CG Gateway Redirect (MPI)
Hosted Payment Page

Implementation & Integration
API Document

Version: 2.4
Date: 19/02/2020

Version	Writer	Date	
Document			
2.0	Itamar Lask	17/12/2015	Full document
2.1	Tzvika Velich	12/07/2016	Page 21, add note on query result when transaction did not approved by the buyer.
2.2	Tzvika Velich & Miriam Rosenthal	14/09/2016	Page 27, enhancing numberOfPayments to include custom list. Adding payments interest tag feature.
2.3	Tzvika Velich	17/12/2016	Adding asynchronous callback report and re-write error codes
2.4	Tzvika Velich	19/02/2020	Update ampersand sign for dynamic URLs

CreditGuard Ltd. 18 Ben Gurion st., Givat Shmuel T: +972 3 7370700 F: +972 3 7370790

Preface

1. Overview

1.1 CreditGuard's CG Redirect (MPI) & CG Gateway solution enables merchants to perform credit card transactions without having to deal with credit card data.

1.2 The concept is based on securely receiving transaction data from the merchant, redirecting card holders to a secured payment web page, and then securely receiving the transaction outcome (success/failure).

1.3 This document will outline the API a merchant has to implement for interacting with the CG Redirect (MPI).

2. Related Documents

2.1 CG Gateway XML API

3. Terms

3.1 CG Gateway – Credit Guard's payment gateway

3.2 CG Redirect – Credit Guard's Merchant Plugin Server (MPI)

3.3 MPI – "CG Redirect server"

3.4 Merchant – System/Website that performs payment requests.

3.5 End User/Card Holder/Buyer – Customer who performs online shopping on the merchant web site.

3.6 PayPal - Payment method using "PayPal Express Checkout"

3.7 Credit Card - Payment method using credit card details

4. General Flow

4.1 The Buyer browses the merchant web site, adding items to the shopping cart.

4.2 When checkout is required, the merchant sends transaction information to CG Redirect by either HTTPS Post or Web service request (SOAP over HTTPS).

4.3 CG Redirect validates the request and returns a unique one-time Payment Page URL back to merchant (the URL is valid for 600 seconds, cannot be reused once invoked and transaction was attempted).

4.4 The merchant uses this URL and redirects the user to CG Redirect responsive payment page.

4.5 The buyer enters credit card information at this payment page (including card number, expiration date, CVV2/CVC, Israeli ID number, installments etc')

4.6 CG Redirect uses credit card data, and transaction information to perform the actual payment transaction through CG Gateway.

4.7 At the end of the transaction, the buyer is redirected to predefined success/failure landing web page on the merchant's web site (these landing pages can be predefined on the merchant setup, or sent dynamically on the initial URL request).

4.8 The redirect page includes transaction identification, transaction return code, transaction authorization number etc')

4.9 At any time, the merchant can then query the system via an additional API and get the transaction full details.

4.10 Please note that CG Redirect can be used in order to perform any supported CG Gateway transaction type by using the appropriate initial request. This is achieved by populating a specific tag on the initial URL request so CG Redirect can be used in order to perform a one phase sale, authorization only, or tokenization only action – please refer to CG Gateway XML API for further details.

5. Process Diagram
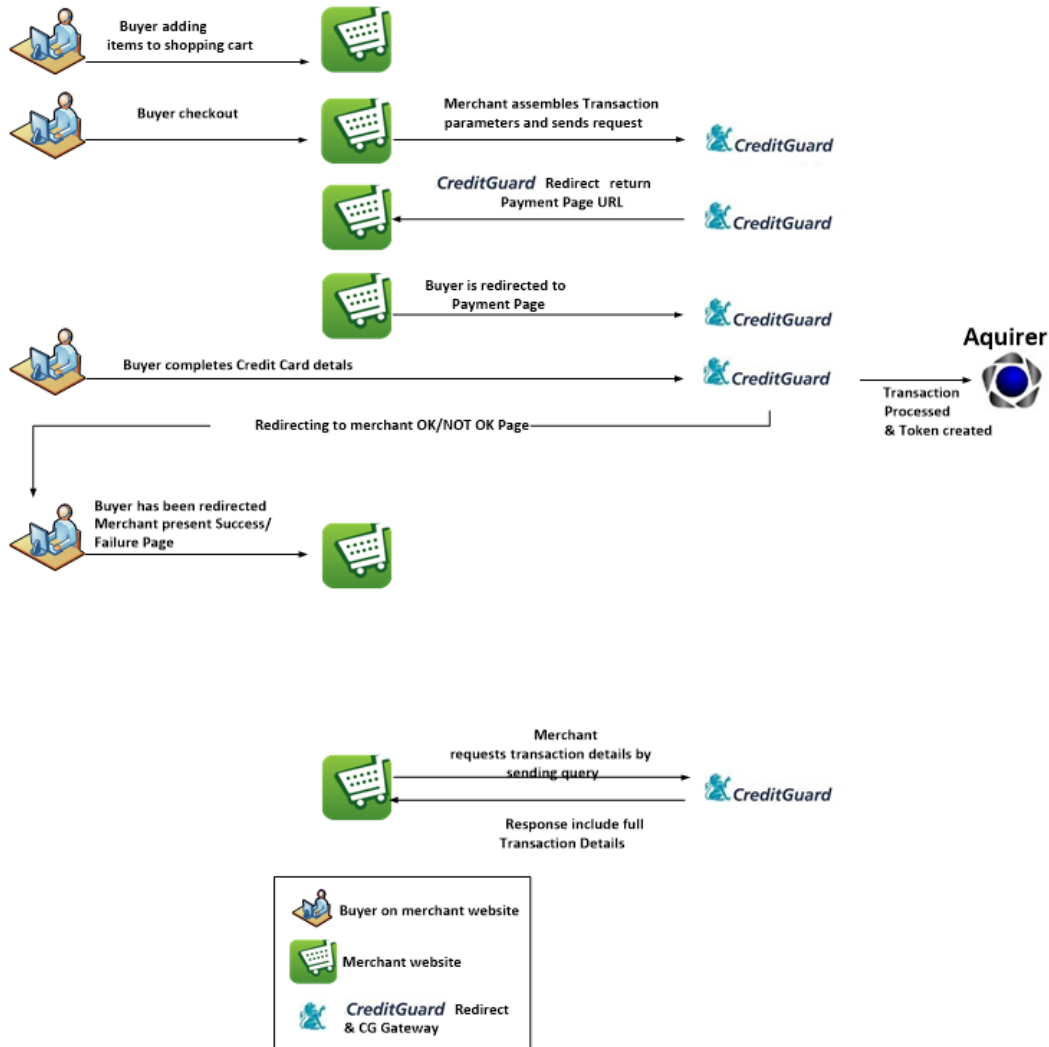
How To Send Requests to CG Gateway ?

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.

1.3   This is achieved by using a uniform interface with three major parts:

    1.3.1   username

    1.3.2   password

    1.3.3   int_in – request content field, containing the full transaction internal CG Gateway XML API protocol.

2. HTTPS Post Request Interface

2.1   The merchant system should post 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

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.

3.10 CG Gateway sample WSDL:

```xml
<?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>
        </schema>
    </wsdl:types>
    <wsdl:message name="ashraitTransactionRequest">
        <wsdl:part element="impl:ashraitTransaction" name="parameters" />
    </wsdl:message>
    <wsdl:message name="ashraitTransactionResponse">
        <wsdl:part element="impl:ashraitTransactionResponse" name="parameters" />
    </wsdl:message>
    <wsdl:portType name="Relay">
        <wsdl:operation name="ashraitTransaction">
            <wsdl:input message="impl:ashraitTransactionRequest"
name="ashraitTransactionRequest" />
```

```
            <wsdl:output message="impl:ashraitTransactionResponse"
name="ashraitTransactionResponse" />
        </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:binding>
    <wsdl:service name="RelayService">
        <wsdl:port binding="impl:RelaySoapBinding" name="Relay">
            <wsdlsoap:address location="https://server_name/xpo/services/Relay" />
        </wsdl:port>
    </wsdl:service>
</wsdl:definitions>
```

Simple Credit Card Transaction Setup

1. Overview

1.1 This section will detail how to perform a simple one phase sale transaction process using CG redirect hosted payment page.
How to initiate a call for getting the hosted page URL, open that URL and once the card holder is redirected back to the merchant site, query the CG Gateway system for the transaction outcome.

2. Transaction Setup

2.1 The following is the XML that should be posted as the INT_IN request variable (via https or web service).

2.2 For posting the request please refer to appendix A.

2.3 When sending this request, a transaction is set up in the system background and a URL is retrieved, in the page opened when accessing the url, the card holder is adding required transaction information (card number, exp date, etc') in the purchase process.

2.4 Setting up a transaction can be used to perform a one phase sale, two phase sale, or only tokenization (card number to card id or vise versus), all according to the "mpiValidation" tag value – in the following example we perform a one phase sale (validation is "AutoComm").

2.5 For additional transaction types please refer to "CG Gateway XML API" document.

2.6 **XML Request**

```xml
<ashrait>
    <request>
            <version>1001</version>
            <language>EN</language>
            <dateTime/>
            <command>doDeal</command>
            <requestid/>
            <doDeal>
                    <terminalNumber>096XXXXX</terminalNumber>
                    <cardNo>CGMPI</cardNo>
                    <successUrl></successUrl>
                    <errorUrl></errorUrl>
                    <cancelUrl></cancelUrl>
                    <total>10000</total>
                    <transactionType>Debit</transactionType>
                    <creditType>RegularCredit</creditType>
                    <currency>ILS</currency>
                    <transactionCode>Phone</transactionCode>
                    <validation>TxnSetup</validation>
                    <firstPayment></firstPayment>
                    <periodicalPayment></periodicalPayment>
                    <numberOfPayments></numberOfPayments>
                    <user>request identifier</user>
                    <mid>123456</mid>
                    <uniqueid>837683509-84983498539</uniqueid>
                    <mpiValidation>AutoComm</mpiValidation>
                    <description>added description to payment page</description>
                    <email>test@creditguard.co.il</email>
                    <customerData>
                        <userData1/>
                        <userData2/>
                        <userData3/>
```

```
                    <userData4/>
                    <userData5/>
                    <userData6/>
                    <userData7/>
                    <userData8/>
                    <userData9/>
                    <userData10/>
                </customerData>
            </doDeal>
    </request>
</ashrait>
```

