Paysafe

Integration Guide

Bank Payouts

Table of Contents

1. About

The Paysafe payout service enables merchants to make third-party payouts to individuals and corporation using their integrated and prefunded Paysafe Digital Wallet. This document details the payout options available through Paysafe and offers instructions on integrating them.

The available payout options are:

- **Unlinked Payouts:** Merchants can send third-party payouts to individuals and corporations via bank transfer, without any connection to an original transaction or initial deposit.
- **Linked Payouts:** Merchants can instruct bank transfers to customers who have made an initial QCO deposit to the merchant.

Payout formats vary by country and currency. See the format guide for details here.

1.1 At a glance

- Streamline payments by combining both pay-ins and payouts with a single provider.
- Track sales and cash flow easily with our merchant portal's clear, unified financial view.
- Save on currency exchange fees by using local payouts in EEA countries.
- Enjoy faster payouts through Paysafe's local banking networks, reducing the need for multiple providers.

Supported countries and currencies

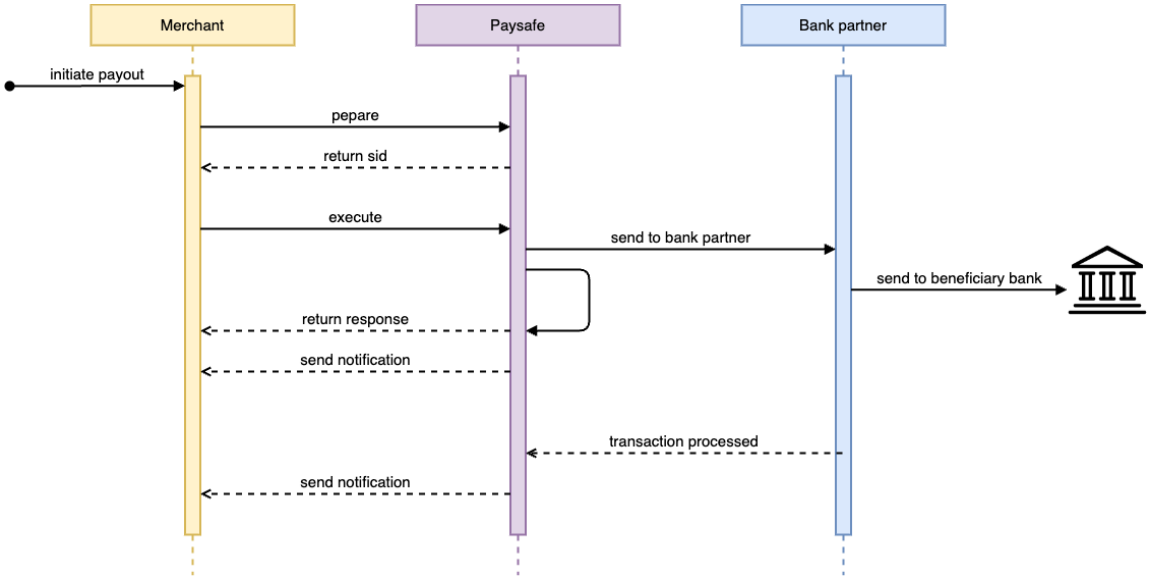
Region	Country	Currency	Payment Rail
EU	Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Liechtenstein, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden	EUR	SEPA Instant where available, if not then payouts will be processed via SEPA

2. Prerequisites

- Integration via Skrill Payout API
- An active Digital Wallet with Paysafe
- Enable Bank Payouts by contacting Merchant Support
- Sufficient balance in your Skrill Digital Wallet
- Whitelisted IP Address

3. Integration options

- Skrill Payouts API integration



Best Practice

- Ensure you meet all technical prerequisites before making payouts, including:
  - Setting up the status_url to receive status notifications.
  - Contacting merchant services to whitelist your preferred IP address for API calls.
- Check your wallet balance to confirm sufficient funds and top up if necessary.
- Provide accurate beneficiary details, such as names and addresses, to ensure smooth transaction processing.

4. Unlinked bank payouts

4.1 Unlinked Payout Execution

Unlinked Payouts with Paysafe are processed in two steps as follows:

- Preparation request
- Execution request

4.1.1 Preparation request

The preparation request or session initiation call validates several payment aspects before processing it. Key validations include authorization and authentication from the merchant initiating the payment and the accuracy of the payment parameters. Note that different countries or regions may require different parameters, and the preparation call checks those criteria.

URL: POST https://pay.skrill.com/json

4.1.2 Preparation Request parameters

Name	Required	Format	Description
action	Yes	enumerated value [PAYOUT]	The action of the request. Currently, only **PAYOUT** is supported.
instrument_type	Yes	enumerated value [BANK]	Type of payout payment. Currently, only **BANK** is supported.
prepare_only	Yes	number, enumerated value [0, 1]	The only supported value is [**1**]; [0] is reserved for future use.
amount	Yes	floating point numeric	Requested amount to be transferred (e.g. 20 / 20.50).
currency	Yes	3-letter currency code	3-letter code of the currency of the amount according to ISO 4217.
transaction_id	Yes	string (max length 100)	Unique reference or identification number for the payout in the merchant system. (it must be unique for each payment).
payment_description	Yes	string (max length 200)	Purpose of the payout.
pay_from_email	Yes	email	The processing email of your merchant account
merchant_id	No	numeric	Unique ID of your Skrill account.

