6/5/2025

GLODIPAY SERVER TO SERVER (S2S) API

VERSION 1.5.5

Table of Contents

INTRODUCTION 2
ENDPOINTS 3
API SPECIFICATION 4
SERVER TO SERVER (S2S) 4
Request: 4
paymentMethod 4
paymentFilter 4
Response 6
Data will be included when calling the callbackUrl 7
TRANSACTION QUERY 8
Request 8
Response 8
NOTIFICATION 9
Payload 9
Response 10
Test Cards with merchant ID start with 11xxxxxxxx 11
Test Cards with merchant ID start with 14xxxxxxxx 11
APPENDIX 11
Payment Methods 11
Status Codes 12
Country Codes 13
Card Type 20
Code example 21
PHP 21
NodeJS 22

INTRODUCTION

This document involves description of:

- GLODIPAY Server to Server (S2S): supports merchants to accept credit/debit cards

ENDPOINTS

Environment	Base Endpoint
Test	https://payment-sandbox.gpayprocessing.com
Production	Get it from the API Keys page of the GLODIPAY Portal

API SPECIFICATION

SERVER TO SERVER (S2S)

Request:

- Endpoint: /v1/card/api
- Method: POST
- Content Type: application/json

Parameter	Field Type	Required	Description
merchantId	String L (1,50)	M	Merchant's id
orderRef	String L (1,250)	M	Merchant **unique Identify** transaction
amount	Float	M	Invoice amount
currency	String L (3)	M	Invoice currency, ISO 4217 standard specifies three-letter ("Alpha-3") codes. e.g. USD
cancelUrl	String L (1,300)	M	(Frontend) The URL the customer will be directed to if they decide to cancel payment and return to your website. Please make sure that the URL is under https!
callbackUrl	String L (1,300)	M	(Frontend) URL where the customer is directed to after a successful payment. Please make sure that the URL is under https!
notificationUrl	String L (1,300)	M	Endpoint on your server for GLODIPAY to send webhooks. Please make sure that the URL is under https!
errorUrl	String L (1,300)	M	(Frontend) URL where the customer is redirected in case of an error in the payment. Please make sure that the URL is under https!
orderDescription	String	M	A short description of the underlying transaction or trade to be displayed on the redirection screen
metadata	JSON	O	Set of key-value pairs that can be attached to the checkout session
transactionDocuments	JSON	O	Supporting documents for the transaction
paymentMethod	String	M	Specify the payment methods you want displayed for the session
paymentFilter	JSON	O	Specify the payment methods you want removed for the session
feeBySeller	Number min(0), max(100)	O	An integer between 0 and 100. Percentage of processing fee to be paid by the merchant. GLODIPAY will charge the merchant an amount.
billingFirstName	String	M	Billing Details - Address, Recipient Name
billingLastName	String	M	
billingStreet1	String	M	
billingStreet2	String	M	
billingCity	String	M	
billingEmail	String	M	
billingState	String	M	

billingCountry	String	M	Address country (Country (ISO 3166-1 alpha_2 country code)
billingPostalCode	String	M	Postal Code
billingPhoneCountryCode	String	M	Calling country code (for example, '1' for United States, '91' for India)
billingPhoneNumber	String	M	Phone Number
brandName	String L (1,255)	O	Override the Brand Name to be displayed on the GLODIPAY hosted checkout screen
colorMode	String L (1,255)	O	You can pass up to 3 color codes, separated by ---. Examples: Color names: darkgreen---lightgreen---mediumseagreen HEX codes: #2e7d32---#e8f5e9---#81c784 RGBA values: rgba(46,125,50,1)---rgba(232,245,233,1)---rgba(129,199,132,1)
logoSource	String L (1,255)	O	Override the Logo to be displayed on the GLODIPAY hosted checkout screen
customerIp	String	M	IP address of the customer
signature	String	M	Signature is calculated in the following steps: 1. Concatenate all parameters values except the signature (arranged in ascending order of parameter's name) to a string 2. RSA sign the hash by merchant's private key

