

Part G: API reference (v1)

Status: October 2026, API version 1.

Base URL: `https://dffms.org/api/v1`

Table of contents

1. [Overview](#)
2. [Authentication](#)
3. [Authorization, roles and approvals](#)
4. [Response envelope and error format](#)
5. [Pagination and filtering](#)
6. [Idempotency](#)
7. [Rate limits and size limits](#)
8. [Webhooks](#)
9. [Versioning](#)
10. [Endpoint index](#)
11. [Endpoint reference](#)
 - [11.1 Platform: health, readiness, OpenAPI](#)
 - [11.2 Public contact form](#)
 - [11.3 Authentication \(Better Auth\)](#)
 - [11.4 Generic resources](#)
 - [11.5 Web push subscriptions](#)
 - [11.6 Finance applications](#)
 - [11.7 Financing cases](#)
 - [11.8 Finance transactions](#)
 - [11.9 Reconciliation](#)
 - [11.10 Reporting exports](#)
 - [11.11 Marketplace decision](#)
 - [11.12 Provider webhooks](#)
12. [Appendix A: Generic resource catalogue](#)

Placeholder conventions used in every example:

- Session cookie: `<session-cookie>` . Tenant: `<tenantId>` . Idempotency key: `<idempotency-key>` .
 - Identifiers are UUIDs in the code (validated with `z.uuid()`), not prefixed strings. Examples use placeholder UUIDs of the shape `0190f2c4-6b3e-7a1d-8c2f-00000000000N` , or angle-bracket names such as `<applicationId>` .
 - Person examples use `@demo.dffms.org` addresses only.
 - Responses are abbreviated where a record has many columns; `...` marks omitted fields.
-

1. Overview

- **Purpose.** `/api/v1` serves integrations, webhooks and future clients. Server Components and Server Actions serve the browser UI; this API is a parallel surface that enforces the same tenant, role, scope and step-up rules.
 - **Protocol.** HTTPS, JSON request and response bodies (`application/json; charset=utf-8`), except:
 - `GET /api/v1/openapi` returns a raw OpenAPI document (no envelope).
 - `GET /api/v1/health` returns HTML when the client sends `Accept: text/html` .
 - `/api/v1/auth/*` returns Better Auth's own JSON, not the DFFMS envelope.
 - `POST /api/v1/webhooks/{provider}` accepts provider payloads (JSON, or form-encoded for Twilio; the route itself reads the raw body).
 - **Tenancy.** Business data is tenant-scoped. Most routes take the tenant from the `X-DFFMS-Tenant-Id` header (UUID). Some routes (reporting, marketplace decision, push subscriptions) do not read this header; see [Section 2.3](#).
 - **Data model conventions.**
 - Money is a string of integer minor units (for NGN, kobo), for example `"2500000"` for NGN 25,000.00. Currency is an ISO 4217 code in uppercase.
 - Database `bigint` values are serialised as decimal strings.
 - Dates are `YYYY-MM-DD` ; timestamps are ISO 8601 with offset.
 - **Background processing.** Provider calls and report generation run asynchronously through a transactional outbox (see [Section 8](#)).
-

2. Authentication

2.1 User sessions (session cookie)

- Authentication is by **Better Auth** session cookie, with `basePath /api/v1/auth`.
- Cookie prefix is `dffms`. In production secure cookies are enabled; the cookie name is `__Secure-dffms.session_token`. Cookies are host-only (cross-subdomain cookies are disabled).
- The Better Auth plugins configured are `twoFactor` and `passkey`. No bearer-token plugin is configured, so the API reads only the session cookie. (Inferred from the plugin list; the code does not state this explicitly.)
- Sign-in methods:
 - Email and password. `requireEmailVerification` is `true`, so unverified accounts cannot sign in. Minimum password length 6, maximum 128.
 - Google sign-in, when enabled for the platform. `disableImplicitSignUp` is `true`, so Google sign-in does not create accounts.
 - Passkeys (WebAuthn), through `@better-auth/passkey`.
- Session lifetime is 4 hours. The session is refreshed every 15 minutes of activity. `freshAge` is 5 minutes (Better Auth default in this configuration).
- Staff accounts are invite-only (accepted invitation converts to a membership). Public accounts have `accessType PUBLIC_USER`.

Example of the cookie header sent on every authenticated request:

```
Cookie: __Secure-dffms.session_token=<session-cookie>
```

2.2 Multi-factor authentication and step-up

Two different MFA concepts appear in the code and must not be confused:

Concept	Where enforced	Rule
Tenant MFA enrolment	<code>resolveActorFromRequest</code> , <code>requireAuthorization</code> , <code>requireReportingAccess</code> , <code>requirePlatformSuperAdmin</code>	If the tenant's security policy sets <code>requireMfaEnrollment</code> , the account must have 2FA enrolled, otherwise <code>MFA_REQUIRED</code> . Default policy value is <code>false</code> .
Step-up for high-risk actions	<code>authorize()</code> ; finance service <code>requireStepUp</code>	Actions <code>approve</code> , <code>reverse</code> , <code>refund</code> , <code>payout</code> , <code>configure</code> , <code>export</code> require an MFA-enrolled account and a session younger than the action's freshness window.

Freshness windows as coded:

Path	Window	Failure
Domain <code>authorize()</code> step-up actions	15 minutes since <code>session.createdAt</code>	403 with reason <code>fresh_auth_required</code> (mapped to <code>FRESH_AUTH_REQUIRED</code> or <code>FORBIDDEN</code> depending on route)
Finance service (<code>requireStepUp</code>) and finance wrapper (<code>requireFreshSession</code>)	5 minutes since <code>session.createdAt</code>	401 <code>FRESH_AUTH_REQUIRED</code> from the service; the wrapper returns 401 <code>UNAUTHENTICATED</code>
Platform Super Administrator access management and marketplace decision	5 minutes (<code>requireFreshSession</code>)	401 <code>FRESH_AUTH_REQUIRED</code>

Passkeys: when a tenant enforces MFA enrolment, `requirePlatformSuperAdmin` also requires at least one registered passkey for the Platform Super Administrator (`MFA_REQUIRED` , 401).

To satisfy a fresh-authentication requirement, the client signs in again (`POST /api/v1/auth/sign-in/email` , or the passkey or Google flow) and then calls the endpoint within the window.

2.3 Tenant selection

- `X-DFFMS-Tenant-Id: <tenantId>` is required on: generic resources (`/api/v1/{resource}`), finance applications, financing cases, finance transactions, reconciliation sync.
 - The header must be a UUID. Missing or invalid values return 400 `VALIDATION_ERROR` on generic and finance-application routes, and 422 `VALIDATION_ERROR` on the finance-transaction and reconciliation routes (see [Section 4](#)).
 - The actor must hold an active role assignment in that tenant. Otherwise 403 `FORBIDDEN` .
- Reporting, marketplace decision and push-subscription routes **ignore** the header. They resolve the tenant from the account:
 - Staff: the oldest `ACTIVE` membership of the account (`getPortalAccess`).
 - Public users (push subscriptions): the public marketplace tenant `00000000-0000-0000-0000-000000000000` .
 - Platform Super Administrator decision: the portal access tenant as above.
- Tenant identifiers are never inferred from client-supplied record identifiers. Record lookups are constrained by `tenantId` in every query.

2.4 Unauthenticated endpoints

- `GET /api/v1/health` , `GET /api/v1/ready` , `GET /api/v1/openapi` .
- `POST /api/v1/contact` : no session, but a mutation Origin check applies.
- `POST /api/v1/webhooks/{provider}` : no session; authenticated by provider HMAC signature.
- `/api/v1/auth/*` : Better Auth endpoints, each with its own rules.

2.5 Origin checks (CSRF defence for mutations)

Every state-changing request that uses a session cookie must send an `Origin` header that is in the allowed-origin set.

- Requests from an origin that is not allowed are rejected: 400 `VALIDATION_ERROR` on the generic-resource and finance-application routes, 403 `ORIGIN_DENIED` elsewhere.
- CORS preflight (`OPTIONS`) on `/api/*` from an origin that is not allowed returns 403.

Example:

```
Origin: https://dffms.org
```

3. Authorization, roles and approvals

3.1 Role abbreviations

Abbr.	Role code	Notes
PSA	PLATFORM_SUPER_ADMIN	Bypasses most role checks; still subject to MFA and fresh-auth rules on step-up actions.
CSA	CLIENT_SYSTEM_ADMIN	Tenant administrator.
PA	PROGRAMME_ADMIN	
SCA	STATE_COORDINATOR_ADMIN	Can approve <code>finance_application</code> decisions (committee).
IA	INSTITUTION_ADMIN	
FEO	FINANCE_ENTRY_OFFICER	Enters transactions and requests corrections.
FA	FINANCE_APPROVER	Approves and rejects transactions; finance committee decisions.
MA	MONITORING_ANALYST	Read and some write access to monitoring resources and reports.
AC	AUDITOR_COMPLIANCE	Read-only in the domain policy.
EDV	EXECUTIVE_DONOR_VIEWER	Read-only.
IV	INVESTOR_VIEWER	Read-only.
SUP	SUPPORT	Read-only.

The domain policy treats `AUDITOR_COMPLIANCE` , `EXECUTIVE_DONOR_VIEWER` , `INVESTOR_VIEWER` and `SUPPORT` as read-only roles.

3.2 Authorisation layers

1. **Role matrix** (`authorize()`): a per-role, per-resource-kind write permission map; `approve` , `reverse` , `refund` , `payout` , `configure` , `export` are finance or configuration actions.
2. **Scope (ABAC)**: records carry `programmeId` , `stateId` , `sapzId` , `institutionId` . Generic list queries filter by the actor's scope arrays. Some resources use relation-based data scope (for example, a case register is visible only through its financing case).
3. **Saved role policies** (`permission_policies`): consulted only by the finance-application, financing-case and decision routes (`actorPolicyPermits`). The generic resource route does not consult them.
4. **Row-level security** in PostgreSQL is the final tenant isolation boundary (`withTenantDatabase`).

3.3 Maker-checker

- **Finance transactions:** the creator cannot approve or reject their own transaction (`MAKER_CHECKER_VIOLATION` , 403). PSA is exempt.
 - **Corrections (reversal, refund):** created as `PENDING_APPROVAL` and need an approval through `/approve` , subject to the same maker-checker rule.
 - **Committee decision on a finance application:** the requester of the pending approval and the applicant cannot decide (`MAKER_CHECKER_REQUIRED` , 403). PSA is **not** exempt here.
 - **Financing case creation:** requires a completed committee approval with a recorded decider.
-

4. Response envelope and error format

4.1 Success

```
{
  "data": { "...": "..." },
  "error": null,
  "requestId": "<request-id>"
}
```

Responses carry `Cache-Control: no-store` and `X-Request-Id` . `X-Request-Id` is echoed when supplied (validated against `^[A-Za-z0-9._:-]{8,128}$` ; otherwise a UUID is generated).

4.2 Error

```
{
  "data": null,
  "error": {
    "code": "VALIDATION_ERROR",
    "message": "Resource validation failed",
    "fields": { "name": "Required" }
  },
  "requestId": "<request-id>"
}
```

`fields` is present only for validation errors. Keys are field paths; `request` is used when the path is empty.

4.3 Error codes

Codes below are the values found in the code. Each endpoint section lists the codes it can return.

