



SPARROW

Transact, Grow, SPARROW.

SPARROW Gateway

Custom Payment Redirect

Version 3.2.0 (Build 7373)

Released September 2016

Revision History

Date	Revision	Comments	Author
2015-06-09	1.0	Initial document created	Blinova Alexandra

Table of Contents

SPARROW Gateway 1

Custom Payment Redirect 1

Revision History 2

Overview 4

Architecture 4

Process Workflow 4

Integration 4

 Setting up the Account 4

Checkout Process 6

 Getting the Token 6

 Processing Payment 6

 Getting the Payment Result 6

 GetCheckoutToken 8

 Request: 8

 Response: 8

 GetCheckoutPaymentResult 9

 Request: 9

 Response: 9

Appendix B – Return Values 11

 ReturnCode 11

 PaymentStatus 11

Overview

Custom white-label redirect page for uComm Global, Inc. uComm's redirect page will communicate with CardinalCommerce and associated banks to process 3DS cards.

Architecture

The architecture will remain the same way the current redirect exists today. Interaction with the service should be performed through the POST Http Request.

Process Workflow

- 1.) Cardholder enters in payment information into the custom redirect page
- 2.) The data is sent as a lookup request to CardinalCommerce to valid enrollment.

*If card is **not** enrolled:*

Cardinal will return an enrollment status of 'N' back with an **ECI** value.

Redirect page will append the ECI value to the transaction and process the payment for the merchant.

If card is enrolled:

- Cardinal will return parameters which represent the URL for the issuing bank (ACS URL) and the Payer Authentication Request (PaReq).

- Request page will POST the form data to the ACS URL. This form POST will include the PaReq payload, a TermURL and an MD value (if no MD, you can leave blank but the field still needs to be there). Through the posting to the ACS URL, the cardholder will be redirected to the bank page to enter in their bank password.

- Once submitted, the ACS will post the cardholder and the encrypted data back to the TermURL (on the Sparrow server).

- The Sparrow server will then take the encrypted value which is the PaRes (Payer Authentication Response) and send that in the Authenticate Request message to CardinalCommerce to decrypt.

- CardinalCommerce will decrypt and send back the Authenticate Response which will include the PaResStatus, the Signature Status along with the ECI, CAVV and XID values.

- The redirect page will append the ECI, CAVV and XID values to the transaction and then process the payment for the merchant.

(Sample code will be provided that represents this process)

Integration

Setting up the Account

Before attempting to process checkout payments, the merchant account which will be receiving payments must be additionally setup.

To do that, visit the Integration page available from the Administration menu, select the account you want to setup, and enter the URL of the page where customers will be redirected after processing the payment.

Select Merchant Account:	<input type="text" value="Credit Card Processing"/>
Allow user to enter quantity	<input type="checkbox"/>
Return URL	<input type="text" value="https://example-merchant.com/PaymentResult.aspx"/>
CSS File	<input type="text"/> <input type="button" value="Browse"/> <input type="button" value="Upload"/> <input type="button" value="Preview"/>
FBI Tools notification emails	<input type="text"/>

FTP	
URL	<input type="text" value="ftp://"/>
Username	<input type="text"/>
Password	<input type="text"/>
<input type="button" value="Test Connection"/>	

<input type="button" value="Save"/>	<input type="button" value="Cancel"/>
-------------------------------------	---------------------------------------

The page available at the entered URL should have the functionality of checking the result of the payment. The details can be found below in *Getting the Payment Result* section. Note that it is critical to have this URL field filled or otherwise payments will not be processed via redirect method.

To integrate an account:

1. Select the Merchant Account to integrate
2. Enter the Return URL
3. Check the "Allow user to enter quantity" checkbox if you would like to allow customers to enter custom quantity for the products they're buying
4. If you want to customize the look of the redirect page, you can upload your own CSS File
5. Click Save

Checkout Process

Getting the Token

First action to initiate the checkout process is getting the checkout token. This is achieved by calling **GetCheckoutToken** method of Checkout API web service. This method has the following parameters:

Parameter	Format	Description
transactionKey	String	Private key of the merchant account that will receive the payment
amount	Decimal	Total amount of the payment
items	XML	Products forming the payment. This is an optional parameter. See Appendix A for detailed description of items format
variableQuantity	Boolean	This is an optional flag signaling a redirect page if the customer is allowed to change the quantity of each product in the purchase. Changing the quantity will affect the total amount of the payment. The default value is false.

The returned token consists of the following fields:

Field	Format	Description
Public key	String, 16..128 characters	This is the public part of the token key which can be exposed to end users. Public key will be specified as an URL query parameter two times during the checkout process
Private key	String, 16..128 characters	This is the private part of the token key which must be kept secret. Private key will be used to retrieve the results of the payment
Expiration Date	<date>T<time> Example:	This is the date and time of the moment when the checkout token will expire and all further actions with it will not be possible

Processing Payment

To process the payment, redirect the customer to the payment URL with **public** token key as a parameter. Here is an example of the resulting URL:

<https://secure.sparrowone.com/Payments/Payment.aspx?Token=214A3E3C8265F6EE>

The Checkout Payment page will then ask the customer of his payment, billing and shipping information and allow him to process the payment. In the end, the customer will be redirected to the page set up in the merchant account settings, with **public** token key as a parameter.

