOptieStandaard	AtIDataTypes.xsd
atITaxatieInfoBasic	AtIDataTypes.xsd
atITaxatieInfoAdvanced	AtIDataTypes.xsd
atITaxatieOnline	AtIDataTypes.xsd
atdKoppel	AtdDataTypes.xsd
wegenbelasting	MrbDataTypes.xsd

NOTE: one XSD can support one or more items!



## 6. Support

---

This manual has been made to help you as much as possible. If you do have any problems or questions regarding programming the webservice please do not hesitate to contact our Customer Care Centre.

### Openingtimes and telephonenumber

Monday to Friday	08.00 till 21.00 o'clock	Tel: 088 – 893 7001
Zaterdag	09.00 till 17.00 o'clock	Tel: 088 – 893 7001

**Fax** 072 - 572 1604

### E-mail

[info@vwe.nl](mailto:info@vwe.nl)

**DISCLAIMER:** VWE only delivers support on the services as described in this manual. Although VWE supplies an PHP example "as is". We cannot give full support on PHP or other programming languages.



## 7. Appendix 1 PHP example standaard request

**The example below is for inspiration. We can't provide any support on this example.**

```
<?php
$connectionHost = "acceptatie-interdata.vwe.nl"; // vervang door interdata.vwe.nl voor productie!
$connectionPath = "/DataAanvraag.asmx/standaardDataRequest";
$_connect = fsockopen('ssl://'.$connectionHost, 443, $errno, $errstr);

if (!$_connect)
{
    die('Geen verbinding');
}

// Vul in <naam>, <wachtwoord> en <berichtsoort> de gegevens in die u van VWE heeft ontvangen
// In <kenteken> kunt u het te bevragen kenteken invullen
$postData = 'requestXml=<bericht>
    <authenticatie>
        <naam>[Vul gebruikersnaam in]</naam>
        <wachtwoord>[Vul wachtwoord in]</wachtwoord>
        <berichtsoort>[Vul uw berichtnaam in]</berichtsoort>
        <referentie>[Vul uw referentie in (optioneel)]</referentie>
    </authenticatie>
    <parameters>
        <kenteken>[Vul kenteken zonder streepjes in]</kenteken>
    </parameters>
</bericht>';

fputs($_connect, "POST ".$connectionPath." HTTP/1.1\r\n");
fputs($_connect, "Host: ".$connectionHost."\r\n");
fputs($_connect, "Content-Type: application/x-www-form-urlencoded\r\n");
fputs($_connect, "Content-Length: ".strlen($postData)."\r\n");
fputs($_connect, "Connection: close\r\n");
fputs($_connect, "\r\n"); // all headers sent
fputs($_connect, $postData);

$bCatch = false;
$postResult = "";
while (!feof($_connect))
{
    $line = fgets($_connect, 128);

    if ($bCatch == true)
        $postResult .= $line;

    if ($line == "\r\n")
        $bCatch = true;
}

// Root node toevoegen om XML te kunnen parsen
$postResult = str_replace('<?xml version="1.0" encoding="utf-8"?>', '<root>', $postResult). '</root>';

// Parse volledige reponse
$response_xml = simplexml_load_string($postResult);

// Parse de antwoordbericht node
$xml_antwoordbericht = simplexml_load_string($response_xml->string);

// Haal de childs op van de atITaxatieOnline node. Hiervoor moet de atl: namespace bekend worden
gemaakt
```





```
$ns_dc = $xml_antwoordbericht->rubrieken->atITaxatieOnline-  
>children('http://www.xmlmode.nl/interdata/atl');  
echo 'kenteken: ' . $ns_dc->kenteken . '<br>';  
echo 'atlCode: ' . $ns_dc->atlCode . '<br>';  
echo 'uitvoering: ' . $ns_dc->uitvoering . '<br>';  
echo 'ranking: ' . $ns_dc->ranking . '<br>';  
echo 'merk: ' . $ns_dc->merk . '<br>';  
echo 'model: ' . $ns_dc->model . '<br>';  
echo 'dagwaardeVerkoop: ' . $ns_dc->dagwaardeVerkoop . '<br>';  
echo 'dagwaardeInruil: ' . $ns_dc->dagwaardeInruil . '<br>';  
echo 'dagwaardeHandel: ' . $ns_dc->dagwaardeHandel . '<br>';  
  
fclose($_connect);  
?>
```



## 8. Appendix 1 PHP example maatwerk/tailormade request

**The example below is for inspiration. We can't provide any support on this example.**

```
<?php
$connectionHost = "acceptatie-interdata.vwe.nl";
$connectionPath = "/DataAanvraag.asmx/DataRequest";
$kenteken      = '74PHTJ';
$_connect = fsockopen('ssl://'.$connectionHost, 443, $errno, $errstr);

if (!$_connect)
{
    echo 'Geen verbinding';
}

$postData = 'XMLBody=<bericht>z
    <authenticatie>
    <naam>xxxxxx</naam>
    <wachtwoord>xxxxx</wachtwoord>
    <berichtsoort>Vul uw berichtnaam in</berichtsoort>
    <referentie>test</referentie>
</authenticatie>
<parameters>
    <kenteken>fill in a licenceplate number without spaces or hyphen</kenteken>
</parameters>
</bericht>';

fputs($_connect, "POST ".$connectionPath." HTTP/1.1\r\n");
fputs($_connect, "Host: ".$connectionHost."\r\n");
fputs($_connect, "Content-Type: application/x-www-form-urlencoded\r\n");
fputs($_connect, "Content-Length: ".strlen($postData)."\r\n");
fputs($_connect, "Connection: close\r\n");
fputs($_connect, "\r\n"); // all headers sent
fputs($_connect, $postData);

$bCatch = false;
$postResult = "";
while (!feof($_connect))
{
    $line = fgets($_connect, 128);

    if ($bCatch == true)
        $postResult .= $line;

    if ($line == "\r\n")
        $bCatch = true;
}

// Root node toevoegen om XML te kunnen parsen
$postResult = str_replace('<?xml version="1.0" encoding="utf-8"?>', '<root>', $postResult);
$postResult .= '</root>';

// Parse volledige reponse
$response_xml = simplexml_load_string($postResult);

// Parse de antwoordbericht node
$xml_antwoordbericht = simplexml_load_string($response_xml->string);

print_r($xml_antwoordbericht);
```



```
// Haal de childs op van de atITaxatieOnline node. Hiervoor moet de atl: namespace bekend worden  
gemaakt  
//$ns_dc = $xml_antwoordbericht->berichtsoort->PP-BASIS->children('http://www.xmlmode.nl/interdata');  
  
//echo 'merk: ' . $ns_dc->merk . '<br>';  
  
fclose($_connect);  
?>
```