2.7   **Request Tags**

2.7.1   Most of the tags are fixed values, and should be submitted exactly as in the example above.

2.7.2   The following are the variable tags details:

Tag Name	Type	Value	Value Mandatory	Description
terminalNumber	AlphaNumeric		Yes	Terminal number as defined for the merchant at CG Gateway within the integration process
cardNo	AlphaNumeric(19)	**CGMPI**	Yes	Card number in regular transaction, must be CGMPI for transaction setup
successUrl	String (256)	Must be a Legal URL	No	OK page URL – overrides setup parameter
errorUrl	String (256)	Must be a Legal URL	No	NOTOK page URL - overrides setup parameter
cancelUrl	String (256)	Must be a Legal URL	No	CANCEL page URL – when present the cancel button will appear on the payment page - overrides setup parameter
total	Numeric(8)		Yes	The total amount of the transaction in cents (agorot). 100 ILS will be 10000
currency	AlphaNumeric(3)	**ILS** USD GBP EUR JPY Etc'	Yes	Currency code, According to ISO-4217
transactionType	AlphaNumeric	**Debit**|Credit	Yes	Please see XML API
creditType	AlphaNumeric	**RegularCredit** Payments IsraCredit SpecialCredit SpecialAlpha PaymentsClub Etc'	Yes	Please see XML API
transactionCode	AlphaNumeric	**Phone** | Magnetic	Yes	All card non present transactions and e-commerce must be marked as 'Phone' Please see XML API for further details
validation	AlphaNumeric	**TxnSetup**	Yes	Indicates a request for MPI page and transaction setup
firstPayment	Numeric(20)		No	Amount of first payment
periodicalPayment	Numeric(20)		No	Each periodical payments amount

numberOfPayments	Numeric(2)		No	In case CreditType isn't Payments please see XML API. In case CreditType is payments: if firstPayment and periodicalPayment are filled please see XML API otherwise numberOfPayments indicates the maximum number of payments the card holder can choose on the payment page.
user	String (19)	User data	No	Transaction identifier (X Field)
mid	String (15)	Merchant ID	Yes	Allocated by CG Gateway for the merchant – identifies the redirect page & design
uniqueid	String (64)	Unique transaction ID	Yes	Generated by merchant. **This ID must be unique** at least within the last 24 hours. It is advisable to generate a new uuid for each transaction.
mpiValidation	AlphaNumeric	**AutoComm** Token Verify AutoCommHold CardNo	Yes	Indicates what transaction will be performed by the MPI and setup the hosted page transaction accordingly. A regular sale is AutoComm For tokenization only use 'Token'. For authorization only on a two phase sale use 'Verify' for reverse tokenization (when enabled) use 'CardNo', for other transaction type refer to "validation" tag XML API documentation
description	AlphaNumeric(256)	Description of sold goods/transaction	No	Will be optionally displayed on the payment page
email	AlphaNumeric (32)	Customer email	No	
userData1	AlphaNumeric (256)	User defined field	No	User defined attribute
userData2	AlphaNumeric (256)	User defined field	No	User defined attribute
userData3	AlphaNumeric (256)	User defined field	No	User defined attribute
userData4	AlphaNumeric (256)	User defined field	No	User defined attribute
userData5	AlphaNumeric (256)	User defined field	No	User defined attribute
userData6	AlphaNumeric (256)	User defined field	No	User defined attribute
userData7	AlphaNumeric (256)	User defined field	No	User defined attribute
userData8	AlphaNumeric (256)	User defined field	No	User defined attribute
userData9	AlphaNumeric (256)	User defined field	No	User defined attribute
userData10	AlphaNumeric (256)	User defined field	No	User defined attribute

2.8 **XML Response**

```xml
<ashrait>
    <response>
            <command>doDeal</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>1000</version>
            <language>Eng</language>
            <doDeal>
                    <status>000</status>
                    <statusText>Permitted transaction.</statusText>
                    <terminalNumber>0962922</terminalNumber>
                    <cardNo>CGMPI</cardNo>
                    <cardName></cardName>
                    <cardExpiration></cardExpiration>
                    <cardType code=""></cardType>
                    <creditCompany code=""></creditCompany>
                    <cardBrand code=""></cardBrand>
                    <cardAcquirer code=""></cardAcquirer>
                    <serviceCode></serviceCode>
                    <transactionType code="01">RegularDebit</transactionType>
                    <creditType code="1">RegularCredit</creditType>
                    <currency code="1">ILS</currency>
                    <transactionCode code="50">Phone</transactionCode>
                    <total>10000</total>
                    <balance></balance>
                    <starTotal></starTotal>
                    <firstPayment></firstPayment>
                    <periodicalPayment></periodicalPayment>
                    <numberOfPayments></numberOfPayments>
                    <clubId></clubId>
                    <clubCode></clubCode>
                    <validation code="106">TxnSetup</validation>
                    <commReason code=""></commReason>
                    <idStatus code=""></idStatus>
                    <cvvStatus code=""></cvvStatus>
                    <authSource code="6">MPIServer</authSource>
                    <authNumber>123456</authNumber>
                    <fileNumber></fileNumber>
                    <slaveTerminalNumber></slaveTerminalNumber>
                    <slaveTerminalSequence></slaveTerminalSequence>
                    <creditGroup></creditGroup>
                    <pinKeyIn></pinKeyIn>
                    <pfsc></pfsc>
                    <eci></eci>
                    <cavv code=""></cavv>
                    <user>request  identifier</user>
                    <addonData></addonData>
                    <supplierNumber></supplierNumber>
                    <intIn>Bxxxxxxx4580C200D011150E0123456J106TxxxxXX field</intIn>
                    <intOt></intOt>
                    <mid>123456</mid>
                    <uniqueid>837683509-84983498539</uniqueid>
                    <mpiValidation>autoComm</mpiValidation>
                    <email>test@creditguard.co.il</email>
                    <token>459c35c1-80d1-4fa7-b6ac-60c33987b958</token>
    <mpiHostedPageUrl>https://cgmpiuat.creditguard.co.il/CGMPI_Server/PerformTransacti
on?txId=459c35c1-80d1-4fa7-b6ac-60c33987b958</mpiHostedPageUrl>
            </doDeal>
    </response>
</ashrait>
```

2.9 Response Tags

2.9.1 The response has many tags that are relevant to different CG Gateway extended functionality capabilities.

2.9.2 Most of the request tags are also returned in the response.

2.9.3 As for performing an e-commerce & tokenization transaction, the relevant tags are highlighted in red. For all tags documentation please refer to "CG Gateway XML API" documentation.

2.9.4 The following are the necessary tags for integration:

Tag Name	Type	Value	Description
result	Numeric(4)		Request result code – please refer to appendix B for detailed result code list. '000' indicats a success
message	AlphaNumeric (32)		Success/Error text
mpiHostedPageUrl	AlphaNumeric (256)		The entire payment page URL.
token	AlphaNumeric (36)		The payment URL generated transaction id

3. Hosted Payment Page

3.1 Once the transaction setup was performed, the merchant now holds the payment page URL retrieved from the response (mpiHostedPageUrl tag).

3.2 Once retrieved, the merchant should redirect the card holder to the payment page.

3.3 CG Redirect page is a responsive payment page, that can be opened either as a full redirect page or within an iFrame on the merchant web site. CG Redirect transaction setup just results in the provided URL, and the merchant opens it as required.

3.4 A retrieved payment page may look in this form (server DNS name may change according to the dedicated hosted page server):
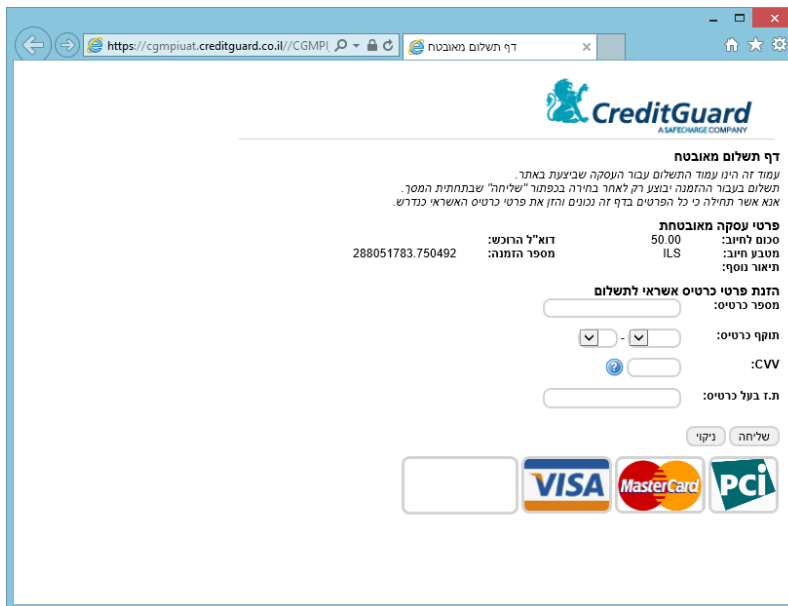
```
https://cgmpiuat.creditguard.co.il/CGMPI_Server/PerformTransaction?txId=<transactionID>
```

or specifically like this:

```
https://cgmpiuat.creditguard.co.il/CGMPI_Server/PerformTransaction?txId=459c35c1-80d1-4fa7-b6ac-
60c33987b958
```
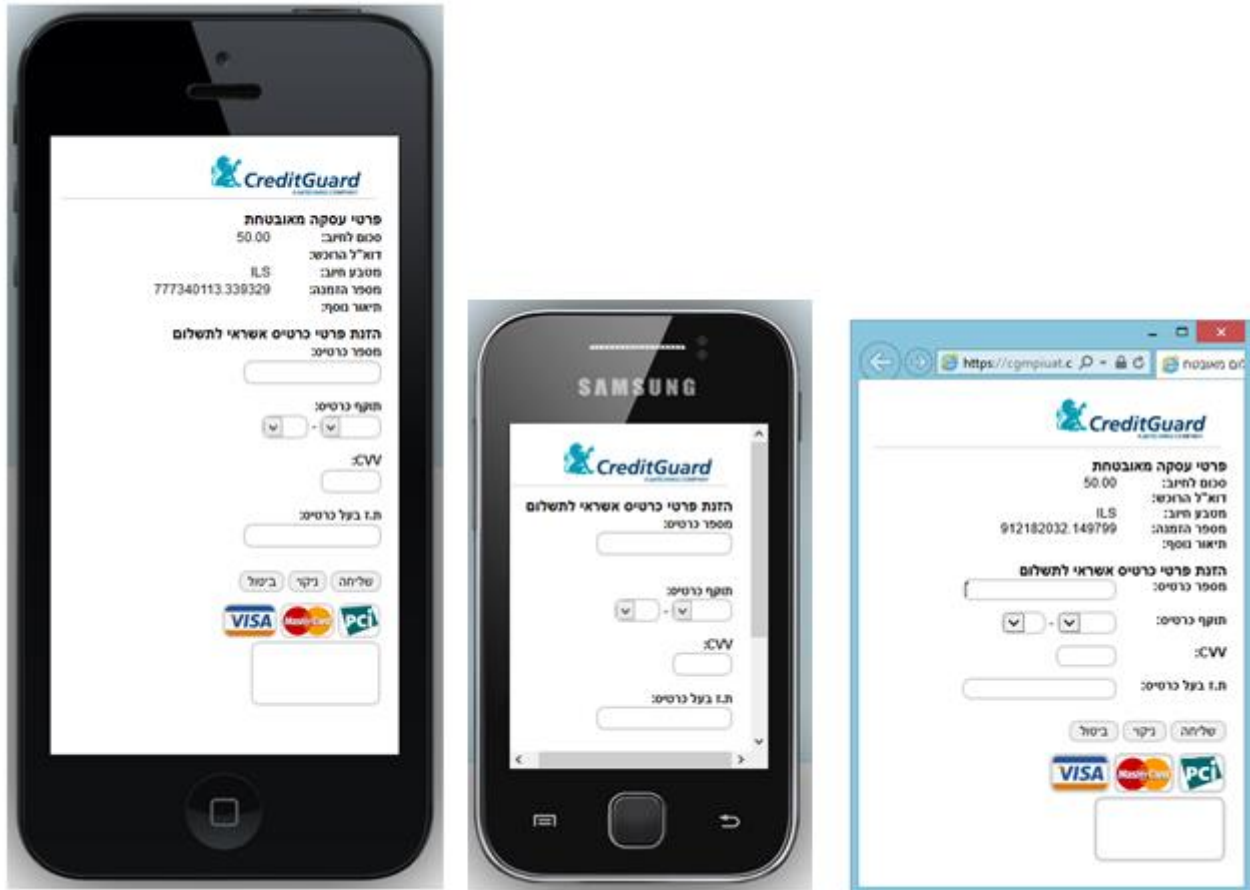
3.5 **Payment Page**

3.5.1 Once redirected to the payment page, the default page looks like this:

3.5.2 When submitted (PAY button is pressed) the actual transaction is performed.

3.5.3 When done, the card holder is redirected back the OK_URL or NOTOK_URL on the merchant web site (predefined landing pages set on the merchant configuration or dynamically sent on the request).

3.5.4 The redirection to the landing pages will contain additional parameters that specify the request/transaction status.

