



SPARROW

Transact, Grow, SPARROW.

Merchant Public Services

Settlement API

Version 3.2.0 (Build 7373)

Released September 2016

Revision History

Date	Revision	Comments	Author
2016-03-10	1.0	Initial document created	Vlad Volosuk
2016-03-23	1.1	Action route URL is corrected	Vlad Volosuk

Table of Contents

Revision History 2

Overview 4

Architectural Overview 4

 Action Route 4

 Action Request 5

 Action Response 6

 JSON Objects 7

 XML Objects 7

 Authentication 7

Account Service 7

 Authenticate Action 7

Terminal Service 8

 Account Settle Action 8

Overview

The Sparrow Gateway provides a set of services which can be used for integrating the gateway with merchant's IT systems. Merchant Public API provides the following RESTfull services:

- **Account** service – it is auxiliary service to obtain the authentication token required for invoking other Merchant Public API services
- **Transaction** service - this service allows querying transactions processed by the merchant in the Sparrow Gateway
- **Terminal** service – this service provides control functions over transaction processing

This document provides architectural overview of Merchant Public API and describes the services included into this API.

Architectural Overview

Merchant Public API consists of a number of *Actions* which are logically grouped into services. This section describes the structure of HTTP requests which are used for invoking *Actions*.

The legend:

- **<TERM>** indicates a parameter that will be described below of the snippet
- **[any-text]** indicates an optional part that can be not provided in requests for some *Actions* (see the specification of the particular *Action*)

Action Route

A route to an *Action* is a URL-based path that uniquely identifies the *Action* for invocation and invocation parameters:

```
<URL>/api/public/merchant/<SERVICE>/<ACTION> [ /<PATH-PARAMETERS> ] [ ?<QUERY-PARAMETERS> ]
```

Term	Description
<URL>	The address of the Sparrow Gateway server where the service is hosted. Merchants have to use the following URL for the production (life) integrations: https://secure.sparrowone.com
<SERVICE>	The name of the service where the <i>Action</i> is located. Examples: account , transactions , terminal . See documentation of services below in this document.
<ACTION>	The name of the <i>Action</i> . See documentation of services below in this document. Note: normally the name of the action must be always presented in <i>Action Route</i> . However, the service can have a default Action. In this case, this action is invoked without the name in the <i>Action Route</i> . Example: authenticate action in the account service.
<PATH-PARAMETERS>	Path parameters are used to clarifies an entity for which the action has

	to be invoked
<QUERY-PARAMETERS>	Additional parameters provided in the form of HTTP query string

Action Request

An *Action* can be invoked by a HTTP request:

```
<HTTP-VERB> <ACTION-ROUTE> HTTP/1.1
Content-Type: <REQUEST-CONTENT-TYPE>
Accept: <RESPONSE-CONTENT-TYPE>
[other HTTP headers]

<REQUEST-DATA>
```

Term	Description
<HTTP-VERB>	<p>HTTP verb for invoking the service <i>Action</i>. Possible values are:</p> <ul style="list-style-type: none"> ▪ GET ▪ POST ▪ PUT ▪ DELETE <p>It is not mandatory for <i>Actions</i> to support all 4 verbs (but at least one verb must be supported). The verb can be interrupted as a sub-action supported by the <i>Action</i>. See the service documentation to select the proper verb for the <i>Action</i>.</p>
<ACTION-ROUTE>	An URL-based path that uniquely identifies the <i>Action</i> for invocation and invocation parameters. See the section above for details.
<REQUEST-CONTENT-TYPE>	<p>The content type of the <REQUEST-DATA>. Possible values are:</p> <ul style="list-style-type: none"> ▪ <i>application/xml</i> ▪ <i>application/json</i> <p>See <i>JSON Objects</i> and <i>XML Objects</i> sections below for object format details.</p>
<RESPONSE-CONTENT-TYPE>	<p>The expected content type for the response. Possible values are:</p> <ul style="list-style-type: none"> ▪ <i>application/xml</i> ▪ <i>application/json</i> <p>If <i>Accept</i> HTTP header is not provided in the request, then the response will be returned in JSON format.</p> <p>See <i>JSON Objects</i> and <i>XML Objects</i> sections below for object format details.</p>
<REQUEST-DATA>	<p>Request data object formatted according <REQUEST-CONTENT-TYPE>. Names of request parameters are case-sensitive.</p> <p>See the <i>Action</i> documentation for parameters supported by the <i>Action</i> in <REQUEST-DATA>.</p>