Code	Typical HTTP status	Meaning
UNAUTHENTICATED	401	No session. Also used for "fresh authentication required" in the finance wrapper.
MFA_REQUIRED	401 or 403	Account has no MFA enrolment where required. Status depends on the route (see endpoint sections).
FRESH_AUTH_REQUIRED	401 or 403	Session is older than the freshness window.
FORBIDDEN	403	Role, scope or policy does not permit the action.
ORIGIN_DENIED	403	Origin not in the allowed set.
IDEMPOTENCY_KEY_REQUIRED	400	Idempotency-Key header missing or malformed.
IDEMPOTENCY_CONFLICT	409	Same idempotency key used with different data, or a duplicate transaction reference.
VALIDATION_ERROR	400, 422	Request body or parameter failed validation. Status varies by route.
NOT_FOUND	404	Record missing, or outside the actor's scope. Scope-denied and missing records are indistinguishable.
INVALID_STATE	409	Record not in a state that allows the action.
STALE_STATE	409	Expected lifecycle value no longer matches (optimistic concurrency).
CONFLICT	409	Unique constraint violation on a generic create.
FEATURE_UNAVAILABLE	501 or 503	Resource is gated, reporting is disabled, or a dependency is unavailable.
PROVIDER_DISABLED	503	Webhook provider secret not configured.
NOT_READY	503 (readiness) or 409 (report download)	Dependency unavailable, or report not ready.
RATE_LIMITED	429	Contact-form limit.
LIMIT_REACHED	429	Report export concurrency or daily limit.
PAYLOAD_TOO_LARGE	413	Body exceeds the route's limit.

Code	Typical HTTP status	Meaning
INVALID_CONTENT_TYPE	400 or 415	Body is not <code>application/json</code> .
INTERNAL_ERROR	500	Unhandled error. Message is generic by design.
WORKER_FAILED	500	Cron worker failed.
UNAUTHORIZED	401	Worker secret mismatch or missing.

Fields that appear on specific routes:

- Finance, financing and application errors use a domain code in `error.code`, for example `UNBALANCED_JOURNAL`, `ACCOUNT_INVALID`, `INVALID_STATE_TRANSITION`, `FINAL_TERMS_INVALID`, `ELIGIBILITY_FAILED`. Those are listed per endpoint.
- Better Auth errors follow Better Auth's own format.

5. Pagination and filtering

5.1 Pagination (generic list and reporting list)

Parameter	Type	Default	Rules
<code>limit</code>	integer	25	Digits only, 1 to 100. Otherwise 400 <code>VALIDATION_ERROR</code> .
<code>cursor</code>	string	none	Generic lists: a UUID (the last <code>id</code> of the previous page). Reporting: a UUID. Anything else is 400.

Response shape (generic list):

```
{
  "data": {
    "items": [ { "id": "...", "...": "..." } ],
    "nextCursor": "0190f2c4-6b3e-7a1d-8c2f-00000000000a",
    "total": null
  },
  "error": null,
  "requestId": "<request-id>"
}
```

Reporting list returns `{ "items": [...], "nextCursor": <uuid or null> }` (no `total` field).

Ordering is by `id` descending (time-ordered UUIDv7 per the architecture notes). `nextCursor` is `null` when there is no further page. `total` is always `null`.

5.2 Filtering

- The generic list endpoint reads only `limit` and `cursor`. There are no field filters in code.
 - Scope filtering is applied server-side from the actor's assignments (for example, a state coordinator sees only records in their states). Clients cannot widen it.
 - Some lists are restricted to the caller's own rows (`exports : requester`; `notifications :` recipient).
-
-

6. Idempotency

Idempotency is implemented per endpoint, not globally. The table is the authoritative summary; section 11 repeats the detail for each endpoint.

Endpoint	Key location	Required	Replay behaviour	Conflict
POST /finance/transactions	Body field idempotencyKey (16 to 128 chars)	Yes (schema)	Same key and same fingerprint: returns the existing transaction with status 201	Same fingerprint IDEMPOTENCY
POST /finance/transactions/{id}/reversal and /refund	Body field idempotencyKey	Yes (schema)	Same key and fingerprint: returns the existing correction	Different fingerprint IDEMPOTENCY
POST /finance/transactions/{id}/approve , /reject	None	n/a	Not idempotent. A second call finds the transaction no longer pending: 409 INVALID_STATE	n/a
POST /finance-applications	Header Idempotency-Key (8 to 128 chars, [A-Za-z0-9._:-])	Yes	Same key and fingerprint: returns the existing application (201)	Different fingerprint IDEMPOTENCY
POST /finance-applications/{id}/transition	Header	Yes	Same key: returns the current application. No fingerprint comparison.	n/a
POST /finance-applications/{id}/decision	Header	Yes	Same key: returns the current application. No	n/a

Endpoint	Key location	Required	Replay behaviour	Cont
			fingerprint comparison.	
POST /financing-cases	Header	Yes	Same key: returns the existing case. No fingerprint comparison.	n/a
POST /financing-cases/{id}/transition	Header	Yes	Same key: returns the existing transition record.	n/a
POST /financing-cases/{id}/agreements	Header	Yes	Same key: returns the existing agreement.	n/a
POST /{resource} (generic)	None read	n/a	Not implemented. A duplicate natural key returns 409 CONFLICT	n/a
PATCH /{resource} (generic)	None	n/a	Not applicable	n/a
POST /reconciliation/sync	None	n/a	Not idempotent: every call queues a new job	n/a
POST /reporting/jobs	None	n/a	Not idempotent: every call queues a new job	n/a
POST /marketplace/applications/{id}/decision	None	n/a	Second call finds the request decided: 409 INVALID_STATE	n/a

Endpoint	Key location	Required	Replay behaviour	Cont
POST /webhooks/{provider}	Provider event id	n/a	Duplicate (tenant, provider, eventId) : returns 200 with duplicate: true	n/a
POST /contact	None	n/a	Not idempotent	n/a

Header name used by the code is Idempotency-Key . Send 16 to 128 characters (A-Z a-z 0-9 . _ : -).

7. Rate limits and size limits

7.1 Rate limits

Scope	Limit	Response
Better Auth sign-in (/sign-in/email)	10 per 60 s	Better Auth 429
Better Auth sign-up (/sign-up/email)	5 per 60 s	Better Auth 429
Better Auth password reset (/request-password-reset)	3 per 300 s	Better Auth 429
Better Auth two-factor (/two-factor/*)	10 per 60 s	Better Auth 429
Better Auth default	60 per 60 s	Better Auth 429
Better Auth 2FA account logout	5 failed attempts, locked 15 minutes	Better Auth error
Contact form	5 inquiries in 15 minutes	429 RATE_LIMITED
Report exports, concurrency	3 active jobs per tenant by default (max 10)	429 LIMIT_REACHED
Report exports, daily	20 jobs per tenant per UTC day by default (max 100)	429 LIMIT_REACHED
Report exports, storage and operations ceiling	Platform storage safety ceiling	503 FEATURE_UNAVAILABLE

Additional network-level protections apply and are not published.

7.2 Size limits

Route	Limit	Code
Generic resources, finance transactions, finance applications, financing cases, reconciliation, reporting jobs, contact	256 KiB JSON body (declared length and actual bytes)	<code>PAYLOAD_TOO_LARGE</code> (finance wrapper) or <code>VALIDATION_ERROR</code> (generic helper)
Marketplace decision	8 KiB (declared <code>Content-Length</code>)	413 <code>PAYLOAD_TOO_LARGE</code>
Push subscription	8 KiB (declared <code>Content-Length</code>)	413 <code>PAYLOAD_TOO_LARGE</code>
Provider webhook	1 MiB (declared and actual bytes)	413 <code>PAYLOAD_TOO_LARGE</code>
Report export <code>expectedMaxBytes</code>	50,000,000 bytes per export	400 <code>VALIDATION_ERROR</code>

Generic and finance routes also reject JSON nested deeper than 12 levels, arrays longer than 1,000 items, and strings longer than 10,000 characters (`validateJsonValue`).

8. Webhooks

8.1 Providers

Only the following provider names are accepted in the path (case-insensitive):

Path segment	Kind	Accepted	Signature header	Algorithm and encoding
flutterwave	Payments	Yes	flutterwave-signature	HMAC-SHA256 over the raw body, base64
paystack	Payments	Yes	x-paystack-signature	HMAC-SHA512 over the raw body, hex
fincra	Payments	Yes	signature	HMAC-SHA512, hex
squad	Payments	Yes	x-squad-signature	HMAC-SHA512, hex
termii	SMS status	Verified, then always 503	x-termii-signature	HMAC-SHA512, hex
twilio	SMS status	Verified, then always 503	x-twilio-signature	HMAC-SHA1 of URL, parameters and content type, base64 (constant-time compare)

Any other provider name returns 404 `NOT_FOUND` .

Signature comparison is constant-time (`constantTimeEqual`). A missing signature fails verification.

SMS status callbacks (`termii` , `twilio`) are verified and then rejected with 503

`FEATURE_UNAVAILABLE` because a durable tenant-bound dispatch record is not implemented. Invalid signatures on those providers still return 401.

8.2 Processing flow for payment webhooks

1. Route reads the raw body (max 1 MiB) and sets `x-dffms-webhook-url` to the request URL (used by Twilio verification).
2. Provider adapter verifies the signature. Failure: 401 `INVALID_WEBHOOK` .
3. The adapter extracts the provider `eventId` and `reference` . Missing reference: 401.
4. The transaction is looked up by `reference` . Zero or several matches: 401 `INVALID_WEBHOOK` ("unknown or ambiguous").
5. An inbox row is inserted with unique key (`tenant` , `provider` , `providerEventId`); a conflict means `duplicate: true` and no further work.
6. An outbox job `payment.webhook.received` is written.
7. The response returns **200 immediately**. `after()` then processes up to 5 outbox jobs for that tenant.

8. The worker re-verifies the payment **with the provider API** (`verifyPayment` , `verifyPayout` or `verifyRefund`), checks amount and currency against the stored transaction, and only then posts ledger entries. Provider redirects or webhook bodies alone never move money.

