zCheck integration guide

zPayLink developer kit, 2026-10-15. Everything below is generated from the verified zPayLink developer documentation; examples use fictional values.

1. Product overview

Identity verification (KYC): create a verification, send your user to the hosted page (or upload documents server to server), then read the checks and the decision and receive signed webhooks.

> zCheck accepts test keys only while it is in development: live verification is switched off. With the default policy no verification is approved automatically; every case ends in manual review. Results in test mode are simulated (simulated: true). OCR, face comparison and liveness need providers that zPayLink connects per organization, and there is no government database check.

Aspect	Status (2026-10-15)
Implemented	yes
Reachable from the internet	yes (see the connectivity note)
Testing	yes, test keys with simulated results
Production	no: live verification is switched off
Disabled or restricted	live keys refused (403 live_processing_disabled); OCR, face and liveness need providers connected by zPayLink; no government database checks

Connectivity: zPayLink servers answer from most networks, but some international networks currently cannot open a connection to them. If your servers time out, test from another network and send zPayLink your public IP and a traceroute.

2. Prerequisites

- Organization admins and developers create keys in the zCheck console (Developers, API keys). The key is shown once; rotation keeps the old key valid for a grace period (24 hours by default, up to 7 days). Scopes: verifications:read, verifications:write (both by default) and verifications:submit for server-to-server uploads.
- A server-side environment (the APIs do not allow browser calls) with HTTPS outbound access to the base URL.
- An HTTPS endpoint on a public address for webhooks (where the product sends them).

3. Authentication

Send the key on every request: `Authorization: Bearer <key>`. Key format: `zck_test_…`.

```
export ZCHECK_API_KEY="<your key>"   # keep it in a secret store or environment variable
curl https://check.zpaylink.com/api/v1/verifications -H "Authorization: Bearer $ZCHECK_API_KEY"
```

Environment	Base URL
Test mode (test keys)	`https://check.zpaylink.com/api/v1`

4. Step-by-step integration

1. Ask zPayLink to create your organization; an admin or developer then creates a test key (zck_test_) in the zCheck console.
2. Create a verification with POST /verifications (external_customer_id, redirect_url, optional sandbox.scenario in test mode).
3. Send the user to verification_url; the page cannot be embedded in an iframe.
4. Add a webhook endpoint in the console, verify ZCheck-Signature and handle verification.pending_review, verification.approved and verification.rejected.
5. Read the checks with GET /verifications/{id}/result. With the default policy every case ends in manual review.

5. Idempotency, retries and rate limits

Create, cancel and resubmission accept an Idempotency-Key header (8 to 100 characters: letters, digits and _ - : .), scoped to your organization and mode and kept for 2 days. The same request replays the stored response; a different body returns 409 idempotency_conflict, and a request still in flight returns 409 idempotency_in_progress.

300 requests per minute per key (429 rate_limited with Retry-After: 60). The edge limits each IP to 20 requests per second on the API and 10 per second on uploads.

- Retry network errors, 5xx and 429 with the same idempotency key and growing pauses; on 429 wait at least Retry-After seconds.
- Do not retry 4xx validation errors unchanged.

6. API reference

Method	Path	Endpoint	Scope
POST	`/verifications`	Create a verification	`verifications:write`
GET	`/verifications`	List verifications	`verifications:read`
GET	`/verifications/{id}`	Retrieve a verification	`verifications:read`
GET	`/verifications/{id}/status`	Verification status	`verifications:read`

Method	Path	Endpoint	Scope
GET	`/verifications/{id}/result`	Verification result	`verifications:read`
GET	`/verifications/{id}/events`	Verification timeline	`verifications:read`
POST	`/verifications/{id}/cancel`	Cancel a verification	`verifications:write`
POST	`/verifications/{id}/resubmission`	Request resubmission	`verifications:write`
POST	`/verifications/{id}/session`	Issue a new link	`verifications:write`
POST	`/verifications/{id}/consent`	Server-to-server: record consent	`verifications:submit`
PUT	`/verifications/{id}/customer`	Server-to-server: customer details	`verifications:submit`
POST	`/verifications/{id}/files/{kind}`	Server-to-server: upload an image	`verifications:submit`
POST	`/verifications/{id}/submit`	Server-to-server: submit	`verifications:submit`
GET	`/events`	List webhook events	`verifications:read`

Create a verification: `POST /verifications`

Creates a verification and returns a one-time link to the hosted page. Every field is optional; unknown fields are rejected.