M = Mandatory
O = Optional
C = Condition

Response

- Method: Form POST
- Content type: application/json

Parameter	Field Type	Required	Description
status	String	M	**success, redirect, error**
message	String	M	

errors	JSON	O	Response with validation failed
			{
			"status": "error",
			"message": "Invalid request data.",
			"errors": [
			{
			"field": "billingEmail",
			"message": [
			"The billing email field is required."
			]
			},
			{
			"field": "customerIp",
			"message": [
			"The customer ip field is required."
			]
			}
			]
			}

data	JSON	O	Response with Authenticate
			{
			"status": "redirect",
			"message": "Please redirect the user to complete the payment.",
			"data": {
			"transactionId": "01jwz0ty1640apxvmyzqpvc18a",
			"url": "https://payment-sandbox.gpayprocessing.com/card/3ds/01jwz0ty1640apxvmyzqpvc18a"
			}
			}
			
			Response bypasses 3DS authentication
			{
			"status": "success",
			"message": "The transaction has been successfully completed.",
			"data": {
			"transactionId": "01jwz13qfcx4z61ded3jcj0tf2"
			}
			}

Data will be included when calling the callbackUrl

- Method: GET

GLODIPAY will sending payment result to merchant server's endpoint as specified in **callbackUrl**.

Parameter	Field Type	Required	Description
payload	Base 64 String	M	Base 64 string contains JSON.

Payload Base 64 decode JSON fields

Parameter	Field Type	Required	Description
status	String	M	General status of transaction.
transactionId	String	O	GLODIPAY Identify transaction
ref	String	O	Merchant Identify transaction
amount	Float	O	
currency	String	O	Invoice currency, ISO 4217 standard specifies three-letter ("Alpha-3") codes. e.g. USD
signature	String	O	Signature is calculated in the following steps: 1. Concatenate all parameters values except the signature (arranged in ascending order of parameter's name) to a string 2. RSA sign the hash by merchant's private key

TRANSACTION QUERY

Request

- Function: query the info of a transaction.
- Endpoint: /v1/checkout/query
- Method: POST
- Data format: JSON

Parameter	Field Type	Required	Description
merchantId	String	M	Merchant's id
transactionId	String	M	GLODIPAY Identify transaction
signature	String	M	Signature is calculated in the following steps: 1. Concatenate all parameters values except the signature (arranged in ascending order of parameter's name) to a string 2. RSA sign the hash by merchant's private key

Response

The same with NOTIFICATION payload

NOTIFICATION

- Function: send instant payment notification to merchant over API
- Method: POST
- Data format: JSON
- Return JSON payload {“returnCode”:”100”} in 30 seconds to let GLODIPAY recognize that you have received the Webhook. If not, GLODIPAY will send the Webhook 10 times. The first time is 30 minutes after the transaction is initiated, and then every 5 minutes thereafter if transaction is completely succeeded.

Payload

Parameter	Field Type	Required	Description
merchantId	String	M	Merchant’s id
transactionId	String	M	GLODIPAY Identify transaction
transactionNumber	String	M	GLODIPAY Transaction Number
ref	String	M	Merchant Identify transaction
currency	String	M	Invoice currency, ISO 4217 standard specifies three-letter ("Alpha-3") codes. e.g. USD
amount	Float	M	
paidAmount	Float	O	Paid amount
settlementAmount	Float	O	Settlement amount
estimationSettlementAt	ISO 8601 datetime format	O	Estimation Settlement Datetime E.g: 2023-12-16T02:13:37+00:00
fees	JSON	O	Fees JSON
paymentMethodDetails	JSON	O	
status	String	M	Status of transaction
statusCode	Number	M	Status code of transaction
metadata	JSON	O	Set of key-value pairs of checkout session
transactionDocuments	JSON	O	Documents of the transaction
message	String	O	Transaction message
transactionCreatedAt	ISO 8601 datetime format	M	Datetime from GLODIPAY system E.g: 2023-12-16T02:13:37+00:00
originalTransactionCreatedAt	ISO 8601 datetime format	M	Original Datetime E.g: 2023-12-16T02:13:37+00:00
signature	String	M	Signature is calculated in the following steps: 1. Concatenate all parameters values except the signature (arranged in ascending order of parameter’s name) to a string 2. RSA sign the hash by merchant’s private key

