SEB Payment Gateway LinkPay Documentation

2019-12-19

Overview of the LinkPay product

What is LinkPay?

Read more about [LinkPay.](#)

Payment options with LinkPay

[See payment options](#)with LinkPay.

How to use LinkPay

Creating a Template

See instructions how to create template [here](#).

Receiving a Callback for the LinkPay payment

In order to receive automatic notifications to Merchants' system a Merchant's URL for accepting callback requests must be defined in SEB Portal under E-Shop configuration (*Settings -> E-Shop settings -> LinkPay Notification URL*).

Defining a LinkPay callback URL will ensure that once the payment attempt is completed by the customer, successfully or not, SEB system will send a callback message (HTTP form POST) to the URL defined.

Callback consists of the following parameters:

Parameter	Description
nonce	Random unique value to prevent replay attacks
timestamp	Expressed as seconds from January 1, 1970 UTC
api_username	Merchant API username. The value can be found in Merchant Portal in the Settings section.
transaction_result	Should only be considered as additional

	information about the different phases of the payment. Possible values ● 'completed' ● 'cancelled' ● 'failed' The definitive outcome of the payment attempt is reflected by payment_state parameter based on which the order status should be updated in merchant system.
linkpay_token	Link token, reference to link used for the payment
linkpay_reference	Reference to payment performed with the link
account_id	Processing account ID that was used to process the transaction
order_reference	Order reference as defined by merchant or combined of linkpay_token/linkpay_reference (e.g.hvcjro/gwwwza)
payment_reference	Payment reference ID
fraud_score	Payment fraud score, calculated by SEB fraud prevention engine, in numeric format
stan	Payment STAN reference number, in numeric format. Returned only in case the payment was sent for processing to card processor and didn't fail already on SEB level.
state_3ds	Payment 3D-Secure authentication status. Possible values are: ● '3ds' - authentication completed successfully ● 'attempted' - authentication attempted but not completed ● 'failed' - authentication failed ● 'no3ds' - no 3DS coverage for payment ● 'unknown' - 3DS status unknown
payment_state	Current status of the payment which

	reflects the definitive payment outcome based on which the order status should be updated in merchant system. Possible scenarios: ● 'settled' or 'authorised' (successful payments) ● 'cancelled' or 'waiting_for_3ds_response' (cancelled/incomplete payments) ● 'failed' (failed payments)
transaction_time	Time of the transaction
amount	Transaction amount, dot as decimal separator
customer_name	Customer name value
customer_email	Customer email value
invoice_number	Value defined by merchant or entered by customer
custom_field_1	Custom field value defined by merchant or entered by customer
custom_field_2	Custom field value defined by merchant or entered by customer
custom_field_3	Custom field value defined by merchant or entered by customer
custom_field_4	Custom field value defined by merchant or entered by customer
cc_last_four_digits	Last four digits of the card number.
cc_year	Card expiration year (as entered by the cardholder in payment form), in YYYY format (e.g. 2017)
cc_month	Card expiration month (as entered by the cardholder in payment form), in mm format (e.g. 1 or 12)
cc_holder_name	Cardholder name (as entered by the cardholder in payment form)
cc_type	Card type. Possible values are 'visa' and

	'master_card'
cc_issuer	Card issuing organization
cc_funding_score	Indicates whether the card is 'Debit', 'Credit' or 'Prepaid'
cc_cobrand	Indicates co-branding if card is co-branded
cc_product	Indicates specific card product (e.g 'Gold', 'Platinum', 'Visa Classic', 'Visa Electron', 'Businesscard' etc.)
cc_issuer_country	Card issuing bank country, in alpha-2 format (e.g. 'EE')
hmac_fields	Comma-separated alphabetically ordered list of parameter names that were used in HMAC calculation.
hmac	HMAC checksum of the redirect parameters. See "SEB Payment API Documentation" document for the HMAC calculation reference.

Getting overview of LinkPay payments

Payment information to customers:
- Once the customer has completed a payment with the LinkPay link they will be notified about the payment outcome (either successful or failed) with all payment details on a payment receipt.
- The receipt will be sent also to email if this information is asked or pre-filled with the order details on LinkPay checkout page.

Information to Merchant:
- Once the payment attempt is completed (successfully or not), SEB e-commerce system will send a callback message to the URL defined.
- All payment attempts are visible in SEB Portal under Payment Management list
  - Additionally, under specific payment details, a reference to the Link details are given
    - with the crosslink to the Portal Link fields ('Token')
    - with the to customer receipt page ('Reference')

- with the option to open up Link specifics attributes ('Form attributes').