8.3 Webhook responses

```
{
  "data": { "accepted": true, "duplicate": false, "eventId": "<provider-event-id>" },
  "error": null,
  "requestId": "<request-id>"
}
```

9. Versioning

- The version is the path segment `/api/v1` . The OpenAPI document declares `info.version` `1.0.0` .
 - There is no version header, content negotiation or sunset header in code.
 - The Better Auth path (`/api/v1/auth`) and the generic resource names are part of the same v1 surface.
 - The code does not define a deprecation policy. Not specified in code.
-

10. Endpoint index

#	Method	Path	Auth	Pu
1	GET	/api/v1/health	None	Liv (JS HT
2	GET	/api/v1/ready	None	Da anc cor rea
3	GET	/api/v1/openapi	None	Op doc
4	POST	/api/v1/contact	Origin check	Su pul cor inq
5	GET, POST, PATCH, PUT, DELETE	/api/v1/auth/{operation}	Better Auth	Sig sig ses 2F, pas pas res
6	GET	/api/v1/{resource}	Session and tenant	Lis (52 res
7	POST	/api/v1/{resource}	Session, tenant, Origin	Cre rec res
8	PATCH	/api/v1/{resource}	Session, tenant, Origin	Up rec res
9	GET	/api/v1/push-subscriptions	Session	Pu ava anc key

#	Method	Path	Auth	Pu
10	POST	/api/v1/push-subscriptions	Session, Origin	Sa bro pus sub
11	DELETE	/api/v1/push-subscriptions	Session, Origin	Re pus sub
12	GET	/api/v1/finance-applications	Session, tenant	Lis app
13	POST	/api/v1/finance-applications	Session, tenant, Idempotency-Key	Su fin app
14	POST	/api/v1/finance-applications/{applicationId}/transition	Session, tenant, Idempotency-Key	No cor life tra
15	POST	/api/v1/finance-applications/{applicationId}/decision	Session, tenant, MFA, fresh, Idempotency-Key	Co dec
16	GET	/api/v1/financing-cases	Session, tenant	Lis cas
17	POST	/api/v1/financing-cases	Session, tenant, MFA, fresh, Idempotency-Key	Cre cas app app
18	POST	/api/v1/financing-cases/{financingCaseId}/transition	Session, tenant, Idempotency-Key (MFA and fresh for	Life tra

#	Method	Path	Auth	Pu
			high-risk targets)	
19	POST	/api/v1/financing-cases/{financingCaseId}/agreements	Session, tenant, Idempotency-Key	Re exe agi
20	POST	/api/v1/finance/transactions	Session, tenant, MFA, fresh	Cre tra dra (pe api
21	POST	/api/v1/finance/transactions/{transactionId}/approve	Session, tenant, MFA, fresh	Ap tra
22	POST	/api/v1/finance/transactions/{transactionId}/reject	Session, tenant, MFA, fresh	Re tra
23	POST	/api/v1/finance/transactions/{transactionId}/reversal	Session, tenant, MFA, fresh	Re rev
24	POST	/api/v1/finance/transactions/{transactionId}/refund	Session, tenant, MFA, fresh	Re refi
25	POST	/api/v1/reconciliation/sync	Session, tenant, fresh	Qu set rec run
26	GET	/api/v1/reporting/jobs	Session, MFA if tenant policy	Lis exp
27	POST	/api/v1/reporting/jobs	Session, MFA, Origin	Scl rep

#	Method	Path	Auth	Pu
28	GET	/api/v1/reporting/jobs/{jobId}/download	Session, MFA if tenant policy	Sh do UF
29	POST	/api/v1/marketplace/applications/{applicationId}/decision	Platform Super Administrator, MFA, fresh	Ac rev dec pul rec
30	POST	/api/v1/webhooks/{provider}	Provider signature	Ac prc we

Rows 6 to 8 cover the generic resource route. Row 5 covers Better Auth.

11. Endpoint reference

11.1 Platform: health, readiness, OpenAPI

GET /api/v1/health

- **Purpose.** Confirms the application process can answer requests. Does not check dependencies.
- **Authentication.** None.
- **Role.** None.
- **Parameters.** None. Send `Accept: text/html` to receive a human-readable page instead of JSON.
- **Body.** None.

Example:

```
curl -s https://dffms.org/api/v1/health
```

Response (200):

```
{
  "data": {
    "status": "ok",
    "service": "dffms-web",
    "scope": "application-liveness",
    "timestamp": "2026-10-26T08:00:00.000Z"
  },
  "error": null,
  "requestId": "<request-id>"
}
```

- **Errors.** None defined beyond unexpected 500.
- **Side effects.** None.

GET /api/v1/ready

- **Purpose.** Reports whether the database answers a readiness query, and whether optional email and Google sign-in are configured.
- **Authentication.** None.
- **Parameters.** None.

```
curl -s https://dffms.org/api/v1/ready
```

Response (200):

```
{
  "data": {
    "status": "ready",
    "dependencies": {
      "database": "ready",
      "email": "configured",
      "googleOAuth": "disabled"
    }
  },
  "error": null,
  "requestId": "<request-id>"
}
```

- **Errors.** 503 `NOT_READY` ("A required dependency is unavailable") when the database query fails.
- **Side effects.** None.

GET /api/v1/openapi

- **Purpose.** Returns the OpenAPI 3.1 description of this API.
- **Authentication.** None.
- **Response.** Raw OpenAPI JSON (no envelope). Headers: `Cache-Control: public, max-age=300, s-maxage=3600`, `X-Content-Type-Options: nosniff`.

```
curl -s https://dffms.org/api/v1/openapi | head -c 300
```

- **Notes.** The document does not list `/api/v1/contact` or `OPTIONS`.

11.2 Public contact form

POST /api/v1/contact

- **Purpose.** Records a public enquiry and notifies staff.
- **Authentication.** None. Requires a valid mutation `Origin` (see [Section 2.6](#)).
- **Role.** None.
- **Headers.** `Content-Type: application/json`, `Origin`.

Request body:

Field	Type	Required	Rules
<code>name</code>	string	Yes	Trimmed; 2 to 120 characters
<code>email</code>	string	Yes	Valid email; trimmed; lower-cased; max 320
<code>organization</code>	string	No	Trimmed; max 180; empty string allowed
<code>subject</code>	string	Yes	Trimmed; 5 to 180
<code>message</code>	string	Yes	Trimmed; 20 to 5,000

Unknown fields are rejected (`.strict()`).

```
curl -s -X POST https://dffms.org/api/v1/contact \
-H 'Content-Type: application/json' \
-H 'Origin: https://dffms.org' \
-d '{
  "name": "Ada Example",
  "email": "ada@demo.dffms.org",
  "organization": "Demo Agro Cooperative",
  "subject": "Question about input financing",
  "message": "We would like to know the eligibility criteria for the smallholder input window."
}'
```

Response (201):

```
{
  "data": {
    "id": "0190f2c4-6b3e-7a1d-8c2f-000000000101",
    "status": "submitted",
    "message": "Your inquiry has been submitted and recorded successfully."
  },
  "error": null,
  "requestId": "<request-id>"
}
```

Errors:

HTTP	Code	Meaning
403	REQUEST_REJECTED	Origin missing or not allowed, or body is not JSON (lib <code>RequestValidationError</code>).
422	VALIDATION_ERROR	Field validation failed; <code>fields</code> lists paths (unknown fields appear under <code>request</code>).
429	RATE_LIMITED	Five or more inquiries from the same email in the last 15 minutes.
500	INTERNAL_ERROR	Database insert failed.

Side effects:

- Inserts a `contact_inquiries` row with status `PENDING` .
- Queues an in-app notification, a push notification and an email to Platform Super Administrators, with an action link to `/admin/inquiries/{id}` . Failures here are swallowed; the inquiry is still recorded.

- Sends an email to the DFFMS support mailbox and an acknowledgement email to the submitter. Email failures do not affect the result.
- No audit event is written by this route.

11.3 Authentication (Better Auth)

ANY /api/v1/auth/{operation}

- **Purpose.** Catch-all for Better Auth. Methods exported: GET, POST, PATCH, PUT, DELETE.
- **Response format.** Better Auth's own JSON. Not wrapped in the DFFMS envelope. The exact body of each operation is not specified in this codebase.

Endpoints enabled by the configuration (paths confirmed from the installed Better Auth 1.7.6 package):

Method and path	Purpose	Notes
POST <code>/sign-up/email</code>	Register a public account	Rate limit 5 per minute. Sends verification email.
POST <code>/sign-in/email</code>	Sign in with email and password	Rate limit 10 per minute. Verified email required.
POST <code>/sign-out</code>	End the session	
GET <code>/get-session</code>	Read the current session	
POST <code>/request-password-reset</code>	Start password reset	Rate limit 3 per 5 minutes. Resets revoke other sessions.
POST <code>/reset-password</code>	Complete password reset	
POST <code>/verify-email</code> and <code>/send-verification-email</code>	Email verification	Token expires in 30 minutes.
GET <code>/sign-in/social</code> , GET <code>/callback/:id</code>	Google sign-in	Only when Google is configured.
POST <code>/two-factor/enable</code> , <code>/disable</code> , <code>/generate-backup-codes</code> , <code>/get-totp-uri</code> , <code>/send-otp</code> , <code>/verify-otp</code> , <code>/verify-totp</code> , <code>/verify-backup-code</code>	Two-factor management and challenge	Rate limit 10 per minute. Account locks for 15 minutes after 5 failed attempts.
POST <code>/passkey/generate-register-options</code> , <code>/passkey/verify-registration</code> , <code>/passkey/generate-authenticate-options</code> , <code>/passkey/verify-authentication</code> , <code>/passkey/authenticate</code> , <code>/passkey/register</code> , GET <code>/passkey/list-user-passkeys</code> , POST <code>/passkey/update-passkey</code> , <code>/passkey/delete-passkey</code>	Passkey management	RP ID is the app host.

Example (sign-in):

```
curl -s -X POST https://dffms.org/api/v1/auth/sign-in/email \
-H 'Content-Type: application/json' \
-H 'Origin: https://dffms.org' \
-d '{"email":"ada@demo.dffms.org","password":"<password>"}' \
-c cookies.txt
```

- **Side effects.**
 - Account-security notices (in-app, email and push) are sent when sign-in protection changes.
 - Session creation records the last successful sign-in.
- **Account types.** `accessType` is `PUBLIC_USER` or `STAFF_USER`; `status` is `ACTIVE`, `SUSPENDED`, `CLOSED` or `ARCHIVED`. Users cannot set these fields through sign-up.

11.4 Generic resources

GET /api/v1/{resource}

- **Purpose.** Lists records of one resource type, tenant-scoped and ABAC-filtered. The resource list is in [Appendix A](#).
- **Authentication.** Session cookie.
- **Headers.** `X-DFFMS-Tenant-Id` (required).
- **Role.** Any active role in the tenant for the action `read` in the domain policy. Some resources add a resource-specific role list (see Appendix A). Some resources are gated.
- **Path parameters.** `resource`: one or more segments (for example `programmes`, `admin/providers`).
- **Query parameters.** `limit` (1 to 100, default 25), `cursor` (UUID).

```
curl -s "https://dffms.org/api/v1/programmes?limit=10" \
-H 'Cookie: __Secure-dffms.session_token=<session-cookie>' \
-H 'X-DFFMS-Tenant-Id: <tenantId>'
```

Response (200):

```

{
  "data": {
    "items": [
      {
        "id": "0190f2c4-6b3e-7a1d-8c2f-000000000020",
        "tenantId": "<tenantId>",
        "code": "AGRO-PRG-01",
        "name": "Agro-Input Support Programme",
        "status": "ACTIVE",
        "createdAt": "2026-10-01T09:00:00.000Z",
        "updatedAt": "2026-10-02T10:00:00.000Z"
      }
    ],
    "nextCursor": null,
    "total": null
  },
  "error": null,
  "requestId": "<request-id>"
}

```

Errors:

HTTP	Code	Meaning
400	VALIDATION_ERROR	Missing or invalid tenant header, invalid <code>limit</code> or <code>cursor</code> .
401	UNAUTHENTICATED	No session.
403	MFA_REQUIRED	Tenant requires MFA enrolment and the account has none.
403	FORBIDDEN	No active role in the tenant, or the resource's role list excludes the actor.
404	NOT_FOUND	Unknown resource path.
501	FEATURE_UNAVAILABLE	Resource is gated (currently <code>benchmarks</code>).

Side effects: none.

POST /api/v1/{resource}

- **Purpose.** Creates a record. Supported only for the nine resources marked `crud` in [Appendix A](#): `organizations` , `programmes` , `sapzs` , `institutions` , `commodities` , `value-chains` , `risk-assessments` , `indicators` , `reports` .

- **Authentication.** Session cookie, `X-DFFMS-Tenant-Id` , `Origin` , `Content-Type: application/json` .
- **Role.** The domain write permission for that resource kind (see Appendix A), or Platform Super Administrator.
- **Body.** Fields per resource (see Appendix A for the rules). Unknown fields are rejected with 400 `VALIDATION_ERROR` . Fields are checked by type and by the rules below.