3.5.5 Please Note the transaction page is fully customizable, and also implement a

![CreditGuard logo - A SAFECHARGE COMPANY]

3.5.6 Responsive design so the same payment page will be presented correctly on different displays and devices according to the display size and features:


3.5.1 Additional payment pages are also available for supporting no CVV2 and no personal ID transaction terminals.

4. Transaction Landing Pages

4.1 Once the payment page details are completed, the transaction is then performed by CG Gateway, and the card holder is being redirected to the appropriate landing page: to the OK Page if the transaction was successful, and to the NOT OK Page if an error or decline was received within the transaction process.
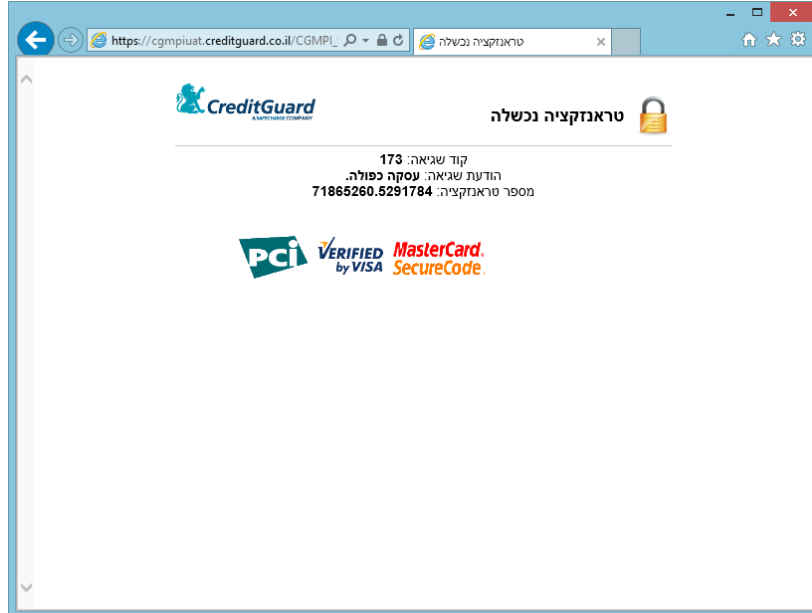
4.2 OK Page

4.2.1 OK Page example:
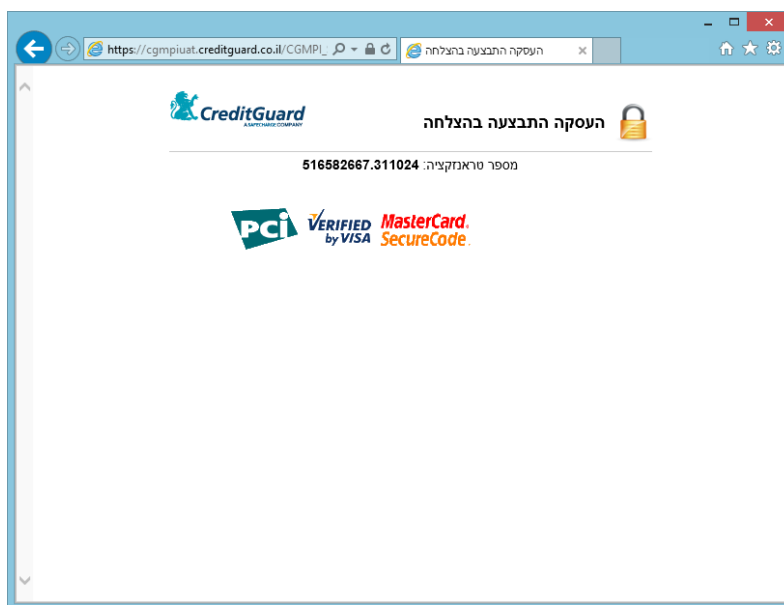
4.2.2 OK Page URL Parameters (URL request string variables):

Field Name	Value	Description
uniqueID	Unique transaction ID	The unique transaction ID as merchant gave on transaction creation
lang	EN|HE	Payment GUI's language
cardToken	Card Token	Credit Card Token (when enabled)
cardExp	Card Expiration	Card Expiration
personalId	Payer ID	Payer personal ID
cardMask	Card mask	Payer credit card number (masked)
txId	Transaction ID	Given by Credit Guard
authNumber	Authorization Number	Given by the acquiring Bank
numberOfPayments	Number of payments	The number of payments per the transaction
firstPayment	First payment amount	The amount of the first payment.
periodicalPayment		The amount of each periodical payment.
responseMAC	Calculated MAC	Verification MAC for ensuring transaction response validity

4.3 NOT OK Page

4.3.1 Alternatively, you might be redirected to the NOTOK Page.
NOTOK Page Example:

4.3.2 NOT OK Page URL Parameters (URL request string variables):

Field Name	Value	Description
uniqueID	Unique transaction ID	The unique transaction ID as merchant gave on transaction creation
lang	EN|HE	Payment GUI's language
authNumber	Authorization Number	Given by the acquirer
cardToken	Card Token	Credit Card Token (when enabled)
cardExp	Card Expiration	Card Expiration
personalId	Payer ID	Payer personal ID
cardMask	Card mask	Payer credit card number (masked)
txId	Transaction ID	Given by Credit Guard
numberOfPayments	Number of payments	The number of payments per the transaction
firstPayment	First payment amount	The amount of the first payment.
periodicalPayment		The amount of each periodical payment.
errorCode	Error Code	Numeric error code
errorText	Error Text	Error Description
responseMAC	Calculated MAC	Verification MAC for ensuring transaction response validity

5. Using Dynamic Landing Page URLs

5.1 Usually landing pages are statically configured on the CG Gateway configuration level, yet CG Gateway and CG Redirect offer the flexibility of dynamic landing page URLs.

5.2 This is achieved by setting the success, error and cancel URLs on the transaction request level, overriding the setup configuration values.

5.3 This is done by using the following transaction setup request tags:

5.3.1 **successUrl** – Defines the OK page URL. End user will be redirected to this page in case of successful transaction.

5.3.2 **errorUrl** – Defines the NOTOK page URL. End user will be redirected to this page in case of transaction failure.

5.3.3 **cancelUrl** – Defines the CANCEL page URL. End user will be redirected to this page if he chooses to cancel the transaction.

5.4 These fields can be set with a legal URL only.

5.4.1 The URL must begin with a http:// or https://

5.4.2 The URL cannot contain Hebrew characters.

5.4.3 The URL can have additional parameters in its end. In that case each ampersand sign <&> must be replaced by the string <&>. Ampersand sign is used separate parameters in URLs.

5.4.4 Please note A URL with parameters must end with <&6amp;>

5.5 Using Dynamic URLs - the following are examples for URLs of various types a merchant can send within the transaction setup request, that will override the pre-defined URLs on the gateway setup :

```
<successUrl>https://www.mymerchant.com/OKPage.jsp?aa=bb&amp;</successUrl>
<errorUrl>https://127.0.0.1/NOTOKPage.jsp</errorUrl>
<cancelUrl>http://www.mymerchant.com/CANCELPage?</cancelUrl>
```

6. Transaction Validation

6.1 Now, once the card holder is redirected back to the merchant site, the merchant system must validate the transaction response, to eliminate interference within the process and to ensure the received response was indeed received from CG Gateway.

6.2 The above can be performed using two different options:

6.2.1 Perform additional CG Gateway transaction Query – query CG Gateway for the transaction outcome.

6.2.2 Perform response MAC calculation – hence ensure the retrieved response indeed arrived from CG Gateway.

6.3 The following sections will detail how to perform just that.

7. Transaction Query

7.1 An additional feature CG Gateway offers is querying the gateway for the full transaction details.

7.2 The can be optionally implemented by the merchant.

7.3 Of course, the query is using the same CG Gateway HTTPS Post or Web service mechanism, using the following XML to perform the transaction.

7.4 The query result include the transaction details that was approved by the end user/buyer.

7.5 Note that the query result will end with a general error (*transaction ID was not found*) as long as the transaction hasn't completed (or in other words – as long as the buyer did not approve the transaction in the payment page).

7.6 **XML Request**

```
<ashrait>
    <request>
            <requestId>723232323</requestId>
            <version>1000</version>
            <language>ENG</language>
            <dateTime/>
            <command>inquireTransactions</command>
            <inquireTransactions>
                    <terminalNumber>0962922</terminalNumber>
                    <mainTerminalNumber/>
                    <queryName>mpiTransaction</queryName>
                    <mid>123456</mid>
                    <mpiTransactionId>
                        459c35c1-80d1-4fa7-b6ac-60c33987b958
                    </mpiTransactionId>
                    <userData1></userData1>
                    <userData2></userData2>
                    <userData3></userData3>
                    <userData4></userData4>
                    <userData5></userData5>
            </inquireTransactions>
    </request>
</ashrait>
```

7.7 Request Tags

7.7.1 The following are the necessary tags for querying the system:

Tag Name	Type	Value	Value Mandatory	Description
terminalNumber	AlphaNumeric		Yes	Terminal number as defined for the merchant at CG Gateway
queryName	AlphaNumeric (32)	mpiTransaction	Yes	Indication of MPI query
mpiTransactionId	AlphaNumeric (32)		Yes	The token received from CG Gateway as the transaction identifier, used as part of the payment page URL
mid	String (15)	Merchant ID	Yes	Allocated by CG Gateway for the merchant – identifies the MPI page

7.8 XML Response

```xml
<ashrait>
    <response>
            <command>inquireTransactions</command>
            <dateTime>2011-08-04 14:00</dateTime>
            <requestId>723232323</requestId>
            <tranId>5030</tranId>
            <result>000</result>
            <message>Permitted transaction.</message>
            <userMessage>Permitted transaction.</userMessage>
            <additionalInfo></additionalInfo>
            <version>1000</version>
            <language>Eng</language>
            <inquireTransactions>
                    <row>
                            <mpiTransactionId>
                              459c35c1-80d1-4fa7-b6ac-60c33987b958
                            </mpiTransactionId>
                            <uniqueid>
                              837683509-84983498539
                            </uniqueid>
                            <amount>10000</amount>
                            <currency>ILS</currency>
                            <authNumber>0123456</authNumber>
                            <cardId>1018812649454580</cardId>
                            <personalId>200553857</personalId>
                            <cardExpiration>1212</cardExpiration>
                            <languageCode>EN</languageCode>
                            <statusCode>0</statusCode>
                            <statusText>SUCCEEDED</statusText>
                            <errorCode>00</errorCode>
                            <errorText>SUCCESS</errorText>
                            <cgGatewayResponseCode>000</cgGatewayResponseCode>
                            <cgGatewayResponseText>
                               Permitted transaction.
                            </cgGatewayResponseText>
                            <cgGatewayResponseXML>
                                    <ashrait>
                                    <response>
                                    <command>doDeal</command>
                                    <dateTime>2011-08-04 12:10</dateTime>
                                    <requestId></requestId>
                                    <tranId>227079</tranId>
                                    <result>000</result>
                                    <message>Permitted transaction.</message>
                                    <userMessage>Permitted transaction.</userMessage>
                                    <additionalInfo></additionalInfo>
                                    <version>1000</version>
                                    <language>Eng</language>
```