Action Response

Here is a format of *Action* responses:

```
HTTP/1.1 <HTTP-CODE> <HTTP-CODE-DESCRIPTION>
Content-Type: <RESPONSE-CONTENT-TYPE>
[other HTTP headers]

<RESPONSE-DATA>
```

Term	Description
<HTTP-CODE>	<p>A standard HTTP response code. Here are codes (with HTTP descriptions) commonly used in Merchant Public API:</p> <ul style="list-style-type: none"> ▪ 200 OK The action was executed successfully ▪ 403 Forbidden The current user is not allowed to execute this <i>Action</i> ▪ 404 Not Found The Action was requested for the entity that is not found on the system ▪ 400 Bad Request The Action was executed with an error. More information about the error is returned in <RESPONSE-DATA>. ▪ 500 Internal Server Error It is unwanted, but possible response that indicates about internal (unexpected) error during processing the <i>Action</i>. <p>Particular <i>Actions</i> may support other HTTP response codes, see the <i>Action</i> documentation for details.</p>
<HTTP-CODE-DESCRIPTION>	<p>A general message that clarifies the <HTTP-CODE>. See above for examples.</p>
<RESPONSE-CONTENT-TYPE>	<p>The content type of <RESPONSE-DATA>. Possible values are:</p> <ul style="list-style-type: none"> ▪ <i>application/xml</i> (see <i>XML Objects</i> section for details) ▪ <i>application/json</i> (see <i>JSON Objects</i> section for details) <p>See <i>JSON Objects</i> and <i>XML Objects</i> sections below for object format details.</p>
<RESPONSE-DATA>	<p>Response parameters formatted according <REQUEST-CONTENT-TYPE>.</p> <p>Names of response parameters are case-sensitive.</p> <p>See the <i>Action</i> documentation for parameters supported by the <i>Action</i> in <REQUEST-DATA>.</p>

JSON Objects

Action requests and responses can be formatted as JSON objects:

```
{
  "Username": "JoeDoe",
  "Password": "pwd"
}
```

See *Action* documentation for supported request/response parameters.

XML Objects

Action requests and responses can be formatted as XML objects:

```
<LoginResponse>
  <AuthToken>TIUPQV+R1R5RGORA4IZVUIXIYEP+OFNB</AuthToken>
</LoginResponse>
```

See *Action* documentation for supported request/response parameters.

The name of the root element is composed as shown below:

- The root element for *Action* requests is named **<ActionName>Request** (e.g. LoginRequest)
- The root element for *Action* responses is named **<ActionName>Response** (e.g. LoginResponse)

where **<ActionName>** is the name of the current *Action*, the first letter is the upper case, others – in the low case.

Authentication

Only authenticated users are allowed to invoke service *Actions*. To confirm the authentication, the parameter **token** has to be specified in the query string during *Action* invocation.

The token value should be obtained via **Authentication** action of **Account** service (see the below in this document). That action is the only action that can be invoked without authentication.

Account Service

The **Account** service is auxiliary service that allows retrieving the authentication token required for invoking other *Actions*.

Authenticate Action

This action is used to obtain the authentication token for the specified user.

Name	Value
Service	Account
Action Name	(default)
Method	POST
Path Parameters	No
Query Parameters	No
Request	The following items must be provided in <REQUEST-DATA> : <ul style="list-style-type: none"> ▪ Username ▪ Password

Response	The action returns the following items in <RESPONSE-DATA> : <ul style="list-style-type: none"> ▪ AuthToken – an authentication token for subsequent service actions calls
-----------------	--

Terminal Service

The **Terminal** service provides a set of *Actions* over payment transactions.

Account Settle Action

This action is used to settle all transactions for the specified account which require settlement.

Name	Value
Service	Terminal
Action Name	Settle
Method	POST
Path Parameters	No
Query Parameters	The following items must be provided in <QUERY-PARAMETERS> : <ul style="list-style-type: none"> ▪ token – the authentication token
Request	The following items must be provided in <REQUEST-DATA> : <ul style="list-style-type: none"> ▪ Mkey – the merchant key associated with the account where transactions need to be settled
Response	The action returns the following items in <RESPONSE-DATA> : <ul style="list-style-type: none"> ▪ Status – the status of the settlement operation ▪ Message – additional text message that contain details about the settlement operation