Field kinds used by the generic validator:

Kind	Rule
<code>string</code>	Trimmed; max length as configured
<code>code</code>	<code>^[A-Z0-9][A-Z0-9._/-]{1,63}\$</code>
<code>uuid</code>	UUID
<code>date</code>	<code>YYYY-MM-DD</code>
<code>datetime</code>	ISO 8601
<code>integer</code>	Integer with configured bounds
<code>enum</code>	One of the listed values
<code>boolean</code>	<code>true</code> or <code>false</code>
<code>json</code>	Any JSON value (object or array)

Record status enum (`status` fields): `DRAFT` , `PENDING` , `APPROVED` , `REJECTED` , `ACTIVE` , `SUSPENDED` , `CLOSED` , `ARCHIVED` .

Example:

```
curl -s -X POST https://dffms.org/api/v1/programmes \
-H 'Cookie: __Secure-dffms.session_token=<session-cookie>' \
-H 'X-DFFMS-Tenant-Id: <tenantId>' \
-H 'Origin: https://dffms.org' \
-H 'Content-Type: application/json' \
-d '{
  "code": "AGRO-PRG-01",
  "name": "Agro-Input Support Programme",
  "description": "Input financing for smallholder farmers.",
  "startsOn": "2026-01-01",
  "endsOn": "2028-12-31",
  "status": "DRAFT"
}'
```

Response (201): the created record, with `id` , `tenantId` , `cellId` , the submitted fields and timestamps. `Idempotency-Replayed` is never set for generic creates (see [Section 6](#)).

Errors:

HTTP	Code	Meaning
400	VALIDATION_ERROR	Field errors (<code>fields</code>), invalid JSON, body not an object, origin missing or not allowed (this helper returns 400), unknown field.
400	VALIDATION_ERROR	Secret-like key names in the body (for example <code>password</code> , <code>token</code> , <code>apiKey</code>) are rejected with a message to store secrets in the deployment secret store.
401	UNAUTHENTICATED	No session.
403	MFA_REQUIRED / FORBIDDEN	As for GET.
409	CONFLICT	Unique constraint (for example, duplicate <code>code</code>).
501	FEATURE_UNAVAILABLE	Resource is read-only. The role check runs first, so an unauthorised caller gets 403 instead.

Side effects: writes one audit event with action `CREATE` , resource type and resource id, and the list of changed field **names** (values are not stored in the audit event). `risk-assessments` records the creator as `assessedBy` and forces status `DRAFT` .

PATCH `/api/v1/{resource}`

- **Purpose.** Updates a record. Same nine `crud` resources as POST.
- **Body.** `id` (UUID, required) plus any fields in the resource's update list. `id` is removed before validation.
- **Rules.** At least one mutable field is required. The record must be inside the actor's scope. Missing record: 404.

Example:

```
curl -s -X PATCH https://dffms.org/api/v1/programmes \
-H 'Cookie: __Secure-dffms.session_token=<session-cookie>' \
-H 'X-DFFMS-Tenant-Id: <tenantId>' \
-H 'Origin: https://dffms.org' \
-H 'Content-Type: application/json' \
-d '{"id":"0190f2c4-6b3e-7a1d-8c2f-000000000020","status":"ACTIVE"}'
```

Response (200): the updated record. Not found: `404 NOT_FOUND`, "`<label>` was not found".

Errors: as for POST, plus 400 when `id` is missing or not a UUID, and 400 `VALIDATION_ERROR` "At least one mutable field is required". No version or ETag check is implemented (last write wins). Audit action is `UPDATE` with changed field names.

OPTIONS /api/v1/{resource}

The route exports an `OPTIONS` handler that answers preflight for allowed origins. Allowed methods listed: `GET`, `POST`, `PATCH`, `OPTIONS`.

11.5 Web push subscriptions

GET /api/v1/push-subscriptions

- **Purpose.** Reports whether browser push is configured and returns the public VAPID key.
- **Authentication.** Session cookie. Public users and staff with an active portal membership.
- **Headers.** None required beyond the cookie.

```
curl -s https://dffms.org/api/v1/push-subscriptions \
-H 'Cookie: __Secure-dffms.session_token=<session-cookie>'
```

Response (200):

```
{
  "data": { "configured": true, "publicKey": "<vapid-public-key>" },
  "error": null,
  "requestId": "<request-id>"
}
```

Errors: 401 UNAUTHENTICATED . Side effects: none.

POST /api/v1/push-subscriptions

- **Purpose.** Stores the browser's push subscription for the signed-in account.
- **Authentication.** Session cookie; `Origin` must be allowed (`assertTrustedOrigin`).
- **Body.** The browser `PushSubscription` JSON:

Field	Type	Required	Rules
<code>endpoint</code>	string	Yes	Valid URL, max 2,048
<code>keys.p256dh</code>	string	Yes	32 to 512 characters
<code>keys.auth</code>	string	Yes	8 to 256 characters

```
curl -s -X POST https://dffms.org/api/v1/push-subscriptions \
-H 'Cookie: __Secure-dffms.session_token=<session-cookie>' \
-H 'Origin: https://dffms.org' \
-H 'Content-Type: application/json' \
-d '{"endpoint":"https://<push-service-host>/<subscription-id>","keys":{"p256dh":"<p256dh-key>"
```

Response (201): `{"data":{"subscribed":true},"error":null,"requestId":"..."}`

Errors:

HTTP	Code	Meaning
401	UNAUTHENTICATED	No session or no portal access.
403	ORIGIN_DENIED	Origin not allowed.
413	PAYLOAD_TOO_LARGE	Declared length above 8 KiB.
422	VALIDATION_ERROR	Subscription fails validation ("Invalid push subscription").
500	INTERNAL_ERROR	Also returned for malformed JSON, because the body is parsed before validation and a parse error is not mapped.

Side effects: upserts a push subscription row for (tenant, user, endpoint) . No audit event.

DELETE /api/v1/push-subscriptions

- **Purpose.** Removes the subscription for the given endpoint.
- **Body.** {"endpoint":"https://<push-service-host>/<subscription-id>"} (URL, required).
- **Response (200).** {"data":{"subscribed":false},"error":null,"requestId":"..."}
- **Errors.** 401, 403 ORIGIN_DENIED , 422 VALIDATION_ERROR (endpoint not a URL), 500 INTERNAL_ERROR (malformed JSON, as above).
- **Side effects.** Deletes the row matching the user and endpoint.

11.6 Finance applications

Facility finance applications are submitted against an authoritative product version, move through a lifecycle, and reach a committee decision.

Lifecycle values: DRAFT , SUBMITTED , ELIGIBILITY_REVIEW , DUE_DILIGENCE , APPRAISAL , COMMITTEE_REVIEW , APPROVED , DECLINED , WITHDRAWN , EXPIRED . The allowed transitions are defined by canTransitionApplication and are not repeated here.

Financing instruments: DEBT , PPP_FINANCE , GUARANTEE , TRADE_FINANCE , EQUITY , VENTURE_CAPITAL .

GET /api/v1/finance-applications

- **Purpose.** Lists finance applications visible to the actor.
- **Authentication.** Session cookie and X-DFMS-Tenant-Id .

- **Role.** Finance read set: PSA, CSA, PA, SCA, IA, FEO, FA, AC, EDV, IV. Results are filtered by the application data scope.
- **Parameters.** `limit` , `cursor` (see [Section 5](#)).
- **Response.** Same shape as the generic list. `idempotencyKey` is redacted in list responses.
- **Errors.** As for generic list (400, 401, 403).

POST /api/v1/finance-applications

- **Purpose.** Submits a new finance application linked to a published product version. Applicant eligibility is evaluated at submission.
- **Authentication.** Session cookie, `X-DFFMS-Tenant-Id` , `Origin` , `Content-Type: application/json` , `Idempotency-Key` header.
- **Role.** `create` on `finance_application` : CSA, PA, SCA, IA, FEO, FA (or PSA). No step-up.
- **Body.**

Field	Type	Required	Rules
<code>productVersionId</code>	UUID	Yes	Must be a published, non-superseded version
<code>windowId</code>	UUID	No	Must be an active product window
<code>stateId</code>	UUID	No	State of operation
<code>instrument</code>	enum	Yes	<code>DEBT</code> , <code>PPP_FINANCE</code> , <code>GUARANTEE</code> , <code>TRADE_FINANCE</code> , <code>EQUITY</code> , <code>VENTURE_CAPITAL</code>
<code>ticketCode</code>	string	Yes	<code>^[A-Z0-9_]{2,64}\$</code>
<code>applicantCategory</code>	string	Yes	Trimmed; 2 to 120
<code>intendedActivity</code>	string	Yes	Trimmed; 10 to 500
<code>requestedAmountMinor</code>	string	Yes	Digits only (<code>^\d+\$</code>); must be greater than zero
<code>currency</code>	string	No	ISO 4217 uppercase; default <code>NGN</code>

Unknown fields are ignored (the schema strips them).

```
curl -s -X POST https://dffms.org/api/v1/finance-applications \
-H 'Cookie: __Secure-dffms.session_token=<session-cookie>' \
-H 'X-DFFMS-Tenant-Id: <tenantId>' \
-H 'Origin: https://dffms.org' \
-H 'Idempotency-Key: fa-2026-10-26-0001' \
-H 'Content-Type: application/json' \
-d '{
  "productVersionId": "0190f2c4-6b3e-7a1d-8c2f-000000000030",
  "stateId": "0190f2c4-6b3e-7a1d-8c2f-000000000031",
  "instrument": "DEBT",
  "ticketCode": "SMALLHOLDER_INPUTS",
  "applicantCategory": "Smallholder cooperative",
  "intendedActivity": "Purchase of certified seed and fertiliser for the 2027 season.",
  "requestedAmountMinor": "500000000",
  "currency": "NGN"
}'
```

Response (201): the stored application row (id , tenantId , reference such as DFFMS-APP-YYYYMMDD-XXXXXXX , lifecycle set to SUBMITTED , eligibilitySnapshot , idempotencyKey , timestamps).

Errors:

HTTP	Code	Meaning
400	IDEMPOTENCY_KEY_REQUIRED	Header missing or malformed.
400	VALIDATION_ERROR	Field errors (<code>fields</code>), tenant header problems, origin problems.
400	INVALID_AMOUNT	Requested amount is zero.
400	PRODUCT_VERSION_UNAVAILABLE	Product version not published or not in the authoritative register.
400	PRODUCT_VERSION_SUPERSEDED	Version replaced by a later term sheet; apply under the current version.
400	WINDOW_UNAVAILABLE	Window not active.
400	STATE_UNAVAILABLE	State not valid for the tenant.
400	ELIGIBILITY_FAILED	Blocking eligibility rule failed. <code>fields</code> maps rule codes to messages.
401	UNAUTHENTICATED	No session.
403	FORBIDDEN	Role cannot create applications.
403	MFA_REQUIRED	Tenant requires MFA enrolment.
409	IDEMPOTENCY_CONFLICT	Key reused with a different request.
500	INTERNAL_ERROR	Unhandled error.

Side effects: inserts the application (reference `DFFMS-APP-<date>-<8 hex>`), an eligibility snapshot, and a submission audit event `FINANCE_APPLICATION_SUBMITTED` . Queues notifications to application team roles (`PROGRAMME_ADMIN` , `STATE_COORDINATOR_ADMIN` , `INSTITUTION_ADMIN` , `FINANCE_ENTRY_OFFICER`) and related escalation events.

