Z-Pay Swift
Fast & secure remittance support

Partner API Integration Guide

Topup. Bundles. Balance. Request status.

A practical offline reference for connecting your own server to Z-Pay Swift.

> PRODUCTION BASE URL
>
> `https://zpayswift.com/api`

Live financial API - no sandbox. Create requests can hold real wallet funds. Keep API keys private and server-side. All IDs, numbers, amounts and credentials in this guide are examples or placeholders, not account data.

Inside this guide

01 Authentication and safe request flow

02 Check Balance

03 Bundle Offers

04 Create Topup

05 Create Bundle

06 Request Status

07 Errors, status and recovery

08 Reusable PHP client

09 Reusable Node.js client

Documentation revision: 2026-11-02 | Malaysia Time (MYT). This PDF is a snapshot. Check the live API Docs for the latest routes, permissions and limits.

Authentication & safe flow

One private partner API key. No app key, PIN, login password or user session.

REQUEST HEADERS

```
Authorization: Bearer YOUR_API_KEY
Accept: application/json
Content-Type: application/json  # POST only
```

1. Get access

Admin must enable API access and the required Topup/Bundle service. Generate a full key in API Keys and store it privately on your server. A masked key cannot authenticate. X-API-KEY is supported as an alternative header. Never put keys in URLs or source control.

2. Verify a read first

Call GET balance. No account UID is accepted: the key selects the account. Both MYR and BDT wallets are supported; always read wallet_currency.

3. Create once, then track

For bundles, select a current offer_id first. Use one persisted idempotency_key per business order. Save data.request_id from the create response; then poll Request Status with backoff. HTTP 200/ok=true is acceptance, not delivery.

4. Treat unknown POST outcomes carefully

A timeout or 5xx may mean that a POST was accepted but its response was lost. Do not automatically resend. If the request ID is known, check status; otherwise review History Logs or contact support before any deliberate retry.

Currency and price

Topup amount is in BDT, including for an MYR account. Read wallet_debit_amount and wallet_debit_currency for the actual wallet hold. Bundle prices, exchange rates, fees and commissions are resolved by the server, not submitted by the client.

Idempotency is not unconditional replay

Send idempotency_key in the JSON body. Keep the same API key, order ID and payload for the same business order. Existing create validation runs before duplicate lookup; changed balance, rates or offers may prevent a replay. Do not reuse an order ID for another recipient, amount or offer.

Supported scope

Only the five documented partner endpoints are covered. No sandbox, MFS creation, Add Money, user creation or webhook endpoint is provided by this guide. Panel features use separate session/CSRF flows.

Check Balance

Read your own available and reserved wallet balances.

GET ENDPOINT

```
https://zpayswift.com/api/public_api/balance.php
```

No query parameters or body. The API key selects your account.

No UID or account ID is accepted. The API key selects the account.

Amounts use wallet_currency. This is a read, not a deposit or transfer.

cURL (BASH) | REQUEST EXAMPLE

```
: "${ZPAY_API_KEY:?Set your private server environment variable}"
ZPAY_BASE_URL="https://zpayswift.com/api"
curl --request GET \
  --url "$ZPAY_BASE_URL/public_api/balance.php" \
  --header "Authorization: Bearer $ZPAY_API_KEY" \
  --header 'Accept: application/json'
```

Illustrative response

Actual currency, prices and values come from the server. Additional fields may be returned.

HTTP 200 | JSON

```
{
  "ok": true,
  "code": "SUCCESS",
  "message": "Illustrative example.",
  "data": {
    "available_balance": 1000,
    "hold_balance": 50,
    "wallet_currency": "MYR",
    "updated_at": 1800000000,
    "checked_at": 1800000010
  }
}
```

Bundle Offers

List active bundle offers visible to your account.