<https://secure.sparrowone.com/Payments/Payment.aspx?Token=eAaNOpIu7DH5h58E3qDNLewDTayu+4Fz>

Getting the Payment Result

The results of the payment will not be provided to the merchant page during the redirect. To get the results, call the **GetCheckoutPaymentResult** method of the Checkout API. This method

requires the **private** token key as a parameter. The fields returned by this method are the following:

Field	Format	Description
ResultCode	Integer	Result code of the GetCheckoutPaymentResult operation. See Appendix B for all possible codes
Message	String	Message describing the result of this operation, i.e. "Success". This is not the description of payment result
PaymentInfo	XML	A structure containing information about payment result. See the table below for field explanations

PaymentInfo element contains the following fields (see full description in XML schema, examples can be found in Appendix A):

Field	Format	Description
Amount	Decimal	Total payment amount (sum of prices for all products)
FailedPaymentAttempts	Integer	Amount of payment attempts made by customer which were declined by the processor
PaymentStatus	String	Status of the payment. See Appendix B for the list of possible values
PaymentDate	<date>T<time>	Date of the transaction, if applicable
PaymentMessage	String	Message returned by the processor
PaymentTransactionId	64-bit integer	The gateway's ID of the transaction, if the transaction exists
Items	XML	A copy of product list provided to GetCheckoutToken method

If the method runs successfully (i.e. specified token exists, is not expired etc.), it returns the ResultCode of 0. Note that even if the method is successful, this does not mean that the payment was successful. Check the <PaymentInfo> element contents to get the result of the payment itself. Result codes different from zero mean that an error has happened, and <PaymentInfo> element of the response won't contain any data. You can find the full list of result codes and their meanings in the Appendix B.

Appendix A – Web Methods Description

This section contains request and response formats for both Checkout API methods. Full schema can be found at <https://secure.sparrowone.com/Services/CheckoutApi.asmx?wsdl>

GetCheckoutToken**Request:**

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetCheckoutToken xmlns="https://secure.sparrowone.com/Services/">
      <transactionKey>string</transactionKey>
      <amount>decimal</amount>
      <items>
        <CheckoutPaymentItem>
          <Description>string</Description>
          <SkuNumber>string</SkuNumber>
          <UnitPrice>decimal</UnitPrice>
          <Quantity>int</Quantity>
        </CheckoutPaymentItem>
        <CheckoutPaymentItem>
          <Description>string</Description>
          <SkuNumber>string</SkuNumber>
          <UnitPrice>decimal</UnitPrice>
          <Quantity>int</Quantity>
        </CheckoutPaymentItem>
        ...
      </items>
      <variableQuantity>boolean</variableQuantity>
    </GetCheckoutToken>
  </soap:Body>
</soap:Envelope>
```

Response:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetCheckoutTokenResponse
xmlns="https://secure.sparrowone.com/Services/">
      <GetCheckoutTokenResult>
        <ResultCode>int</ResultCode>
        <Message>string</Message>
        <Token>
          <PublicKey>string</PublicKey>
          <PrivateKey>string</PrivateKey>
          <ExpirationTime>dateTime</ExpirationTime>
        </Token>
      </GetCheckoutTokenResult>
    </GetCheckoutTokenResponse>
```



```

</soap:Body>
</soap:Envelope>

```

GetCheckoutPaymentResult

Request:

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetCheckoutPaymentResult
xmlns="https://secure.sparrowone.com/Services/">
      <privateKey>string</privateKey>
    </GetCheckoutPaymentResult>
  </soap:Body>
</soap:Envelope>

```

Response:

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetCheckoutPaymentResultResponse
xmlns="https://secure.sparrowone.com/Services/">
      <GetCheckoutPaymentResultResult>
        <ResultCode>int</ResultCode>
        <Message>string</Message>
        <PaymentInfo>
          <Amount>decimal</Amount>
          <Items>
            <CheckoutPaymentItem>
              <Description>string</Description>
              <SkuNumber>string</SkuNumber>
              <UnitPrice>decimal</UnitPrice>
              <Quantity>int</Quantity>
            </CheckoutPaymentItem>
            <CheckoutPaymentItem>
              <Description>string</Description>
              <SkuNumber>string</SkuNumber>
              <UnitPrice>decimal</UnitPrice>
              <Quantity>int</Quantity>
            </CheckoutPaymentItem>
            ...
          </Items>
          <FailedPaymentAttempts>int</FailedPaymentAttempts>
          <PaymentTransactionId>long</PaymentTransactionId>
          <PaymentDate>dateTime</PaymentDate>
          <PaymentStatus>string</PaymentStatus>
          <PaymentMessage>string</PaymentMessage>
        </PaymentInfo>
      </GetCheckoutPaymentResultResult>
    </GetCheckoutPaymentResultResponse>
  </soap:Body>
</soap:Envelope>

```

```
</GetCheckoutPaymentResultResponse>  
</soap:Body>  
</soap:Envelope>
```

Appendix B – Return Values

ReturnCode

The following table describes possible values of ReturnCode field in all web method responses.

Return Code	Description
0	Success
9	Transaction Key parameter is missing
10	Invalid Transaction Key specified
11	Private Key parameter is missing
12	Specified checkout token was not found (invalid private key)
13	Checkout token has expired
14	Invalid checkout amount: sum of prices for each product is not equal to the total payment amount
15	Specified account cannot process transactions. Possible reason is that account is closed
100	Unknown processing error, contact support for details

PaymentStatus

The following table describes possible values of PaymentStatus subelement of PaymentInfo element in **GetCheckoutPaymentResult** method's response.

Status	Description
Successful	Payment was successful. Transaction has been created.
Declined	Processor has declined the payment. PaymentMessage contains details. Transaction has been created.
Error	There was an error processing the message, either at gateway or processor's side. Transaction may or may not exist in the case of this payment status.