

Understanding Paysera Recurring Billing Integration: A Comprehensive Guide

This report provides a detailed analysis of Paysera's recurring billing system, explaining the core concepts, technical requirements, and implementation steps. Based on the official documentation, this guide will help developers understand how to efficiently integrate and utilize Paysera's recurring billing functionality.

Introduction to Paysera Recurring Billing

Recurring billing is a payment method where merchants can automatically charge cardholders for specified goods or services on a prearranged schedule without requiring customer interaction for each transaction^{[1] [2]}. This system requires the merchant to obtain the cardholder's permission once upfront for subsequent recurring charges^{[1] [2]}. Essentially, it enables seamless automated payment collection after the customer's initial successful transaction, making it ideal for subscription-based business models^{[3] [4]}.

Key Terminology

Before diving into the integration process, it's important to understand the terminology used in Paysera's API documentation:

- **User:** The customer who makes payments in your e-commerce project^{[1] [2]}
- **Payment Request:** A resource that serves as an entry point in any payment flow, which can be processed either through user interaction or automatically via recurring billing^{[1] [2]}
- **Recurring billing:** A payment flow method allowing merchants to charge users without their interaction, possible only after the first successful user payment^{[1] [2] [3]}

Integration Process

Preparation Phase

Step 1. Read the Documentation

Before proceeding with integration, thoroughly review the documentation to ensure Paysera's recurring billing solution aligns with your business requirements^{[1] [2]}.

Step 2. Get Paysera Account

Recurring billing API is only available in Production environment. Register for a Paysera account to obtain access to the platform^[1] ^[2].

Step 3. Create Checkout Project

Create a Checkout project within your Paysera account and complete all required information fields^[1] ^[2].

Step 4. Contact Paysera

Reach out to Paysera with details about how you plan to implement the Recurring Billing API and provide your `project ID`. Paysera will evaluate whether your business model is compatible with their recurring billing service^[1] ^[2].

Step 5. Obtain API Credentials

If your business meets Paysera's requirements, they will configure an API client and provide you with credentials necessary for integration^[1] ^[2].

Step 6. Configure Your Business

Contact Paysera to reconfigure your business settings for collecting real payments through the recurring billing system^[1] ^[2].

Authentication System

All API calls to Paysera must be authenticated using MAC access authentication, which is based on the OAuth 2.0 Message Authentication Code (MAC) Tokens specification^[1] ^[2].

When integrating, you'll receive credentials including:

```
business_id: Opb2XVb-gEh4aGcR09Ko5Wb8V_6vueDM
mac_id: wkVd93h2uS
mac_key: IrdTc8uQodU7PRpLzzLTW6wqZA06tAMU
mac_algorithm: hmac-sha-256
```

The `mac_key` must be treated as confidential at all times^[1] ^[2].

For each API request, you must construct an authentication header containing:

- `id`: The client identifier
- `ts`: Current UNIX timestamp
- `nonce`: A randomly generated value
- `mac`: Hash calculated from the request parameters and your `mac_key`
- `ext`: Optional extension parameters^[1] ^[2]

Recurring Billing Flow

The recurring billing process consists of three essential steps^{[3] [4]}:

1. Payment Request Initialization

This is the starting point for both standard and recurring payment flows. For recurring billing, you must create a Payment Request with specific parameters including the `token_strategy` parameter set to "required"^{[3] [4] [5]}.

2. Payment Request Authorization with Token

For the initial payment, the user completes the transaction through Paysera's interface. For subsequent recurring payments, you authorize the Payment Request using the token generated from the initial transaction, eliminating the need for user interaction^{[3] [4]}.

3. Payment Request Capturing

After authorization, you must capture the Payment Request to finalize the billing process^{[3] [4]}.

Implementation Example

Initial Payment (To Obtain Token)

Step 1: Create Payment Request

```
POST /checkout/rest/v1/payment-requests HTTP/1.1
Host: checkout-eu-a.paysera.com
Content-Type: application/json; charset=utf-8
Authorization: MAC id="wkVd93h2uS", ts="1343811600", nonce="nQnNaSNyubfPErjR055yaaEYo9YZ1

{
  "business_id": "LcKB4rs-pNVLgQKxZTaGzFZRAuiQPuyE",
  "order_id": "12345446",
  "price": {
    "amount": "10.00",
    "currency": "EUR"
  },
  "description": "Payment for goods on order 12345446 on the website https://shop.com.",
  "method_key": "card",
  "payer": {
    "email": "payer@email.com"
  },
  "locale": "en",
  "accept_url": "https://shop.com/accept-url",
  "cancel_url": "https://shop.com/cancel-url",
  "callback_url": "https://shop.com/callback-url",
  "token_strategy": "required"
}
```

The response will include an `authorization_url` to which you should redirect the user^{[3] [4] [5]}.