merchant_client_id	Yes	string (max length 512)	Unique ID which you (the merchant) use to identify the individual or business in your system (i.e. customer id). If you are sending to a payee without an account enter “000000000”
merchant_client_registration_date	No	string (ISO 8601 Date format)	Specify the date the consumer registered their account or first interacted with your merchant system. Format: ISO 8601 (with time zone), e.g., YYYY-MM-DD or YYYY-MM-DDTHH:mm.sss.
merchant_client_kyc_level	No	number, enumerated value [0, 1]	Indicate whether the merchant has done KYC or KYB using legal documents (e.g. passport, driving license). 0 – not verified; 1 – verified with documents
merchant_client_registration_country	No	string (3-letter ISO-3166 code)	The country where you registered the payee.
firstname	Yes	string (max length 50)	For payouts to Individuals: Payee's first name For payouts to corporations: Company Name
lastname	Yes	string (max length 50)	For payouts to Individuals: Payee's last name For payouts to corporations: The rest of the company name including the entity legal form code (e.g. Ltd.)
date_of_birth	No	string (ddMMyyyy format)	Example: 25011999 (Day-Month-Year with no separators) Not relevant when sending to a corporate account.
phone_number	No	string (only numeric characters, max length 50)	Payee's phone number
country	Yes	string (3-letter ISO-3166 code)	Payee's country 3-letter ISO-3166 code (e.g. GBR)
state	No	string (max length 50)	Payee's country states for countries that have states or regions (e.g. Central London)
city	No	string (max length 50)	Payee's city (e.g. London)
address	Yes	string (max length 800)	Payee's address
postal_code	No	string (max length 30)	Payee's postal (zip) code
recipient_type	Yes	enumerated value [INDIVIDUAL, CORPORATE]	Parameter denoting whether the payment recipient is an "INDIVIDUAL" or "CORPORATE".

sign	Yes	string (max length depends on chosen hashing algorithm)	Security checksum ensures that the sender of the request is the merchant himself. Instructions for signature calculation
pay_to_email	Yes	email (max length 100)	Payee's email address
account_num	No/Yes*	string (max length 50)	Payee's bank account number (UK payouts reserved field)
bank_code	No/Yes*	string (max length depends on country)	Payee's bank code is required for some countries: - UK – the sort code of the recipient bank
iban	No/Yes*	string (max length 34)	Payee's iban
swift	No/Yes*	string (max length 11)	Payee's bank swift code
instrument_token	No/Yes**	UUID string (exactly 36)	A UUID representing the payment instrument token id, which can be used for making subsequent payments without the need to supply payment instrument, payee, or merchant client fields
instrument_country	Yes	string (3-letter ISO-3166 code)	3-letter ISO-3166 code of the country in which the bank account of the payee is located (e.g. GBR)
status_url	No	string (max length 400)	URL to which the transaction details are posted after the payout status update. Alternatively, payout status can be acquired by a separate call to the Skrill Merchant Query Interface
status_url2	No	string (max length 400)	Secondary URL to which the transaction details are posted after the payout status update.
merchant_fields	No	string (max length 240)	A comma-separated list of field names is passed back to your status URL when a status notification is sent.

Note: ** Mandatory Bank Fields depending on the Bank Instrument country.*
*** Required when using a payment instrument token to make a subsequent payout with fewer fields.*

Example Preparation Request:

POST https://pay.skrill.com/json
Content-Type: application/json


```
{
   "action": "payout",
   "instrument_type": "BANK",
   "prepare_only": 1,
   "amount": "20.45",
   "currency": "GBP",
   "pay_from_email": "merchant_email@mail.com",
   "pay_to_email": "payee1@mail.com",
   "merchant_id": "1234567890",
   "transaction_id": "merchant_ref_id",
   "payment_description": "merchant payment description",
   "account_num": "9876543",
   "bank_code": "200052",
   "instrument_country": "GBR",
   "firstname": "fname",
   "lastname": "lname",
   "address": "address",
   "country": "GBR",
   "recipient_type": "INDIVIDUAL",
   "merchant_client_id": "1241",
   "merchant_client_kyc_level":"1",
   "merchant_client_registration_country": "GBR",
   "merchant_client_registration_date": "2016-08-22T14:30+02:00",
   "status_url": "https://your.webhook.site",
   "sign": "ad34df771d38ba82c4f271115675a6c1ffa5642527a50045b8614f57e186f813"
}
```

Example Preparation Request with Instrument Token:

POST https://pay.skrill.com/json
Content-Type: application/json


```
{
   "action": "payout",
   "instrument_type": "BANK",
   "prepare_only": 1,
   "amount": "20.45",
   "currency": "GBP",
   "pay_from_email": "merchant_email@mail.com",
   "pay_to_email": "payee1@mail.com",
   "merchant_id": "1234567890",
   "transaction_id": "merchant_ref_id",
   "payment_description": "merchant payment description",
   "status_url": "https://your.webhook.site",
   "instrument_token": "fc71683d-efa6-4ffc-a81d-2b7eaee38f75",
   "sign": "ad34df771d38ba82c4f271115675a6c1ffa5642527a50045b8614f57e186f813"
}
```


4.1.3 Preparation Response

Successful: 200 OK
Content-Type: text/html

Format	Description
string	Unique payment session identifier used to execute the payment

Error: 4XX error code
Content-Type: application/json

Name	Format	Description
code	string	General purpose error code in which area the problem originated
message	string	Descriptive message showing in more detail the problem
parameter	string (optional)	Parameter that failed validation

Example Preparation Response (Success):