fees JSON fields

Parameter	Field Type	Required	Description
operate	Float	O	Total fees
buyer	Float	O	Buyer fees amount
seller	Float	O	Seller fees amount
rolling	Float	O	Rolling fees amount
estimationRollingReleaseAt	ISO 8601 datetime format	O	Estimation Rolling Release Datetime E.g: 2025-04-27T02:47:02+00:00

paymentMethodDetails JSON fields

Parameter	Field Type	Required	Description
displayName	String	O	Payment method label
group	String	O	Payment method group type
family	String	O	Payment method family type
type	String	O	Name of payment method type You can use this value to specify Payment method in POST PAYMENT of field paymentMethod or paymentFilter
Value of *type*	JSON	O	

Response

- Data format: JSON

Parameter	Field Type	Required	Description
returnCode	String	R	Return code “100” to let GLODIPAY recognize that you have received the webhook.
description	String L (1,1500)	O	Description for the response

Test Cards with merchant ID start with 11xxxxxxx

Card Number	Expire	CVV	Password	
4242424242424242	12/30	123		Success
4000000000001018	12/30	123		Fail
4012888888881881	01/30	029	Success: 123456 Fail: 111111	3DS Payment
5111111111111118	01/30	029	Success: 123456 Fail: 111111	3DS Payment
4141414141414141	12/30	123	Success: 123456 Fail: 111111	3DS Payment

Test Cards with merchant ID start with 14xxxxxxx

Card Number	Expire	CVV	Password	
4111111111111111	01/30	029		Success
5555555555554444	01/30	029		Success
4012888888881881	01/30	029	Success: 123456 Fail: 111111	3DS Payment
5111111111111118	01/30	029	Success: 123456 Fail: 111111	3DS Payment
4141414141414141	12/30	123	Success: 123456 Fail: 111111	3DS Payment

APPENDIX

Payment Methods

NAME	DESCRIPTION
card	Credit or Debit cards

Status Codes

CODE	DESCRIPTION
1	incomplete
2	pending
3	error
4	failed
5	under_review
6	successful
7	released
8	refund_initiated
9	refund_failed
10	refund_under_review
11	refund_successful
12	refund_partially_failed
13	refund_partially_successful
14	canceled
15	rejected
16	expired
17	documents_uploaded
18	void_initiated
19	void_under_review
20	void_successful
21	void_failed
22	void_partially_successful
23	void_partially_failed

Country Codes

Code	Country name (using title case)
AD	Andorra
AE	United Arab Emirates
AF	Afghanistan
AG	Antigua and Barbuda
AI	Anguilla
AL	Albania
AM	Armenia
AO	Angola
AQ	Antarctica
AR	Argentina
AS	American Samoa
AT	Austria
AU	Australia
AW	Aruba
AX	Åland Islands
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BD	Bangladesh
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BH	Bahrain
BI	Burundi
BJ	Benin
BL	Saint Barthélemy
BM	Bermuda
BN	Brunei Darussalam
BO	Bolivia, Plurinational State of
BQ	Bonaire, Sint Eustatius and Saba
BR	Brazil
BS	Bahamas
BT	Bhutan

Code	Country name (using title case)
BV	Bouvet Island
BW	Botswana
BY	Belarus
BZ	Belize
CA	Canada
CC	Cocos (Keeling) Islands
CD	Congo, Democratic Republic of the
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CK	Cook Islands
CL	Chile
CM	Cameroon
CN	China
CO	Colombia
CR	Costa Rica
CU	Cuba
CV	Cabo Verde
CW	Curaçao
CX	Christmas Island
CY	Cyprus
CZ	Czechia
DE	Germany
DJ	Djibouti
DK	Denmark
DM	Dominica
DO	Dominican Republic
DZ	Algeria
EC	Ecuador
EE	Estonia
EG	Egypt
EH	Western Sahara
ER	Eritrea
ES	Spain