- All LinkPay payments with specific token can be exported to CSV file from menu LINKS -> button CSV EXPORT

Creating Terms of Service for LinkPay

Read more about creating Terms of Service here.

LinkPay Subscriptions

What is LinkPay Subscription?

Read about LinkPay Subscription here.

Please note: the functionality is disabled by default. Please contact support@every-pay.com to be able to use Subscriptions functionality with your SEB account.

Creating a Subscription Plan

Linkpay Subscriptions assume that there are already one or more Plans configured in Subscriptions section in the Merchant Portal.
A Plan enables to define a template with the service name, cost, currency and the frequency for the Subscription to be billed on recurring basis.

When creating a Plan, the following fields must be specified:
- **Plan name**: To identify the plan e.g. Gold Plan, Yearly plan etc.
- **Cycle price**: The amount the customers will be charged.
- **Processing account:** Defines the currency (and whether 3DS is used for initial payment) in which the customer will be charged.

- **Calendar synchronized:** If opted 'yes' then the period always starts on corresponding calendar period (beginning of week, month, year).
- **Cycle unit:** How often the customer will be charged.
- **Cycle length:** This defines the frequency of the payments for how often the customer will be charged.
- **Billing day of period:** Enables to define when to bill the Subscription from the customer:
  - indicates the day from the beginning of period on which the payment will be processed;
  - applies for all periods;
  - by default set to '1';
  - will be defined by entering the positive number manually.
- **Breakoff day:** Enables to define a day which will be considered as a maximum allowed day for the current period signup, later Subscriptions are handled as pre-payments for the following period. E.g if "Billing day of month" is defined as '05' and "Breakoff day" is defined as '20' then for all Subscriptions registered until 20.01 (included) the first automated recurring payment will be processed on 05.02 and for Subscriptions registered starting from 21.01 the first automated recurring payment will be processed on 05.03.
- **Plan end date:** The duration of agreement. Can be left undefined.
- **Trial period:** Free introductory window of time for customers on a recurring billing plan. Can be set by using 'Trial period length' and 'Trial period unit'.

Creating Subscription Link/Template

LinkPay links can be tied to multiple Subscription Plans and Plans can have multiple Links (e.g. multi-language links for the same Plan).
The link language is stored together with the Subscription data, so that future communication with the user continues in the correct language.
Creating a Link for Subscription is similar to creating a regular Link or Template, except for one or more of the Subscriptions Plans must be selected from the dropdown.

To create a Subscription Link:
- Enable **Subscription link checkbox** (if not already pre-selected with the template).
- Pick one or more of the **Subscription plans in the link creation / editing view**, if it is not already preset by the template . The selected plan **can be changed later** – this **will affect only new subscriptions** created through this link, all existing subscriptions will continue to use the Plan they were created with.
- Fields **'Customer name' and 'Customer email' must be always present for Subscription Links**. This data is needed in order to tie the Plan to a Customer. The information can be viewed in Portal on Customers and Subscriptions page.
- Other fields can be defined as needed.

The created Subscription payment link allows customer to sign up for the service and/or make the initial payment. Following payments are then initiated automatically by SEB ecommerce system using the schedule and amount described in the Plan.

Subscription Link limits

Subscription link is subject to the same limits as a regular LinkPay link – if the allowed number of Subscriptions has been reached, then the link will deactivate. This counts the initial Subscriptions created via this link, not active subscriptions at any given time.

LinkPay Subscription lifecycle events

After a customer follows the subscription link and successfully creates a Subscription, the following life cycle begins:

1. Optional trial period
2. First payment
3. Periodic payments as per the Plan
4. Subscription ends:
   - Cancelled Subscription
     - by merchant
     - by failed payments (triggered automatically)
     - by customer
   - Subscription expires on Plan end date

LinkPay Subscription Callbacks

In order to receive automatic notifications to Merchants' system a Merchant's URL for accepting callback requests must be defined in SEB Portal under E-Shop configuration (Settings -> E-Shop settings -> Subscription notification URL). Defining a Subscription notification URL will ensure that notifications on subscription lifecycle and payment events are sent to the specified URL. Those notifications enable to guard access to the resources or services based on the subscription state changes.

A Subscription can change its state according to the success of the periodic payments or other reasons. Subscription lifecycle event notifications are sent to the merchant if there are any changes in the Subscription's state. For example, Subscription can become cancelled or reactivated and knowledge of those events must also be relayed to the merchant.

Also for every payment for this Subscription, a notification is sent that indicates the payment outcome.