GET ENDPOINT

`https://zpayswift.com/api/public_api/bundle_offers.php`

Parameter	Type and meaning
`operator`	**Optional** | string Operator filter, for example GP.
`limit`	**Optional** | integer Page size: 1 to 10. Default: 10.
`cursor`	**Optional** | string Opaque next_cursor from the previous response.
`status`	**Optional** | string Use ACTIVE. Offers must still be active and unexpired.

The total field is the current page count, not an all-time total.

Continue with pagination.next_cursor while pagination.has_more is true, even if a filtered page is empty.

cURL (BASH) | REQUEST EXAMPLE

```
: "${ZPAY_API_KEY:?Set your private server environment variable}"
ZPAY_BASE_URL="https://zpayswift.com/api"
curl --request GET \
  --url "$ZPAY_BASE_URL/public_api/bundle_offers.php" \
  --header "Authorization: Bearer $ZPAY_API_KEY" \
  --header 'Accept: application/json' \
  --get \
  --data-urlencode 'limit=10'
```

Bundle Offers | Response

Read the response fields before taking the next action.

Illustrative response

Actual currency, prices and values come from the server. Additional fields may be returned.

HTTP 200 | JSON


```
{
  "ok": true,
  "code": "SUCCESS",
  "message": "Illustrative example.",
  "data": {
    "uid": "YOUR_ACCOUNT_ID",
    "total": 1,
    "items": [
      {
        "offer_id": "OFFER_ID_FROM_RESPONSE",
        "operator": "GP",
        "bundle_name": "Example bundle",
        "amount": 100
      }
    ],
    "pagination": {
      "limit": 10,
      "count": 1,
      "has_more": false,
      "next_cursor": ""
    }
  }
}
```


Reading the page

items: current active/unexpired offers. Copy an actual offer_id from the response; the example ID is not usable.

total: the count on this page, not an all-time total.

pagination.has_more / next_cursor: follow the opaque cursor while has_more is true. A filtered page can be empty and still have another page.

Next: create a bundle

Use the chosen offer_id with bundle_number and your persisted idempotency_key. Do not send your own price, currency or commission. Availability is checked again by the server at creation.

Create Topup

Submit a Bangladesh mobile topup request. This is a live financial operation.

POST ENDPOINT

`https://zpayswift.com/api/public_api/topup_create.php`

Parameter	Type and meaning
`topup_number`	**Required** | string Bangladesh mobile number, for example 01712345678. Keep it as a string.
`operator`	**Required** | string Common codes: GP, ROBI, BL, AIRTEL, TT. Availability is configured by Admin.
`amount`	**Required** | number Recharge amount in BDT. Current service and account limits are checked by the server.
`idempotency_key`	**Optional** | string Recommended: your unique, persisted business-order ID. Keep it and the same payload on any deliberate retry.
`note`	**Optional** | string Optional reference or note.

The amount is BDT, including for MYR accounts. Read wallet_debit_amount and wallet_debit_currency for the actual wallet hold.

ok=true with status=PENDING means accepted, not delivered. Save data.request_id.

LIVE POST: Replace every {{placeholder}} before running. No automatic retries.

cURL (BASH) | REQUEST EXAMPLE

```
: "${ZPAY_API_KEY:?Set your private server environment variable}"
ZPAY_BASE_URL="https://zpayswift.com/api"
curl --request POST \
  --url "$ZPAY_BASE_URL/public_api/topup_create.php" \
  --header "Authorization: Bearer $ZPAY_API_KEY" \
  --header 'Accept: application/json' \
  --header 'Content-Type: application/json' \
  --data '{
  "topup_number": "{{topup_number}}",
  "operator": "GP",
  "amount": 20,
  "idempotency_key": "{{client_request_id}}",
  "note": "Partner topup"
}'
```

Create Topup | Response

Read the response fields before taking the next action.

Illustrative response

Actual currency, prices and values come from the server. Additional fields may be returned.