POST /api/v1/finance-applications/{applicationId}/transition

- **Purpose.** Moves an application through a non-committee lifecycle step (for example `SUBMITTED` to `ELIGIBILITY_REVIEW`).
- **Authentication.** Session cookie, `X-DFFMS-Tenant-Id` , `Origin` , `Idempotency-Key` header.
- **Role.** update on `finance_application` (CSA, PA, SCA, IA, FEO, FA) plus saved role policy check for `finance:update` . No step-up.
- **Path.** `applicationId` : UUID.

- **Body.**

Field	Type	Required	Rules
expectedLifecycle	enum	Yes	Current lifecycle (optimistic check)
targetLifecycle	enum	Yes	Must not be APPROVED or DECLINED
reason	string	Yes	Trimmed; 10 to 1,000

Unknown fields are ignored.

```
curl -s -X POST https://dffms.org/api/v1/finance-applications/0190f2c4-6b3e-7a1d-8c2f-000000000004
-H 'Cookie: __Secure-dffms.session_token=<session-cookie>' \
-H 'X-DFFMS-Tenant-Id: <tenantId>' \
-H 'Origin: https://dffms.org' \
-H 'Idempotency-Key: fa-trans-2026-10-26-0001' \
-H 'Content-Type: application/json' \
-d '{"expectedLifecycle":"SUBMITTED","targetLifecycle":"ELIGIBILITY_REVIEW","reason":"Submitted
```

Response (200): the application row after the transition.

Errors:

HTTP	Code	Meaning
400	IDEMPOTENCY_KEY_REQUIRED	Header missing or malformed.
400	VALIDATION_ERROR	Body or path invalid.
401	UNAUTHENTICATED	No session.
403	FORBIDDEN	Role or saved policy does not permit.
404	NOT_FOUND	Missing, or outside the actor's scope (not distinguished).
409	COMMITTEE_DECISION_REQUIRED	Target is APPROVED or DECLINED ; use /decision .
409	INVALID_STATE_TRANSITION	Transition not allowed by the lifecycle rules.
409	STALE_STATE	expectedLifecycle no longer matches.

Side effects: updates lifecycle, may open a pending approval row for committee review, writes a transition event and audit FINANCE_APPLICATION_TRANSITIONED . Replays with the same key return the current application without checking the body.

POST /api/v1/finance-applications/{applicationId}/decision

- **Purpose.** Records the independent committee decision on an application in COMMITTEE_REVIEW . Approval records the exact amount and final terms; soft rule failures can be carried as rule exceptions.
- **Authentication.** Session cookie, X-DFFMS-Tenant-Id , Origin , Idempotency-Key header, MFA enrolled, session age within 15 minutes.
- **Role.** approve on finance_application : FA or SCA (or PSA), plus saved policy finance:approve .
- **Path.** applicationId : UUID.
- **Body.**

Field	Type	Required	Rules
decision	enum	Yes	APPROVED or DECLINED
reason	string	Yes	Trimmed; 10 to 1,000
approvedAmountMinor	string	Required for APPROVED	Digits only; greater than zero
approvedTerms	object	Required for APPROVED	Shape depends on instrument (below)

Credit terms (DEBT , PPP_FINANCE , GUARANTEE , TRADE_FINANCE):

Field	Type	Rules
pricingStructure	string	^[A-Z0-9_]{2,64}\$
interestRateBps	integer	0 to 2,000
tenorMonths	integer	1 to 300
graceMonths	integer	0 to 60
contributionBps	integer	0 to 10,000, optional
coverageBps	integer	0 to 10,000, optional
repaymentStructure	string	3 to 240
conditions	string[]	Up to 50 items, each 3 to 500; default []

Equity terms (EQUITY , VENTURE_CAPITAL):

Field	Type	Rules
ownershipBps	integer	1 to 10,000
valuationMinor	string	Digits, optional
milestones	string[]	1 to 50 items, each 3 to 500
exitPlan	string	3 to 1,000, optional
conditions	string[]	Up to 50 items, each 3 to 500; default []

```
curl -s -X POST https://dffms.org/api/v1/finance-applications/0190f2c4-6b3e-7a1d-8c2f-000000000004
-H 'Cookie: __Secure-dffms.session_token=<session-cookie>' \
-H 'X-DFFMS-Tenant-Id: <tenantId>' \
-H 'Origin: https://dffms.org' \
-H 'Idempotency-Key: fa-decision-2026-10-26-0001' \
-H 'Content-Type: application/json' \
-d '{
  "decision": "APPROVED",
  "reason": "Committee approved the request with the conditions below.",
  "approvedAmountMinor": "450000000",
  "approvedTerms": {
    "pricingStructure": "<pricingStructure>",
    "interestRateBps": 1200,
    "tenorMonths": 36,
    "graceMonths": 6,
    "repaymentStructure": "Quarterly amortising repayments after grace",
    "conditions": ["Insurance cover bound before first drawdown"]
  }
}'
```

Response (200): the application row after the decision.

Errors:

HTTP	Code	Meaning
400	IDEMPOTENCY_KEY_REQUIRED	Header missing or malformed.
400	VALIDATION_ERROR	Body invalid, or <code>approvedTerms</code> does not match the instrument shape.
400	APPROVED_TERMS_REQUIRED	Approval without amount or terms. (Returned with the default status 400.)
400	INVALID_AMOUNT	Approved amount zero or negative.
400	FINAL_TERMS_INVALID	A hard rule failed. <code>fields</code> lists all failures with <code>committeeExceptionAllowed</code> .
401	UNAUTHENTICATED	No session.
403	MFA_REQUIRED	No MFA enrolment.
403	FRESH_AUTH_REQUIRED	Session older than 15 minutes.
403	FORBIDDEN	Role or saved policy does not permit.
403	MAKER_CHECKER_REQUIRED	Decider is the requester of the approval or the applicant. Applies to PSA too.
404	NOT_FOUND	Missing or outside scope.
409	INVALID_STATE	Application not in <code>COMMITTEE_REVIEW</code> .
409	APPROVAL_NOT_FOUND	No pending committee approval.
409	STALE_STATE	Application changed before the decision was recorded.

Side effects: records the decision, audit `FINANCE_APPLICATION_APPROVED` or `FINANCE_APPLICATION_DECLINED`, resolves escalations, and queues notifications to the decider roles and applicant team. Soft failures create product rule exceptions in state `PENDING` for later review. Replays with the same key return the current application.

11.7 Financing cases

A financing case is created from a committee-approved application and then moves through disbursement and repayment lifecycle steps.

Lifecycle values: `CONDITIONS_PRECEDENT` , `DOCUMENTATION` , `READY_TO_DISBURSE` , `PARTIALLY_DISBURSED` , `ACTIVE` , `COMPLETED` , `RESTRUCTURED` , `DEFAULTED` , `RECOVERY` , `CANCELLED` , `WRITTEN_OFF` .

High-risk targets (require approve authority, MFA and fresh session): `READY_TO_DISBURSE` , `PARTIALLY_DISBURSED` , `ACTIVE` , `COMPLETED` , `RESTRUCTURED` , `DEFAULTED` , `RECOVERY` , `WRITTEN_OFF` .

GET /api/v1/financing-cases

- **Purpose.** Lists financing cases visible to the actor.
- **Authentication.** Session cookie and `X-DFFMS-Tenant-Id` .
- **Role.** Finance read set, with case data scope.
- **Parameters.** `limit` , `cursor` .
- **Errors.** As for generic list.

POST /api/v1/financing-cases

- **Purpose.** Creates a controlled financing case from an application whose committee approval is complete.
- **Authentication.** Session cookie, tenant header, `Origin` , `Idempotency-Key` , MFA enrolled, session age within 15 minutes.
- **Role.** `approve` on `financing_case` : FA (or PSA).
- **Body.**

Field	Type	Required	Rules
<code>applicationId</code>	UUID	Yes	Application must be <code>APPROVED</code>
<code>beneficiary.displayName</code>	string	Yes	2 to 200
<code>beneficiary.type</code>	string	Yes	2 to 100
<code>beneficiary.externalReference</code>	string	No	Max 120
<code>beneficiary.gender</code>	enum	No	<code>FEMALE</code> or <code>MALE</code> (for inclusion indicators)
<code>beneficiary.youth</code>	boolean	No	Youth inclusion indicator
<code>institutionId</code>	UUID	No	Institution must exist

```
curl -s -X POST https://dffms.org/api/v1/financing-cases \
-H 'Cookie: __Secure-dffms.session_token=<session-cookie>' \
-H 'X-DFFMS-Tenant-Id: <tenantId>' \
-H 'Origin: https://dffms.org' \
-H 'Idempotency-Key: fc-create-2026-10-26-0001' \
-H 'Content-Type: application/json' \
-d '{
  "applicationId": "0190f2c4-6b3e-7a1d-8c2f-000000000040",
  "beneficiary": {
    "displayName": "Demo Agro Cooperative",
    "type": "COOPERATIVE",
    "externalReference": "COOP-000123",
    "gender": "FEMALE",
    "youth": false
  },
  "institutionId": "0190f2c4-6b3e-7a1d-8c2f-000000000050"
}'
```

Response (201): the financing case row (id , tenantId , applicationId , lifecycle CONDITIONS_PRECEDENT at creation, idempotencyKey , timestamps).

Errors:

HTTP	Code	Meaning
400	IDEMPOTENCY_KEY_REQUIRED	Header missing or malformed.
400	VALIDATION_ERROR	Body invalid.
401	UNAUTHENTICATED	No session.
403	MFA_REQUIRED / FRESH_AUTH_REQUIRED / FORBIDDEN	As for application decisions.
404	APPLICATION_NOT_FOUND	Application missing in the tenant.
404	INSTITUTION_NOT_FOUND	Institution missing.
409	APPLICATION_NOT_APPROVED	Application not APPROVED .
409	COMMITTEE_APPROVAL_REQUIRED	No completed committee approval with a decider.
409	APPROVED_TERMS_MISSING	Committee decision lacks a valid immutable term snapshot.

Side effects: creates the case, its initial workflow rows, outbox entries and audit `FINANCING_CASE_CREATED` . A replay with the same key returns the existing case.

POST /api/v1/financing-cases/{financingCaseId}/transition

- **Purpose.** Moves a case to the next lifecycle state, checking conditions precedent and restructuring approval where relevant.
- **Authentication.** Session cookie, tenant header, `Origin` , `Idempotency-Key` . For high-risk targets: MFA enrolled and session within 15 minutes.
- **Role.** `approve` (FA, or PSA) for high-risk targets; otherwise `update` on `financing_case` (CSA, PA, IA, FEO, FA, or PSA).
- **Body.**

Field	Type	Required	Rules
<code>expectedLifecycle</code>	enum	Yes	Current lifecycle
<code>targetLifecycle</code>	enum	Yes	Target lifecycle
<code>reason</code>	string	Yes	10 to 1,000

Errors:

HTTP	Code	Meaning
400	IDEMPOTENCY_KEY_REQUIRED	Header missing or malformed.
400	VALIDATION_ERROR	Body invalid.
401	UNAUTHENTICATED	No session.
403	MFA_REQUIRED / FRESH_AUTH_REQUIRED / FORBIDDEN	Step-up or role failure.
404	NOT_FOUND	Missing or outside scope.
409	INVALID_STATE_TRANSITION	Transition not permitted.
409	STALE_STATE	expectedLifecycle no longer matches.
409	(condition code)	A condition-precedent check failed; the code is specific to the failing rule.
409	RESTRUCTURING_APPROVAL_REQUIRED	Target RESTRUCTURED without an approved restructuring.