Code	Country name (using title case)
ET	Ethiopia
FI	Finland
FJ	Fiji
FK	Falkland Islands (Malvinas)
FM	Micronesia, Federated States of
FO	Faroe Islands
FR	France
GA	Gabon
GB	United Kingdom of Great Britain and Northern Ireland
GD	Grenada
GE	Georgia
GF	French Guiana
GG	Guernsey
GH	Ghana
GI	Gibraltar
GL	Greenland
GM	Gambia
GN	Guinea
GP	Guadeloupe
GQ	Equatorial Guinea
GR	Greece
GS	South Georgia and the South Sandwich Islands
GT	Guatemala
GU	Guam
GW	Guinea-Bissau
GY	Guyana
HK	Hong Kong
HM	Heard Island and McDonald Islands
HN	Honduras
HR	Croatia
HT	Haiti
HU	Hungary
ID	Indonesia
IE	Ireland
IL	Israel

Code	Country name (using title case)
IM	Isle of Man
IN	India
IO	British Indian Ocean Territory
IQ	Iraq
IR	Iran, Islamic Republic of
IS	Iceland
IT	Italy
JE	Jersey
JM	Jamaica
JO	Jordan
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KH	Cambodia
KI	Kiribati
KM	Comoros
KN	Saint Kitts and Nevis
KP	Korea, Democratic People's Republic of
KR	Korea, Republic of
KW	Kuwait
KY	Cayman Islands
KZ	Kazakhstan
LA	Lao People's Democratic Republic
LB	Lebanon
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
LY	Libya
MA	Morocco
MC	Monaco

Code	Country name (using title case)
MD	Moldova, Republic of
ME	Montenegro
MF	Saint Martin (French part)
MG	Madagascar
MH	Marshall Islands
MK	North Macedonia
ML	Mali
MM	Myanmar
MN	Mongolia
MO	Macao
MP	Northern Mariana Islands
MQ	Martinique
MR	Mauritania
MS	Montserrat
MT	Malta
MU	Mauritius
MV	Maldives
MW	Malawi
MX	Mexico
MY	Malaysia
MZ	Mozambique
NA	Namibia
NC	New Caledonia
NE	Niger
NF	Norfolk Island
NG	Nigeria
NI	Nicaragua
NL	Netherlands, Kingdom of the
NO	Norway
NP	Nepal
NR	Nauru
NU	Niue
NZ	New Zealand
OM	Oman
PA	Panama

Code	Country name (using title case)
PE	Peru
PF	French Polynesia
PG	Papua New Guinea
PH	Philippines
PK	Pakistan
PL	Poland
PM	Saint Pierre and Miquelon
PN	Pitcairn
PR	Puerto Rico
PS	Palestine, State of
PT	Portugal
PW	Palau
PY	Paraguay
QA	Qatar
RE	Réunion
RO	Romania
RS	Serbia
RU	Russian Federation
RW	Rwanda
SA	Saudi Arabia
SB	Solomon Islands
SC	Seychelles
SD	Sudan
SE	Sweden
SG	Singapore
SH	Saint Helena, Ascension and Tristan da Cunha
SI	Slovenia
SJ	Svalbard and Jan Mayen
SK	Slovakia
SL	Sierra Leone
SM	San Marino
SN	Senegal
SO	Somalia
SR	Suriname
SS	South Sudan

Code	Country name (using title case)
ST	Sao Tome and Principe
SV	El Salvador
SX	Sint Maarten (Dutch part)
SY	Syrian Arab Republic
SZ	Eswatini
TC	Turks and Caicos Islands
TD	Chad
TF	French Southern Territories
TG	Togo
TH	Thailand
TJ	Tajikistan
TK	Tokelau
TL	Timor-Leste
TM	Turkmenistan
TN	Tunisia
TO	Tonga
TR	Türkiye
TT	Trinidad and Tobago
TV	Tuvalu
TW	Taiwan, Province of China
TZ	Tanzania, United Republic of
UA	Ukraine
UG	Uganda
UM	United States Minor Outlying Islands
US	United States of America
UY	Uruguay
UZ	Uzbekistan
VA	Holy See
VC	Saint Vincent and the Grenadines
VE	Venezuela, Bolivarian Republic of
VG	Virgin Islands (British)
VI	Virgin Islands (U.S.)
VN	Viet Nam
VU	Vanuatu
WF	Wallis and Futuna