200 OK
Content-Type: text/html

c56acb417b2e22563424373cf58d0137

Example Preparation Response (Error):

```
400 Bad Request
Content-Type: application/json

{
   "code": "MISSING_PARAMETER",
   "parameter": "iban",
   "message": "Missing iban parameter."
}
```

4.1.4 Preparation Response error messages

Code	Error message	Description
MISSING_PARAMETER	Missing <parameter_name> parameter.	Missing required a payout field parameter. Example: Missing instrument_type parameter.
INVALID_PARAMETER	<parameter_name> is invalid.	Invalid parameter Example: amount is invalid.
NOT_MATCHING_PAYMENT_INSTRUMENT_COUNTRY	Not matching payment instrument country.	Instrument country doesn't match the country from IBAN
SWIFT_NOT_MATCH_WITH_IBAN	swift does not match bank associated with IBAN	SWIFT code does not match bank associated with IBAN
UNSUPPORTED_BANK	Unsupported bank	Used payment instruments are not supported for payouts
INVALID_PARAMETER	<parameter_name> field with length=<parameter_length> exceeds the <parameter_max_length> character limit	The parameter exceeds the specified length.
INVALID_PARAMETER	<parameter_name>=<parameter_value> has invalid format	The parameter is in invalid format. Example: bank_code=00243 has invalid format

INVALID_PARAMETER	<parameter_name> field {<parameter_value>} does not match the possible values	document_type or account_type fields do not match the possible values Example: document_type field {Ced} does not match the possible values
INVALID_PARAMETER	account_type=<parameter_value> does not match the required fields for country <instrument_country_value>	Passed account_type or document_type is not supported for the given instrument country. Example: account_type=EVP does not match the required fields for country ECU
INVALID_PARAMETER	<parameter_name> has invalid format for type: <account_type/document_type value>	The parameter has an invalid format for the specified account_type/document_type.
INVALID_PARAMETER	account_type=<account_type value> requires Document type RUT for country <instrument_country value>	When the provided bank_code is for "Banco Estado" and the bank_account_type is Rut, the provided document_type must be RUT.
ALREADY_PAID	Already paid	When a given payment, identified by the parameter transaction_id was already completed, and a second attempt is made with the already paid transaction_id.
COUNTRY_NOT_SUPPORTED_FOR_PAYOUTS	Payout to requested country is not supported	When the payout attempt is for a payment instrument from a country not supported by the system.
BAD_REQUEST	IP not allowed	When the request is from a non-authorized IP address.
BAD_REQUEST	The initial payout transaction is not confirmed	The used token belongs to payout that is not confirmed/settled.

GENERAL_ERROR	General Error	Error occurred during session initiation.
GENERAL_ERROR	Token does not belong to the merchant	This token is associated with a different merchant and cannot be used for your transactions to ensure security and proper authorization.
GENERAL_ERROR	Token does not belong to the payee	This token is linked to a specific beneficiary and cannot be used to make payments to a different beneficiary.
GENERAL_ERROR	The initial payout transaction does not belong to the merchant	The initial payout transaction associated with this token does not belong to the given merchant, ensuring transactions are tied to the correct merchant.

***Note**: If errors persist, please contact your dedicated relationship manager or merchant services.*

4.1.5 Execution Request

To execute the payment with the data provided in the session initiation call, send a POST request using the returned sid parameter and the sign parameter from the session initiation

URL: POST https://pay.skrill.com/payout

4.1.6 Execution Request Parameters

Name	Required	Format	Description
sid	Yes	string	Session identifier returned from Session Initiation call.
sign	Yes	string	Security check sum ensuring the sender of the request is the merchant himself

Example Execution Request:

```
POST https://payout.skrill.com/json
Content-Type: application/json

{
   "sid": "c56acb417b2e22563424373cf58d0137",
   "sign": "AD34DF771D38BA82C4F271115675A6C1FFA5642527A50045B8614F57E186F813"
}
```

4.1.7 Execution Response Parameters

Successful: 200 OK
Content-Type: application/json

Name	Format	Description
id	number	Payment identification number
transactionId	number	Transaction identification number
status	string, enumerated	Status of the executed payment: **PENDING, SCHEDULED, PROCESSED, FAILED**
metadata	object	Additional details of the executed payment

Metadata:

Name	Format	Description
amount	floating point numeric	Actual payment amount deducted from merchant account (excluding fees).
currency	3-letter currency code	Currency of the deducted amount according to ISO 4217.
instrument	string	Identification number of the payment instrument which can be used for future payments without needing to specify bank details again.
token	string	A UUID representing the payment instrument token id, which can be used for subsequent payments without the need to provide payment instrument, payee, or merchant client fields.

Error: 4XX error code
Content-Type: application/json

Name	Format	Description
code	number	Numeric error code of the problem
message	string	Brief description of the problem
description	string	Detailed description of the problem

Example Execution Response (Success):


```
{
   "id": "5024699328",
   "transactionId": "5086059195",
   "status": "SCHEDULED",
   "metadata": {
      "amount": "10.225838",
      "currency": "EUR",
      "instrument": "BANKWIRE"
   }
}
```


Example Execution Response (Error):


```
{
   "error": {
      "code": "7007",
      "message": "Limits not met",
      "details": [
         "5dfdc472-d4e4-4c50-b5ca-1ca0ead67666",
         "Limits not met.",
         "Limits not met"
      ]
   }
}
```


4.1.8 Execution response error messages

Code	Message	Description
7001	Invalid request.	The request provided does not match the required format.
7002	Invalid signature.	The provided request signature does not match the required format.
7003	Session expired.	The session has expired.
7004	Invalid session data.	Invalid session data.
7005	Insufficient balance	The balance is insufficient to create the payout.
7006	Account restricted	The account is restricted.
7007	Limits not met	Limits not met.
7008	Per transaction limit exceeded	Per transaction limit exceeded.
7009	Daily limit exceeded	Daily limit exceeded.
7010	Weekly limit exceeded	Weekly limit exceeded.