Item	Value
Method and path	`POST /verifications`
Base URL	`https://check.zpaylink.com/api/v1`
Authentication	`Authorization: Bearer <key>` , scope `verifications:write`
Idempotency-Key header	optional
Success	`201 Created`

Body (JSON)

Name	Type	Required	Description
`external_customer_id`	`string`		Your user id, up to 64 characters
`external_reference`	`string`		Your reference, up to 100 characters
`document_types`	`string[]`		Subset of national_id_card, birth_certificate, passport, driving_license allowed by your policy
`redirect_url`	`string`		HTTPS URL for the return button, up to 500 characters
`locale`	`string`		"fa" (default) or "en"
`expires_in_minutes`	`integer`		10 to 10080, capped by the policy (default 1440)
`customer`	`object`		first_name, last_name (1 to 60) and national_id (valid 10-digit Iranian ID); prefilled values cannot be changed by the user
`metadata`	`object`		Up to 4 KB; values string (200), number or boolean

Name	Type	Required	Description
`sandbox.scenario`	`string`		Test keys only: approve, name_mismatch, nid_mismatch, fuzzy_name, ocr_unreadable, face_mismatch, face_borderline, liveness_spoof, document_expired, suspicious_document, provider_error, provider_timeout

Example request

```
curl -X POST https://check.zpaylink.com/api/v1/verifications \
  -H "Authorization: Bearer $ZCHECK_API_KEY" \
  -H "Idempotency-Key: $(uuidgen)" \
  -H "Content-Type: application/json" \
  -d '{"external_customer_id":"user-2041","external_reference":"onboarding-2041","document_types":["national_id_card","passport"],"redirect_url":"https://app.example.com/kyc/done","locale":"fa","metadata":{"plan":"standard"},"sandbox":{"scenario":"approve"}}'
```

Example response (201 Created)

```
{
  "id": "6f1c2a9e-3b47-4d1a-9c55-2e8b7a0d4f13",
  "reference": "ZC-7K2M9QX4",
  "external_customer_id": "user-2041",
  "external_reference": "onboarding-2041",
  "mode": "test",
  "status": "CREATED",
  "attempt": 1,
  "document_type": null,
  "simulated": true,
  "decision": null,
  "created_at": "2026-10-15T09:00:00.000Z",
  "updated_at": "2026-10-15T09:00:00.000Z",
  "completed_at": null,
  "submitted_at": null,
  "decided_at": null,
  "allowed_document_types": ["national_id_card", "passport"],
  "liveness_mode": "selfie",
  "session_expires_at": "2026-10-16T09:00:00.000Z",
  "max_attempts": 3,
  "policy_version": 1,
  "customer": null,
  "metadata": { "plan": "standard" },
  "verification_url": "https://check.zpaylink.com/v/HOSTED_TOKEN"
}
```

Errors

- `422 validation_error`
- `403 live_processing_disabled`
- `409 idempotency_conflict`
- `409 idempotency_in_progress`

Notes

- verification_url is returned on creation only (null on an idempotent replay); POST /verifications/{id}/session issues a new link.

List verifications: `GET /verifications`

Newest first with cursor pagination (next_cursor).

