

API-Only User *Migration*

How-To Guide

This guide provides the technical steps to migrate your GoContact API integrations to API-only users before enforcement on 21 October 2026 (release 26.20). After that date, only API-only users can authenticate against the GoContact public API; normal UI users and SSO users will be blocked (HTTP 403).

Prerequisites

PREREQUISITE	AVAILABLE FROM	ACTION
API-only user	26.06	Create or designate a non-SSO user with the API access only flag enabled
interaction_id in Report Designer	26.14	Add interaction_id to the relevant Campaign, Queue, or Script report source (recording migration only)
New public recordings endpoint in Swagger	26.14	Migrate to GET /api/voice/interactions/{interaction_id}/recordings and validate before 26.20 (recording migration only)

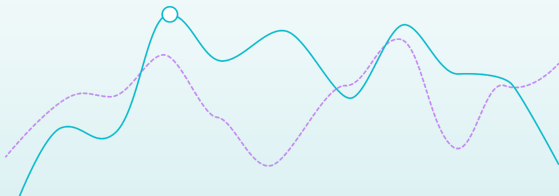
Step 1 — Create an API-only user

1. Go to Users & Groups > User detail > User permissions.
2. Enable the API access only toggle.
3. Confirm the user is a non-SSO user — SSO users are not eligible for the API-only configuration.

Notes on behavior:

- The API-only user can authenticate and access GoContact via the public API, but cannot log in to the GoContact web UI.
- Voice, Tickets, and channel-permission UI sections are not relevant to this user type.
- The JWT/session includes the API access flag so backend services enforce access.
- Existing users are not changed unless an admin explicitly enables API-only access.

Do not reuse a human UI user or an SSO user for API access. Where a single user is shared today between UI and scripts, split it into one UI user and one API-only user.



Step 2 — Migrate your integrations to the API-only user

CURRENT STATE	REQUIRED ACTION
Normal (human) UI user used for API	Create/use a dedicated non-SSO API-only user, swap credentials, retest
Shared user used by scripts and UI	Split into one UI user and one API-only user
SSO-backed user used for API	Create a new non-SSO API-only user and migrate credentials

After swapping credentials, retest against the endpoints your integration uses. Any public API call made with a non-API-only user will return 403 after enforcement.

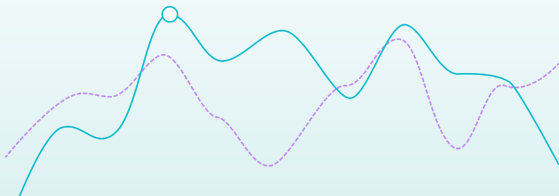
Step 3 — Migrate call recording retrieval (if applicable)

Note: The recordings endpoint specification below reflects the confirmed design agreed with engineering. The published Swagger in release 26.14 (29 July 2026) is the final source of truth — please re-confirm exact field names against the Swagger once 26.14 ships.

The legacy recording mechanism relies on a GoContact web session. API-only users have no web session, so the legacy mechanism is incompatible and stops working for API workflows after enforcement. Migrate to the new public endpoint.

What changes

ASPECT	LEGACY (CURRENT)	NEW ENDPOINT (26.14+)
Calls required	3 (sequential and dependent)	1
Authentication	User login → session cookie	Bearer token from an API-only user
State	Stateful — session cookie, requires an intermediate validation call	Stateless — token validated per call
Call identifier	Manually built file path	interaction_id (UUID) — single, stable field
Where to get the identifier	Not exposed via API; reconstructed externally (reports, CDRs)	Exported directly from Report Designer (Campaign, Queue, Script) or the reporting API
Script recordings	Not distinguished from the rest	Identified by type: "script-*" and a script_id field
Response format	Plain text or direct binary file	Structured JSON (array of objects)
File URL	Direct path, no defined expiry	Pre-signed URL (link.url) with expiry (link.expireAt)
Bulk downloads	Not recommended — the legacy path was never intended for mass downloads	One call per interaction; more predictable for automation (iterate per interaction_id)



Recognizing the legacy flow you're replacing

If your current recording retrieval works like the following, it is the legacy mechanism and must be migrated:

- 1. It logs in with a username and password to obtain a web session.
- 2. It relies on that session to authorize the download.
- 3. It downloads each recording from a file path.

Critical limitation: in the legacy flow the recording file path is not returned by any API call — it has to be known, constructed, or obtained separately (for example from reports or CDRs), which makes the integration fragile and dependent on internal file-naming conventions. The new endpoint removes this: you resolve everything from a single stable `interaction_id`.

3.1 Obtain the `interaction_id`

From release 26.14, `interaction_id` is a system field in Report Designer under the Campaign, Queue, and Script report sources. Export it from a report (or via the reporting API) to obtain the UUID for each voice interaction. This UUID is the path parameter for the recordings endpoint.

