

CG Gateway Web service

Architecture and API

Version: 2.3

Date: 01/03/2018

CreditGuard 2018 LTD ©

All rights reserved

Version	Writer	Date	
Document			
1.0	Tzvika Velich	26/03/2008	First version
2.0	Itamar Lask	08/08/2008	Fixes and formatting
2.1	Itamar Lask	10/08/2009	Add https support documentation
2.2	Tzvika Velich	29/06/2011	Fix URLs for https support
2.3	Tzvika Velich	01/03/2018	Remove deprecated & update documentation

Change List

Version #	Date	Description	Pages	Changer Name
1	26/03/2008	First version	All document	Tzvika Velich
2	27/03/2008	review version	All document	Itamar Lask
3	30/03/2008	Adding XPO Web Service API	All document	Tzvika Velich
4	30/03/2008	Change formatting and data	All document	Itamar Lask
5	08/08/2008	Change formatting and data	All document	Itamar Lask
6	10/08/2009	Add HTTPS support documentation		Itamar Lask
7	29/06/2011	Fix URLs for https support		Tzvika Velich
8	01/03/2018	Remove deprecated java example	Pages 6 – 13	Tzvika Velich
9	01/03/2018	Update WSDL	Pages 7 – 9	Tzvika Velich

1. General

1.1 This document describes the CG Gateway web module interfaces:

1.1.1 HTTPS POST interface.

1.1.2 Web service API.

1.2 The CG Gateway Solution

1.2.1 CreditGuard's CG Gateway solution enables you to conveniently work with various clearance authorities using standard XML request/response format.

1.2.2 CG Gateway system receives these requests through HTTPS post or Web Service API.

2. CG Gateway HTTPS Post interface

2.1 Usage

2.1.1 When accessing CG Gateway via HTTPS Post, one must implement the HTTPS Post method on the client side.

2.1.2 Always use the server name when accessing the web service (which should be the server and the certificate name) – in order to prevent certificate authentication errors.

2.1.3 Accessing the HTTPS interface is done via the following URL:

https://server_name/xpo/Relay

2.1.4 Request Parameters (submitted via HTTPS Post):

2.1.4.1 user=<username>

2.1.4.1.1 The name of the CG Gateway user.

2.1.4.2 password=<password>

2.1.4.2.1 The password of the CG Gateway user

2.1.4.3 int_in =<request details according to XML API standards>

2.1.4.3.1 The request payload formatted as XML.

2.1.4.3.2 The XML API full spec is detailed in additional documentation provided by CreditGuard for the specific request/command type.

2.1.5 Response

2.1.5.1 The response is formatted as a single string containing the XML response.

3. CG Gateway Web Service

3.1 Usage

- 3.1.1 Always use the server name when accessing the web service (which should be the server and the certificate name)
- 3.1.2 The WSDL location is -
https://server_name/xpo/services/Relay?wsdl
- 3.1.3 The access to CG Gateway Web Service itself -
https://server_name/xpo/services/Relay

3.2 Overview

- 3.2.1 This section will elaborate on the main CG Gateway WS function called “ashraitTransaction”.
- 3.2.2 ashraitTransaction is a generic function which allows to use CG Gateway functionality through standard web service calls.
- 3.2.3 The function interface has three string input parameters (user, password and a XML string called int-in).
- 3.2.4 The function response output is the corresponding XML formatted string.
- 3.2.5 For further details on the XML input and output format please refer to additional documentation provided by CreditGuard for the specific request/command type.

3.3 CG Gateway Web Service function – ashraitTransaction

3.3.1 ashraitTransaction input parameters:

3.3.1.1 user – type String, the name of the CG Gateway user.

3.3.1.2 password – type String, the password of the CG Gateway user.

3.3.1.3 int-in – type String, the request payload formatted as XML.

3.3.1.3.1 The XML API full spec is detailed in the "CG Gateway XML API" document.

3.3.1.3.2 The following is a sample XML request:

```
<ashrait>
  <request>
    <command>doDeal</command>
    <requestid/>
    <dateTime></dateTime>
    <version>1000</version>
    <language>HEB</language>
    <doDeal>
      <terminalNumber>0962922</terminalNumber>
      <cardNo>64818357</cardNo>
      <track2></track2>
      <cardExpiration>1010</cardExpiration>
      <cardId></cardId>
      <total>13391</total>
      <firstPayment></firstPayment>
      <periodicalPayment></periodicalPayment>
      <numberOfPayments></numberOfPayments>
      <transactionType>Debit</transactionType>
      <creditType>RegularCredit</creditType>
      <authNumber>55555</authNumber>
      <currency>ILS</currency>
      <cvv></cvv>
      <mainTerminalNumber></mainTerminalNumber>
      <transactionCode>Phone</transactionCode>
      <validation>autoComm</validation>
      <user>6666</user>
      <customerData/>
    </doDeal>
  </request>
</ashrait>
```

3.3.2 ashraitTransaction WSDL file

- 3.3.2.1 The WSDL will be found at - https://server_name/xpo/services/Relay?wsdl
- 3.3.2.2 Following is the WSDL file specifying the ashraitTransaction API.
- 3.3.2.3 The <wsdlsoap:address> tag at the end of the WSDL file should be set to the WS server location.