Item	Value
Method and path	`GET /verifications`
Base URL	`https://check.zpaylink.com/api/v1`
Authentication	`Authorization: Bearer <key>`, scope `verifications:read`
Success	`200 OK`

Query parameters

Name	Type	Required	Description
`status`	`string`		Comma-separated statuses, e.g. MANUAL_REVIEW,APPROVED
`external_customer_id`	`string`		Exact match
`external_reference`	`string`		Exact match
`created_from / created_to`	`ISO date`		Creation time range
`limit`	`integer`		1 to 100, default 25
`cursor`	`string`		next_cursor of the previous page

Example request

```
curl https://check.zpaylink.com/api/v1/verifications \
  -H "Authorization: Bearer $ZCHECK_API_KEY"
```

Example response (200 OK)

```
{
  "data": [
    { "id": "6f1c2a9e-3b47-4d1a-9c55-2e8b7a0d4f13", "reference": "ZC-7K2M9QX4", "status":
"MANUAL_REVIEW", "mode": "test" }
  ],
  "has_more": false,
  "next_cursor": null
}
```

Errors

- `422 validation_error`

Notes

- List items are full verification objects; the example is shortened.

Retrieve a verification: `GET /verifications/{id}`

Status, decision summary, configuration and the masked customer. Never contains images or the full national ID.

Item	Value
Method and path	`GET /verifications/{id}`
Base URL	`https://check.zpaylink.com/api/v1`
Authentication	`Authorization: Bearer <key>`, scope `verifications:read`
Success	`200 OK`

Path parameters

Name	Type	Required	Description
`id`	`uuid`	yes	Verification id

Example request

```
curl https://check.zpaylink.com/api/v1/verifications/6f1c2a9e-3b47-4d1a-9c55-2e8b7a0d4f13
\
  -H "Authorization: Bearer $ZCHECK_API_KEY"
```

Example response (200 OK)

```
{
  "id": "6f1c2a9e-3b47-4d1a-9c55-2e8b7a0d4f13",
  "reference": "ZC-7K2M9QX4",
  "external_customer_id": "user-2041",
  "external_reference": "onboarding-2041",
  "mode": "test",
  "status": "CREATED",
  "attempt": 1,
  "document_type": null,
  "simulated": true,
  "decision": null,
  "created_at": "2026-10-15T09:00:00.000Z",
  "updated_at": "2026-10-15T09:00:00.000Z",
  "completed_at": null,
  "submitted_at": null,
  "decided_at": null,
  "allowed_document_types": ["national_id_card", "passport"],
  "liveness_mode": "selfie",
  "session_expires_at": "2026-10-16T09:00:00.000Z",
  "max_attempts": 3,
  "policy_version": 1,
  "customer": null,
  "metadata": { "plan": "standard" }
}
```

Errors

- `404 not_found`

Verification status: `GET /verifications/{id}/status`

A light status read for polling.

Item	Value
Method and path	`GET /verifications/{id}/status`
Base URL	`https://check.zpaylink.com/api/v1`
Authentication	`Authorization: Bearer <key>`, scope `verifications:read`
Success	`200 OK`

Path parameters

Name	Type	Required	Description
`id`	`uuid`	yes	Verification id

Example request

```
curl https://check.zpaylink.com/api/v1/verifications/6f1c2a9e-3b47-4d1a-9c55-2e8b7a0d4f13/status \
  -H "Authorization: Bearer $ZCHECK_API_KEY"
```

Example response (200 OK)

```
{
  "id": "6f1c2a9e-3b47-4d1a-9c55-2e8b7a0d4f13",
  "reference": "ZC-7K2M9QX4",
  "status": "PROCESSING",
  "attempt": 1,
  "updated_at": "2026-10-15T09:12:40.000Z"
}
```

Errors

- `404 not_found`

Verification result: `GET /verifications/{id}/result`

The decision and every check: document_quality, ocr_extraction, first_name_match, last_name_match, national_id_match, face_match, liveness, presentation_attack and document_authenticity. A check is pass, fail, suspicious, inconclusive, unavailable, not_required or error.