Side effects: updates lifecycle, writes a transition event and audit FINANCING_CASE_TRANSITIONED , and may queue notifications. Replays with the same key return the stored transition.

POST /api/v1/financing-cases/{financingCaseId}/agreements

- **Purpose.** Records an executed financing, guarantee, investment or security agreement against a case. Records are immutable once written.
- **Authentication.** Session cookie, tenant header, Origin , Idempotency-Key . No step-up.
- **Role.** create on financing_case : CSA, PA, IA, FEO, FA (or PSA).
- **Body.**

Field	Type	Required	Rules
agreementType	enum	Yes	FINANCING_AGREEMENT , GUARANTEE_AGREEMENT , INVESTMENT_AGREEMENT , SECURITY_DOCUMENT
version	string	Yes	1 to 40; ^[A-Za-z0-9] [A-Za-z0-9._/-]*\$
executedOn	date	Yes	ISO date (YYYY-MM-DD)
signatories	string	Yes	5 to 1,000
executionReference	string	No	Max 160

```
curl -s -X POST https://dffms.org/api/v1/financing-cases/0190f2c4-6b3e-7a1d-8c2f-000000000060/agr
-H 'Cookie: __Secure-dffms.session_token=<session-cookie>' \
-H 'X-DFFMS-Tenant-Id: <tenantId>' \
-H 'Origin: https://dffms.org' \
-H 'Idempotency-Key: fc-agr-2026-10-26-0001' \
-H 'Content-Type: application/json' \
-d '{
  "agreementType": "FINANCING_AGREEMENT",
  "version": "v1.0",
  "executedOn": "2026-10-20",
  "signatories": "Director, Demo Agro Cooperative; Accounting Officer, DFFMS",
  "executionReference": "EXEC-2026-0001"
}'
```

Response (201): the agreement record.

Errors:

HTTP	Code	Meaning
400	IDEMPOTENCY_KEY_REQUIRED	Header missing or malformed.
400	VALIDATION_ERROR	Body invalid.
401	UNAUTHENTICATED	No session.
403	FORBIDDEN	Role not permitted. This route has no step-up check.
404	NOT_FOUND	Case missing in the tenant.
409	CASE_CLOSED	Case is COMPLETED , CANCELLED or WRITTEN_OFF .
409	(agreement rule code)	Agreement failed validation (for example, duplicate version of the same type).

Side effects: inserts the agreement, writes audit `FINANCING_AGREEMENT_EXECUTED` . A replay with the same key returns the stored agreement.

11.8 Finance transactions

Transactions are double-entry journal postings with maker-checker approval. Status values stored: `INITIATED` , `PENDING_APPROVAL` , `PROCESSING` , `SUCCEEDED` , `FAILED` , `REVERSED` , `PARTIALLY_REFUNDED` , `REFUNDED` .

Common rules for all five finance transaction routes:

- Body must be `application/json` , at most 256 KiB, and non-empty.
- Session cookie and `X-DFFMS-Tenant-Id` are required.
- The session must be **created within the last 5 minutes** (`requireFreshSession`). Failure returns 401 `UNAUTHENTICATED` with message "Fresh authentication is required".
- Approve, reject, create, reversal and refund also require MFA enrolment (`requireStepUp`). Failure: 401 `MFA_REQUIRED` .
- Provider-backed transactions are not posted when approved. They are posted only after provider verification.

POST /api/v1/finance/transactions

- **Purpose.** Creates a balanced transaction draft that awaits independent approval.
- **Role.** Route roles: CSA, PA, SCA, IA, FEO, FA (or PSA).
- **Body.** Strict object (unknown fields rejected with 422).

Field	Type	Required	Rules
reference	string	Yes	8 to 100; <code>^[A-Za-z0-9][A-Za-z0-9._:/-]+\$</code> ; unique per tenant
idempotencyKey	string	Yes	16 to 128; <code>^[A-Za-z0-9][A-Za-z0-9._:-]+\$</code>
type	enum	Yes	<code>COLLECTION</code> , <code>DISBURSEMENT</code> , <code>ADJUSTMENT</code>
amountMinor	string	Yes	Positive integer string, up to 19 digits
currency	string	Yes	ISO 4217 uppercase
provider	enum	Collection and disbursement: Yes (or manual transfer, see rules)	<code>FLUTTERWAVE</code> , <code>PAYSTACK</code> , <code>FINCRA</code> , <code>SQUAD</code> , <code>GENERIC</code>
beneficiaryId	UUID	No	
programmeId	UUID	No	Scope; defaults from financing case metadata if present
sapzId	UUID	No	Scope
institutionId	UUID	No	Scope
entries	array	Yes	2 to 100 journal lines: <code>accountId</code> (UUID), <code>direction</code> (<code>DEBIT</code> or <code>CREDIT</code>), <code>amountMinor</code> (positive integer string), <code>memo</code> (max 500, optional)
metadata	object	No	Free-form. Manual bank transfers require <code>channel</code> <code>MANUAL_BANK_TRANSFER</code> and <code>bankTransferReference</code> (4 to 120)

Rules:

- Total debits and total credits must each equal `amountMinor` (`UNBALANCED_JOURNAL` , 422).
- Each ledger account must exist, belong to the tenant, be `ACTIVE` , and use the transaction currency (`ACCOUNT_INVALID` , 422).
- Collections and disbursements require a provider. `GENERIC` is accepted only as a manual bank transfer with evidence in `metadata` .
- `ADJUSTMENT` does not require a provider.

```
curl -s -X POST https://dffms.org/api/v1/finance/transactions \
-H 'Cookie: __Secure-dffms.session_token=<session-cookie>' \
-H 'X-DFFMS-Tenant-Id: <tenantId>' \
-H 'Origin: https://dffms.org' \
-H 'Content-Type: application/json' \
-d '{
  "reference": "TXN-2026-000145",
  "idempotencyKey": "txn-create-2026-10-26-000145",
  "type": "COLLECTION",
  "amountMinor": "2500000",
  "currency": "NGN",
  "provider": "PAYSTACK",
  "programmeId": "0190f2c4-6b3e-7a1d-8c2f-000000000020",
  "entries": [
    { "accountId": "0190f2c4-6b3e-7a1d-8c2f-000000000070", "direction": "DEBIT", "amountMinor":
    { "accountId": "0190f2c4-6b3e-7a1d-8c2f-000000000071", "direction": "CREDIT", "amountMinor"
  ]
}'
```

Response (201): the stored transaction row, abbreviated:

```
{
  "data": {
    "id": "0190f2c4-6b3e-7a1d-8c2f-000000000080",
    "tenantId": "<tenantId>",
    "reference": "TXN-2026-000145",
    "idempotencyKey": "txn-create-2026-10-26-000145",
    "type": "COLLECTION",
    "status": "PENDING_APPROVAL",
    "amountMinor": "2500000",
    "currency": "NGN",
    "provider": "PAYSTACK",
    "createdBy": "<userId>",
    "approvedBy": null,
    "postedAt": null,
    "metadata": { "requestHash": "<sha256>", "ledgerDraft": [ "..." ] },
    "...": "..."
  },
  "error": null,
  "requestId": "<request-id>"
}
```

Errors:

HTTP	Code	Meaning
400	INVALID_BODY / INVALID_CONTENT_TYPE	Body not JSON, empty, or wrong content type.
401	UNAUTHENTICATED	No session, or session older than 5 minutes.
401	MFA_REQUIRED	No MFA enrolment.
403	FORBIDDEN	Role not permitted.
403	ORIGIN_DENIED	Origin not allowed.
409	IDEMPOTENCY_CONFLICT	Key reused with different data, or reference already exists.
413	PAYLOAD_TOO_LARGE	Body over 256 KiB.
422	VALIDATION_ERROR	Schema error (fields), including missing provider or invalid manual transfer evidence. Also returned for an invalid or missing tenant header.
422	UNBALANCED_JOURNAL	Debits or credits do not equal amountMinor .
422	ACCOUNT_INVALID	Account missing, inactive, another tenant's, or wrong currency.
500	INTERNAL_ERROR	Unhandled error.

Side effects: approval row (PENDING , step 1), outbox `finance.transaction.pending_approval` , audit `FINANCIAL_TRANSACTION_CREATED` (amount and reference only), in-app notification to the creator, and an approval-required notification to finance checker roles.

POST /api/v1/finance/transactions/{transactionId}/approve

- **Purpose.** Approves a pending transaction. For `GENERIC` (no provider), the ledger is posted immediately. For provider-backed transactions, the status moves to `PROCESSING` ; a payment-execution job is queued, and posting waits for provider verification.
- **Role.** `FINANCE_APPROVER` (or PSA).
- **Path.** `transactionId` : UUID. An invalid value returns 422 `VALIDATION_ERROR` after authorization.
- **Body.** `{"reason": "<5 to 500 characters>"}` (strict).

```
curl -s -X POST https://dffms.org/api/v1/finance/transactions/0190f2c4-6b3e-7a1d-8c2f-000000000008
-H 'Cookie: __Secure-dffms.session_token=<session-cookie>' \
-H 'X-DFFMS-Tenant-Id: <tenantId>' \
-H 'Origin: https://dffms.org' \
-H 'Content-Type: application/json' \
-d '{"reason":"Evidence matched; approved for payment."}'
```

Response (200): the transaction row with `status` `PROCESSING` (provider) or `SUCCEEDED` with `postedAt` set (GENERIC), and `approvedBy` set.

Errors:

HTTP	Code	Meaning
401	<code>UNAUTHENTICATED</code> / <code>MFA_REQUIRED</code>	Fresh session or MFA missing.
403	<code>FORBIDDEN</code>	Not an approver.
403	<code>MAKER_CHECKER_VIOLATION</code>	Creator is approving their own transaction (PSA exempt).
404	<code>NOT_FOUND</code>	Transaction missing in the tenant.
409	<code>INVALID_STATE</code>	Not <code>PENDING_APPROVAL</code> , or no pending approval row.
422	<code>VALIDATION_ERROR</code>	Reason invalid, or transaction id not a UUID.
422	<code>UNBALANCED_JOURNAL</code> , <code>ACCOUNT_INVALID</code>	Re-validation of the stored draft failed.

Side effects: updates approval to `APPROVED` , writes the ledger entries (GENERIC only), applies financing-case effects on posting, resolves escalations, writes audit `FINANCIAL_TRANSACTION_APPROVED_AND_POSTED` or `FINANCIAL_TRANSACTION_APPROVED_FOR_PROVIDER` , queues notifications, and enqueues an outbox job (`finance.transaction.posted` or `finance.payment.execution_requested`).

POST /api/v1/finance/transactions/{transactionId}/reject

- **Purpose.** Declines a pending transaction. No ledger entries are written.
- **Role.** `FINANCE_APPROVER` (or PSA).
- **Body.** `{"reason": "<5 to 500 characters>"}` (strict).

Response (200): the transaction row with `status` `FAILED` .

Errors: as for approve, except there is no `UNBALANCED_JOURNAL` path.

Side effects: approval `REJECTED` ; transaction `FAILED` ; outbox `finance.transaction.rejected` ; audit `FINANCIAL_TRANSACTION_REJECTED` (with reason); notification to the creator; escalations resolved.