Code	Country name (using title case)
WS	Samoa
YE	Yemen
YT	Mayotte
ZA	South Africa
ZM	Zambia
ZW	Zimbabwe

Card Type

No.	Card Type	Code
1	VISA	visa
2	MASTER CARD	mastercard
3	AMERICAN EXPRESS	amex
4	JCB	jcb
5	MAESTRO	maestro
6	DISCOVER	discover
7	UNION PAY	union-pay
8	DINERS	diners

Code example

PHP

```
<?php
function generateSignature(array $data): string
{
   $privateKey = openssl_pkey_get_private("-----BEGIN PRIVATE KEY-----
MIIJQQIBADANBgkqhkiG9w0BAQEFAASCCSswggknAgEAAoICAQDwgPXMA8tv7boG
...
8S7wZ3XcsaLzlzMXrxhykwsDFBFg
-----END PRIVATE KEY-----
");
   ksort($data, SORT_NATURAL);
   array_walk_recursive(
      $data,
      static function(&$field) {
         $field = trim($field);
      }
   );
   openssl_sign(json_encode($data), $signature,
      $privateKey, 'md5WithRSAEncryption'
   );
   return base64_encode($signature);
}

function verifySignature(array $data): bool
{
   $publicKey = openssl_pkey_get_public("-----BEGIN PUBLIC KEY-----
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAxu+5iLOm7KCXoGxUMHX5
...
PR1ofGsEO8/N/qJuggQc2P8CAwEAAQ==
-----END PUBLIC KEY-----
");
   $dataWithoutSignature = array_filter($data, static function ($key) {
      return $key !== 'signature';
   }, ARRAY_FILTER_USE_KEY);
   $signature = $data['signature'];
   ksort($dataWithoutSignature, SORT_NATURAL);
   array_walk_recursive(
      $dataWithoutSignature,
      static function(&$field) {
         $field = trim($field);
      }
   );
   $result = openssl_verify(json_encode($dataWithoutSignature), base64_decode($signature), $publicKey,
'md5WithRSAEncryption');
   return $result === 1;
}
```

NodeJS

```
function generateSignature(data) {
   const crypto = require('crypto');
   const privateKey = `-----BEGIN PRIVATE KEY-----
MIIJQQIBADANBgkqhkiG9w0BAQEFAASCCSswggknAgEAAoICAQDwgPXMA8tv7boG
...
8S7wZ3XcsaLzlzMXrxhykwsDFBFg
-----END PRIVATE KEY-----
`;
   const clonedData = JSON.parse(JSON.stringify(data));

   Object.keys(clonedData).forEach(key => {
      if (typeof clonedData[key] === 'number') {
         clonedData[key] = String(clonedData[key]);
      }
   });

   const sortedData = {};
   Object.keys(clonedData)
      .sort((a, b) => a.localeCompare(b, undefined, { numeric: true, sensitivity: 'base' }))
      .forEach(key => {
         sortedData[key] = clonedData[key];
      });

   const trimRecursive = (obj) => {
      if (typeof obj !== 'object' || obj === null) {
         return typeof obj === 'string' ? obj.trim() : obj;
      }

      if (Array.isArray(obj)) {
         return obj.map(item => trimRecursive(item));
      }

      const result = {};
      for (const key in obj) {
         if (Object.prototype.hasOwnProperty.call(obj, key)) {
            result[key] = trimRecursive(obj[key]);
         }
      }
      return result;
   };

   const trimmedData = trimRecursive(sortedData);

   const jsonData = JSON.stringify(trimmedData).replace(/\//g, '\\/');

   const sign = crypto.createSign('md5WithRSAEncryption');
   sign.update(jsonData);
   const signature = sign.sign(privateKey, 'base64');

   return signature;
}
```