Item	Value
Method and path	`GET /verifications/{id}/result`
Base URL	`https://check.zpaylink.com/api/v1`
Authentication	`Authorization: Bearer <key>` , scope `verifications:read`
Success	`200 OK`

Path parameters

Name	Type	Required	Description
`id`	`uuid`	yes	Verification id

Example request

```
curl https://check.zpaylink.com/api/v1/verifications/6f1c2a9e-3b47-4d1a-9c55-2e8b7a0d4f13/result \
  -H "Authorization: Bearer $ZCHECK_API_KEY"
```

Example response (200 OK)


```
{
  "id": "6f1c2a9e-3b47-4d1a-9c55-2e8b7a0d4f13",
  "reference": "ZC-7K2M9QX4",
  "status": "MANUAL_REVIEW",
  "final": false,
  "simulated": true,
  "attempt": 1,
  "policy_version": 1,
  "decision": {
    "outcome": "MANUAL_REVIEW",
    "automatic": true,
    "reasons": [{ "code": "auto_approve_disabled", "message": "All checks passed. Automatic approval is off in this policy: a reviewer decides." }],
    "manual_reason": null,
    "engine_version": "zcheck-decision-1",
    "decided_at": "2026-10-15T09:13:05.000Z"
  },
  "checks": {
    "national_id_match": { "status": "pass", "score": null, "threshold": null, "raw_score": null, "calibrated": false, "provider": "Simulated (sandbox)", "simulated": true, "reasons": [], "entered_last4": "1234", "extracted_last4": "1234", "match_type": "exact" }
  },
  "providers": ["Simulated (sandbox)"]
}
```


Errors

- `404 not_found`

Notes

- checks contains one entry per check; the example shows one.

Verification timeline: `GET /verifications/{id}/events`

Every step of the verification in order; reviewer names and internal notes are not included.

Item	Value
Method and path	`GET /verifications/{id}/events`
Base URL	`https://check.zpaylink.com/api/v1`
Authentication	`Authorization: Bearer <key>`, scope `verifications:read`
Success	`200 OK`

Path parameters

Name	Type	Required	Description
`id`	`uuid`	yes	Verification id

Example request

```
curl https://check.zpaylink.com/api/v1/verifications/6f1c2a9e-3b47-4d1a-9c55-2e8b7a0d4f13/events \
  -H "Authorization: Bearer $ZCHECK_API_KEY"
```

Example response (200 OK)

```
{
  "data": [
    { "type": "created", "at": "2026-10-15T09:00:00.000Z", "actor": "api" },
    { "type": "submitted", "at": "2026-10-15T09:12:40.000Z", "actor": "customer" },
    { "type": "review_required", "at": "2026-10-15T09:13:05.000Z", "actor": "system" }
  ]
}
```

Errors

- `404 not_found`

Cancel a verification: `POST /verifications/{id}/cancel`

Cancels any verification that is not yet final (APPROVED, REJECTED, EXPIRED or CANCELLED).

Item	Value
Method and path	`POST /verifications/{id}/cancel`
Base URL	`https://check.zpaylink.com/api/v1`
Authentication	`Authorization: Bearer <key>`, scope `verifications:write`
Idempotency-Key header	optional
Success	`200 OK`

Path parameters

Name	Type	Required	Description
`id`	`uuid`	yes	Verification id

Body (JSON)

Name	Type	Required	Description
`reason`	`string`		Optional, up to 300 characters

Example request

```
curl -X POST https://check.zpaylink.com/api/v1/verifications/6f1c2a9e-3b47-4d1a-9c55-2e8b7a0d4f13/cancel \
  -H "Authorization: Bearer $ZCHECK_API_KEY" \
  -H "Idempotency-Key: $(uuidgen)" \
  -H "Content-Type: application/json" \
  -d '{"reason":"User closed the account"}'
```

Example response (200 OK)

```
{ "id": "6f1c2a9e-3b47-4d1a-9c55-2e8b7a0d4f13", "reference": "ZC-7K2M9QX4", "status": "CANCELLED" }
```

Errors

- `409 invalid_state`
- `404 not_found`

Notes

- Returns the full verification object; the example is shortened.