CreditGuard Ltd. 18 Ben Gurion st., Givat Shmuel T: +972 3 7370700 F: +972 3 7370790

```xml
                                        <doDeal>
                                        <status>000</status>
                                        <statusText>Permitted transaction.</statusText>
                                        <terminalNumber>0962922</terminalNumber>
                                        <cardId>1018812649454580</cardId>
                                        <cardBin>458045</cardBin>
                                        <cardMask>458045******4580</cardMask>
                                        <cardLength>16</cardLength>
                                        <cardNo>xxxxxxxxxxxx4580</cardNo>
                                        <cardName></cardName>
                                        <cardExpiration>0114</cardExpiration>
                                        <cardType code="0">Local</cardType>
                                        <creditCompany code="11">Isracard</creditCompany>
                                        <cardBrand code="1">Mastercard</cardBrand>
                                        <cardAcquirer code="1">Isracard</cardAcquirer>
                                        <serviceCode>000</serviceCode>
                                        <transactionType
code="02">AuthDebit</transactionType>
                                        <creditType code="1">RegularCredit</creditType>
                                        <currency code="1">ILS</currency>
                                        <transactionCode code="50">Phone</transactionCode>
                                        <total>24000</total>
                                        <balance></balance>
                                        <starTotal>0</starTotal>
                                        <firstPayment></firstPayment>
                                        <periodicalPayment></periodicalPayment>
                                        <numberOfPayments></numberOfPayments>
                                        <clubId></clubId>
                                        <clubCode></clubCode>
                                        <validation code="5">AutoComm</validation>
                                        <commReason code="5">VerifyOnly</commReason>
                                        <idStatus code="1">Valid</idStatus>
                                        <cvvStatus code="1">Valid</cvvStatus>
                                        <authSource code="2">CreditCompany</authSource>
                                        <authNumber>4528125</authNumber>
                                        <fileNumber>13</fileNumber>
                                        <slaveTerminalNumber>001</slaveTerminalNumber>
                                        <slaveTerminalSequence>001</slaveTerminalSequence>
                                        <creditGroup></creditGroup>
                                        <pinKeyIn></pinKeyIn>
                                        <pfsc></pfsc>
                                        <eci></eci>
                                        <cavv code=" "></cavv>
                                        <user></user>
                                        <addonData></addonData>
                                        <supplierNumber>5032560</supplierNumber>
                                        </doDeal>
                                        </response>
                                        </ashrait>
                        </cgGatewayResponseXML>
                        <queryErrorText>SUCCESS</queryErrorText>
                        <xRem></xRem>
                </row>
                <totals>
                <pageNumber></pageNumber>
                <pagesAmount></pagesAmount>
                <queryResultId></queryResultId>
                <total> </total>
                <totalMatch></totalMatch>
                </totals>
        </inquireTransactions>
    </response>
</ashrait>
```

7.9 **Response Tags**

7.9.1 Please note the XML response include an external response (the **inquireTransactions** response as a transaction by itself) and internal data block, appearing under the **cgGatewayResponseXML** tag including all of the financial transaction response tags. When one try to extract the full financial transaction response and details – the internal block should be referred to.

7.9.2 Some of the most important values are also populated under dedicated tags of the external response block:

Tag Name	Type	Value	Description
uniqueid	AlphaNumeric		As sent in the original transaction setup request
cardId	AlphaNumeric (36)		The transaction's credit card token (Card Id)
personalId	AlphaNumeric (9)		The card holder personal ID if inserted in the payment page
expirationDate	AlphaNumeric (4)		The card expiration date in format MMYY
cgGatewayResponseCode	Number(4)		CG Gateway transaction code (as performed from the URL Page)
cgGatewayResponseText	AlphaNumeric		CG Gateway corresponding text message
cgGatewayResponseXML	AlphaNumeric		The entire XML response as returned from CG Gateway

7.9.1 For further implementation and manipulation (cancel or refund the financial transaction) it is advised the merchant will also store the transaction identifier provided by the gateway, that can be retrieved from the "tranId" tag value on the **cgGatewayResponseXML** block:

Tag Name under cgGatewayResponseXML	Type	Value	Description
tranId	Number (20)		CG Gateway internal unique transaction identifier

7.9.2 Full tags documentation can be obtained from "CG Gateway XML API" Document.

8. Response MAC Validation

8.1 In order to securely depend upon the returned values to the OK/NOT OK Pages, a response MAC is also returned.

8.2 The response MAC is basically a one way hash signature on the returned values, including a shared secret in the based hash string (which is the API user web service password).

8.3 In order to validate the returned values, the following returned values shpuld be concatenated and hashed in the exact following order:

8.3.1 API assigned password

8.3.2 txId (assigned transaction id)

8.3.3 errorCode (returned error code or "000" in case of a successful transaction)

8.3.4 cardToken (if received – an empty string otherwise)

8.3.5 cardExp (if received – an empty string otherwise)

8.3.6 personalId (if received – an empty string otherwise)

8.3.7 uniqueId (merchant unique identifier of the original request)

8.4 The above concatenated string should be hashed (using sha-256) and transformed to base64 form, and the retrieved value should be compared on the merchant side to a similar calculation performed by the merchant.

8.5 If the retrieved value and the calculated values are identical, the signature is hence validated.

8.6 **MAC Calculation example:**

    8.6.1 Lets assume the following values form the transaction details:

        8.6.1.1 API password id = "Password"

        8.6.1.2 txId = "973f89bf-d6d1-45e0-8d71-a3af77d9575e"

        8.6.1.3 errorCode is "000" (successful transaction)

        8.6.1.4 cardToken is "1030241464051111"

        8.6.1.5 cardExp is "1212"

        8.6.1.6 personalId is not retrieved hence the value is an empty string ""

        8.6.1.7 sent uniqueId is "uniqueid"

        8.6.1.8 The base string for MAC calculation is then:

```
Password973f89bf-d6d1-45e0-8d71-a3af77d9575e0001030241464051111112l2uniqueid
```

        8.6.1.9 The **SHA-256** function invoked on the above string and the result is encoded as **hexadecimal** string.

        8.6.1.10 The resulting MAC string is then:

```
2ee3a85b0db3665cd5d8ef3d0583fd3f3a58f4f9f49d3dcbb776048ed717c96d
```

9. Credit Card Tokenization

9.1 Credit Card Tokenization concept is basically providing the merchant with a card token (which is not the credit card number) the merchant can store and use for future transactions to CreditGuard's gateway instead of sending the real full credit card number.

9.2 When CreditGuard's tokenization is enabled for the merchant, when a Credit Card transaction is initially performed by CG Gateway, a token is generated for the credit card and returned on the transaction response under the "**cardId**" tag (returned on both the landing page URL and the transaction query response).

9.3 This **cardId** token is a unique identifier for this specific credit card and can be stored by the merchant (along with the card expiration date and potentially the card holder personal ID) for future use.

9.4 CreditGuard's cardId token format:

    9.4.1 Digits only string.

    9.4.2 16 digits long – no matter what was the original credit card length.

    9.4.3 The **cardId** last 4 digits are the original card number last 4 digits.

    9.4.4 The **cardId** number confirms with the LUHN algorithm for control digit.

9.5 **Please note** that the cardId replaces **ONLY** the credit card number for future transactions, expiration date and Israeli personal ID number when applicable should be always stored by the merchant as well and sent on consecutive transaction when necessary.

Adding Installments to the Payment Page (עסקת תשלומים)

1. Overview

1.1 When working is Israel, Payments or Installments transaction is an option the israeli acquirers offer.

1.2 If the terminal allow payments transaction, it is possible to utilize this and automatically present a choice for number of payments to the card holder on the redirect payment page.

1.3 This is done simply by specifying the value '**Payments**' on the '**creditType**' tag.

1.4 In addition the '**numberOfPayments**' tag should be included with a value of the following options:

    1.4.1 Stating the maximum number of payments for the card holder to choose from.
```
<numberOfPayments>12</numberOfPayments>
```
The number of payments dropdown list will include all from 1 to the maximum number of payments.

    1.4.2 State a list for the possible values of '**numberOfPayments**' where each list item can be a single number or a range of numbers.
```
<numberOfPayments>1-12 24 36</numberOfPayments>
```
this way the number of payments dropdown list will include values as specified by the list values.
For example the above list will create the following options in the dropdown list:
1,2,3,4,5,6,7,8,9,10,11,12,24,36

1.5 That exact functionality also applies when using '**SpecialCredit**' as the '**creditType**' tag value for the Israeli special credit supported transaction type.

1.6 The above tags ('**creditType', 'numberOfPayments**') are mandatory for supporting installments in the payment page.

1.7 An optional tag '**paymentsInterest**' can be included in the XML request
```
<paymentsInterest>0 250 650</paymentsInterest>
```

    1.7.1 This option adds to the total amount a specific interest that can change for each value of number of payments that is chosen in the dropdown list.

    1.7.2 The interest will be displayed in the payment page separately.

    1.7.3 The interest tag value is defined as a list of amounts and the number of list elements will be same as the number of list elements in '**numberOfPayments**'.

    1.7.4 The interest item value is the absolute interest amount in cents/agorot (100 ILS will be 10000).
Zero (0) value is considered as no interest.

1.8 Once this is done, the page will present a dropdown list of number of payments the card holder can use to complete the transaction.

1.9 The page will also automatically calculate and present the amount of the first and periodical payments, and if exist an interest will also automatically calculate the general amount of transaction.

1.10 Once the card holder choose the number of payments and perform the transaction, the merchant will be able to see this data either on the landing page redirect

parameters, or by performing a transaction query after the landing page was invoked.

2. Transaction Setup

2.1 For using this functionality, a transaction setup request should be specified as follows:

2.2 **XML Request**

2.2.1   First option:

```
<ashrait>
    <request>
        <version>1000</version>
        <language>HEB</language>
        <dateTime/>
        <command>doDeal</command>
        <doDeal>
            <terminalNumber>096XXXXX</terminalNumber>
            <mainTerminalNumber/>
            <cardNo>CGMPI</cardNo>
            <successUrl></successUrl>
            <errorUrl></errorUrl>
            <cancelUrl></cancelUrl>
            <total>5000</total>
            <transactionType>Debit</transactionType>
            <creditType>Payments</creditType>
            <currency>ILS</currency>
            <transactionCode>Phone</transactionCode>
            <authNumber/>
            <numberOfPayments>12</numberOfPayments>
            <firstPayment></firstPayment>
            <periodicalPayment></periodicalPayment>
            <validation>TxnSetup</validation>
            <dealerNumber/>
            <user></user>
            <mid>10763</mid>
            <uniqueid>130825221.407718</uniqueid>
            <mpiValidation>autoComm</mpiValidation>
            <email></email>
            <clientIP/>
            <customerData>
                <userData1/>
                <userData2/>
                <userData3/>
                <userData4/>
                <userData5/>
                <userData6/>
                <userData7/>
                <userData8/>
                <userData9/>
                <userData10/>
            </customerData>
        </doDeal>
    </request>
</ashrait>
```
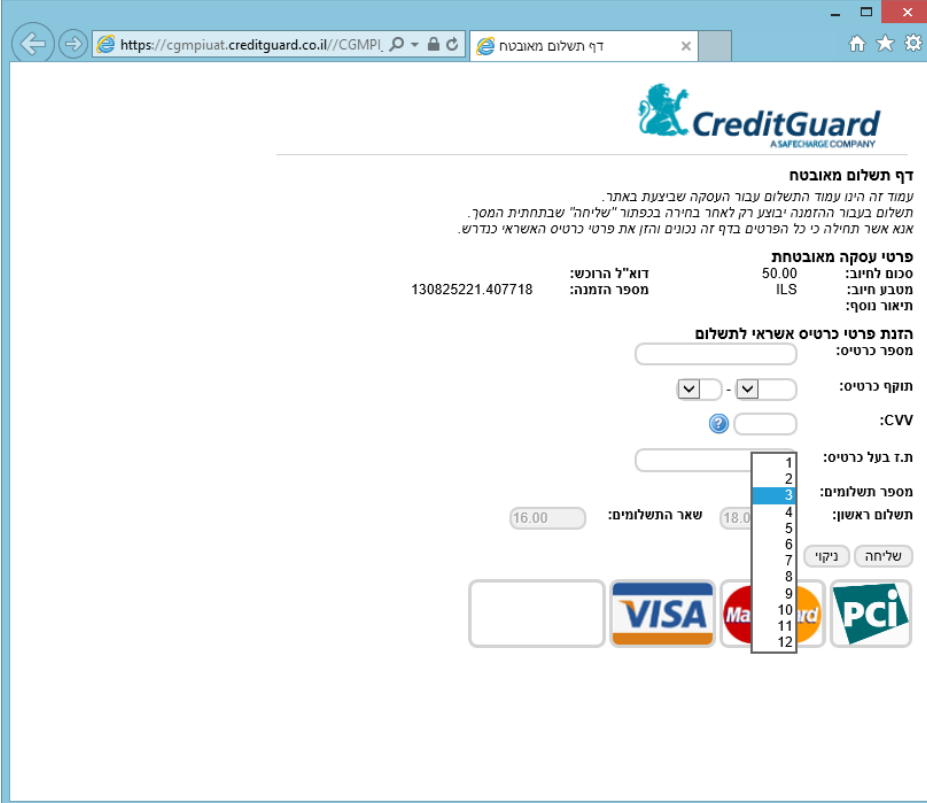
2.2.2   Second option

```
<ashrait>
    <request>
        <version>1000</version>
        <language>HEB</language>
        <dateTime/>
        <command>doDeal</command>
        <doDeal>
            <terminalNumber>096XXXXX</terminalNumber>
            <mainTerminalNumber/>
            <cardNo>CGMPI</cardNo>
```

