CG Gateway
Session Authentication

Implementation & Integration
API Document

Version: 1.0
Date: 10/08/2016

Version	Writer	Date	
Document			
1.0	Tzvika Velich & Itamar Lask	17/01/2016	All document
1.1	Yahalom Emet	10/08/2016	Adding new error codes
1.2	Hai mizrachi	13/10/2023	Adding scope use

Preface

1. Overview

1.1 Standard CreditGuard's Gateway interface require using user/password authentication for each API call.

1.2 Some situations require the capability to create an API call without using the user or password parameters values for security reasons (Mobile device usage etc').

1.3 CreditGuard's session authentication module enables merchants to perform requests to CG Gateway server with minimal exposure of the API's username and password.

1.4 This interface requires a two stage process:

1.4.1 Obtain session id by executing a getSessionId initial API call to CG Gateway (server to Server call, using the standard API authenticating by user and password).

1.4.2 Create a consecutive API call to CG Gateway, using the acquired sessionId instead of the user/password credentials.

1.5 When doing so, the consecutive API call will obtain the initial API username/password credentials for executing the request.

1.6 On CG Gateway merchant and API credentials now include two additional settings:

1.6.1 Session expiration (seconds) – determines for which period of time a provided session id is valid from the initial or last API call. Default value is 600 seconds.

1.6.2 Session reuse (Boolean 1|0) – determines whether a session id can be used for multiple consecutive API calls. Default value is 0 – hence a session id can only be used once.

1.6.3 Session can limit request by command, validation and type, Use according to the demand command using scope.

1.7 This document will outline the API a one should implement for using session authentication.

2. Related Documents

2.1 CG Gateway XML API 1.33

3. Terms

3.1 CG Gateway – Credit Guard's payment gateway

3.2 Merchant – System/Website that performs payment requests.

3.3 Credit Card - Payment method using credit card details

Implementing Session Authentication

1. Obtaining Session Id for Authentication

1.1 Using The Command getSessionId using the standard user/password authentication interface from a secured environment.

1.2 **Request:**

```
<ashrait>
    <request>
            <version>1001</version>
            <language>ENG</language>
            <dateTime/>
            <command>getSessionId</command>
            <requestid/>
            <getSessionId>
                    <customerData>
                       <userData1/>
                       <userData2/>
                       <userData3/>
                       <userData4/>
                       <userData5/>
                       <userData6/>
                       <userData7/>
                       <userData8/>
                       <userData9/>
                       <userData10/>
                    </customerData>
            </getSessionId>
    </request>
</ashrait>
```

1.3 **Request TAGs:**

Tag Name	Type	Value	Value Mandatory	Description
userData1	AlphaNumeric (256)	User defined field	No	User defined attribute, this value will be included in the response as it was sent.
userData2	AlphaNumeric (256)	User defined field	No	
userData3	AlphaNumeric (256)	User defined field	No	
userData4	AlphaNumeric (256)	User defined field	No	
userData5	AlphaNumeric (256)	User defined field	No	
userData6	AlphaNumeric (256)	User defined field	No	
userData7	AlphaNumeric (256)	User defined field	No	
userData8	AlphaNumeric (256)	User defined field	No	
userData9	AlphaNumeric (256)	User defined field	No	
userData10	AlphaNumeric (256)	User defined field	No	

1.4 **Response**:

```
<ashrait>
    <response>
            <command>getSessionId</command>
            <dateTime>2011-08-04 13:59</dateTime>
            <requestId></requestId>
            <tranId>5028</tranId>
            <result>000</result>
            <message>Permitted transaction.</message>
            <userMessage>Permitted transaction.</userMessage>
            <additionalInfo></additionalInfo>
            <version>1001</version>
            <language>Eng</language>
            <getSessionId>
                    <status>000</status>
                    <sessionId>1c42ade3-d80b-4516-5f9c-3d4bab0fc3cb</sessionId>
                    <sessionExpiration>600</sessionExpiration>
                    <sessionReUse>0</sessionReUse>
                    <customerData>
                       <userData1/>
                       <userData2/>
                       <userData3/>
                       <userData4/>
                       <userData5/>
                       <userData6/>
                       <userData7/>
                       <userData8/>
                       <userData9/>
                       <userData10/>
                    </customerData>
            </getSessionId>
    </response>
</ashrait>
```

1.5 **Possible Response Codes**

1.5.1 000 – permitted transaction

1.5.2 **405 – SSL HTTPS customers are not permitted to access the system.**

this is a standard generic authentication failure response returned by CG Gateway when user/password or session authentication fails.

1.5.3 455- merchant does not support session id

1.5.4 456 – merchant session timeout

1.5.5 457 – session id generation failed

1.5.6 XXX – gateway general error codes (308, 303, 354, 444, please refer to API XML document for complete documentation)

1.6 Response TAGs

Tag Name	Type	Value	Description
sessionId	GUID AlphaNumeric (36)		The returned sessionid value
sessionExpiration	Numeric		The session duration validity (seconds)
sessionReUse	Numeric	0 | 1	Documents if the session can be reused in consequtive API call. When merchant configuration set to reuse session any API call will extend the session expiration validity by the value returned in field sessionExpiration.
userData1	AlphaNumeric (256)	User defined field	User defined attribute, this value will be included in the response as it was sent.
userData2	AlphaNumeric (256)	User defined field	
userData3	AlphaNumeric (256)	User defined field	
userData4	AlphaNumeric (256)	User defined field	
userData5	AlphaNumeric (256)	User defined field	
userData6	AlphaNumeric (256)	User defined field	
userData7	AlphaNumeric (256)	User defined field	
userData8	AlphaNumeric (256)	User defined field	
userData9	AlphaNumeric (256)	User defined field	
userData10	AlphaNumeric (256)	User defined field	

Limit requests by using scope

When use Command getSessionId can limit the request by use scope tag

to limit the created session to a specific request type be command, validation, transactionType.