Request resubmission: `POST /verifications/{id}/resubmission`

Asks the user to try again (from MANUAL_REVIEW, REJECTED, ERROR or PROCESSING): a new attempt and a new link. The reason is shown to the user.

Item	Value
Method and path	`POST /verifications/{id}/resubmission`
Base URL	`https://check.zpaylink.com/api/v1`
Authentication	`Authorization: Bearer <key>`, scope `verifications:write`
Idempotency-Key header	optional
Success	`200 OK`

Path parameters

Name	Type	Required	Description
`id`	`uuid`	yes	Verification id

Body (JSON)

Name	Type	Required	Description
`reason`	`string`	yes	At least 3 characters

Example request

```
curl -X POST https://check.zpaylink.com/api/v1/verifications/6f1c2a9e-3b47-4d1a-9c55-2e8b7a0d4f13/resubmission \
  -H "Authorization: Bearer $ZCHECK_API_KEY" \
  -H "Idempotency-Key: $(uuidgen)" \
  -H "Content-Type: application/json" \
  -d '{"reason":"The document photo is blurred."}'
```

Example response (200 OK)

```
{ "id": "6f1c2a9e-3b47-4d1a-9c55-2e8b7a0d4f13", "status": "PENDING_UPLOAD", "attempt": 2, "verification_url": "https://check.zpaylink.com/v/HOSTED_TOKEN" }
```

Errors

- `409 invalid_state`
- `409 retry_limit`
- `422 validation_error`

Issue a new link: `POST /verifications/{id}/session`

A new hosted-page link (the previous one stops working), for CREATED, PENDING_UPLOAD, PENDING_LIVENESS or EXPIRED verifications.

Item	Value
Method and path	`POST /verifications/{id}/session`
Base URL	`https://check.zpaylink.com/api/v1`
Authentication	`Authorization: Bearer <key>`, scope `verifications:write`
Success	`200 OK`

Path parameters

Name	Type	Required	Description
`id`	`uuid`	yes	Verification id

Example request

```
curl -X POST https://check.zpaylink.com/api/v1/verifications/6f1c2a9e-3b47-4d1a-9c55-2e8b7a0d4f13/session \
  -H "Authorization: Bearer $ZCHECK_API_KEY"
```

Example response (200 OK)

```
{
  "verification_url": "https://check.zpaylink.com/v/HOSTED_TOKEN",
  "session_expires_at": "2026-10-16T10:00:00.000Z"
}
```

Errors

- `409 invalid_state`

Server-to-server: record consent: `POST /verifications/{id}/consent`

First step when you collect documents in your own app instead of the hosted page.

Item	Value
Method and path	`POST /verifications/{id}/consent`
Base URL	`https://check.zpaylink.com/api/v1`
Authentication	`Authorization: Bearer <key>`, scope `verifications:submit`
Success	`200 OK`

Path parameters

Name	Type	Required	Description
`id`	`uuid`	yes	Verification id

Body (JSON)

Name	Type	Required	Description
`accepted`	`boolean`	yes	Must be true
`consent_version`	`string`		The version of the text the user accepted

Example request

```
curl -X POST https://check.zpaylink.com/api/v1/verifications/6f1c2a9e-3b47-4d1a-9c55-2e8b7a0d4f13/consent \
  -H "Authorization: Bearer $ZCHECK_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{"accepted":true,"consent_version":"2026-10"}'
```

Example response (200 OK)

```
{ "ok": true }
```

Errors

- `422 consent_required`
- `409 invalid_state`

Server-to-server: customer details: `PUT /verifications/{id}/customer`

Name, national ID and document type, after consent.