CreditGuard Ltd. 18 Ben Gurion st., Givat Shmuel T: +972 3 7370700 F: +972 3 7370790

```xml
                    <successUrl></successUrl>
                    <errorUrl></errorUrl>
                    <cancelUrl></cancelUrl>
                    <total>5000</total>
                    <transactionType>Debit</transactionType>
                    <creditType>Payments</creditType>
                    <currency>ILS</currency>
                    <transactionCode>Phone</transactionCode>
                    <authNumber/>
                    <numberOfPayments>1-12 24 36</numberOfPayments>
                    <paymentsInterest>0 250 6500</paymentsInterest>
                    <firstPayment></firstPayment>
                    <periodicalPayment></periodicalPayment>
                    <validation>TxnSetup</validation>
                    <dealerNumber/>
                    <user></user>
                    <mid>10763</mid>
                    <uniqueid>130825221.407718</uniqueid>
                    <mpiValidation>autoComm</mpiValidatx`ion>
                    <email></email>
                    <clientIP/>
                    <customerData>
                        <userData1/>
                        <userData2/>
                        <userData3/>
                        <userData4/>
                        <userData5/>
                        <userData6/>
                        <userData7/>
                        <userData8/>
                        <userData9/>
                        <userData10/>
                    </customerData>
            </doDeal>
        </request>
</ashrait>
```

Note that if sent in 'numberOfPayments' tag only one number and not string, it will be the maximum number of payments like first option.

3. Hosted Payment Page

3.1   The generated payment page will now present a choice of number of installments (up to 12 installments as specified on the request).

3.2   One can also notice, that when the card holder choose to use 3 installments, the page automatically calculates the first and periodical payments.

3.3   One can also notice, that when the card holder choose to use 24 installments, the
        page automatically calculates the first and periodical and **interest** payments.

דף תשלום מאובטח

עמוד זה הינו עמוד התשלום עבור העסקה שביצעת באתר.
תשלום בעבור ההזמנה יבוצע רק לאחר בחירה בכפתור "שליחה" שבתחתית המסך.
אנא אשר תחילה כי כל הפרטים בדף זה נכונים והזן את פרטי כרטיס האשראי כנדרש.

פרטי עסקה מאובטחת

דוא"ל הרוכש:	סכום לחיוב:	82.50
מספר הזמנה: 734539210.061817	מטבע חיוב:	
	תיאור נוסף:	

הזנת פרטי כרטיס לתשלום

מספר כרטיס:	
תוקף כרטיס:	☑ - ☑
CVV:	❓
ת.ז בעל כרטיס:	
מספר תשלומים	
תשלום ראשון	שאר התשלומים 3.00
סכום הריבית	2.50
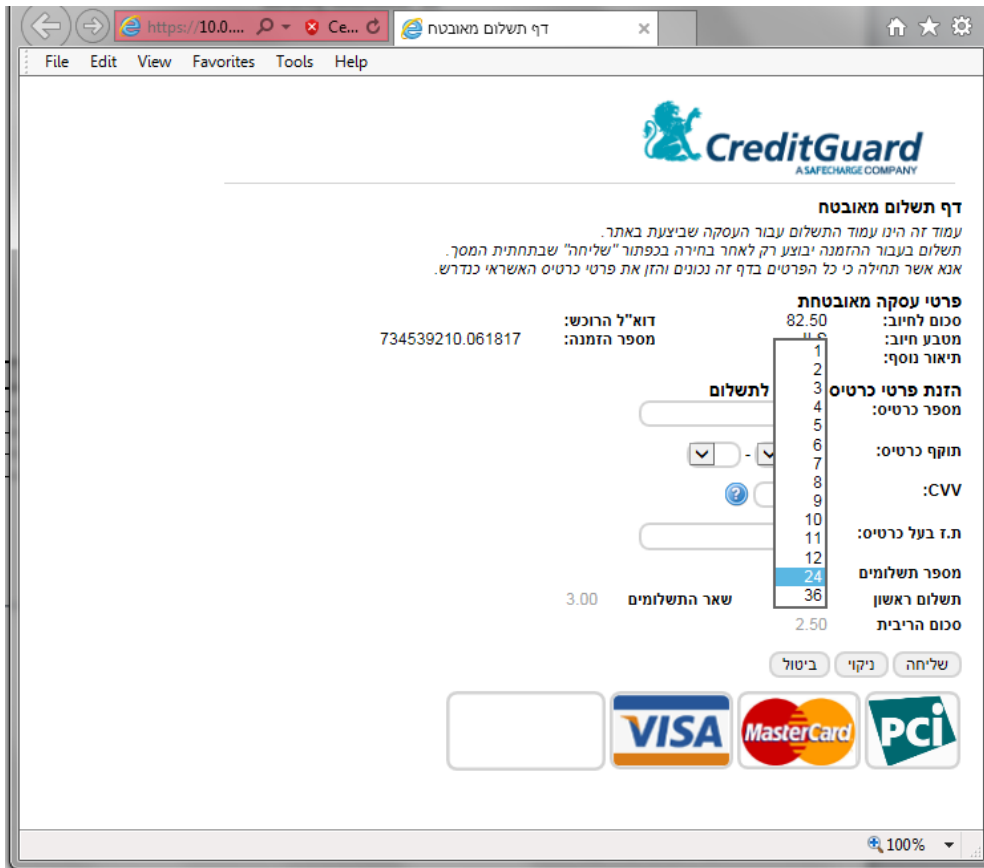
שליחה   ניקוי   ביטול

CreditGuard Ltd. 18 Ben Gurion st., Givat Shmuel T: +972 3 7370700 F: +972 3 7370790

3.4 Once redirected back to the merchant site, the number of payments used on the transaction now appear on the redirect URL (according to SHVA logic – the number of additional installments to the first installment) and can also be retrieved by performing the transaction query:

```
https://cgmpiuat.creditguard.co.il/CGMPI_Server/merchantPages/merchant/NOTOK.jsp?Error
Code=162&ErrorText= תקרה 0 לסוג כרטיס זה בעסקת
תשלומים.&uniqueID=130825221.407718&lang=HE&responseMac=9cd474e394a16812b0ad91a0e026ea7
6447c6b1d5cdfb1135d3cd1e8807d05f4&cardToken=1074946602134580&cardExp=0317&personalId=0
00000000&cardMask=458045******4580&txId=854fbfbd-22dc-4a9f-90e9-
88e90b5bd26f&numberOfPayments=2&firstPayment=1800&periodicalPayment=1600
```

Adding PayPal Option to the Payment Page

1. Overview

1.1 CG Gateway and CG Redirect also support using the PayPal express checkout payment method as an optional added service.

1.2 Please note that due to PayPal restriction, The PayPal option works **only if the payment page is opened as a full frame** and not within an embedded iFrame.

1.3 When a merchant wishes to add this capability, the merchant must first create a PayPal account and provide its credentials to CreditGuard. Once this is done, CreditGuard defines a PayPal terminal on CG Gateway (a terminal with the prefix of 'pp') and a logical terminal for the merchant (a terminal with the prefix of 'cg').

1.4 The PayPal terminal will be used to view all PayPal transactions performed by the merchant on CG Console, and the logical terminal will be used to define the routing logic per the merchant - credit card transactions will be routed to SHVA terminal, and PayPal transactions to the PayPal terminal according to the payment method the buyer/card holder choose on the payment page.

1.5 The PayPal service is optional and should be added to the merchant contract with CreditGuard prior to configuration and activation.

2. Transaction flow

2.1 Transaction flow diagram:
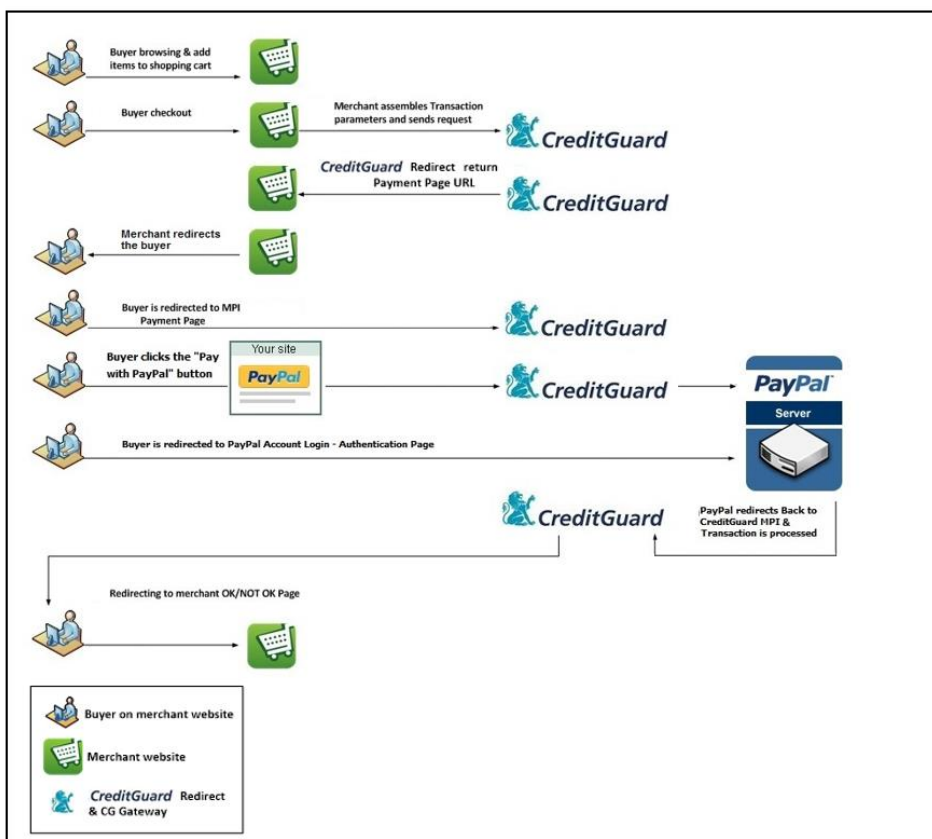
2.2   Flow description:

2.2.1   The merchant issue a transaction setup request to CG Gateway, now specifying the logical terminal as the transaction terminal, and providing requested additional acquirer data tags (required by PayPal).

2.2.2   A URL is retrieved, and the merchant redirects the buyer to the payment page.

2.2.3   The buyer will now see the option to check out with PayPal on the payment page, in addition to the option to pay with credit card.

2.2.4   If the buyer now choose to pay with PayPal, a second automatic redirect will be performed to PayPal page to complete the transaction.

2.2.5   Once the transaction is now completed, the buyer will be redirected back to the merchant site success or failure page.

2.2.6   Now it is advised that the merchant will perform the transaction query to CG Gateway, getting the full transaction details.

