

PayByLink API — English Guide

Version 1.3

PayByLink API — English Guide

****Version:**** 1.3 • ****Updated:**** 2025-09-11

This guide explains how to integrate with the PayByLink payment gateway, covering:

- Standard payment link generation (`/api/v1/transfer/generate`)
- Transaction cancellation (`/api/v1/transfer/cancel`)
- BLIK white label background authorization (`/api/v1/transfer/blikauth`)
- Webhook notifications for payments (`notifyURL`) and BLIK status (`notifyStatusURL`)
- Signature calculation and verification
- Error handling and recommended best practices

All requests use ****JSON**** payloads and ****UTF-8**** strings, and must be sent via ****HTTP POST**** with `Content-Type: application/json`. The production base URL is: `https://secure.paybylink.pl`

> Replace placeholders like `YOUR\_SHOP\_ID`, `YOUR\_SECRET\_KEY`, and URLs with your actual values.

Signature Calculation (Requests)

Every request must include a `signature` field. Compute the SHA-256 (hex) of the concatenation:

```
SECRET_KEY | field1 | field2 | ... | lastNonEmptyField
```

- Use a literal pipe `|` as a separator.
- ****Omit empty fields**** entirely from the concatenation.
- ****Order matters:**** follow the exact order of fields listed in each endpoint spec.
- Ensure `price` (and other monetary fields) use two decimal format: `%.2f` (e.g. `10.00`).

Signature Verification (Webhooks)

Verify webhook signatures the same way, using your `SECRET\_KEY` and the exact order specified for each webhook.

Create Standard Payment Link

****POST**** `/api/v1/transfer/generate`

****Request JSON fields (non-empty fields must be included in signature):****

- `shopId` (int, required)
- `price` (number, required; two decimals)
- `control` (string, optional)
- `description` (string, optional)
- `email` (string, optional)
- `notifyURL` (string URL, optional)
- `returnUrlSuccess` (string URL, optional)
- `returnUrlSuccessTidPass` (bool, optional; if true, `tid` is appended to `returnUrlSuccess`)
- `hideReceiver` (bool, optional)
- `customFinishNote` (string, optional)
- `signature` (string, required)

****Signature order:****

```
SECRET_KEY|shopId|price|control|description|email|notifyURL|returnUrlSuccess
```

****Success 200 Response:****

```
``json
{
  "url": "https://secure.pbl.pl/transfer/XYZ...",
  "transactionId": "XYZ..."
}
```

****Error Response:****

```
``json
{ "errorCode": 403, "error": "invalid signature" }
```

Cancel a Transaction

****POST**** `/api/v1/transfer/cancel`

****Request:****

- `shopId` (int, required)
- `transactionId` (string, required)
- `customReason` (string, optional)
- `signature` (string, required)

****Signature order:****

```
SECRET_KEY|shopId|transactionId|customReason
```

****Response 200:****

```
```json
{ "cancelled": true }
```
or
```json
{ "cancelled": false, "cancelError": "Transaction already cancelled" }
```

**## BLIK White Label (Background) Authorization**

**\*\*POST\*\*** `/api/v1/transfer/blikauth`

Create a BLIK transaction using the customer's 6 digit BLIK code, without redirecting to the hosted page. Use this when you capture the BLIK code in your own checkout and authorize **\*\*in the background\*\***.

**\*\*Request fields (non-empty fields must be included in signature):\*\***

- `shopId` (int, required)
- `price` (decimal/number, required; two decimals)
- `code` (string, required; 6 digits from customer's banking app)
- `customerIP` (string, required; IP of the customer initiating the transaction)
- `control` (string, optional)
- `notifyPaymentURL` (string URL, optional; payment confirmation webhook)
- `notifyStatusURL` (string URL, optional; BLIK status webhook)
- `signature` (string, required)

**\*\*Signature order:\*\***

```
SECRET_KEY|shopId|price|code|customerIP|control|notifyPaymentURL|notifyStatusURL
```

**\*\*Success 200 Response:\*\***

```
```json
{ "transactionId": "w3WUQX...JLJ" }
```

****BLIK Status Webhook (`notifyStatusURL`) Examples:****

```
```json
{ "transactionId": "...", "control": "", "price": "1.01", "status": "CONFIRMED", "signature": "..." }
{ "transactionId": "...", "control": "", "price": "1.01", "status": "TIMEOUT", "signature": "..." }
{ "transactionId": "...", "control": "", "price": "1.01", "status": "USER_DECLINED", "signature": "..." }
{ "transactionId": "...", "control": "", "price": "1.01", "status": "WRONG_TICKET", "signature": "..." }
```

**## Webhook Notifications**

**### Payment Notification (`notifyURL`)**

If you set `notifyURL` when generating a standard payment, PayByLink sends a POST with JSON:

- `transactionId` (string, 32)
- `control` (string; your custom value)
- `email` (string)
- `amountPaid` (float) – format with two decimals
- `notificationAttempt` (int)
- `paymentType` (string; e.g., `TRANSFER`, `PAYPAL`)
- `apiVersion` (int)
- `signature` (string) – verify it

**\*\*Acknowledge\*\*** with HTTP **\*\*200 OK\*\*** and body `OK` (Content-Type `text/plain`).

If not acknowledged, delivery is retried with **\*\*exponential backoff\*\*** up to **\*\*241 attempts\*\***.

**\*\*Payment webhook signature string (SHA-256 hex):\*\***

```
SECRET_KEY|transactionId|control|email|amountPaid|notificationAttempt|paymentType|apiVersion
```

**### BLIK Status Notification (`notifyStatusURL`)**

For BLIK white label, PayByLink sends a POST with JSON:

- `transactionId` (string, 32)
- `control` (string)
- `price` (decimal string, two decimals)
- `status` (string; examples: `CONFIRMED`, `TIMEOUT`, `USER\_DECLINED`, `WRONG\_TICKET`, etc.)
- `signature` (string) – verify it

**Acknowledge** with HTTP **200 OK** and body `OK`.

**BLIK status webhook signature string (SHA256 hex):**

```
SECRET_KEY|transactionId|control|price|status
```

#### ## Best Practices & Security

- **Always validate signatures** on incoming webhooks before updating order state.
- **Use TLS** on all endpoints. Only call `https://secure.paybylink.pl`.
- **Do not log secrets** or full card/BLIK codes.
- **Store only necessary data** and comply with applicable regulations.
- **Idempotency:** Use your unique `control` per order to avoid duplicates.

HTTP	Meaning
200	OK — Operation succeeded
400	JSON decode error
401	Required field(s) missing
402	Field length/size invalid
403	Invalid request signature
404	Shop ID not found
500	Internal error while processing