



SPARROW

Transact, Grow, SPARROW.

SPARROW Gateway

Developer Data Vault Payment Type API

Version 3.2.0 (Build 7373)

Released September 2016

Revision History

Date	Revision	Comments	Author
2016-05-09	1.0	Initial document created	Blinova Alexandra

Table of Contents

SPARROW Gateway 1
Developer Data Vault Payment Type API 1
Revision History.....2
Overview..... 4
Architecture..... 4
 Merchant Private Key and Payment Types 4
Integration..... 4
 Setting up the redirect settings 4
Data Vault payment type API Process 6
 Getting Callback Result 10
 Appendix A – Return Values 11
 Return Message 11

Overview

The Data Vault payment type API allows merchants to add and update Data Vault customer payment type.

Architecture

Service for integration is available at the following URL:

<https://secure.sparrowone.com/Services/PaymentTypeRedirectTokenHandler.ashx>

Interaction with the service should be performed through the POST Http Request.

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:

`+ubfjSpQWaT65U01xcNSuQ6BGKF16166`

Further, each Private Key represents the payment type (e.g. ACH, Credit Card, etc.) and merchant account targeted for processing the transaction. This methodology allows merchants to have multiple merchant accounts and payment types for processing payments.

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

The following image shows the Transaction Routing Management page available from the Administration menu:

Enabled	Display in VT	Type	Name	Transaction Key	Target	Payment Type	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Direct	Account 1	V5N2MOWNK7SGJ2LZLNADHSDR1	MA185	Star Card	Edit Delete
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Direct	Account 2	6CZBW62E2MFD854S7141HAWH	MA185	Star Card	Edit Delete
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Direct	Account 3	SJACVAISZD5P64D6JK1A8ENC	MA191	Star Card	Edit Delete
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Direct	Account 4	4L2RVNI377KUFLJLTXRSBW0	MA191	Star Card	Edit Delete
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Direct	Account 5	HDJ82W2BN18WN0EJEL0LJLK	MA193	Star Card	Edit Delete
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Direct	Account 6	XBCHIWPKV6VYWH5F6CBKONX	MA193	Star Card	Edit Delete
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Direct	Account 7	LYUOA8VBNW70VRFTSHTU2EG	MA247	Star Card	Edit Delete
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Direct	Account 8	VPC85J645F24BASFU1VBO5MY	MA247	Star Card	Edit Delete

Integration

Setting up the redirect settings

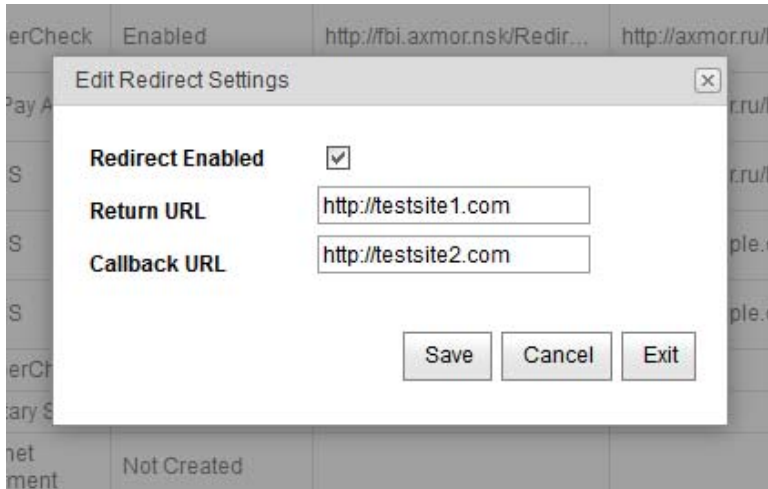
Before attempting to use the Data Vault payment type API, the merchant account which will be used, must be additionally setup.

Merchant account type (e.g. ACH, Credit Card, etc.) should be the same as a type of the payment type that merchant wants to create or update.

To set redirect settings for the account go to the 'Redirect Settings' page available from the Administration menu.