7011	Monthly limit exceeded	Monthly limit exceeded.
7012	Country blocked for payouts.	The bank account's country is blocked for payouts.
7013	Payout to this country is not enabled for your account.	Payout to this country is not enabled for your account.
7014	Account not found.	Accounts could not be found.
7015	Recipient country blocked for payouts.	Recipient country blocked for payouts.
7017	Recipient country is blocked for gaming.	The recipient's country is blocked for gaming.
7018	Recipient Instrument country is blocked for gaming.	The recipient's instrument country is blocked for gaming.

4.2 Mandatory Bank Fields Per Country

Mandatory Bank Fields	Countries
iban swift	Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Liechtenstein, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden
account_num bank_code	UK

5. Linked bank payouts

Linked (Quick Checkout) payouts allow Skrill merchants to transfer funds (such as winnings) to customers by referencing the transaction ID of the customer's initial payment. The funds are sent directly to the payment instrument used for the original transaction.

Restrictions:

Linked payouts are available only for:

- Rapid Transfer payments
- NETELLER payments
- Paysafecard payments
- Sofort payments
- EPS payments
- MyBank payments
- Bank Payouts (EMEA, NA, LATAM)
- Giropay payments

Merchants who wish to use Linked payouts must have this functionality enabled by the Merchant Services team.

Linked payouts use the same URL as Unlinked Payouts and are implemented by sending POST requests to: https://pay.skrill.com/json

The process consists of two steps:

- Send a payout preparation request to initiate a session.
- Execute the Linked payout within the session.

Skrill returns a JSON response after each step, containing the result of the performed action.

5.1 Linked Payout execution

Linked Payouts with Paysafe are processed in two steps as follows:

- Preparation request
- Execution request

5.1.1 Preparation request

You must include the parameters described below in your POST request.

5.1.2 Preparation Request parameters

Name	Required	Format	Description
action	Yes	enumerated value [PAYOUT]	The action of the request. Currently only **PAYOUT** is supported.
instrument_type	Yes	enumerated value [BANK]	Type of payout payment. Currently only **BANK** is supported.
prepare_only	Yes	number, enumerated value [0, 1]	Currently the only supported value is [1]. [0] is reserved for future use.
amount	Yes	floating point numeric	Requested amount to be transferred (e.g. 20 / 20.50).
currency	Yes	3-letter currency code	3-letter code of the currency of the amount according to ISO 4217.
transaction_id	Yes	string (max length 100)	Unique reference or identification number for the payout in the merchant system. (Must be unique for each payment).
payment_description	Yes	string (max length 200)	Purpose of the payout,
pay_from_email	Yes	email	The processing email of your merchant account
merchant_id	No	numeric	Unique ID of your Skrill account.

payin_transaction_id	Yes	string (max length 100)	The Skrill transaction ID of the original payment. OR the Merchant's transaction ID of the original payment.
merchant_client_id	No/Yes**	string (max length 512)	Unique ID which you (the merchant) use to identify the client in your system (i.e., customer id).
merchant_client_registration_date	No/Yes**	string (ISO 8601 Date format)	Date when you registered the payee account in your system-ISO 8601 (with time zone). Examples: YYYY-MM-DD or YYYY-MM-DDTHH:mm:ss.sss
merchant_client_kyc_level	No/Yes**	number, enumerated value [0, 1]	KYC status of the payee, determined by a KYC verification from your system: 0 – not verified; 1 – verified with documents
merchant_client_registration_country	No/Yes**	string (3-letter ISO-3166 code)	The country the user has been registered with in the merchant's system.
date_of_birth	No/Yes*	string (**ddMMyyyy** format)	Example: 25011999 (Day-Month-Year with no separators)
sign	Yes	string (max length depends on chosen hashing algorithm)	Security check sum ensures that the sender of the request is the merchant himself. Instructions for signature calculation
account_num	No/Yes*	string (max length 50)	Payee's bank account number (UK payouts reserved field)
account_type	No/Yes*	string, enumerated (refer to link)	Type of the beneficiary bank account (LATAM payouts reserved field; example for Brazil - CPF)
bank_code	No/Yes***	string (max length depends on country)	Payee's bank code required for some countries. Required for Bank Payouts to Chile and Ecuador
status_url	No	string (max length 400)	URL to which the transaction details are posted after the payout status update. Alternatively, payout status can be acquired by a separate call to the Skrill Merchant Query Interface
merchant_fields	No	string (max length 240)	A comma-separated list of field names passed back to your status URL when status notification is sent.

document_type	No/Yes*	string, enumerated (see Document Type accepted by Skrill table, column "Code")	Type of the ID document provided (LATAM payouts reserved field)
document_number	No/Yes*	string (max length 50)	The identification number of the document (LATAM payouts reserved field)

Note: *Required for Bank Payouts to LATAM
**Applicable only for some payment options. Merchant Client fields are used for additional risk screening.
***Required for Bank Payouts to Chile and Ecuador.