HTTP 200 | JSON


```
{
  "ok": true,
  "code": "SUCCESS",
  "message": "Illustrative example.",
  "data": {
    "request_id": "YOUR_REQUEST_ID",
    "uid": "YOUR_ACCOUNT_ID",
    "status": "PENDING",
    "operator": "GP",
    "topup_number": "01712345678",
    "amount": 20,
    "topup_currency": "BDT",
    "wallet_debit_amount": 20,
    "wallet_debit_currency": "BDT",
    "total_debit": 20,
    "created_at": 1800000000
  }
}
```


Store these fields

request_id: save it with your business-order ID before displaying a final result.

status: PENDING means accepted and waiting, not successful delivery.

wallet_debit_amount / wallet_debit_currency: read the actual wallet denomination and amount. Do not infer an MYR debit from the BDT face value.

Next: track the service outcome

Read GET /public_api/request_status.php with the saved request_id. SUCCESS and FAILED are final; pending/processing are not. A status read is not an instruction to resubmit the create request.

If the POST response is missing or unreadable, the request may still exist. Review status/history before any deliberate retry.

Create Bundle

Submit a bundle request using an active offer_id. This is a live financial operation.

POST ENDPOINT

`https://zpayswift.com/api/public_api/bundle_create.php`

Parameter	Type and meaning
`offer_id`	**Required** | string Copy offer_id from Bundle Offers. Prices and availability are resolved by the server.
`bundle_number`	**Required** | string Bangladesh mobile number as a string.
`idempotency_key`	**Optional** | string Recommended: a unique, persisted business-order ID.
`note`	**Optional** | string Optional reference or note.

Do not submit your own price, commission, currency or account UID. The server decides these.

ok=true with status=WAITING_ADMIN means accepted, not completed. Save data.request_id.

LIVE POST: Replace every {{placeholder}} before running. No automatic retries.

cURL (BASH) | REQUEST EXAMPLE

```
: "${ZPAY_API_KEY:?Set your private server environment variable}"
ZPAY_BASE_URL="https://zpayswift.com/api"
curl --request POST \
  --url "$ZPAY_BASE_URL/public_api/bundle_create.php" \
  --header "Authorization: Bearer $ZPAY_API_KEY" \
  --header 'Accept: application/json' \
  --header 'Content-Type: application/json' \
  --data '{
  "offer_id": "{{offer_id}}",
  "bundle_number": "{{bundle_number}}",
  "idempotency_key": "{{client_request_id}}",
  "note": "Partner bundle"
}'
```

Create Bundle | Response

Read the response fields before taking the next action.

Illustrative response

Actual currency, prices and values come from the server. Additional fields may be returned.

HTTP 200 | JSON


```
{
  "ok": true,
  "code": "SUCCESS",
  "message": "Illustrative example.",
  "data": {
    "request_id": "YOUR_REQUEST_ID",
    "uid": "YOUR_ACCOUNT_ID",
    "status": "WAITING_ADMIN",
    "offer_id": "OFFER_ID_FROM_RESPONSE",
    "bundle_number": "01712345678",
    "amount": 100,
    "wallet_debit_amount": 100,
    "wallet_debit_currency": "BDT",
    "created_at": 1800000000
  }
}
```


Store these fields

request_id: save it with your business-order ID before displaying a final result.

status: WAITING_ADMIN means accepted and waiting, not successful delivery.

wallet_debit_amount / wallet_debit_currency: read the actual wallet denomination and amount. Do not infer an MYR debit from the BDT face value.

Next: track the service outcome

Read GET /public_api/request_status.php with the saved request_id. SUCCESS and FAILED are final; pending/processing are not. A status read is not an instruction to resubmit the create request.

If the POST response is missing or unreadable, the request may still exist. Review status/history before any deliberate retry.

Request Status

Read the canonical status of your own Topup or Bundle request.

GET ENDPOINT

```
https://zpayswift.com/api/public_api/request_status.php
```

Parameter	Type and meaning
`request_id`	**Required** | string The data.request_id returned by Create Topup or Create Bundle.