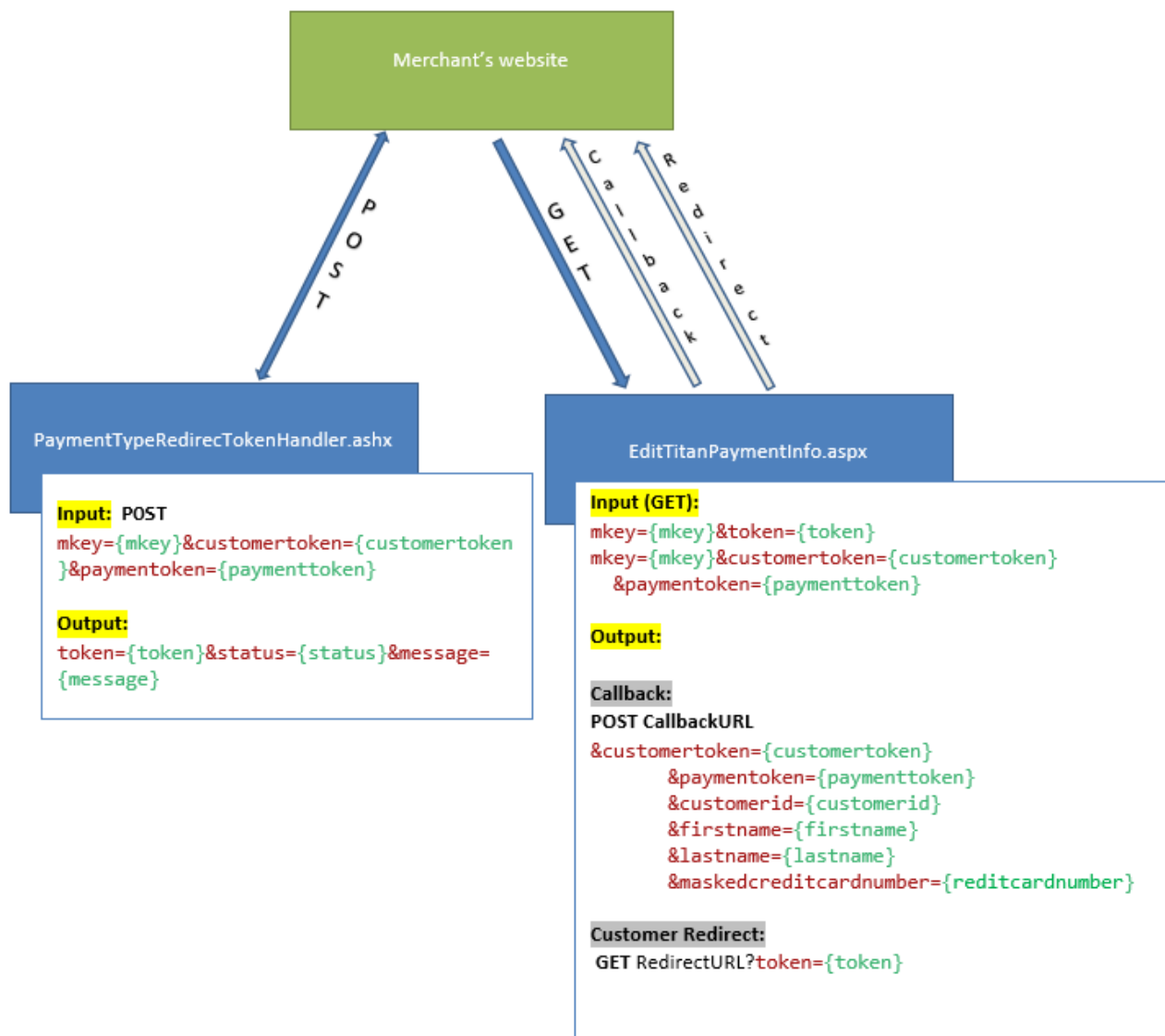
To integrate the account:

1. Press Edit for the Merchant Account from the list
2. Check the 'Redirect Enabled' setting
3. Enter the Return URL (after the payment information is saved Sparrow system will redirect user to the Return URL)
4. Enter the Callback URL (After the payment information is saved Sparrow system will send the POST request to the Callback URL)
5. Press Save or Exit



The image shows a screenshot of a web application interface. In the foreground, a modal dialog box titled "Edit Redirect Settings" is open. The dialog has a close button in the top right corner. It contains three labeled fields: "Redirect Enabled" with a checked checkbox, "Return URL" with a text input field containing "http://testsite1.com", and "Callback URL" with a text input field containing "http://testsite2.com". At the bottom of the dialog are three buttons: "Save", "Cancel", and "Exit". The background shows a table with columns for "Merchant Account", "Status", and "URL". The first row shows "erCheck", "Enabled", and "http://fbi.axmor.nsk/Redir...". The second row shows "Pay A", "Not Created", and "http://axmor.ru/tr".

Data Vault payment type API Process



First, merchants posts PaymentTypeRedirectTokenHandler.ashx and received a **token** which can be used to redirect customer to the EditDataVaultPaymentInfo.aspx. The purpose of this token is to be used for redirect only if customer is not created. Please note that this **token** may not be used for payments.

