



# SPARROW

*Transact, Grow, SPARROW.*

## **SPARROW Gateway**

Developer API

**Version** 3.2.0 (Build 7373)

**Released** September 2016

---

## Revision History

Date	Revision	Comments	Author
2016-02-26	2.0	Initial document created	Alexandra Blinova

***Table of Contents***

---

SPARROW Gateway ..... 1

Developer API ..... 1

Revision History.....2

Overview..... 4

Architecture..... 4

    Merchant Private Key and Payment Types ..... 4

Integration..... 6

    Setting up the Account ..... 6

Checkout Process ..... 7

    Getting the Token ..... 7

    Processing Payment..... 7

    Getting the Payment Result..... 7

Appendix A – Web Methods Description ..... 8

    GetCheckoutToken ..... 9

        Request: ..... 9

        Response: ..... 9

    GetCheckoutPaymentResult..... 10

        Request: ..... 10

        Response: ..... 10

Appendix B – Return Values ..... 12

    ReturnCode ..... 12

    PaymentStatus..... 12

## Overview

There are two primary options in which transactions can be submitted through the Sparrow API. The Payment Redirect Method allows merchants to send consumers to a secure payment page for payment. The second method and most transparent method is the Direct Post Method.

## Architecture

Given the two methods of integration, separate URLs are used for the interaction of the systems. Each has a different purpose however both achieve the same result. Here are the URLs for both:

- <https://secure.sparrowone.com/Payments/Payment.aspx>
- [https://secure.sparrowone.com/Payments/Services\\_API.aspx](https://secure.sparrowone.com/Payments/Services_API.aspx)

The Payment.aspx is a secure payment page hosted on the Sparrow gateway. Consumers can be redirected from the merchant website to the payment gateway for customer information collection and payment processing.

There is an additional web service used to provide information for redirect payments. It is located at the following URL:

- <https://secure.sparrowone.com/Services/CheckoutApi.asmx>

Description of all its methods can be found in Appendix A and in sections below.

Services\_API.aspx is an API implementation of secure payments. In this fashion, merchants can seamlessly integrate gateway payment functions into their checkout process.

### ***Merchant Private Key and Payment Types***

When sending a payment transaction request to Sparrow (on either implementation) a Merchant Private Key is required. A sample Merchant Private Key follows:

`MTA6MzM6NTYgUE0tLVBheVRhY3RpeA==`

Further, each Private Key represents the payment type and merchant account targeted for processing the transaction. This methodology allows merchants to have multiple merchant accounts and payment types for processing transactions. The transactions can be targeted at a particular account based on it's characteristics. For example, to process a Visa or MasterCard payment the calling program will send the private key associated with the V/MC merchant account. (Discover and American Express can also be passed through a charge card key if the merchant accepts them.)

To generate a private key, select Add Transaction Routing and enter the appropriate information on the Transaction Routing Management page:

The screenshot shows the 'Transaction Routing Management' page for user166. The page features a sidebar with navigation options like Home, Payments, Reports, Customers, Data Vault, FBI Tools, Invoices, Security, Administration, and System Configuration. The main content area displays a table of transaction routing rules with columns for Enabled, Display in VT, Type, Name, Transaction Key, Target, and Payment Type. Each row includes 'Edit' and 'Delete' links. Below the table are buttons for 'Add Transaction Routing', 'Save', and 'Cancel'. The footer contains copyright information for SparrowOne, LLC and version 2.7 (Build 6246).

Enabled	Display in VT	Type	Name	Transaction Key	Target	Payment Type	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Direct	Account 1	V5N2MOWNK7SGJLZJH4HSR1	MA185	Star Card	<a href="#">Edit</a> <a href="#">Delete</a>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Direct	Account 2	6CZBW92E2MF084S7141HAWH	MA185	Star Card	<a href="#">Edit</a> <a href="#">Delete</a>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Direct	Account 3	SJACVWSZD5P84DKJK1A8ENC	MA191	Star Card	<a href="#">Edit</a> <a href="#">Delete</a>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Direct	Account 4	4L2RVN377KJLJLTXR8BWB0	MA191	Star Card	<a href="#">Edit</a> <a href="#">Delete</a>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Direct	Account 5	HDJ82VQ8N16WNEJEL0LJLK	MA193	Star Card	<a href="#">Edit</a> <a href="#">Delete</a>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Direct	Account 6	X8CHWPKVWVYWH5F6CBK0XK	MA193	Star Card	<a href="#">Edit</a> <a href="#">Delete</a>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Direct	Account 7	LYUJAB8BNW70VRFTSHTU2EG	MA247	Star Card	<a href="#">Edit</a> <a href="#">Delete</a>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Direct	Account 8	VPC85J545F248ASFU1VB05MY	MA247	Star Card	<a href="#">Edit</a> <a href="#">Delete</a>

The following image shows the account management page in the Merchant Accounts section:

The screenshot shows the 'Merchant Accounts' page for user166. The sidebar is identical to the previous page. The main content area displays a table of merchant accounts with columns for Merchant Account Name, Payment Type, Creation Time, Time Zone, and Status. Each row includes 'Edit' and 'Delete' links. Below the table is a pagination control showing 'Page 1 of 1 (4 items)' and a 'Page size' dropdown set to 10. The footer contains copyright information for SparrowOne, LLC and version 2.7 (Build 6246).

Merchant Account Name	Payment Type	Creation Time	Time Zone	Status	
MA185	Star Card	Wednesday, August 17, 2011	CMT -06:00	Live	<a href="#">Edit</a> <a href="#">Delete</a>
MA191	Star Card	Monday, October 10, 2011	GMT -12:00	Live	<a href="#">Edit</a> <a href="#">Delete</a>
MA193	Star Card	Thursday, October 20, 2011	GMT -12:00	Live	<a href="#">Edit</a> <a href="#">Delete</a>
MA247	Star Card	Friday, June 22, 2012	CMT -12:00	Live	<a href="#">Edit</a> <a href="#">Delete</a>

## 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.

The screenshot shows a web form for setting up a merchant account. It includes the following fields and controls:

- Select Merchant Account:** A dropdown menu with "Credit Card Processing" selected.
- Allow user to enter quantity:** A checkbox that is currently unchecked.
- Return URL:** A text input field containing "https://example-merchant.com/PaymentResult.aspx".
- CSS File:** A text input field, a "Browse" button, an "Upload" button, and a "Preview" button.
- FBI Tools notification emails:** A text input field.
- FTP Section:**
  - URL:** A text input field containing "ftp://".
  - Username:** A text input field.
  - Password:** A text input field.
  - Test Connection:** A button below the FTP fields.
- Save and Cancel:** Two buttons at the bottom right of the form.

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>
```

```

    </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>
```

## 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.