SUCCESS and FAILED are final. PENDING, WAITING_ADMIN and PROCESSING are not final.

UNKNOWN status_group has is_final=null: stop automatic retries and review it. A successful GET does not imply the service succeeded.

Suggested polling: once every 5-10 seconds with backoff and a bounded timeout. This is client guidance, not a published API quota.

cURL (BASH) | REQUEST EXAMPLE

```
: "${ZPAY_API_KEY:?Set your private server environment variable}"
ZPAY_BASE_URL="https://zpayswift.com/api"
curl --request GET \
  --url "$ZPAY_BASE_URL/public_api/request_status.php" \
  --header "Authorization: Bearer $ZPAY_API_KEY" \
  --header 'Accept: application/json' \
  --get \
  --data-urlencode 'request_id=YOUR_REQUEST_ID'
```

Illustrative response

Actual currency, prices and values come from the server. Additional fields may be returned.

HTTP 200 | JSON

```
{
  "ok": true,
  "code": "SUCCESS",
  "message": "Illustrative example.",
  "data": {
    "request_id": "YOUR_REQUEST_ID",
    "type": "TOPUP",
    "status": "SUCCESS",
    "status_group": "SUCCESS",
    "is_final": true,
    "updated_at": 1800000000,
    "checked_at": 1800000010
  }
}
```

Errors, status & recovery

Inspect both the HTTP result and the JSON envelope. An API call and a service outcome are different.

401 | API_KEY_REQUIRED / INVALID_API_KEY

Use an active full API key in the Authorization header. A masked key is not usable.

403 | API_KEY_DISABLED / API_NOT_ALLOWED

Check key status and ask Admin to enable API access. Topup and Bundle also have separate service permissions.

404 | OFFER_NOT_FOUND / REQUEST_NOT_FOUND / USER_NOT_FOUND

Check the ID. Request status only returns your own Topup/Bundle requests.

409 | Financial operation conflict

The request may already be in progress. Check its existing status/history before any further POST.

422 | VALIDATION_ERROR / INSUFFICIENT_BALANCE

Read message and data.field when present. Correct the input or balance only after checking for an existing request.

500 / 503 | WALLET_UNAVAILABLE / STATUS_UNAVAILABLE / SERVICE_UNAVAILABLE

For POST, the result may be unknown. Do not automatically resend. GET can be retried with backoff.

ERROR ENVELOPE

```
{
  "ok": false,
  "code": "VALIDATION_ERROR",
  "message": "Invalid topup number",
  "data": {
    "field": "topup_number"
  }
}
```

Some failures return data: [] instead of an object. Do not assume data.field exists.

Canonical request states

PENDING / WAITING_ADMIN: accepted and waiting. **PROCESSING:** in progress. **SUCCESS / FAILED:** terminal. An UNKNOWN status_group has is_final=null and needs review; never treat it as success.

Suggested polling: one GET every 5-10 seconds, with backoff and a bounded timeout. No fixed public API quota is advertised. Status reads do not perform refunds or settlement.

Reusable PHP client

PHP with cURL. Keep this code on your server. All calls shown below are examples, not executed by the PDF.