Important : If validation or transactionType was transferred, scopeCmd must be transferred

1.1 creating a sessionId that can be used as a token only for a dodeal type command

Request:

```
<ashrait>
    <request>
            <version>1001</version>
            <language>ENG</language>
            <dateTime/>
            <command>getSessionId</command>
            <requestid/>
            <getSessionId>
                    <scope>
                       <scopeCmd>doDeal</scopeCmd>
                    </scope>
            </getSessionId>
    </request>
</ashrait>
```

1.2 creating a sessionId that can be used as a token only for a dodeal type command and - normal type validation

Request:

```
<ashrait>
    <request>
            <version>1001</version>
            <language>ENG</language>
            <dateTime/>
            <command>getSessionId</command>
            <requestid/>
            <getSessionId>
                    <scope>
                       <scopeCmd>doDeal</scopeCmd>
                        <validation>normal</validation>
                    </scope>
            </getSessionId>
    </request>
</ashrait>
```

1.3 creating a sessionId that can be used as a token only for a dodeal type command and - validation of the normal type and transactionType of the Credit type

Request:

```
<ashrait>
    <request>
            <version>1001</version>
            <language>ENG</language>
            <dateTime/>
            <command>getSessionId</command>
            <requestid/>
            <getSessionId>
                    <scope>
                       <scopeCmd>doDeal</scopeCmd>
                        <validation>normal</validation>
                       <transactionType>Credit</transactionType>
                    </scope>
            </getSessionId>
    </request>
</ashrait>
```

How to Send Requests to CG Gateway?

1. Overview

1.1 CG Gateway support two major generic interfaces for any request-response interaction: HTTPS Post and Web Service (SOAP over HTTPS).

1.2 Both interfaces are generic and used for sending any request (transaction, cancelation, refund, transaction setup, query etc') to CG Gateway.

2. HTTPS Post Request Interface (user/password authentication)

2.1 The merchant system should use a HTTPS POST mechanism for sending the request.

2.2 Always use the server full provided DNS name when accessing the service (which should point to the server and the certificate name) – this prevents certificate authentication errors.

2.3 Accessing the HTTPS interface is done via the following URL:
https://server_name/xpo/Relay

2.3.1 Server name will be assigned to the merchant within the integration process.

2.4 Request

2.4.1.1 Request Parameters (submitted via HTTPS Post):

2.4.1.2 user=<username>

2.4.1.2.1 the name of the CG Gateway API user.

2.4.1.3 password=<password>

2.4.1.3.1 The password of the CG Gateway API user

2.4.1.4 int_in =<transaction details according to XML API standards detailed on the relevant sections>

2.5 Response

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

3. CG Gateway Web Service Interface (user/password authentication)

3.1 Alternatively, the merchant can use the web service interface.

3.2 The service exposes the generic WSDL at:
https://server_name/xpo/services/Relay?wsdl

3.3 The actual CG Gateway Web Service URL is:
https://server_name/xpo/services/Relay

3.4 Server name will be assigned to the merchant within the integration process.

3.5 The main CG Gateway WS function is called "ashraitTransaction".

3.6 ashraitTransaction is a generic function which allows to use CG Gateway functionality through a standard web service call.

3.7 The function interface has three string input parameters :

3.7.1 user – type String, the name of the CG Gateway API user.

3.7.2 password – type String, the password of the CG Gateway API user.

3.7.3 Int_in – type String, XML or SHVA Ashrait96 Int_In formatted request.

3.8 The function response output is the corresponding XML formatted string.

3.9 For further details on the XML input and output format please refer to the relevant sections describing the necessary XML request format.

4. HTTPS Post Request Interface (session authentication)

4.1 Alternatively the merchant system should use a HTTPS POST mechanism for sending the request using the session authentication.

4.2 Always use the server full provided DNS name when accessing the service (which should point to the server and the certificate name) – this prevents certificate authentication errors.

4.3 Accessing the HTTPS interface is done via the following URL:
https://server_name/xpo/Relay

4.3.1 Server name will be assigned to the merchant within the integration process.

4.4 **Request**

Request Parameters (submitted via HTTPS Post):

4.4.1.1 sessionId=<sessionId>

4.4.1.1.1 A valid session id value that was previously returned using the getSessionId request.

4.4.1.2 int_in =<transaction details according to XML API standards detailed on the relevant sections>

4.5 **Response**

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

5. CG Gateway Web Service Interface (session authentication)

5.1 Alternatively, the merchant can use the web service interface.

5.2 The service exposes the generic WSDL at:
https://server_name/xpo/services/Relay?wsdl

5.3 The actual CG Gateway Web Service URL is:
https://server_name/xpo/services/Relay

5.4 Server name will be assigned to the merchant within the integration process.

5.5 The main CG Gateway WS function is called "**ashraitSessionTransaction**".

5.6 **ashraitSessionTransaction** is a generic function which allows to use CG Gateway functionality through a standard web service call.

5.7 The function interface has two string input parameters:

5.7.1 **sessionId** – type String, A valid session id value that were previously returned in getSessionId command.

5.7.2 **Int_in** – type String, XML or SHVA Ashrait96 Int_In formatted request.

5.8 The function response output is the corresponding XML formatted string.

5.9 For further details on the XML input and output format please refer to the relevant sections describing the necessary XML request format.