This is set during the installation.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://xpo.xor.com"
xmlns:apachesoap="http://xml.apache.org/xml-soap" xmlns:impl="http://xpo.xor.com"
xmlns:intf="http://xpo.xor.com" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <!--WSDL created by Apache Axis version: 1.4
  Built on Apr 22, 2006 (06:55:48 PDT)-->
  <wsdl:types>
    <schema elementFormDefault="qualified" targetNamespace="http://xpo.xor.com"
xmlns="http://www.w3.org/2001/XMLSchema">
      <element name="ashraitTransaction">
        <complexType>
          <sequence>
            <element name="user" type="xsd:string"/>
            <element name="password" type="xsd:string"/>
            <element name="int_in" type="xsd:string"/>
          </sequence>
        </complexType>
      </element>
      <element name="ashraitTransactionResponse">
        <complexType>
          <sequence>
            <element name="ashraitTransactionReturn" type="xsd:string"/>
          </sequence>
        </complexType>
      </element>
      <element name="ashraitSessionTransaction">
        <complexType>
          <sequence>
            <element name="sessionId" type="xsd:string"/>
            <element name="int_in" type="xsd:string"/>
          </sequence>
        </complexType>
      </element>
      <element name="ashraitSessionTransactionResponse">
        <complexType>
          <sequence>
            <element name="ashraitSessionTransactionReturn" type="xsd:string"/>
          </sequence>
        </complexType>
      </element>
    </schema>
  </wsdl:types>

  <wsdl:message name="ashraitTransactionRequest">
    <wsdl:part element="impl:ashraitTransaction" name="parameters">
    </wsdl:part>
  </wsdl:message>

  <wsdl:message name="ashraitSessionTransactionRequest">
    <wsdl:part element="impl:ashraitSessionTransaction" name="parameters">
    </wsdl:part>
  </wsdl:message>
</wsdl:definitions>
```

```

</wsdl:message>

<wsdl:message name="ashraitSessionTransactionResponse">

  <wsdl:part element="impl:ashraitSessionTransactionResponse" name="parameters">

    </wsdl:part>

  </wsdl:message>

<wsdl:message name="ashraitTransactionResponse">

  <wsdl:part element="impl:ashraitTransactionResponse" name="parameters">

    </wsdl:part>

  </wsdl:message>

<wsdl:portType name="Relay">

  <wsdl:operation name="ashraitTransaction">

    <wsdl:input message="impl:ashraitTransactionRequest"
name="ashraitTransactionRequest">

      </wsdl:input>

      <wsdl:output message="impl:ashraitTransactionResponse"
name="ashraitTransactionResponse">

        </wsdl:output>

      </wsdl:operation>

    <wsdl:operation name="ashraitSessionTransaction">

      <wsdl:input message="impl:ashraitSessionTransactionRequest"
name="ashraitSessionTransactionRequest">

        </wsdl:input>

        <wsdl:output message="impl:ashraitSessionTransactionResponse"
name="ashraitSessionTransactionResponse">

          </wsdl:output>

        </wsdl:operation>

      </wsdl:portType>

    <wsdl:binding name="RelaySoapBinding" type="impl:Relay">

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

      <wsdl:operation name="ashraitTransaction">

        <wsdlsoap:operation soapAction=""/>

        <wsdl:input name="ashraitTransactionRequest">

          <wsdlsoap:body use="literal"/>

        </wsdl:input>

        <wsdl:output name="ashraitTransactionResponse">

          <wsdlsoap:body use="literal"/>

        </wsdl:output>

```

```
</wsdl:operation>

<wsdl:operation name="ashraitSessionTransaction">

    <wsdlsoap:operation soapAction=""/>

    <wsdl:input name="ashraitSessionTransactionRequest">

        <wsdlsoap:body use="literal"/>

    </wsdl:input>

    <wsdl:output name="ashraitSessionTransactionResponse">

        <wsdlsoap:body use="literal"/>

    </wsdl:output>

</wsdl:operation>

</wsdl:binding>

<wsdl:service name="RelayService">

    <wsdl:port binding="impl:RelaySoapBinding" name="Relay">

        <wsdlsoap:address
location="https://<SERVER_DOMAIN>/xpo/services/Relay"/>

    </wsdl:port>

</wsdl:service>

</wsdl:definitions>
```