Item	Value
Method and path	`PUT /verifications/{id}/customer`
Base URL	`https://check.zpaylink.com/api/v1`
Authentication	`Authorization: Bearer <key>`, scope `verifications:submit`
Success	`200 OK`

Path parameters

Name	Type	Required	Description
`id`	`uuid`	yes	Verification id

Body (JSON)

Name	Type	Required	Description
`first_name`	`string`	yes	1 to 60 characters
`last_name`	`string`	yes	1 to 60 characters
`national_id`	`string`	yes	Valid 10-digit Iranian national ID
`document_type`	`string`		One of the allowed document types

Example request

```
curl -X PUT https://check.zpaylink.com/api/v1/verifications/6f1c2a9e-3b47-4d1a-9c55-2e8b7a0d4f13/customer \
  -H "Authorization: Bearer $ZCHECK_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{"first_name":"سارا","last_name":"رضایی","national_id":"0012345679","document_type":"national_id_card"}'
```

Example response (200 OK)

```
{ "ok": true }
```

Errors

- `409 consent_required`
- `422 validation_error`
- `409 invalid_state`

Server-to-server: upload an image: `POST /verifications/{id}/files/{kind}`

Uploads one image. National ID cards and driving licences need front and back; passports and birth certificates need the front.

Item	Value
Method and path	`POST /verifications/{id}/files/{kind}`
Base URL	`https://check.zpaylink.com/api/v1`
Authentication	`Authorization: Bearer <key>`, scope `verifications:submit`
Success	`201 Created`

Path parameters

Name	Type	Required	Description
`id`	`uuid`	yes	Verification id
`kind`	`string`	yes	front, back, selfie or liveness

Query parameters

Name	Type	Required	Description
`seq / challenge`	`string`		For live-capture frames only

Body (binary)

The raw image bytes (the type is detected from the content). Up to 8 MB by default; documents at least 800 px on the short side, selfies 480 px; no animated images.

Example request

```
curl -X POST https://check.zpaylink.com/api/v1/verifications/6f1c2a9e-3b47-4d1a-9c55-2e8b7a0d4f13/files/front \
  -H "Authorization: Bearer $ZCHECK_API_KEY" \
  -H "Content-Type: application/octet-stream" \
  --data-binary @document-front.jpg
```

Example response (201 Created)

```
{ "ok": true, "kind": "front", "seq": 0, "width": 1600, "height": 1010, "issues": [] }
```

Errors

- `409 steps_incomplete`
- `413 file_too_large`
- `415 unsupported_media_type`
- `422 resolution_too_low`
- `422 not_required`

Server-to-server: submit: `POST /verifications/{id}/submit`

Starts processing once every required image is uploaded.

Item	Value
Method and path	`POST /verifications/{id}/submit`
Base URL	`https://check.zpaylink.com/api/v1`
Authentication	`Authorization: Bearer <key>`, scope `verifications:submit`
Success	`200 OK`

Path parameters

Name	Type	Required	Description
`id`	`uuid`	yes	Verification id

Example request

```
curl -X POST https://check.zpaylink.com/api/v1/verifications/6f1c2a9e-3b47-4d1a-9c55-2e8b7a0d4f13/submit \
  -H "Authorization: Bearer $ZCHECK_API_KEY"
```

Example response (200 OK)

```
{ "ok": true, "status": "PROCESSING" }
```

Errors

- `409 steps_incomplete`
- `422 uploads_incomplete`

List webhook events: `GET /events`

The events sent to your webhooks, for reconciliation.

Item	Value
Method and path	`GET /events`
Base URL	`https://check.zpaylink.com/api/v1`
Authentication	`Authorization: Bearer <key>` , scope `verifications:read`
Success	`200 OK`

Query parameters

Name	Type	Required	Description
`type`	`string`		One event type
`verification_id`	`uuid`		Only events of this verification
`starting_after`	`string`		An event id (evt_…)
`limit`	`integer`		1 to 100, default 25

Example request

```
curl https://check.zpaylink.com/api/v1/events \
  -H "Authorization: Bearer $ZCHECK_API_KEY"
```

Example response (200 OK)

```
{
  "data": [
    { "id": "evt_9f2c4a7e1b3d5f8a0c2e4b6d", "type": "verification.pending_review", "creat
ed_at": "2026-10-15T09:13:05.000Z", "data": { "verification": { "id": "6f1c2a9e-3b47-4d1a
-9c55-2e8b7a0d4f13", "status": "MANUAL_REVIEW" } } }
  ],
  "has_more": false
}
```

7. Webhooks

> Webhooks carry no names, national IDs or images; read details with the API.