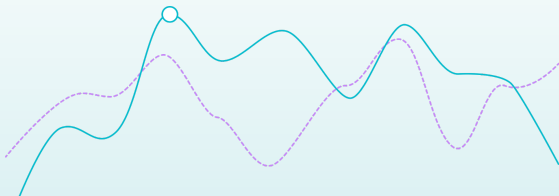
3.2 Call the new public recordings endpoint

```
GET /api/voice/interactions/{interaction_id}/recordings
Authorization: Bearer <API-only user token>

Optional query parameter:
  includeScriptRecordings=true      # default false; set true to also return script-generated recordings
```

The endpoint returns an array of recording objects. The first entry is always the primary recording (from the `recording_log` table), for backward compatibility; any additional recordings follow. Each object includes:

FIELD	DESCRIPTION
<code>contact_id</code>	Associated contact ID
<code>username</code>	GoContact user linked to the recording
<code>owner_id</code>	ID of the owning entity (campaign or queue)
<code>owner_type</code>	Type of owner: "campaign" or "queue"
<code>type</code>	Recording type. Standard recordings use values such as "always-stop". Script-generated recordings use a value starting with "script-" (e.g. "script-auto-stop", "script-manual").
<code>script_id</code>	Present only for script-generated recordings; identifies the originating script
<code>link.url</code>	Pre-signed URL to download the recording file
<code>link.expireAt</code>	Expiry timestamp for the URL



broadvoice

Script recordings are opt-in. By default (`includeScriptRecordings=false`) only the primary recording is returned. To also receive script-generated recordings, pass `includeScriptRecordings=true`. A script recording is identifiable because its type starts with `script-` and it carries an extra `script_id` field.

Example response (with `includeScriptRecordings=true`):

```
[
  {
    "contact_id": 33,
    "username": "agent1",
    "owner_id": 3,
    "owner_type": "campaign",
    "type": "always-stop",
    "link": { "expireAt": "...", "url": "..." }
  },
  {
    "contact_id": 33,
    "username": "agent1",
    "owner_id": 3,
    "owner_type": "campaign",
    "type": "script-auto-stop",
    "script_id": 42,
    "link": { "expireAt": "...", "url": "..." }
  }
]
```

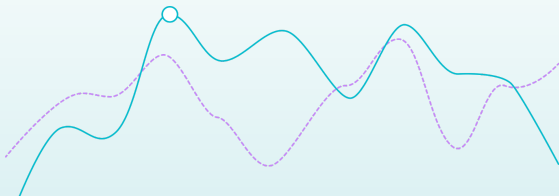
3.3 Download the recording

Use `link.url` to retrieve the file. URLs are time-limited — download or archive the file before `link.expireAt`. Do not cache the URL; call the endpoint fresh for each retrieval.

Bulk and scheduled exports: the endpoint resolves recordings for a single interaction per call. If you currently pull recordings in bulk, migrate that workflow to iterate over `interaction_id` values (one call per interaction) rather than relying on the legacy path-based approach, which was never intended for mass downloads.

3.4 Recordings endpoint error reference

CODE	MEANING	RESOLUTION
200	Success — recording array returned	—
401	Token invalid or not provided	Check API-only user credentials and token
403	User does not have access to recordings	Verify recording permissions on the API-only user
404	Recording not found	Verify <code>interaction_id</code> is correct and a recording exists for that interaction
500	Internal server error	Escalate with endpoint, user, tenant, response body, and timestamp



Narrow workaround (non-API recording retrieval only)

If you do not retrieve recordings through an API-only or automated API flow, you may continue to use the existing legacy mechanism: a normal authenticated GoContact session (PHPSESSID) plus the legacy recordings endpoint. This requires a normal (non-API-only) user with the Access Recordings and Download Recordings permissions and an active web session. It is not compatible with API-only users. Treat this as a temporary path, not a long-term replacement.

Step 4 — Migrate legacy PHP ticket creation (if applicable)

This applies if you create GoContact tickets through the legacy PHP proxy route `POST /fs/html/script-generic/php/proxy.php`. This route depends on a `PHPSESSID` web session and does not accept API-only bearer-token authentication, so it will fail for API-only users after enforcement.

4.1 Identify the integration and owner

For each affected integration, identify who owns the ticket-creation logic (internal team or partner) and confirm it will run through an API-only user.

4.2 Replace the legacy call with the supported Swagger route

```
Legacy (deprecated): POST /fs/html/script-generic/php/proxy.php (ticket creation via session cookie)
Supported:          POST /api/tickets
                    Authorization: Bearer <API-only user token>
```

`POST /api/tickets` is documented in the public Swagger and is the official replacement for the deprecated `POST_create_ticket` behavior.

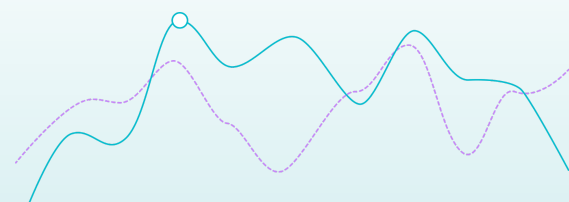
4.3 Authenticate with an API-only user

Use a non-SSO user with the API access only flag enabled (Step 1). Do not reuse a human UI or SSO user.

4.4 Validate before enforcement

Confirm tickets are created successfully through `POST /api/tickets` with the API-only user before 21 October 2026 (release 26.20). After that date the legacy proxy route no longer works for API-only credentials.

The legacy update-ticket flow through the same proxy route follows the same principle: use the documented Swagger ticket endpoints, not `proxy.php`.



Validation *checklist*

(complete before 21 October 2026)

- ☐ A dedicated non-SSO API-only user exists for each integration, script, or reporting job.
- ☐ All integration credentials have been swapped to the API-only user and retested.
- ☐ GoFlow scripts have been updated to use the API-only user.
- ☐ Any SSO-backed API users have been replaced with non-SSO API-only users.
- ☐ Recording retrieval uses GET /api/voice/interactions/{interaction_id}/recordings with interaction_id exported from Report Designer.
- ☐ includeScriptRecordings=true is set where script-generated recordings are needed.
- ☐ Recording files are downloaded before link.expireAt; URLs are not cached.
- ☐ Ticket creation uses POST /api/tickets instead of proxy.php.
- ☐ Every workflow validated end-to-end with the API-only user.