3. Transaction Setup

3.1 The following describes transaction setup XML for combined credit card and PayPal payment page **XML Request:**

```xml
<ashrait>
    <request>
        <version>1000</version>
        <language>HEB</language>
        <dateTime/>
        <requestId/>
        <command>doDeal</command>
        <doDeal>
            <terminalNumber>cg00XXXXXX</terminalNumber>
            <mainTerminalNumber />
            <cardNo>CGMPI</cardNo>
            <successUrl></successUrl>
            <errorUrl></errorUrl>
            <cancelUrl></cancelUrl>
            <total>10000</total>
            <transactionType>Debit</transactionType>
            <creditType>RegularCredit</creditType>
            <currency>ILS</currency>
            <transactionCode>Phone</transactionCode>
            <authNumber/>
            <validation>TxnSetup</validation>
            <dealerNumber/>
            <user>12345</user>
            <mid>999</mid>
            <uniqueid>969356714.709812</uniqueid>
            <mpiValidation>autoComm</mpiValidation>
            <email/>
            <clientIP/>
            <acquirerSelect>all</acquirerSelect>
            <customerData>
                <userData1/>
                <userData2/>
                <userData3/>
                <userData4/>
                <userData5/>
                <userData6/>
                <userData7/>
                <userData8/>
                <userData9/>
                <userData10/>
            </customerData>
            <acquirerData>
                <acquirer>
                    <acquirerId>pp</acquirerId>
                    <itemAmount>10000</itemAmount>
                    <taxAmount>0</taxAmount>
                    <shippingAmount>0</shippingAmount>
                    <handlingAmount>0</handlingAmount>
                    <shipDiscAmount>0</shipDiscAmount>
                    <insuranceAmount>0</insuranceAmount>
                    <itemsDesc/>
                    <item1>as as|143242|Item Full Description|10000|1</item1>
                    <item2></item2>
                    <item3></item3>
                    <item4></item4>
                    <item5></item5>
                    <item6></item6>
                    <item7></item7>
                    <item8></item8>
                    <item9></item9>
                    <item10></item10>
                </acquirer>
            </acquirerData>
        </doDeal>
    </request>
</ashrait>
```

3.2 Request Tags

3.2.1 This section will detail the additional and/or changed request tags used to add the PayPal option to a standard payment page transaction setup request.

Tag Name	Type	Value	Value Mandatory	Description
terminalNumber	AlphaNumeric	cgxxxxxxxx	Yes	Required only a logical terminal number, prefixed by 'cg' as defined for the merchant at CG Gateway
acquirerSelect	AlphaNumeric (3)	all | sh | pp	Yes	"all" – Both Credit Card and SHVA "sh" – present credit Card option only "pp" – present PayPal option Only

3.3 Request Acquirer Data Tags

3.3.1 As shown in the XML request above, in order to support the PayPal payment option, "**acquirerData**" tags and values are mandatory on the request.

3.3.2 Please note that the "acquirerData" tag contains an inner "acquirer" data block.

3.3.3 The following table contains detailed information on each tag in the "acquirer" XML Request block needed for supporting the PayPal payment method:

Tag Name	Type	Value	Value Mandatory	Description
acquirerId	AlphaNumeric(2)	pp	Yes	PayPal Acquirer Id indication
itemAmount	Numeric(8)		No	The price sum for all items (in cents or agorot).
taxAmount	Numeric(8)		No	The tax sum for all items (in cents)
shippingAmount	Numeric(8)		No	The total shipping cost (in cents)
handlingAmount	Numeric(8)		No	The Total Handling cost (in cents)
shipDiscAmount	Numeric(8)		No	The total shipping discount amount (in cents)
insuranceAmount	Numeric(8)		No	The total shipping insurance amount (in cents)
itemsDesc	AlphaNumeric (200)		No	The items description
Item1	AlphaNumeric	name1|number1|desc1|amount1|qty1	Yes	Name1...10 (alphanumeric) Item name
Item2	AlphaNumeric	Name2|number2|desc2|amount2|qty2	No	Number1...10 (numeric) Item number (internal merchant item identification)
Item3	AlphaNumeric	Name3|number3|desc3|amount3|qty3	No	Desc1...10 (alphanumeric) Item description
Item4	AlphaNumeric	Name4|number4|desc4|amount4|qty4	No	Amount1...10 (numeric) Item prices in cents
Item5	AlphaNumeric	Name5|number5|desc5|amount5|qty5	No	Qty1...10 (numeric) Item quantity
Item6	AlphaNumeric	Name6|number6|desc6|amount6|qty6	No	
Item7	AlphaNumeric	Name7|number7|desc7|amount7|qty7	No	
Item8	AlphaNumeric	Name8|number8|desc8|amount8|qty8	No	
Item9	AlphaNumeric	Name9|number9|desc9|amount9|qty2	No	
Item10	AlphaNumeric	Name10|number10|desc10|amount10|qty10	No	

3.3.4 **Please note the following:**
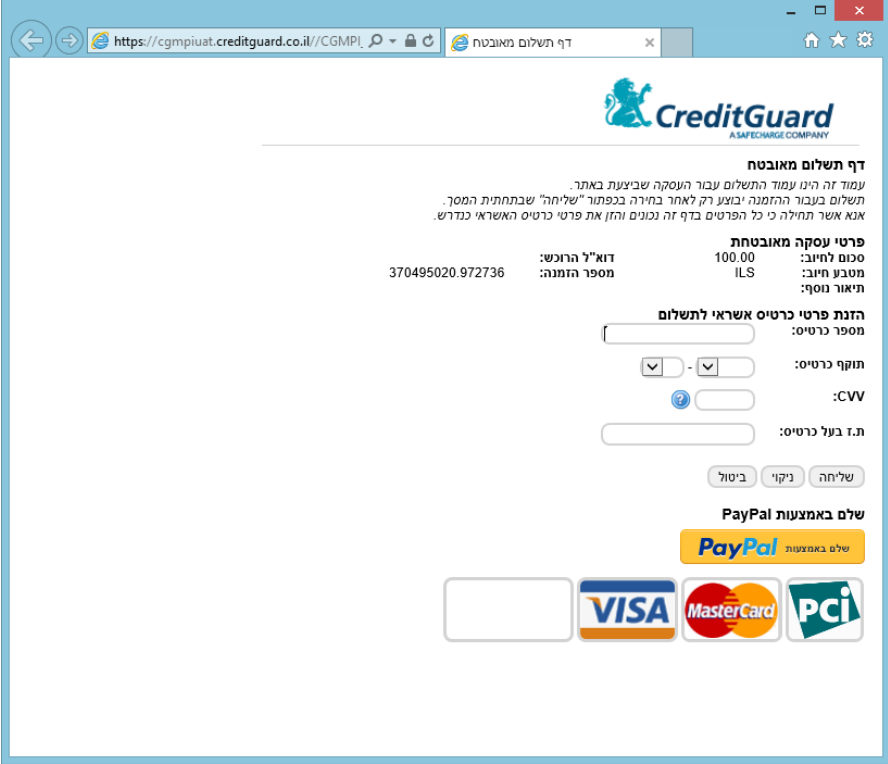
3.3.4.1 The first item ("item1" tag) in the request is mandatory. This will be presented at PayPal's page when the buyer will login for approving the transaction.

3.3.4.2 The other items ("item2" tag till "item10" tag) are optional item tags. These item tags allow the merchant to specify up to 10 different items that will be presented at PayPal's page.

3.3.4.3 The items tag value ("item1" tag value till "item10" tag value) contains the name, number, description, amount and quantity of the item with a pipe ("|" symbol) as a separator.

3.3.4.4 The nonempty values of the tags: "itemAmount", "taxAmount", "shippingAmount", "handlingAmount", "shipDiscAmount" and "insuranceAmount" are checked for validity with the transaction request XML "total" tag value (as described in section above) in the following manner:

itemAmount + taxAmount + shippingAmount + handlingAmount + insuranceAmount – shipDiscAmount = total

3.4 **XML Response**

3.4.1 The XML Response is a standard transaction setup response.

3.4.2 The merchant should retrieve the URL from the tag "**mpiHostedPageUrl**" and redirect the buyer to that payment Page.

4. Hosted Payment Page

4.1   The following is a CG Redirect payment page with PayPal checkout option:

4.2   The following describes the PayPal page presented for the buyer when within the purchase process, the buyer clicks the "Checkout with PayPal" link:

Adding MasterPass Option to the Payment Page

1. Overview

1.1 CG Gateway and CG Redirect also support using the MasterPass digital wallet checkout as an optional added service.

1.2 When a merchant wishes to add this capability, the merchant must first contact CreditGuard support in order to create a merchant account with MasterPass. Once this is done, CreditGuard defines a specific terminal for all MasterPass transactions on CG Gateway and an additional logical terminal for the merchant (a terminal with the prefix of 'cg').

1.3 The MasterPass terminal will be used to populate & view (on the CG Console) all transactions that were originated with MasterPass digital wallet, and the logical terminal will be used to define the routing logic per the merchant - credit card transactions that were accepted by the payment page 'form' will be routed to a different terminal then those transactions that were originated with MasterPass digital wallet. All according to the payment method the buyer/card holder choose on the payment page.

1.4 The MasterPass service is optional and should be added to the merchant contract with CreditGuard prior to configuration and activation.

2. Transaction Setup

2.1 The following describes transaction setup XML for combined credit card and MasterPass payment page **XML Request:**

```
<ashrait>
    <request>
        <version>1000</version>
        <language>HEB</language>
        <dateTime />
        <requestId />
        <command>doDeal</command>
        <doDeal>
            <terminalNumber>cg00962832</terminalNumber>
            <cardNo>CGMPI</cardNo>
            <total>15000</total>
            <transactionType>Debit</transactionType>
            <creditType>RegularCredit</creditType>
            <currency>ILS</currency>
            <transactionCode>Phone</transactionCode>
            <authNumber></authNumber>
            <validation>TxnSetup</validation>
            <mid>9</mid>
            <uniqueid>498150050.141956</uniqueid>
            <mpiValidation>autocomm</mpiValidation>
        </doDeal>
    </request>
</ashrait>
```

2.2 Request Tags

2.2.1 This section will detail the additional and/or changed request tags used to add the MasterPass option to a standard payment page transaction setup request.

| Tag Name	Type	Value	Value Mandatory	Description
terminalNumber	AlphaNumeric	**cgxxxxxxxx**	Yes	Required only a logical terminal number, prefixed by 'cg' as defined for the merchant at CG Gateway

2.3 XML Response

2.3.1 The XML Response is a standard transaction setup response.

2.3.2 The merchant should retrieve the URL from the tag "**mpiHostedPageUrl**" and redirect the buyer to that payment Page.