Step 2: Redirect User to Complete Payment

After the user completes the payment, Paysera will send a notification to your `callback_url` with a `notification_id` [\[3\]](#) [\[4\]](#).

Step 3: Process Notification and Retrieve Token

```
GET /notification/rest/v1/notifications/ABcJDZe-rWzLgQKxZTamdfZRApsrPuyE HTTP/1.1
Host: checkout-eu-a.paysera.com
Authorization: MAC id="wkVd93h2uS", ts="1343811600", nonce="nQnNaSNyubfPErjR055yaaEYo9YZ1
```

The response will contain payment details and the crucial `issued_token` for future recurring payments [\[3\]](#) [\[4\]](#) [\[5\]](#).

Step 4: Mark Notification as Read

```
PUT /notification/rest/v1/notifications/ABcJDZe-rWzLgQKxZTamdfZRApsrPuyE/read HTTP/1.1
Host: checkout-eu-a.paysera.com
Authorization: MAC id="wkVd93h2uS", ts="1343811600", nonce="nQnNaSNyubfPErjR055yaaEYo9YZ1
```

Subsequent Recurring Payments

Step 1: Create New Payment Request

```
POST /checkout/rest/v1/payment-requests HTTP/1.1
Host: checkout-eu-a.paysera.com
Content-Type: application/json; charset=utf-8
Authorization: MAC id="wkVd93h2uS", ts="1343811600", nonce="nQnNaSNyubfPErjR055yaaEYo9YZ1

{
  "business_id": "LcKB4rs-pNVLgQKxZTaGzfZRAuiQPuyE",
  "order_id": "12345447",
  "price": {
    "amount": "10.00",
    "currency": "EUR"
  },
  "description": "Payment for goods on order 12345447 on the website https://shop.com.",
  "method_key": "card",
  "payer": {
    "email": "payer@email.com"
  },
  "locale": "en",
  "accept_url": "https://shop.com/accept-url",
  "cancel_url": "https://shop.com/cancel-url",
  "callback_url": "https://shop.com/callback-url"
}
```

Step 2: Authorize Payment with Saved Token

```
PUT /checkout/rest/v1/payment-requests/PRAFY24n-UBgSkiRCL-h_CVAAvj4MEkj/authorize HTTP/1.1
Host: checkout-eu-a.paysera.com
Content-Type: application/json;charset=utf-8
Authorization: MAC id="wkVd93h2uS", ts="1343811600", nonce="nQnNaSNyubfPErjR055yaaEYo9YZd

{
  "token": "vLUGdTtnDjQs7Yv0fjYQyYfG60m"
}
```

The response will show the payment request status as "authorized" [\[3\]](#) [\[4\]](#).

Step 3: Capture Payment

```
PUT /checkout/rest/v1/payment-requests/PRAFY24n-UBgSkiRCL-h_CVAAvj4MEkj/capture HTTP/1.1
Host: checkout-eu-a.paysera.com
Authorization: MAC id="wkVd93h2uS", ts="1343811600", nonce="nQnNaSNyubfPErjR055yaaEYo9YZd
```

The response will show the payment request status as "captured" [\[3\]](#) [\[4\]](#).

Step 4: Handle Notification

As with the initial payment, Paysera will send a notification to your callback URL, which you should process and mark as read [\[3\]](#) [\[4\]](#).

Alternative: Token Without Fund Transfer

Paysera also offers a method to obtain a token without actually transferring funds, useful for validating payment methods without charging the customer. This is achieved by adding `refund_on_capture: true` to the payment request parameters [\[5\]](#).

Conclusion

Integrating Paysera's recurring billing system involves a series of well-defined steps, from initial account setup to handling recurring transactions. The system provides a secure, efficient mechanism for subscription-based billing models while minimizing user interaction after the initial payment authorization.

When implementing this solution, pay special attention to the security aspects, particularly the handling of the `mac_key`, and ensure proper processing of notifications to complete each payment cycle. Following these guidelines will help establish a reliable recurring billing system for your e-commerce platform.

✱

1. <https://developers.paysera.com/en/checkout/>

2. <https://developers.paysera.com/en/checkout/>

3. <https://developers.paysera.com/lt/checkout/payment-collection/recurring-billing>
4. <https://developers.paysera.com/en/checkout/payment-collection/recurring-billing>
5. <https://developers.paysera.com/en/checkout/payment-collection/payment-flow>