Example Preparation Request:

POST https://pay.skrill.com/json
Content-Type: application/json

```
{
   "action": "payout",
   "instrument_type": "BANK",
   "prepare_only": 1,
   "amount": "20.45",
   "currency": "EUR",
   "pay_from_email": "merchant_email@mail.com",
   "merchant_id": "1234567890",
   "payin_transaction_id ": "129876543"
   "transaction_id": "merchant_payout_ref_id",
   "payment_description": "merchant payment description",
   "merchant_client_id": "1241",
   "merchant_client_kyc_level":"1",
   "merchant_client_registration_country": "GER",
   "merchant_client_registration_date": "2016-08-22T14:30+02:00",
   "status_url": "https://your.webhook.site",
   "sign": "ad34df771d38ba82c4f271115675a6c1ffa5642527a50045b8614f57e186f813"
}
```

Example Preparation Request for LATAM:

```
POST https://pay.skrill.com/json
Content-Type: application/json

{
   "action": "payout",
   "instrument_type": "BANK",
   "prepare_only": 1,
   "amount": "10.45",
   "currency": "USD",
   "pay_from_email": "merchant_email@mail.com",
   "merchant_id": "1234567890",
   "transaction_id": "merchant_payout_ref_id",
   "payin_transaction_id ": "129876543",
   "payment_description": "merchant payment description",
   "merchant_client_id": "1241",
   "merchant_client_kyc_level":"1",
   "merchant_client_registration_country": "ECU",
   "merchant_client_registration_date": "2016-08-22T14:30+02:00",
   "status_url": "https://your.webhook.site",
   "account_type": "Checking",
   "account_num": "12345678",
   "document_type": "RUC",
   "document_number": "1234567A",
   "bank_code":"0024",
   "date_of_birth": "01011994"
   "sign":
"ad34df771d38ba82c4f271115675a6c1ffa5642527a50045b8614f57e186f813",
}
```

5.1.3 Preparation Response

Successful: 200 OK
Content-Type: text/html

Format	Description
string	Unique payment session identifier used to execute the payout

Error: 4XX error code
Content-Type: application/json

Name	Format	Description
code	string	General purpose error code in which area the problem originated
message	string	Descriptive message showing in more detail the problem
parameter	string (optional)	Parameter which failed validation

Example Preparation Response (Success):

```
200 OK
Content-Type: text/html

c56acb417b2e22563424373cf58d0137
```

Example Preparation Response (Error):

```
400 Bad Request
Content-Type: application/json

{
   "code": "MISSING_PARAMETER",
   "parameter": "amount",
   "message": "Missing amount parameter."
}
```

5.1.4 Preparation response error messages

Code	Error message	Description
MISSING_PARAMETER	Missing <parameter_name> parameter.	Missing required a payout field parameter. Example: Missing instrument_type parameter.
INVALID_PARAMETER	<parameter_name> is invalid.	Invalid parameter Example: amount is invalid.
INVALID_PARAMETER	<parameter_name> field with length=<parameter_length> exceeds the <parameter_max_length> character limit	The parameter exceeds the specified length.
INVALID_PARAMETER	<parameter_name>=<parameter_value> has invalid format	The parameter is in an invalid format. Example: bank_code=00243 has invalid format
INVALID_PARAMETER	<parameter_name> field {<parameter_value>} does not match the possible values	document_type or account_type fields do not match the possible values Example: document_type field

		{Ced} does not match the possible values
INVALID_PARAMETER	account_type=<parameter_value> does not match the required fields for country <instrument_country_value>	Passed account_type or document_type is not supported for the given instrument country. Example: account_type=EVP does not match the required fields for country ECU
INVALID_PARAMETER	<parameter_name> has invalid format for type: <account_type/document_type value>	The parameter has an invalid format for the specified account_type/document_type.
INVALID_PARAMETER	account_type=<account_type value> requires Document type RUT for country <instrument_country value>	When the provided bank_code is for "Banco Estado" and the bank_account_type is Rut, the provided document_type must be RUT.
ALREADY_PAID	Already paid	When a given payment, identified by the parameter transaction_id was already completed, and a second attempt is made with the already paid transaction_id.
ORIGINAL_PAYMENT_UNSUPPORTED_FOR_PAYOUT	Original payment is not supported for payout	When payin transaction is not supported for payout.
COUNTRY_NOT_SUPPORTED_FOR_PAYOUTS	Payout to a requested country is not supported	When a requested payout is to not be supported country.
BAD_REQUEST	IP not allowed	When the request is from a non-whitelisted IP address,
GENERAL_ERROR	General Error	Error occurred during session initiation.

5.1.5 Execution Request

To execute the payout with the data provided in the initiation call, execute a POST request using the returned sid parameter, as well as the sign parameter from the session initiation.

URL: POST https://pay.skrill.com/payout

5.1.6 Execution Request parameters

Name	Required	Format	Description
sid	Yes	string	Session identifier returned from Session Initiation call.
sign	Yes	string	Security check sum ensuring the sender of the request is the merchant himself

Example Execution Request:

```
POST https://payout.skrill.com/json
Content-Type: application/json

{
   "sid": "c56acb417b2e22563424373cf58d0137",
   "sign": "AD34DF771D38BA82C4F271115675A6C1FFA5642527A50045B8614F57E186F813"
}
```

5.1.7 Execution Response Parameters

Successful: 200 OK
Content-Type: application/json

Name	Format	Description
id	number	Payment identification number
transactionId	number	Transaction identification number
status	string, enumerated	Status of the executed payment: **PENDING, SCHEDULED, PROCESSED, FAILED**
metadata	object	Additional details of the executed payment

Metadata contents:

Name	Format	Description
amount	floating point numeric	Actual payment amount deducted from merchant account (excluding fees)
currency	3-letter currency code	The currency of the deducted amount, as per ISO 4217.
instrument	string	Identification number of the payment instrument which can be used for future payments without needing to provide bank details again.