3. Hosted Payment Page

3.1 The following is a CG Redirect payment page with MasterPass checkout option:

CreditGuard Ltd. 18 Ben Gurion st., Givat Shmuel T: +972 3 7370700 F: +972 3 7370790
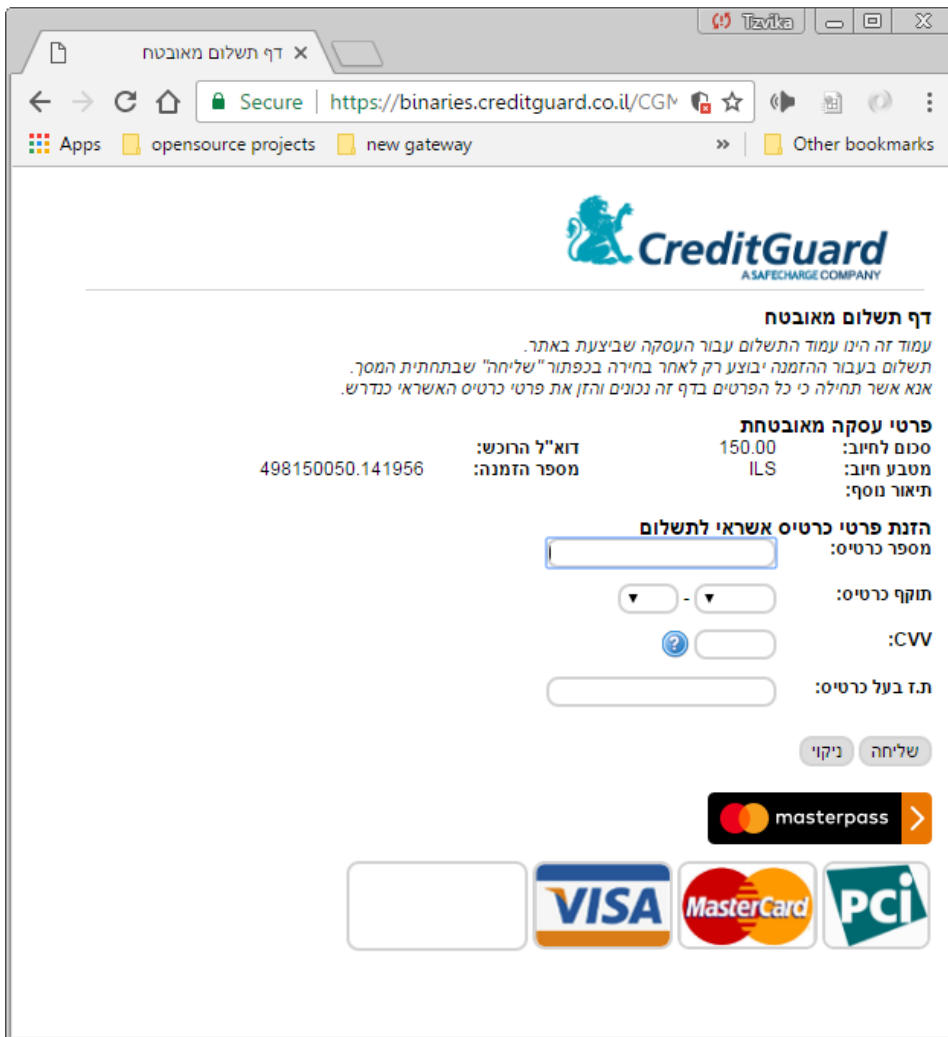
3.2 The following describes the MasterPass page presented for the buyer when within the purchase process, the buyer clicks the "masterpass" link:
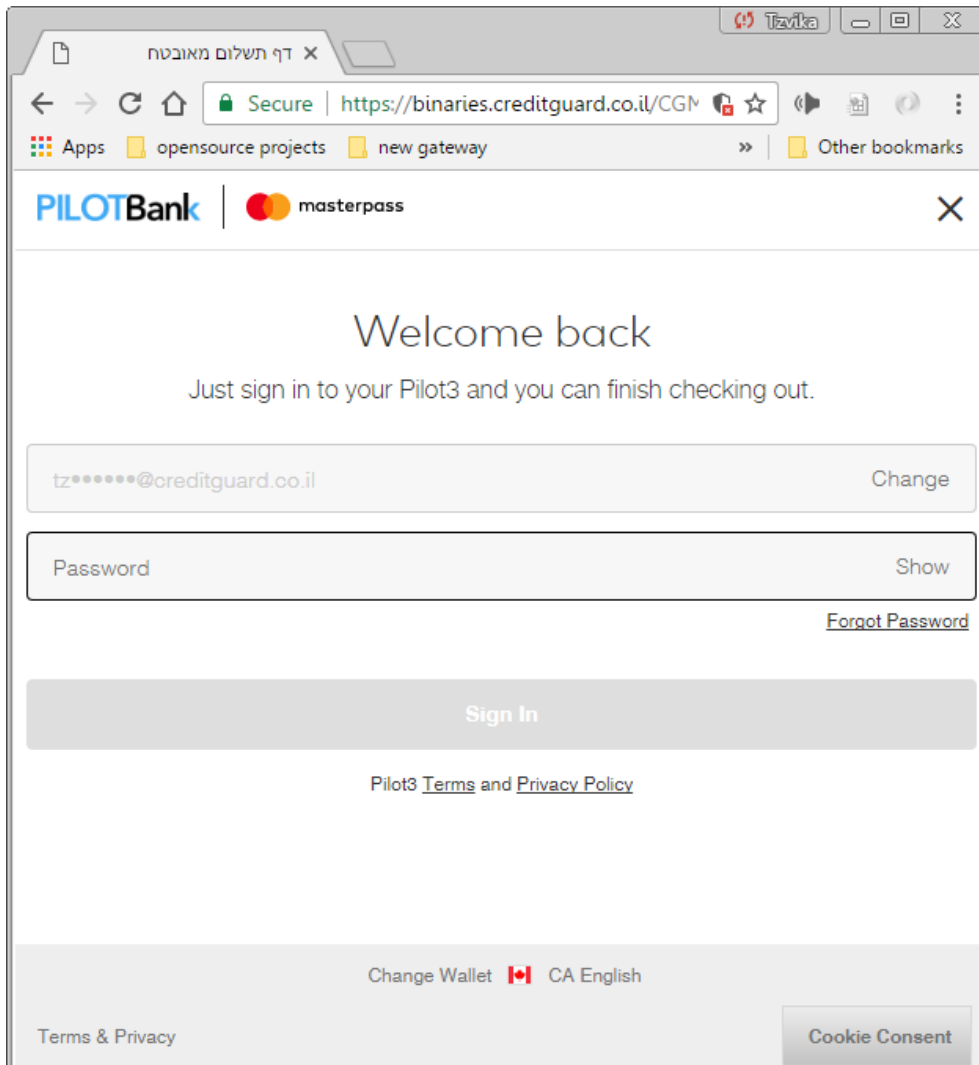
Adding Automatic Invoice Creation to Payment Page Transactions

1. Overview

1.1 CG Gateway and CG Redirect support automatic customer invoice creation as part of the transaction processing.

1.2 This is an additional service that require configuration on the merchant level on CG Gateway.

1.3 Once this option is enabled, the merchant can add additional invoice data block to the transaction setup request, and when a sale transaction will be successfully completed, an automatic invoice generation will occur, optionally sending the invoice as via e-mail to the provided customer email.

1.4 Invoice format and invoice type ("קבלה", "חשבונית מס קבלה" etc') are set upon the invoice account creation on CG Gateway and Invoice Provider.

1.5 The full invoice API include many options, yet we will detail the most common manner of invoice creation within this document. For further details refer to "Invoice integration XML API" documentation.

1.6 Please note, that invoice creation failure will not fail the financial transaction. A credit card transaction will be performed even if an invoice creation failed – yet a separate invoice creation status will be reported on the transaction response.

1.7 The invoices service is optional and should be added to the merchant contract with CreditGuard prior to configuration and activation.

2. Transaction Setup

2.1 **XML Request:**

```xml
<ashrait>
    <request>
        <version>1000</version>
        <language>HEB</language>
        <dateTime/>
        <requestId>444123</requestId>
        <mayBeDuplicate></mayBeDuplicate>
        <command>doDeal</command>
        <doDeal>
            <terminalNumber>096xxxx</terminalNumber>
            <mainTerminalNumber></mainTerminalNumber>
            <cardNo>CGMPI</cardNo>
            <total>2900</total>
            <transactionType>Debit</transactionType>
            <creditType>Payments</creditType>
            <currency>ILS</currency>
            <transactionCode>Phone</transactionCode>
            <authNumber/>
            <numberOfPayments></numberOfPayments>
            <firstPayment></firstPayment>
            <periodicalPayment></periodicalPayment>
            <validation>TxnSetup</validation>
            <dealerNumber></dealerNumber>
            <user>12345</user>
            <invoice>
                <invoiceCreationMethod>wait</invoiceCreationMethod>
                <invoiceDate>2015-12-15</invoiceDate>
                <invoiceSubject>נושא חשבונית לשירות טלפ'קה</invoiceSubject>
                <invoiceDiscount/>
                <invoiceDiscountRate/>
```

```
                    <invoiceItemCode>602|888</invoiceItemCode>
                    <invoiceItemDescription>מוצר ראשון|מוצר שני</invoiceItemDescription>
                    <invoiceItemQuantity>1|1</invoiceItemQuantity>
                    <invoiceItemPrice>2300|600</invoiceItemPrice>
                    <invoiceTaxRate/>
                    <invoiceComments>הערה לחשבונית</invoiceComments>
                    <companyInfo>פרטי חברה</companyInfo>
                    <mailTo>israelisraeli@creditguard.co.il</mailTo>
                    <isItemPriceWithTax>1</isItemPriceWithTax>
                    <ccDate>2015-12-15</ccDate>
                </invoice>
                <successUrl></successUrl>
                <mid>10589</mid>
                <uniqueid>358364283.680179</uniqueid>
                <mpiValidation>autoComm</mpiValidation>
                <email></email>
                <clientIP/>
                <customerData>
                    <userData1/>
                    <userData2/>
                    <userData3/>
                    <userData4/>
                    <userData5/>
                    <userData6/>
                    <userData7/>
                    <userData8/>
                    <userData9/>
                    <userData10/>
                </customerData>
            </doDeal>
        </request>
</ashrait>
```

2.2 **Request Tags**

2.2.1 This section will detail the additional request tags used to initiate invoice creation process to the transaction completion.

2.2.2 Please note the "invoice" data block of the transaction setup.

2.2.3 All related invoice creation attributes reside under this section.

Tag Name	Type	Value	Value Mandatory	Description
invoiceCreationMethod	post|wait	wait	Yes	post – a-synchronous invoice creation with immediate response. wait – synchroneous invoice creation.
invoiceDate	Date	YYYY-MM-DD	No	Invoice Date – Default is current date
invoiceSubject	Alpha Numeric and Space (255)		Yes	Invoice subject
invoiceDiscount	Numeric		No	Invoice discount in agorot/cents – default is 0
invoiceDiscountRate	Percentage XXX.XX		No	Invoice discount percent – default is 0.00
invoiceItemCode	Alpha Numeric and Space (500)		Yes	Item Code (makat)
invoiceItemDescription	Alpha Numeric and Space (500)		Yes	Item description
invoiceItemQuantity	Numeric XXX.XX		Yes	Item quantity
invoiceItemPrice	Numeric (500)		Yes	Item price in agorot/cents
invoiceTaxRate	Percentage XXX.XX		No	Item tax rate (percentage)
invoiceComments	Alpha Numeric and Space (255)		No	Invoice comments
companyInfo	Alpha Numeric and Space (255)		No	Customer full details
mailTo	Char (80)	x@x.x	No	Mailing address for automatic invoice sending
isItemPriceWithTax	Numeric	0|1	Yes	Indication whether the item price include tax

Tag Name	Type	Value	Value Mandatory	Description
ccDate	Date	YYYY-MM-DD	Yes	Payment date

3. Additional Invoice Creation Logic

3.1 Multiple Invoice Item Support:

3.1.1 Creating an invoice with multiple items is possible provided that the following four fields maintain the same item count separated by the pipe symbol ('|"):

3.1.1.1 invoiceItemCode

3.1.1.2 invoiceItemDescription

3.1.1.3 invoiceItemQuantity

3.1.1.4 invoiceItemPrice

For example invoice with two items will include the following tags:

```
<invoiceItemCode>11010|551</invoiceItemCode>
<invoiceItemDescription>שם פריט 1|שם פריט 2<invoiceItemDescription>
<invoiceItemQuantity>3|2<invoiceItemQuantity>
<invoiceItemPrice>1500|2000</invoiceItemPrice>
```

3.2 Amount Total Validation:

3.2.1 In order to produce a valid invoice amount validation is performed.

3.2.2 If amount validation fails the invoice creation will fail.

3.2.3 Method of calculation:

```
 invoiceItemPrice* invoiceItemQuantity = total
```

For multiple items:

```
(invoiceItemPrice * invoiceItemQuantity) + (invoiceItemPrice *
invoiceItemQuantity) = total
```

3.3 Discount Management:

3.3.1 When using a discount in the invoice creation Only one of the two following variables may be set:

3.3.1.1 invoiceDiscount

3.3.1.2 invoiceDiscountRate

3.3.2 Method of calculation:

```
invoiceItemPrice * invoiceItemQuantity - invoiceDiscount = total
invoiceItemPrice * invoiceItemQuantity * (100 – invoiceDiscountRate) / 100 = total
```

Example For two items:

```
total = 99 (Item price and invoice discount in agorot)

   <invoiceDiscount>1100<invoiceDiscount>
   <invoiceItemQuantity>2|2<invoiceItemQuantity>
   <invoiceItemPrice>4000|1500</invoiceItemPrice>

        Or:

   <invoiceDiscountRate>10.00<invoiceDiscountRate>
   <invoiceItemQuantity>2|2<invoiceItemQuantity>
   <invoiceItemPrice>4000|1500</invoiceItemPrice>
```

3.4 Using specific tax calculation:

3.4.1 This is possible by using with a single variable: isItemPriceWithTax.

3.4.2 The indication affects the amounts fields: invoiceItemPrice, invoiceDiscount.

3.4.3 May be set isItemPriceWithTax = 0 - the amount without tax

or isItemPriceWithTax = 1 – the amount with tax.