POST /api/v1/finance/transactions/{transactionId}/reversal

- **Purpose.** Requests a full reversal of a posted (`SUCCEEDED`) transaction. Creates a linked `REVERSAL` transaction in `PENDING_APPROVAL` , which another approver must approve.
- **Role.** `FINANCE_ENTRY_OFFICER` or `FINANCE_APPROVER` (or PSA).
- **Body.** Strict object.

Field	Type	Required	Rules
<code>reference</code>	string	Yes	Same rules as transaction reference
<code>idempotencyKey</code>	string	Yes	16 to 128, same rules as transaction key
<code>reason</code>	string	Yes	10 to 500

```
curl -s -X POST https://dffms.org/api/v1/finance/transactions/0190f2c4-6b3e-7a1d-8c2f-000000000008
-H 'Cookie: __Secure-dffms.session_token=<session-cookie>' \
-H 'X-DFFMS-Tenant-Id: <tenantId>' \
-H 'Origin: https://dffms.org' \
-H 'Content-Type: application/json' \
-d '{
  "reference": "REV-2026-000145",
  "idempotencyKey": "rev-2026-10-26-000145-key",
  "reason": "Duplicate collection posted in error."
}'
```

Response (201): the new `REVERSAL` transaction row (`status` `PENDING_APPROVAL` , `metadata.originalTransactionId` set).

Errors:

HTTP	Code	Meaning
401	UNAUTHENTICATED / MFA_REQUIRED	Session or MFA.
403	FORBIDDEN	Role not permitted.
404	NOT_FOUND	Original transaction missing.
409	INVALID_STATE	Original not SUCCEEDED , or no posted ledger.
409	REVERSAL_EXISTS	A non-failed reversal already exists for this original.
409	IDEMPOTENCY_CONFLICT	Key reused with different data.
422	VALIDATION_ERROR	Body invalid.

Side effects: creates the correction, approval row, outbox `finance.reversal.pending_approval` , audit `FINANCIAL_REVERSAL_REQUESTED` , notifications to the requester and finance checker roles.

POST /api/v1/finance/transactions/{transactionId}/refund

- **Purpose.** Requests a partial or full refund of a posted transaction. Creates a linked `REFUND` in `PENDING_APPROVAL` .
- **Role.** `FINANCE_ENTRY_OFFICER` or `FINANCE_APPROVER` (or PSA).
- **Body.** As for reversal, plus:

Field	Type	Required	Rules
amountMinor	string	Yes	Positive integer string

The sum of non-failed refunds for the original must not exceed the original amount.

```
curl -s -X POST https://dffms.org/api/v1/finance/transactions/0190f2c4-6b3e-7a1d-8c2f-000000000008
-H 'Cookie: __Secure-dffms.session_token=<session-cookie>' \
-H 'X-DFFMS-Tenant-Id: <tenantId>' \
-H 'Origin: https://dffms.org' \
-H 'Content-Type: application/json' \
-d '{
  "reference": "REF-2026-000145-A",
  "idempotencyKey": "ref-2026-10-26-000145-a-key",
  "amountMinor": "500000",
  "reason": "Partial refund for over-collection on the receipt."
}'
```

Response (201): the new `REFUND` transaction row.

Errors: as for reversal, plus:

HTTP	Code	Meaning
409	<code>INVALID_STATE</code>	Original not <code>SUCCEEDED</code> or <code>PARTIALLY_REFUNDED</code> .
422	<code>INVALID_AMOUNT</code>	Missing amount, zero, or above the unrefunded balance.

Side effects: as for reversal, with audit `FINANCIAL_REFUND_REQUESTED` .

11.9 Reconciliation

POST /api/v1/reconciliation/sync

- **Purpose.** Queues an import of settlement data from a provider and matching against ledger records.
- **Authentication.** Session cookie, `X-DFFMS-Tenant-Id` , `Origin` , `Content-Type: application/json` . Session must be created within 5 minutes. MFA enrolment is required only if the tenant policy requires it (no step-up check on this route).
- **Role.** CSA, PA, FA, AC (or PSA).
- **Body.**

Field	Type	Required	Rules
<code>provider</code>	enum	Yes	<code>FLUTTERWAVE</code> or <code>PAYSTACK</code>
<code>from</code>	date	No	ISO date
<code>to</code>	date	No	ISO date; must be on or after <code>from</code>

```
curl -s -X POST https://dffms.org/api/v1/reconciliation/sync \
-H 'Cookie: __Secure-dffms.session_token=<session-cookie>' \
-H 'X-DFFMS-Tenant-Id: <tenantId>' \
-H 'Origin: https://dffms.org' \
-H 'Content-Type: application/json' \
-d '{"provider":"PAYSTACK","from":"2026-10-01","to":"2026-10-25"}'
```

Response (202): the integration job record, with `id` and `status` (queued).

```
{
  "data": { "id": "0190f2c4-6b3e-7a1d-8c2f-000000000090", "status": "QUEUED" },
  "error": null,
  "requestId": "<request-id>"
}
```

Errors:

HTTP	Code	Meaning
401	UNAUTHENTICATED	No session, or session older than 5 minutes.
403	FORBIDDEN	Role not permitted.
401	MFA_REQUIRED	Tenant requires MFA enrolment.
422	VALIDATION_ERROR	Provider not allowed, or date range invalid.
500	INTERNAL_ERROR	Job could not be scheduled.

Side effects: inserts an integration job, enqueues `finance.settlement.sync_requested` , writes audit `SETTLEMENT_SYNC_SCHEDULED` . When the worker completes, it writes `SETTLEMENT_SYNC_COMPLETED` . **Not idempotent:** each call creates a new job.

11.10 Reporting exports

Report exports are gated by configuration. If `REPORTING_EXPORTS_ENABLED` is not `true` , or neither `R2_EXPORTS_ENABLED` nor `OBJECT_STORAGE_EXPORTS_ENABLED` is `true` , every reporting route returns 503 `FEATURE_UNAVAILABLE` .

Catalogue codes (`REPORT_CATALOGUE`): `FINANCIAL_TRANSACTIONS` , `GENERAL_LEDGER` , `AUDIT_TRAIL` , `RECONCILIATION` , `MARKETPLACE_APPLICATIONS` , `FACILITY_ALLOCATION_UTILIZATION` , `FACILITY_INSTRUMENT_BALANCES` , `APPLICATION_PIPELINE` , `CONDITIONS_PRECEDENT_AGING` , `PORTFOLIO_PERFORMANCE` , `GUARANTEE_EXPOSURE` , `INSURANCE_COVERAGE` , `WAREHOUSE_INVENTORY` , `OFFTAKE_DELIVERY_SETTLEMENT` , `INFRASTRUCTURE_PPP` , `FACILITY_KPI_PACK` , `EARLY_WARNING_REGISTER` .

GET /api/v1/reporting/jobs

- **Purpose.** Lists export jobs for the tenant.

- **Authentication.** Session cookie. MFA enrolment only if the tenant policy requires it. No tenant header (tenant from the account).
- **Role.** PSA, CSA, PA, SCA, IA, FA, MA, AC, EDV, IV.
- **Query.** `limit` (1 to 100, default 25), `cursor` (UUID).
- **Scope.** Any read-role member sees all tenant jobs (not limited to the requester). Result rows include `parameters`.

```
curl -s "https://dffms.org/api/v1/reporting/jobs?limit=5" \
-H 'Cookie: __Secure-dffms.session_token=<session-cookie>'
```

Response (200):

```
{
  "data": {
    "items": [
      {
        "id": "0190f2c4-6b3e-7a1d-8c2f-000000000100",
        "definitionId": null,
        "status": "SUCCEEDED",
        "parameters": { "reportCode": "FINANCIAL_TRANSACTIONS", "format": "CSV", "fileLabel": "Oc
        "sizeBytes": "182736",
        "expiresAt": "2026-10-27T08:00:00.000Z",
        "createdAt": "2026-10-26T08:00:00.000Z",
        "updatedAt": "2026-10-26T08:02:00.000Z"
      }
    ],
    "nextCursor": null
  },
  "error": null,
  "requestId": "<request-id>"
}
```

Status values: `QUEUED`, `PROCESSING`, `SUCCEEDED`, `FAILED`, `DEAD_LETTER`.

Errors: 400 `VALIDATION_ERROR`, 401 `UNAUTHENTICATED`, 403 `MFA_REQUIRED` or `FORBIDDEN`, 503 `FEATURE_UNAVAILABLE`, 500 `INTERNAL_ERROR`.

POST /api/v1/reporting/jobs

- **Purpose.** Schedules an export. The file is generated asynchronously and stored privately.
- **Authentication.** Session cookie, `Origin`, `Content-Type: application/json`. **MFA enrolment is always required** for this action. No tenant header.

- **Role.** PSA, CSA, PA, SCA, IA, FA, MA, AC.
- **Body.** Strict object.

Field	Type	Required	Rules
reportCode	string	Yes	<code>^[A-Z][A-Z0-9_]{2,63}\$</code> (see catalogue)
format	enum	Yes	CSV , XLSX , PDF
parameters	object	No	Free-form; default <code>{}</code> . Shape depends on the report (not specified in this API)
expectedMaxBytes	integer	No	1 to 50,000,000; default 5,000,000
definitionId	UUID	No	Must be an active report definition in the tenant
notBefore	ISO datetime	No	Within 30 seconds in the past up to 30 days ahead
fileLabel	string	No	<code>^[A-Za-z0-9][A-Za-z0-9_.-]{0,79}\$</code>

```
curl -s -X POST https://dffms.org/api/v1/reporting/jobs \
-H 'Cookie: __Secure-dffms.session_token=<session-cookie>' \
-H 'Origin: https://dffms.org' \
-H 'Content-Type: application/json' \
-d '{
  "reportCode": "FINANCIAL_TRANSACTIONS",
  "format": "CSV",
  "parameters": { "from": "2026-10-01", "to": "2026-10-31" },
  "expectedMaxBytes": 2000000,
  "fileLabel": "Oct-2026-transactions"
}'
```

Response (202):

```

{
  "data": {
    "id": "0190f2c4-6b3e-7a1d-8c2f-000000000101",
    "status": "QUEUED",
    "format": "CSV",
    "availableAt": "2026-10-26T08:00:00.000Z",
    "expiresAt": "2026-10-27T08:00:00.000Z",
    "expectedMaxBytes": "2000000"
  },
  "error": null,
  "requestId": "<request-id>"
}

```

Errors:

HTTP	Code	Meaning
400	VALIDATION_ERROR	Schema error (<code>fields</code>), or <code>reportCode</code> format invalid.
400	INVALID_REPORT	<code>definitionId</code> unavailable, or <code>notBefore</code> outside the allowed window.
401	UNAUTHENTICATED	No session.
403	MFA_REQUIRED / FORBIDDEN	Not enrolled, or role not permitted.
429	LIMIT_REACHED	Active or daily job limit reached.
503	FEATURE_UNAVAILABLE	Exports disabled, policy values invalid, or storage ceiling reached.

Side effects: inserts a `QUEUED` job with a retention of `REPORTING_RETENTION_SECONDS` (default 86,400; max 604,800), an outbox `report.generate.requested` job, and updates aggregate storage counters. The handler writes no audit row. The worker's generation step is not documented here beyond its outbox topic.