Interaction with the service should be performed through the POST Http Request with following parameters:

Parameter	Required	Format	Description
mkey	Required	String	Secured merchant account key
customertoken	Required	String	Secured Data Vault Customer key
paymenttoken	Optional	String	Secured payment type key. <ul style="list-style-type: none"> if paymenttoken is not provided in the request, new payment type will be created if paymenttoken is provided in the request, corresponding payment type will be updated

The returned request consists of the following fields:

Parameter	Required	Format	Description
token	Required	String	Redirect token for add or edit customer' payment type. Empty when some error occurred.
status	Required	String	Successful/Error
message	Required	String	Message describing the result of this operation

You can find the full list of result messages in the [Appendix A](#).

After the token is generated, merchant system should redirect user to the EditDataVaultPaymentInfo.aspx page. Here is an example of the resulting URL:
<https://secure.sparrowone.com/Users/EditDataVaultPaymentInfo.aspx?token={0}> ({0} token from response from API)

There are two ways how the merchant may force customer to update his payment info:

1. If customer doesn't exist, redirect customer to EditDataVaultPaymentInfo.aspx with a onetime **token** retrieved from PaymentTypeRedirectTokenHandler.ashx
2. If customer exists, redirect customer to EditDataVaultPaymentInfo.aspx with a **customertoken** or **paymenttoken**. (see input usage for EditDataVaultPaymentInfo.aspx)

On the EditDataVaultPaymentInfo.aspx page user will be able to create/update payment type:

Second, once customer updates his payment info on EditDataVaultPaymentInfo.aspx, we perform two actions:

1. Send callback request to the merchant with all necessary data including **customertoken**, **paymenttoken**, maskedcreditcardnumber etc.(see callback output format).
2. Redirect customer to the merchant's website. Since this action interact with the customer, we DO NOT PASS any payment information and tokens in this request (see customer redirect format).

After the payment information is saved, Sparrow system will send the POST request to the Callback URL (which user entered in the [Account Redirect settings](#)). Then Sparrow system will redirect user to the Return URL.

The following table describes request content for different types:

Payment Type	Content
Credit Card / Star Card	firstname={0} lastname={1} maskedcreditcardnumber={2} customertoken={3} paymenttoken={4} customerid={5}
ACH / Check	Account type <i>Business</i> company={0} maskedaccountnumber={1} customertoken={2} paymenttoken={3} customerid={4}
	Account type <i>Personal</i> firstname={0} lastname={1} maskedaccountnumber={2} customertoken={3} paymenttoken={4} customerid={5}

The Return URL will be passed with a get variable of: **token**={0}. This token will be the token that was generated for the redirect. This allows the merchant to store it in their DB and link the returning customer to the proper page.

Redirect and callback URLs are configured in the merchant portal, you can access this page from menu: Administration → Redirect Settings or use a direct link (as described above):
<https://secure.sparrowone.com/Users/Administration/RedirectSettings.aspx>

Third, to perform a tokenized payment, the merchant needs to send in the token field a value from **customertoken** or a **paymenttoken** that are returned in the callback result.

Example 1: **POST: services_api.aspx**

`mkey={mkey}&transtype=sale&amount=1&token={customertoken}`

Example 2: **POST: services_api.aspx**

`mkey={mkey}&transtype=sale&amount=1&token={paymenttoken}`

Getting Callback Result

To get callback result the request with transtype= getpaymentinfocallbackresult should be sent. The table below describes parameters of this request:

Parameter	Required	Format	Description
mkey	Required	String	Secured merchant account key
transtype	Required	getpaymentinfocallbackresult	
token	Required	String	Redirect token for adding or editing customer payment type

The example of the request:

`mkey=XTZ18PUHGPTKZF0CPEU3KJI1&transtype=getpaymentinfocallbackresult&token=XCHKY40WF6SS1I3G`

The returned response consists of the following fields:

Parameter	Required	Format	Description
response	Required	numeric	See the table below for possible responses list
textresponse	Required	String	Textual response
postdata	Required	String	Callback data in URL encoded format

Response field can have the following values:

Value	Description
1	OK, postdata is filled
2	Decline, callback data is not found, postdata is empty
3	Error, postdata is empty

The example of returned response:

`response=1&textresponse=Processed&postdata=firstname%3dJohn%26lastname%3dDoe%26maskedaccountnumber%3d*****1111%26customertoken%3dSPYWOB10G718TQIP%26paymmentoken%3dJT86MSW6RK26PK8W%26customerid%3d`

Appendix A – Return Values

Return Message

Values of 'message' field for response with 'Successful' status is:

- Token has been created successfully.

Possible values of 'message' field for response with 'Error' status are described below:

- Required field 'mkey' is missing.
- Required field 'customertoken' is missing.
- Routing with specified '{0}' was not found. ({0} = mkey)
- Routing must be enabled.
- Merchant account is not configured for redirect.
- Customer with specified '{0}' was not found. ({0} = customertoken)
- Payment information with specified '{0}' was not found. ({0} = paymenttoken)
- Payment information does not belong to the customer.
- Payment information is not active.
- Payment type does not match the type from routing info.