3.4.4 Method of to calculation:

```
(100 + tax) / 100 = amount with tax

invoiceItemPrice For example: invoiceItemPrice with tax = 50, tax = 18%:
50 / 1.18 = 42.37 = invoiceItemPrice without tax

invoiceDiscount for example: invoiceDiscount with tax = 22 , tax = 18%:
118/100 = 22 -> 22 / 1.18 = 18.64 = invoiceDiscount without tax
```

4. Generated Invoice

4.1 When using the specified XML request, a created invoice will look as the following:

Asynchronous Callback Report

1. Overview

1.1 At the end of the transaction, CG Redirect can be configured to issue automatic & asynchronous HTTPS Post report back to a predefined URL on merchant server. This is used to allow for tracking capabilities by merchant.

1.2 The callback report service is optional and should be added to the merchant contract with CreditGuard prior to configuration and activation by CreditGuard.

1.3 The callback report populates all the information about the transaction and its modules (i.e. userData information or invoice information…).

1.4 The callback report also populates information specifying the status according to the CG Redirect flow.

2. Configuring callback reports

2.1 Contact your account manager to start receiving asynchronous callback reports. Alternatively, contact CreditGuard Support Team.

2.2 In your request please supply a predefined URL on your merchant's server which receives and process the callback reports.

3. Integration

3.1 You need to set up a server to receive and accept the notifications we send you only through HTTP POST interface. Your system needs to be able to handle requests and responses containing additional fields, as well as duplicate notifications for the same transaction.

4. Report structure

4.1 CreditGuard reserves the right to introduce new fields in the future.
Therefore, make sure your listening service does not expect a fixed, predefined set of values.

4.2 The report includes the following fields:

Name	Name in POST	Type & format	Description
Unique Order ID	merchantUniqueOrderId	String (64) The merchantUniqueOrderId should be in the values range of 0-9, a-z, A-Z and special characters like: _-.#$: and space. "OrderId" length should be between 1-64.	Unique ID provided by the merchant in the request
MPI Transaction Code	mpiTransactionId	String(36) The mpiTransactionId should be in the values range of 0-9, a-z, A-Z and special character "-". i.e.: 81a133ea-afb7-45f0-a376-0d9f97700d03	Transaction ID generated by CG Redirect and returned in the response
Amount	amount	Numeric (8)	Amount of transaction in cents/agurot
Currency	currency	String (3)	Currency
Language Code	languageCode	String (2)	3 letter code
CG GW response code	cgGatewayResponseCode	Numeric(3)	response code as received from CG GW
CG GW response text	cgGatewayResponseText	String (256)	The corresponding response text as received from CG GW
CardID	creditCardToken	Numeric(16)	CreditID as received from CG GW

CreditGuard Ltd. 18 Ben Gurion st., Givat Shmuel T: +972 3 7370700 F: +972 3 7370790

Authorization Number	authNumber	String(7)	authNumber as received from CG GW
Status Code	statusCode	Numeric (3)	Reflects the transaction request status according to the CG Redirect flow, when: • Value 0 – the deal ended successfully (the user got success message). • Value 1 – the deal has stored successfully after the "Transaction Setup" request ended successfully. • Value 2 – the user was redirected to the "payment page". • Value 3 – the deal were sent to CG Gateway (the user entered his credit card details and authorized the deal) • Value 4 – the request processing has failed during one of the above stages.
Status Text	statusText	String (256)	CG Redirect corresponding text for Status Code field. Reflects the transaction request status according to the CG Redirect flow, when: • SUCCEEDED – the deal ended successfully (the user got success message). • INITIALIZED– the deal has stored successfully after the "Transaction Setup" request ended successfully. • DISPLAYED_TO_USER– the user was redirected to the "payment page". • SENT_TO_GATEWAY– the deal were sent to CG Gateway (the user entered his credit card details and authorized the deal) • FAILED– the request processing has failed during one of the above stages.
Error Code	errorCode	Numeric (3)	CG Redirect error code (if exist)
Error Text	errorText	String (256)	Corresponding CG Redirect error text (if exist)
Gateway Response	cgGatewayResponseXML	String(2000) The field value is in XML format	The full CG response, NOT including card details
X Field	xRem	String(19)	The X field, if sent, from the request, or empty String
personalId	personalId	Numeric (11)	Customer's social number (ID)
CreditCard expiration date	cardExpiration	Numeric (4)	Credit Card expiration date

4.3   Fields will be URL encoded, all UTF-8 strings (application/x-www-form-urlencoded)

4.4 Example:

```
amount = 9000
personalId = 000000000
errorCode = 00
cgGatewayResponseXML = <?xml
version='1.0'?><ashrait><response><command>doDeal</command><dateTime>2016-12-16
09:53</dateTime><requestId></requestId><tranId>6136085</tranId><result>000</result><me
ssage>Permitted transaction.</message><userMessage>Permitted
transaction.</userMessage><additionalInfo></additionalInfo><version>1000</version><lan
guage>Eng</language><doDeal><status>000</status><statusText>Permitted
transaction.</statusText><terminalNumber>0962832</terminalNumber><cardId>1012163979564
580</cardId><cardBin>458045</cardBin><cardMask>458045******4580</cardMask><cardLength>
16</cardLength><cardNo>xxxxxxxxxxxx4580</cardNo><cardName></cardName><cardExpiration>0
318</cardExpiration><cardType
code="1">Foreign</cardType><extendedCardType>Credit</extendedCardType><lifeStyle></lif
eStyle><customCardType></customCardType><creditCompany
code="23">Visa</creditCompany><cardBrand code="2">Visa</cardBrand><cardAcquirer
code="6">Alphacard</cardAcquirer><serviceCode>000</serviceCode><transactionType
code="02">AuthDebit</transactionType><creditType
code="1">RegularCredit</creditType><currency code="1">ILS</currency><transactionCode
code="50">Phone</transactionCode><total>9000</total><balance></balance><starTotal>0</s
tarTotal><firstPayment></firstPayment><periodicalPayment></periodicalPayment><numberOf
Payments></numberOfPayments><clubId></clubId><clubCode>0</clubCode><validation
code="4">AutoComm</validation><commReason code=""></commReason><idStatus
code="3">NotValidated</idStatus><cvvStatus
code="3">NotValidated</cvvStatus><authSource
code="3">VoiceMail</authSource><authNumber>0584653</authNumber><fileNumber>68</fileNum
ber><slaveTerminalNumber>287</slaveTerminalNumber><slaveTerminalSequence>525</slaveTer
minalSequence><creditGroup></creditGroup><pinKeyIn>0</pinKeyIn><pfsc>0</pfsc><eci>0</e
ci><cavv code="
"></cavv><user></user><addonData></addonData><supplierNumber>0225821</supplierNumber><
id>000000000</id><shiftId1></shiftId1><shiftId2></shiftId2><shiftId3></shiftId3><shift
TxnDate></shiftTxnDate><authAmount></authAmount><customerData><userData1>0543334455</u
serData1><userData2>john.doe@gmail.com</userData2><userData3>metadata3</userData3></cu
stomerData></doDeal></response></ashrait>
languageCode = EN
errorText = SUCCESS
creditCardToken = 1012163979564580
merchantUniqueOrderId = 200176417.62743
cgGatewayResponseText = Permitted transaction.
xRem =
statusText = SUCCEEDED
authNumber = 0584653
currency = ILS
mpiTransactionId = d92af620-354f-4d97-bebf-9b9b327a3848
cardExpiration = 0318
cgGatewayResponseCode = 000
statusCode = 0
```

CreditGuard Ltd. 18 Ben Gurion st., Givat Shmuel T: +972 3 7370700 F: +972 3 7370790

Payment Page Error Codes

1. Error Codes

1.1   The payment page error codes are reported by the interfaces as documented above:

    1.1.1   Transaction Setup

    1.1.2   Transaction Landing Page

    1.1.3   Transaction Query

1.2   **Transaction Setup Error Codes**

    1.2.1   Error codes are populated in the "status" XML field.

    1.2.2   Error messages are populated in the "message" XML field.

    1.2.3   User message & additional information are populated in the "userMessage" and "additionalInfo" XML fields.

    1.2.4   The following is a list of dedicated payment page error codes as reported by the "Transaction Setup" interface.

1.3   **Transaction Landing Page**

    1.3.1   Error codes are populated in the "ErrorCode" field.

    1.3.2   Error messages are populated in the "ErrorText" field.

    1.3.3   The following is a list of error codes as reported by the "Transaction Landing Page" interface.

1.4   **Transaction Query**

    1.4.1   Payment Page related error codes are populated in the "ErrorCode" field.

    1.4.2   Payment Page related error messages are populated in the "ErrorText" field.

    1.4.3   Actual payment transaction related error codes are populated in the "cgGatewayResponseCode" field.

    1.4.4   Actual payment transaction related error messages are populated in the "cgGatewayResponseText" field.

1.5   **List of Redirect payment page error codes**

    1.5.1   Please note that a full list of CG Gateway error codes can be found on "CG Gateway XML API" documentation.

Message	Error Code
Permitted transaction.	000
MPI Authentication Failed	681
MPI Transaction validation failed	682
MPI Invalid request or post parameters	683
MPI CG Gateway returned an error	684
MPI Error accessing the CG Gateway	685
MPI Internal Server Error	686
MPI Transaction has expired (timeout)	687
MPI Transaction already submitted	688
MPI Transaction not found	689
PayPal specific error codes	

Transaction type not supported	1082
General error - call support	1900
Token expired	1901
The totals of the cart item amounts do not match order amounts	1902
A successful transaction has already been completed for this token	1903
Refusal	1904
Missing parameter in request - call support	1905
Invalid data	1906
The customer has not yet confirmed payment for this transaction	1907
Authorization is voided	1908
Authorization has expired	1909
Authorization has already been completed	1910
The buyer account is restricted	1911
Currency not supported	1912
Amount specified exceeds allowable limit	1914
Capture was already done	1915
Currency of capture must be the same as currency of authorization	1916
Transaction has already been voided or expired or captured	1917
Transaction expired	1918
Transaction canceled	1919
The amount exceeds the maximum amount for a single transaction	1920
Unable to process your request. Please try later.	1921
Token is missing	1922
General error - please try again	1925
Buyer cannot pay	1926
Cannot charge amount zero	1927
This transaction couldn`t be completed. Please redirect your customer to PayPal	1928
Buyer did not accept billing agreement	1929
Recurring Transaction Id is not valid	1930
Agreement was canceled	1931
Payment has not been authorized by the user	1932
Invalid argument	1933
Invoice Specific error codes	
Cannot Create Invoice	668
Currency not supported for invoice creation	669
Invoice Validation Failed	670
Invoice not found	671
Additional Redirect error codes	

Message (English)	Message (Hebrew)	Error Code
Success	הצלחה	0
Authentication Failed	כשלון בהזדהות לשרת	-10
Transaction validation failed	בעיה בבדיקת ערכי הבקשה	-20
Invalid request or post parameters	פרמטרים לא נכונים לבקשה	-30
CG Gateway returned an error	שגיאה משרת הסליקה	-40
Error accessing the CG Gateway	בעיית גישה לשרת הסליקה	-50
Internal Server Error	בעיית מערכת	-60
Transaction has expired (timeout)	תוקף הבקשה פג	-70
Transaction already submitted	העסקה שנשלחה קיימת במערכת	-80
Transaction not found (used for reports)	העסקה לא נמצאה עבור הדוח	-90
Transaction already cancelled	העיסקה שנשלחה כבר בוטלה	-100