GET /api/v1/reporting/jobs/{jobId}/download

- **Purpose.** Returns a short-lived signed URL for a completed export.
- **Authentication.** Session cookie. MFA only if tenant policy requires.
- **Role.** Read roles (as for list).
- **Path.** `jobId` : UUID. Invalid UUID: 400.

```
curl -s https://dffms.org/api/v1/reporting/jobs/0190f2c4-6b3e-7a1d-8c2f-000000000101/download \
-H 'Cookie: __Secure-dffms.session_token=<session-cookie>'
```

Response (200):

```
{
  "data": {
    "url": "https://<private-bucket-host>/<object-key>?<signature-parameters>",
    "expiresAt": "2026-10-26T08:01:00.000Z"
  },
  "error": null,
  "requestId": "<request-id>"
}
```

- The URL lifetime is `REPORTING_SIGNED_URL_SECONDS` (default 60; max 300).
- Errors: 400 `VALIDATION_ERROR` (bad id), 401, 403, 409 `NOT_READY`, 503 `FEATURE_UNAVAILABLE` (exports disabled or storage ceiling reached).
- `NOT_READY` (409) is returned for unknown jobs, jobs in another tenant, jobs not `SUCCEEDED`, and expired jobs. The response does not distinguish these cases.
- Side effects: none in this handler. No audit row is written by the handler.

11.11 Marketplace decision

POST /api/v1/marketplace/applications/{applicationId}/decision

- **Purpose.** Platform Super Administrator accepts a public marketplace request into eligibility review, or declines it.
- **Authentication.** Session cookie. The account must be Platform Super Administrator; MFA enrolment is required, plus a registered passkey if the tenant policy requires MFA; session must be created within 5 minutes. Tenant header is ignored.
- **Headers.** `Origin` (must be allowed), `Content-Type: application/json` (otherwise 415).
- **Path.** `applicationId`: UUID.
- **Body.**

Field	Type	Required	Rules
<code>decision</code>	enum	Yes	<code>APPROVED</code> or <code>REJECTED</code>
<code>note</code>	string	Required when <code>REJECTED</code>	Trimmed; max 500; at least 5 characters when rejecting

```
curl -s -X POST https://dffms.org/api/v1/marketplace/applications/0190f2c4-6b3e-7a1d-8c2f-00000000
-H 'Cookie: __Secure-dffms.session_token=<session-cookie>' \
-H 'Origin: https://dffms.org' \
-H 'Content-Type: application/json' \
-d '{"decision":"REJECTED","note":"Registration certificate not provided."}'
```

Response (200):

```
{
  "data": {
    "id": "0190f2c4-6b3e-7a1d-8c2f-000000000110",
    "reference": "MKT-2026-0042",
    "status": "REJECTED",
    "updatedAt": "2026-10-26T08:05:00.000Z"
  },
  "error": null,
  "requestId": "<request-id>"
}
```

Errors:

HTTP	Code	Meaning
401	MFA_REQUIRED	MFA or passkey missing.
401	FRESH_AUTH_REQUIRED	Session older than 5 minutes.
401	UNAUTHENTICATED	No session.
403	ORIGIN_DENIED	Origin not allowed.
403	FORBIDDEN	Not Platform Super Administrator.
404	NOT_FOUND	Request not found.
409	INVALID_STATE	Request not PENDING , already decided, or a legacy request that needs manual review.
413	PAYLOAD_TOO_LARGE	Declared length above 8 KiB.
415	INVALID_CONTENT_TYPE	Not JSON.
422	VALIDATION_ERROR	Body invalid, or rejection without a note.
500	INTERNAL_ERROR	Unhandled error.

Side effects:

- **APPROVED** : status `APPROVED` , lifecycle `ELIGIBILITY_REVIEW` . Creates a finance application in the product's tenant, keyed `marketplace:<applicationId>` (once only), with the state resolved from the applicant's state of operation. Writes audit and notifies the applicant.
- **REJECTED** : status `REJECTED` , lifecycle `DECLINED` . Notifies the applicant.

11.12 Provider webhooks

POST /api/v1/webhooks/{provider}

- **Purpose.** Receives provider callbacks. Verifies the signature, records the event durably, and triggers provider-side verification.
- **Authentication.** None. The provider HMAC signature is the authentication (see [Section 8](#)).
- **Path.** `provider` : `flutterwave` , `paystack` , `fincra` , `squad` , `termii` , `twilio` (case-insensitive).
- **Body.** The raw provider body (JSON; Twilio sends form-encoded data). Max 1 MiB.
- **Headers.** The provider's signature header (see [Section 8.1](#)).

Example (Paystack, placeholder signature):

```
curl -s -X POST https://dffms.org/api/v1/webhooks/paystack \
-H 'Content-Type: application/json' \
-H 'x-paystack-signature: <hmac-sha512-hex-of-raw-body>' \
--data-binary @paystack-event.json
```

Response (200):

```
{
  "data": { "accepted": true, "duplicate": false, "eventId": "<provider-event-id>" },
  "error": null,
  "requestId": "<request-id>"
}
```

Errors:

HTTP	Code	Meaning
401	INVALID_WEBHOOK	Signature missing or invalid; no reference; unknown or ambiguous transaction reference; invalid JSON.
404	NOT_FOUND	Provider name not supported.
413	PAYLOAD_TOO_LARGE	Over 1 MiB.
500	INTERNAL_ERROR	Unhandled error.
503	PROVIDER_DISABLED	Provider's webhook secret not configured.
503	FEATURE_UNAVAILABLE	SMS status callbacks (<code>termii</code> , <code>twilio</code>), after successful verification.

Side effects (payment providers): inbox row, outbox `payment.webhook.received` , background processing (see Section 8.2). Duplicates return 200 with `duplicate: true` and create nothing.

12. Appendix A: Generic resource catalogue

Path segment is the value after `/api/v1/` . Mode: `crud` (read, create and update), `read-only` (read only; writes return 403 or 501), `gated` (not available; returns 501). "Any" read means any active role in the tenant. "FR" is the finance read set: PSA, CSA, PA, SCA, IA, FEO, FA, AC, EDV, IV.

Write roles below are the domain write permissions (`WRITE_PERMISSIONS`) that apply to the resource's kind. PSA can always write.

Path segment	Mode	Read roles	Create or update roles (domain)	Notes
organizations	crud	Any	CSA	
programmes	crud	Any	CSA, PA	
sapzs	crud	Any	CSA, PA, SCA	
institutions	crud	Any	CSA, PA, SCA, IA	
commodities	crud	Any	CSA, PA, SCA	Kind reference_data
value-chains	crud	Any	CSA, PA, SCA	Kind reference_data
risk-assessments	crud	Any	CSA, PA, SCA, IA, MA	Create forces status DRAFT ; records assessedBy
indicators	crud	Any	CSA, PA, SCA, MA	
reports	crud	Any	CSA, PA, SCA, IA, MA	Report definitions, not exports
benchmarks	gated	n/a	n/a	501 FEATURE_UNAVAILABLE
transactions	read-only	FR	n/a	Use /finance/transactions to create
disbursements	read-only	FR	n/a	Filtered to type DISBURSEMENT
reversals	read-only	FR	n/a	Filtered to type REVERSAL
approvals	read-only	FR	n/a	
reconciliation	read-only	FR	n/a	
investments	read-only	FR	n/a	Funding records, kind INVESTMENT
loans	read-only	FR	n/a	Funding records, kind LOAN
grants	read-only	FR	n/a	Funding records, kind GRANT
exports	read-only	Any (own rows only)	n/a	Requester's own report jobs

Path segment	Mode	Read roles	Create or update roles (domain)	Notes
notifications	read-only	Any (own rows only)	n/a	Recipient's own notifications
integrations	read-only	PSA, CSA, SUPPORT, AC	n/a	Integration jobs; input , output , error redacted
admin/providers	read-only	PSA, CSA	Configure: CSA (or PSA). Route-level write returns 501 after authorization	Secrets rejected; only id , activeProvider , reason , version , expectedVersion accepted
audit-events	read-only	PSA, CSA, AC	n/a	Excludes integrity rows; hash fields redacted
facilities	read-only	FR	n/a	Authoritative facility register
facility-versions	read-only	FR	n/a	
facility-windows	read-only	FR	n/a	
facility-allocations	read-only	FR	n/a	
finance-applications	read-only (route has its own POST and GET)	FR, with application scope	n/a	Use the dedicated routes
financing-cases	read-only (dedicated routes)	FR, with case scope	n/a	Use the dedicated routes
conditions-precedent	read-only	FR, case scope	n/a	
collateral	read-only	FR, case scope	n/a	

Path segment	Mode	Read roles	Create or update roles (domain)	Notes
insurance	read-only	FR, case scope	n/a	
insurance-claims	read-only	FR, policy scope	n/a	
disbursement-schedules	read-only	FR, case scope	n/a	
repayment-schedules	read-only	FR, case scope	n/a	
warehouses	read-only	FR	n/a	
warehouse-receipts	read-only	FR, warehouse scope	n/a	
inventory	read-only	FR	n/a	Inventory lots
warehouse-inspections	read-only	FR	n/a	
inventory-movements	read-only	FR	n/a	
offtake-agreements	read-only	FR, case scope	n/a	
delivery-obligations	read-only	FR, agreement scope	n/a	
projects	read-only	FR, project scope	n/a	
spvs	read-only	FR, project scope	n/a	Special purpose vehicles
ppps	read-only	FR, project scope	n/a	

Path segment	Mode	Read roles	Create or update roles (domain)	Notes
project-milestones	read-only	FR, project scope	n/a	
project-preparation	read-only	FR, case scope	n/a	
equity-investments	read-only	FR, case scope	n/a	
guarantees	read-only	FR, case scope	n/a	
restructurings	read-only	FR, case scope	n/a	
recoveries	read-only	FR, case scope	n/a	
beneficiary-profiles	read-only	FR, case scope	n/a	

Field rules for the nine crud resources (create fields; * marks required):

Resource	Create fields	Update fields (plus id)
organizations	parentId (UUID, nullable), name * (string 240), type * (string 80), registrationNumber (string 120, nullable), status (enum), metadata (JSON)	parentId , name , type , registrationNumber , status , metadata
programmes	code * (code), name * (string 240), description (string 5,000, nullable), startsOn (date, nullable), endsOn (date, nullable), status (enum)	code , name , description , startsOn , endsOn , status
sapzs	programmeId * (UUID), stateId * (UUID), code * (code), name * (string 240), status (enum)	programmeId , stateId , code , name , status
institutions	organizationId (UUID, nullable), type * (string 80), name * (string 240), licenseNumber (string 120, nullable), status (enum)	organizationId , type , name , licenseNumber , status
commodities	code * (code), name * (string 160), unit (string 60, nullable)	code , name , unit
value-chains	commodityId (UUID, nullable), name * (string 160), stage (string 80, nullable)	commodityId , name , stage
risk-assessments	subjectType * (PROGRAMME , STATE , SAPZ , INSTITUTION , BENEFICIARY), subjectId * (UUID), score (integer 0 to 100), rating (string 60, nullable), status (enum; forced to DRAFT), factors (JSON), assessedAt (datetime, nullable)	score , rating , status , factors , assessedAt
indicators	code * (code), name * (string 200), unit * (string 80), dimensions (JSON)	code , name , unit , dimensions
reports	code * (code), name * (string 200), format * (PDF , XLSX , CSV), querySpec * (JSON), schedule (string 120, nullable), active (boolean)	code , name , format , querySpec , schedule , active

Example response from a read-only resource when the actor lacks a role (for example GET /api/v1/disbursements as MONITORING_ANALYST):

```
{  
  "data": null,  
  "error": { "code": "FORBIDDEN", "message": "You are not permitted to access this resource" },  
  "requestId": "<request-id>"  
}
```