Error: 4XX error code
Content-Type: application/json

Name	Format	Description
code	number	Numeric error code of the problem
message	string	Brief description of the problem
description	string	Detailed description of the problem

Example Execution Response (Success):

```
Content-Type: application/json
{
   "id": "5024699328",
   "transactionId": "5086059195",
   "status": "SCHEDULED",
   "metadata": {
      "amount": "10.225838",
      "currency": "EUR",
      "instrument": "BANKWIRE"
   }
}
```

Example Response (Error):

```
Content-Type: application/json
{
   "error": {
      "code": "7007",
      "message": "Limits not met",
      "details": [
         "5dfdc472-d4e4-4c50-b5ca-1ca0ead67666",
         "Limits not met.",
         "Limits not met"
      ]
   }
}
```

5.1.8 Execution response error messages

Code	Message	Description
7001	Invalid request.	The request provided does not match the required format.
7002	Invalid signature.	The provided request signature does not match the required format.
7003	Session expired.	The session has expired.
7004	Invalid session data.	Invalid session data.
7005	Insufficient balance	The balance is insufficient to create the payout.
7006	Account restricted	The account is restricted.
7007	Limits not met	Limits not met.
7008	Per transaction limit exceeded	Per transaction limit exceeded.
7009	Daily limit exceeded	Daily limit exceeded.
7010	Weekly limit exceeded	Weekly limit exceeded.
7011	Monthly limit exceeded	Monthly limit exceeded.
7012	Country blocked for payouts.	The bank account's country is blocked for payouts.
7013	Payout to this country is not enabled for your account.	Payout to this country is not enabled for your account.
7014	Account not found.	Accounts could not be found.
7015	Recipient country blocked for payouts.	Recipient country blocked for payouts.
7017	Recipient country is blocked for gaming.	The recipient's country is blocked for gaming.
7018	Recipient Instrument country is blocked for gaming.	The recipient's instrument country is blocked for gaming.
7019	Original payment unsupported for payout.	During the Linked Payout flow, the original payment provided is not supported.

7020	Error while getting original deposit.	During the Linked Payout flow, an error while getting the original payment has occurred.
7021	Permission to withdraw denied.	During the Linked Payout flow, the permission to withdrawal checks have failed.

6. Payout Status Notification

Once the payout process is complete, Skrill sends the transaction details to the status_url you provided in your request via a standard HTTP POST. The Skrill server continues to POST the status until it receives an HTTP OK (200) response from your server, or the number of posts exceeds 10.

The table below lists the parameters sent to your status_url page:

Status URL Parameters

Field name	Description	Required	Example value
pay_from_email	Your email address.	Yes	info@merchant.com
pay_to_email	Payee's email address.	Yes	payer123@skrill.com
merchant_id	Unique ID of your Skrill account. (only needed for the calculation of the MD5 signature).	Yes	100005
transaction_id	A unique reference or identification number provided by you in your request.	No [1]	A205220
mb_transaction_id	Skrill's internal unique reference ID for this transaction	Yes	170032056
mb_amount	The total amount of the payout in the processing currency. *Note: If the currency of your Digital Wallet is different, currency conversion may apply.*	Yes	25.46 / 25.4 / 25
mb_currency	Currency of ***mb_amount***.	Yes	GBP
status	Status of the transaction: **-2** failed **2** processed **0** pending **1** scheduled **-1** cancelled **-4** Payout returned by recipient's bank	Yes	2
failed_reason_code	If the payout is ***-2*** (failed), this field will contain a code detailing the failure reason. Failed Reason Codes Table. Note: to receive this code you must contact support to enable this feature.	No [2]	99

md5sig	MD5 signature Please see instructions on how to calculate	Yes	327638C253A4637199CEBA6642371F20
sha2sig	Upper-case Sha2 signature. This is constructed in the same way as the MD5 signature, but with a different hashing algorithm. This parameter is not available by default. To enable this option, send a request to Merchant Service.	No	dbb7101322257a311f08d1c527053058fc7e464e30bcfb4613f09053c22dd1f8
amount	Amount of the payout as your request.	Yes	39.60 / 39.6 / 39
currency	The currency of the payout as your request.	Yes	EUR
payment_type	The payment method used by the system for the payout.	No	BWI, HBW, WPW, ONB
merchant_fields	If you submitted a list of values in the *merchant_fields* parameter, they will be passed back with the status report.	No	field1=value1

- If no *transaction_id* is submitted, the *mb_transaction_id* value will be posted in the report.
- The *failed_reason_code* parameter is enabled upon activation and is part of the response status.

6.1 Language encoding for text parameters

All text fields use UTF-8 encoding. However, note that the Quick Checkout can only display Latin-1 characters.

6.2 Validating the status response

We recommend validating the transaction details in the status response as follow:

1. Create a pending transaction or order for a fixed amount on your website.
2. Redirect the customer to the Quick Checkout page, where they complete the transaction.
3. Skrill will post the transaction confirmation to your ***status_url .*** This will include the ***mb_amount*** (amount) parameter.
4. Your website should validate the parameters received by calculating the md5 signature (see ***MD5 signature on page 8-6***). If successful, compare the value in the confirmation post (amount parameter) with value from the pending transaction or order on your website.