Event	When
`verification.created`	A verification was created
`verification.processing`	The user or your server submitted it
`verification.pending_review`	It needs a manual decision
`verification.approved`	Approved (automatically or by a reviewer)
`verification.rejected`	Rejected (automatically or by a reviewer)
`verification.expired`	The session expired before submission
`verification.failed`	Processing failed (status ERROR)
`verification.cancelled`	Cancelled
`verification.resubmission_requested`	A new attempt was requested

Signature header: `ZCheck-Signature: t=<unix seconds>,v1=<hex HMAC-SHA256(secret, "<t>.<raw body>")>`. Return any 2xx within 10 seconds. Redirects are not followed. First attempt at once; after a failure the next waits 1 minute, 5 minutes, 30 minutes, 2 hours, 6 hours, 12 hours and 24 hours. After 8 failed attempts the delivery stops. Add HTTPS endpoints in the zCheck console per mode and choose their events; the whsec_ secret is shown once. The console can also create a built-in test receiver and send a test.ping.

Full details and verification code: webhooks.md.

8. Error handling

Use the machine-readable code, not the message.

```
{
  "error": {
    "code": "validation_error",
    "message": "customer.national_id is not a valid Iranian national ID.",
    "details": [{ "field": "customer.national_id", "message": "The check digit does not m
atch: the number contains a typing or reading error." }]
  },
  "request_id": "req_4f9a1c7e2b8d3a6f0e5c"
}
```

All codes: error-codes.md.

9. Important integration notes

- Live verification is switched off: live keys are refused with 403 live_processing_disabled.
- The return button sends the user to redirect_url with status=submitted; the outcome is never in the URL. Use webhooks or the API.
- Webhooks carry no names, national IDs or images; read details with the API.

10. Security

- Keep API keys and webhook secrets on your server, in a secret store or environment variables; never in front-end code, mobile apps or source control.
- Use separate keys per environment and per system, with the narrowest scopes you need (zGate read-only keys, zCheck verifications:submit only where you upload server to server).
- Rotate keys when people or systems change: zCheck and zPay keep the old key valid for a grace period during rotation; revoke at once if a key may be exposed.
- Verify every webhook signature with a constant-time comparison and a 5 minute timestamp window, and deduplicate on the event id.
- Webhook endpoints must be HTTPS on a public address; zPayLink refuses private and internal addresses and does not follow redirects.
- Treat webhooks as notifications: confirm amounts and statuses with the API before you credit a user or ship an order.
- zCheck never returns images or full national IDs through the API or webhooks; keep the personal data you collect to the minimum.

11. Troubleshooting

Symptom	What to do
401 unauthorized / Invalid API key	Send exactly "Authorization: Bearer <key>"; check you use a test key on the test URL and a live key on production.

Symptom	What to do
403 insufficient_scope	The key lacks the scope listed for the endpoint; ask for a key with that scope.
422 validation_error / invalid_request	Fix the request: details lists the fields. Unknown body fields are rejected.
409 idempotency_conflict	The Idempotency-Key was used with a different request; use a new key for a new operation.
429 rate_limited	Wait at least Retry-After seconds, then retry with the same Idempotency-Key.
Webhook signature does not verify	Verify the raw body bytes (before JSON parsing), use the secret of that endpoint and keep your server clock in sync (5 minute window).
Connection timeouts from your servers	Some international networks cannot reach zPayLink servers at the moment. Test from another network and contact zPayLink with your server's public IP and a traceroute.