Each lifecycle notification consists of the following parameters:

Parameter	Description
time	Time of the event. Formatted according to the ISO 8601 standard.
subscription_token	Subscriptions unique ID
plan_token	Subscription's plan ID
customer_token	Subscription customer's unique ID
customer_name	Subscription customer's name
customer_email	Subscription customer's email
merchants_subscription_id	Unique subscription ID provided to a subscription by the merchant.
subscription_state	Current state of the subscription. Possible values are: • active • cancelled_by_customer • cancelled_by_merchant • cancelled_by_failed_payment • failed_to_be_created • unsettled
event_name	Subscription lifecycle event name. Possible values are: • subscription_activated • subscription_cancelled_due_to_failed_payment • subscription_creation_failed • subscription_cancelled_by_customer • subscription_cancelled_by_merchant • subscription_unsettled
nonce	Random unique value to prevent replay attacks
timestamp	Time of sending the event. Expressed as

	seconds from January 1, 1970 UTC
api_username	Merchant API username. The value can be found in Merchant Portal in the Settings section.
hmac_fields	Comma-separated alphabetically ordered list of parameter names that were used in HMAC calculation. See "SEB API Documentation" document for the HMAC calculation reference.
hmac	HMAC checksum of the redirect parameters. See "SEB API Documentation" document for the HMAC calculation reference.

Creating URL-changeable LinkPay links

In order to meet more sophisticated needs it is possible to create and pre-fill the links on a system level. This functionality allows sending out unique payment links to customers without manually creating every link separately in SEB portal. It is only needed to define one general link (per use-case) with needed changeable parametres and all required values, such as needed for specific regular invoices as an example, can be pre-filled within URL. Such links can be used to send regular invoices to customers by using period-specific data (e.g variable amount and invoice number) with minimum effort to create the links and easy to use for the customers.

Please note: the functionality is disabled by default. Please contact support@every-pay.com to be able to define url-changeable parametres with your SEB account.

Setting up a general link

Setting up the link can be done via SEB Merchant portal.
- Define the set of payment details to be associated with the link.
  - determine which details you want to be modified by you (for example set unique order reference or invoice number for each link)
  - determine which details you want to predefine and that can't be changed by customer

- determine which details you want to leave open for your customers to fill in
- the field values can have both conditions available together - customer changeable and url changeable
- After creating a link a linkpay_token is generated by SEB system that must be referred to when creating the following payment links

It is also possible to create [LinkPay Template](#) first and define link-specific details by using already defined specifics on a Template.

HMAC calculation for the link

To ensure secure payment processing, Merchant must sign parts of the URL with their `api_secret` which will be verified by SEB during the payment request.

Definitions:
- URL := `schema://hostname:port/path?query_string`
- Signature algorithm: HMAC-SHA256
- Signed parts: `query_string`
- Merchant must include token as part of query_string: `token=<linkpay_token>`
- The query_string is URL-encoded by `RFC3986` specification, with spaces encoded as `%20`.
- Name of signature param: `hmac`

The order of the params in the query_string does not matter – the string is taken as-is, while removing the hmac signature.

Steps to produce hmac:
1. create query_string from LinkPay token: `linkpay_token=<token>`
2. URL-encode all needed params
3. append params to the query string, e.g.
   `linkpay_token=<token>&order_reference=<ref>`
4. calculate HMAC-SHA256 signature of this query_string using the api_secret provided in the Portal.
5. append signature to the query_string:
   `linkpay_token=<token>&order_reference=<ref>&hmac=<signature>`
6. Combine the signed URL with LinkPay prefix to produce full URL:
   `https://pay.every-pay.eu/lp?linkpay_token=<token>&order_reference=<ref>&hmac=<signature>`

```php
<?php
$api_secret = 'abc1234abc1234';
$linkpay_prefix = 'https://pay.every-pay.eu/lp';
$params = [
  'linkpay_token' => 'w23gd4',
```

```php
  'order_reference' => 'ord123',
  'name' => 'John Smith',
  'transaction_amount' => '5.00',
  'email' => 'customer@example.com'
];


$query = http_build_query($params);


$hmac = hash_hmac('sha256', $query, $api_secret);


$url = "${linkpay_prefix}?${query}&hmac=${hmac}";
echo $url;


# outputs:
#
https://pay.every-pay.eu/lp?linkpay_token=w23gd4&order_reference=
ord123&name=Ester%20Tester&transaction_amount=5.00&email=customer
%40example.com&hmac=013f9e8fbabd1b73f4ab8b7b6abdd8f348da67100c4c8
7d7530a67957e75d16f
```