You should also compare other parameters such as ***transaction id*** and ***pay_from_email***. Once you have validated the transaction data you can process the transaction(e.g., by dispatching the goods ordered). To protect against replay attacks or duplicate status posts, you must implement measures to ensure that goods are not dispatched multiple times for the same transaction ID(either transaction_id or mb_transaction_id)

Note: *If you want to restrict the receipt of status responses based on the posting IP address, you should use the full list of Skrill IP ranges, as Skrill may change the IP addresses used from time to time. Any address within the following listed ranges could be used:*

IP Ranges:

- 91.208.28.0/24
- 93.191.174.0/24
- 193.105.47.0/24
- 195.69.173.0/24

If you are implementing a new integration, use the following list, as the first one will be soon deprecated.

IP Ranges:

- 18.156.81.39
- 3.64.161.98
- 18.195.181.168
- 52.16.193.112
- 54.228.179.122
- 34.249.111.249

6.3 Using the Merchant Query Interface

You can use the Merchant Query Interface to repost a status report, to automatically check the status of a payout. For details, see the Skrill API and Merchant Query Interface guide

7. APPENDICES

7.1 SIGN parameter calculation

The value of the **SIGN** field is a message digest, represented as a string of sixty-four hexadecimal digits. It is generated by performing a hashing calculation on a string formed by concatenating the following fields:

- merchant_id
- transaction_id
- the uppercase MD5 value of the ASCII equivalent of the secret word submitted in the **Settings > Developer Settings > API/MQI/GSR/CVT** Management section of your online Skrill account.
- amount
- currency

The default hashing algorithm for these parameters is **HMAC SHA-256**, which requires a secret key to complete the hash. In this case, the secret key is the **MD5 value** of the ASCII equivalent of the **secret word** (in lowercase).

Example (HMAC SHA-256):

Field	Value
merchant_id	299202295
transaction_id	frn123merid
secret word (uppercase)	EE38B95C14D6CC07F48EF550C4474EE3
amount	20.45
currency	GBP
concatenated value	**299202295**frn123merid**EE38B95C14D6CC07F48EF550C4474EE3**20.45**GBP**
secret key for the encryption(secret word lowercase)	ee38b95c14d6cc07f48ef550c4474ee3
sign	AD34DF771D38BA82C4F271115675A6C1FFA5642527A50045B8614F57E186F813

7.2 MD5 signature

The md5sig parameter consists of an MD5 sum on a string built by concatenating the following parameters and converting the result to uppercase:

MD5 signature parameters

Value	Description	Example
merchant_id	Your Skrill account user ID	4637827
mb_transaction_id	The new Skrill transaction ID	5585262
MD5 of secret word	The upper-case MD5 value of the secret word submitted in the **Settings > Developer Settings > API/MQI/GSR/CVT** Management section of your online Skrill account.	327638C253A4637199CEBA6642371F20
mb_amount	Processed amount to the payee.	9.99
mb_currency	Currency of mb_amount.	EUR
status	The status of the payout.	2

Example code:

Concatenated fields in Ruby code:

```
Fields = [merchant_id, mb_transaction_id,
Digest::MD5.hexdigest(secret).upcase, mb_amount, mb_currency, status].join
md5sig == Digest::MD5.hexdigest(fields).upcase
```

Using the example values in the ***Table MD5 signature parameters***, the following MD5 code is returned:

CF9DCA614656D19772ECAB978A56866D

Example status report:

```
merchant_id=290186320&transaction_id=200366670&mb_transaction_id=2003666
70&mb_amoun
t=74.218786&mb_currency=GBP&status=2&md5sig=3ED76725C3E3CE6CE25F16F01
BDFDF1D&amount
=80.0&pay_from_email=payer%40gmail.com&pay_to_email=merchant%40info.com&
currency=EUR
```

7.3 Failed Reason Codes

The table below contains all possible values of the failed_reason_code parameter and their corresponding meanings. Failed reason codes are mappings of codes Skrill receives from external processors and failures due to internal procedures.

Failed Reason Codes

Code	Description
01	Referred by Card Issuer
02	Invalid Merchant. Merchant account inactive.
03	Pick-up card
04	Declined by Card Issuer
05	Insufficient funds
06	Merchant/NETELLER/Processor declined
07	Incorrect PIN
08	PIN tries exceed - card blocked
09	Invalid Transaction
10	Transaction frequency limit exceeded
11	Invalid Amount format. Amount too high. Amount too low. Limit Exceeded.
12	Invalid credit card or bank account
13	Invalid card Issuer
15	Duplicate transaction reference
19	Authentication credentials expired/disabled/locked/invalid. Cannot authenticate. Request not authorized.
20	Neteller member is in a blocked country/state/region/geolocation
22	Unsupported Accept header or Content type
24	Card expired
27	Requested API function not supported (legacy function)
28	Lost/stolen card
30	Format Failure
32	Card Security Code (CVV2/CVC2) Check Failed
34	Illegal Transaction
35	Member/Merchant not entitled/authorized. Account closed. Unauthorized access
37	Card restricted by Card Issuer
38	Security violation
42	Card blocked by Card Issuer
44	Card Issuing Bank or Network is not available

45	Processing error - card type is not processed by the authorization centre
51	System error
58	Transaction not permitted by acquirer
63	Transaction not permitted for cardholder
64	Invalid accountId/country/currency/customer/email/field/merchant reference/ merchant account currency/term length/verification code. Account not found/disabled. Entity not found. URI not found. Existing member email. Plan already exists. Bad request.
67	BitPay session expired
68	Referenced transaction has not been settled
69	Referenced transaction is not fully authenticated
70	Customer failed 3DS verification
80	Fraud rules declined
81	Matching criteria not met
98	Error in communication with provider
99	Other
104	PSC Customer name matching failed
106	Transaction failed due to mismatch of the customer type (B2C vs B2B)