PHP | CLIENT + EXAMPLE CALLS

```php
<?php
function zpayRequest(string $method, string $path, array $params = []): array
{
    $key = getenv('ZPAY_API_KEY');
    if (!$key) { throw new RuntimeException('Set ZPAY_API_KEY'); }
    $url = 'https://zpayswift.com/api' . $path;
    if ($method === 'GET' && $params) {
        $url .= '?' . http_build_query($params);
    }
    $headers = ['Authorization: Bearer ' . $key, 'Accept: application/json'];
    $options = [
        CURLOPT_RETURNTRANSFER => true,
        CURLOPT_CONNECTTIMEOUT => 10,
        CURLOPT_TIMEOUT => 30,
        CURLOPT_CUSTOMREQUEST => $method,
    ];
    if ($method === 'POST') {
        $headers[] = 'Content-Type: application/json';
        $options[CURLOPT_POSTFIELDS] = json_encode($params, JSON_THROW_ON_ERROR);
    }
    $options[CURLOPT_HTTPHEADER] = $headers;
    $ch = curl_init($url);
    curl_setopt_array($ch, $options);
    $raw = curl_exec($ch);
    $http = curl_getinfo($ch, CURLINFO_HTTP_CODE);
    curl_close($ch);
    if ($raw === false) {
        throw new RuntimeException('Outcome unknown; review before retrying POST.');
    }
    $result = json_decode($raw, true, 512, JSON_THROW_ON_ERROR);
    if ($http < 200 || $http >= 300 || ($result['ok'] ?? false) !== true) {
        throw new RuntimeException($result['message'] ?? 'API request failed');
    }
    return $result['data'];
}

$balance = zpayRequest('GET', '/public_api/balance.php');
$offers = zpayRequest('GET', '/public_api/bundle_offers.php', ['limit' => 10]);
$status = zpayRequest('GET', '/public_api/request_status.php', [
    'request_id' => 'YOUR_SAVED_REQUEST_ID',
]);
// Live POST: replace placeholders; keep one stored order ID per business order.
$request = zpayRequest('POST', '/public_api/topup_create.php', [
    'topup_number' => 'YOUR_BD_NUMBER', 'operator' => 'GP', 'amount' => 20,
    'idempotency_key' => 'YOUR_PERSISTED_ORDER_ID',
]);
// Persist $request['request_id']; accepted does not mean delivered.
```

For a bundle POST, use /public_api/bundle_create.php with offer_id, bundle_number and idempotency_key (plus optional note). Consult the endpoint chapter for fields. On any ambiguous POST result, review status/history before retrying.

Reusable Node.js client

Node.js 18+ server-side fetch. No browser or mobile app should receive the private partner key.

NODE.JS | CLIENT + EXAMPLE CALLS

```
// Node.js 18+. Server-side only. Never expose the key to a public browser.
async function zpayRequest(method, path, params = {}) {
  const key = process.env.ZPAY_API_KEY;
  if (!key) throw new Error('Set ZPAY_API_KEY');
  const url = new URL('https://zpayswift.com/api' + path);
  const options = {
    method,
    headers: {Authorization: 'Bearer ' + key, Accept: 'application/json'},
    signal: AbortSignal.timeout(30000),
  };
  if (method === 'GET') {
    for (const [name, value] of Object.entries(params)) {
      url.searchParams.set(name, String(value));
    }
  } else {
    options.headers['Content-Type'] = 'application/json';
    options.body = JSON.stringify(params);
  }
  const response = await fetch(url, options);
  const result = await response.json();
  if (!response.ok || result.ok !== true) {
    throw new Error(result.message || 'API request failed');
  }
  return result.data;
}

const balance = await zpayRequest('GET', '/public_api/balance.php');
const offers = await zpayRequest('GET', '/public_api/bundle_offers.php', {
  limit: 10,
});
const status = await zpayRequest('GET', '/public_api/request_status.php', {
  request_id: 'YOUR_SAVED_REQUEST_ID',
});
// Live POST: replace placeholders; do not automatically retry on errors.
const request = await zpayRequest('POST', '/public_api/bundle_create.php', {
  offer_id: 'YOUR_ACTIVE_OFFER_ID', bundle_number: 'YOUR_BD_NUMBER',
  idempotency_key: 'YOUR_PERSISTED_ORDER_ID',
});
// Persist request.request_id, then read its status until terminal.
// Timeouts/5xx: a POST may have been accepted. Review before any retry.
```

Before going live

Confirm account/service permissions, valid recipient and current offer, correct currency, private key storage, persisted order/request IDs, bounded polling, and manual review of unknown POST outcomes. Download the matching Postman/OpenAPI files from the live Docs page.

Latest documentation: zpayswift.com/subadmin/ > API Docs