7.4 Bank Details for LATAM

7.4.1 Document Types accepted by Skrill

Code	Official document name	Country	Validation (regex)	Example
CPF	Cadastro de Pessoas Físicas	BRAZIL	^[0-9]{11}$	11831541021
CE	Cedula de Identidad	CHILE	^[A-Za-z0-9]{9,20}$	A1234567890
PAS	Pasaporte	CHILE	^[A-Za-z0-9]{9,20}$	CHLAO01001
RUN	Rol Único Nacional	CHILE	^[0-9]{6,8}-[0-9]{1}$	125319092-2
RUT	Rol Único Tributario	CHILE	^[0-9]{6,8}-[0-9]{1}$	125319092-2

RFC	Registro Federal Contribuyentes	MEXICO	^([A-ZÑ&]{4}) ?(?:-?)?(\d{2}(?:0[19]|1[0-2])(?:0[1-9]|[12]\d|3[01])) ?(?:-?)?([A-Z\d]{2})([A\d])$	HEGJ820506M10
DNI	Documento Nacional de Identidad	PERU	^[0-9]{8}$	44851534
PAS	Pasaporte	PERU	^[A-Za-z0-9]{7,12}$	8345627394AB
CE	Carne de Identidad para Extranjeros	PERU	^[A-Za-z0-9]{8,20}$	A847635253910

7.4.2 Bank Account Types accepted by Skrill

Code	Description	Country	Validation	Example
CLABE	Standardized Bank Key or Clave Bancaria Estandarizada	MEX	^\d{18}$	123456789034567896
CPF	CPF	BRA	^[0-9]{11}$	11831541602
Email	Email	BRA	^[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,77}$	test@test.com
Phone	Phone number, without country code	BRA	^[0-9]{11}$	31971212883
EVP	Random Key	BRA	([0-9a-f]{8}-[0-9a-f]{4}-[0-9af]{4}-[0-9a-f]{4}-[0-9a-f]{12})	23e05112-0dfd-4c2bb433-d6420967dbbe
Checking	Interbank account number	PER	^[0-9]{20}$ must start with: 002, 003, 007, 009, 011, 018, 023, 035, 038, 049, 053, 054, 055, 056, 058, 059, 060, 800, 801, 802, 803, 805, 806, 808, 809, 811, 813	00211111111111111111

Savings	Interbank account number	PER	^[0-9]{20}$ must start with: 002, 003, 007, 009, 011, 018, 023, 035, 038, 049, 053, 054, 055, 056, 058, 059, 060, 800, 801, 802, 803, 805, 806, 808, 809, 811, 813	00211111111111111111
Checking	Interbank account number	CHL	^[0-9]{7,20}$	00211111111111111111
Savings	Interbank account number	CHL	^[0-9]{7,20}$	00211111111111111111
Salary	Interbank account number	CHL	^[0-9]{7,20}$	00211111111111111111
Rut	Rol Único Tributario	CHL	When the provided bank_id_code is for "Banco Estado", the provided document_type is RUT and the bank_account_type is Rut: 1. The prefix of bank_account_number must be the same as the prefix of the RUT document_number. (For example, if provided RUT document_number is 12345678-9, the bank_account_number should start with 12345678) 2. the length of the bank account number must be from 8 to 11 digits In other cases: ♦ the length of the bank account number must be7 to 20 digits (^[09]{7,20}$)	When: bank_id_code=012, document_type=RUT, document_number= 12345678-9 bank_account_type= Rut, bank_account_number= 12345678 All other cases: 00211111111111111111

7.5 Bank API

Fetch available banks for LATAM Bank payouts.

Request a list of available Banks by merchant ID:

```
GET pay.skrill.com/banks?paymentOption={paymentOption}&merchantId=
{merchantId}&countryId={countryId}&amount={amount}&currency={currency}
```

Request a list of available Banks by merchant processing e-mail:

```
GET
pay.skrill.com/banks?paymentOption={paymentOption}&merchantEmail={merch
antEmail}&countryId={countryId}&amount={amount}&currency={currency}
```

Example BANK API Response (Success):

```
200 OK

[
 {
  "id": "8194",
  "name": "Banco Estado"
 },

 {
  "id": "8457",
  "name": "Scotiabank Chile"
 },  ...
]
```

Example BANK API Response (Error):

```
{
  "error": {
    "code": "5016",
    "message": "Provided merchant ID or merchant processing e-mail is not existing"
  }
}
```

7.5.1 Table Bank API parameters

Parameter	Description	Required	Example value
paymentOption	Payment option	Yes	ONB
countryId	Skrill country ID	Yes	CHL
merchantId	Merchant ID	Yes/No*	1001
merchantEmail	Merchant processing e-mail	Yes/No*	skrill@merchant.com
currency	Payment amount currency	Yes	CLP
amount	Payment amount	Yes	10000 (min 1)

*Provide either merchantId or merchantEmail.

7.5.2 Table Bank API error codes

Response code	Error message	Description
400	Field error(s)	Missing required field
400	Invalid countryId: {countryId}	Provided Skrill country ID is invalid
400	Invalid merchant	Provided merchant ID or merchant processing e-mail does not exist
400	Invalid amount	Provided amount is not valid
400	Unknown currency: {currency}	Provided currency is not valid
400	Payment method not available: {paymentOption}	Payment method is not supported for this payment