

CHECKLIST

Trader Reactivation API Requirements Checklist

What a trader reactivation API must support before it touches MT4/MT5 or your CRM: auth, list import, campaign control, events, outcomes, webhooks.

WHO IT IS FOR

Integration engineers, CRM managers and broker ops leads evaluating a calling vendor's API for a dormant trader reactivation campaign.

TIME TO COMPLETE

60 minutes with the API docs open

This checklist is the list of questions your integration engineer asks a calling vendor before a single dormant trader gets dialled. Work through it against the vendor's API reference and you'll know whether their API can take a list from your CRM or trading platform, start and pause a campaign from your side, tell you what happened on each call and write the outcome back to the right account. Anything it can't do lands on your team as manual work, so the gaps are worth finding while you can still walk away.

1 Authentication and access control

Trader phone numbers and balances leave your systems through this API. Check how it is locked before you check what it does.

- ☐ **Confirm the API uses per-key or per-token credentials that you can create, name and revoke without contacting the vendor.**

A shared password or a key that only support can rotate is a blocker. Your security review will ask who holds it and how fast you can cut it off.

- ☐ **Check that keys are scoped to one workspace or one brand, so a white label or a second entity can't read another's lists.**

Most brokerages run more than one legal entity. One key per entity keeps the audit trail clean.

- ☐ **Confirm read-only keys exist for reporting and BI tools.**

Your data team wants outcomes and recordings, not the ability to launch a campaign.

- ☐ **Confirm every request runs over HTTPS and the vendor states where data is stored and processed.**

EU and UK client data may need to stay in region. Get the region in writing and confirm it with your compliance officer or data protection lead.

☐ **Check for IP allowlisting on the API and on webhook delivery.**

Not every vendor offers it. If yours doesn't, you'll need webhook signatures (see the webhooks section) to trust inbound calls.

☐ **Ask for an API changelog and a versioning policy with a stated notice period for breaking changes.**

An unversioned API that changes field names on a Friday will break your CRM sync on the Saturday of your busiest campaign.

☐ **Record who at your brokerage owns the keys and where they are stored.**

Key owner: _____

2 List import and contact data

Your dormant list lives in the CRM or in MT4/MT5 exports. The API has to accept it in the shape you already have, with the identifiers you need to join outcomes back later.

☐ **Confirm you can create a list and add contacts by API rather than only through CSV upload in the dashboard.**

A nightly job that pulls yesterday's newly dormant accounts and pushes them is the whole point. Manual uploads don't survive the second week.

☐ **Check the batch size per request and whether large lists can be sent in pages.**

Write the limit down. If the vendor allows 1,000 contacts per call and your dormant book is 40,000 accounts, that's 40 requests and you need a loop with retries.

☐ **Confirm each contact accepts an external ID that you set, such as the CRM record ID or the MT4/MT5 login.**

This is the join key for every outcome, recording and transcript that comes back. Without it you're matching on phone numbers, which change.

☐ **Check for custom fields per contact: last deposit date, lifetime deposits, balance band, preferred language, account manager, lead source.**

The agent script should reference these. If the API only takes name and phone, the call sounds generic and you'll hear it in the disconnect rate.

☐ **Confirm the API validates phone numbers to E.164 and tells you which rows failed, per row, in the response.**

A silent drop of 8% of your list is worse than a rejection you can see and fix.

☐ **Check for update and delete endpoints on a contact, so a trader who deposits today comes off tonight's list.**

Calling someone who reactivated themselves this morning is the most common complaint reactivation campaigns generate.

- ☐ **Confirm a per-contact do-not-call flag and a workspace-level suppression list, both writable by API.**

Your CRM opt-out and your regulatory DNC scrubs have to reach the dialler before the next batch, not after.

- ☐ **Check whether contacts can carry a consent basis and consent date as fields, or at least a tag.**
Segmenting by consent basis lets you run a smaller, safer batch first. Which basis is required for which country is a question for your compliance officer or counsel.

3 Campaign control

The campaign is where the list, the agent instructions and the calling rules meet. You need to drive it from your systems, because your CRM knows when to stop before the dialler does.

- ☐ **Confirm you can create a campaign by API and attach a list, an agent configuration and a caller ID.**

One campaign per segment (funded then dormant, demo only, KYC pending) is the normal shape. Creating them by hand each week doesn't scale past three.

- ☐ **Confirm start, pause, resume and stop endpoints, and that pause takes effect on the next dial rather than at the end of the batch.**

A compliance officer who says stop wants the calls to stop within a minute, not after the remaining 2,000 numbers.

- ☐ **Check for calling windows per country or per contact time zone, set by API.**

Your list spans time zones. The rule should attach to the contact's country, not the campaign's.

- ☐ **Check retry settings: how many attempts, the gap between them, and which outcomes trigger a retry (no answer, busy, voicemail).**

Retrying a trader who said no is a complaint. Retrying one who didn't pick up is normal. The API should let you tell the difference.

- ☐ **Confirm a maximum concurrent calls setting per campaign.**

Your retention desk can take a handoff rate of, say, six warm transfers an hour. Concurrency is how you keep the AI from overrunning them.

- ☐ **Check whether agent instructions and the opening line can be updated by API without recreating the campaign.**

You'll rewrite the opening line after the first 200 calls. Versioning matters less than being able to do it fast.

- ☐ **Confirm campaign-level read endpoints return dialled, connected, in progress and remaining counts.**

This is what your ops dashboard polls. If the numbers only exist in the vendor's UI, someone is copying them into a spreadsheet.

- ☐ **Check for a test or single-call endpoint that dials one number with the campaign config.**

Your team calls their own phones before the list does. Every vendor should let you do that from the API.

4 Call events and live data

Events are what the dialler tells you while a call is happening. They feed your CRM timeline, your agent desk and your monitoring, so check the list against what you'd act on.

☐ **List the call lifecycle events the API emits and match each one to an action on your side.**

The table below is the minimum set for a reactivation campaign. Tick which ones the vendor supports.

☐ **Confirm every event carries your external contact ID, the campaign ID and a stable call ID.**

Without all three you can't tell which trader, which segment and which attempt the event belongs to.

☐ **Check that events include timestamps in UTC with the contact's time zone available separately.**

Reports that mix local times across your MENA, EU and APAC lists are unreadable.

☐ **Confirm a warm transfer or handoff event fires when the AI hands a trader to a human, with the destination number or queue.**

The retention agent needs the trader's context on screen before they say hello. That's a CRM screen-pop triggered by this event.

☐ **Check whether a live transcript or a mid-call intent signal is available, and whether you'd use it.**

A deposit intent flag during the call can alert the account manager in real time. Nice to have, not required for a first campaign.

☐ **Confirm the answering machine and voicemail detection result is reported as its own event or field.**

Voicemails count against your attempt budget. You want them tagged, not mixed in with connected calls.

EVENT	PAYLOAD YOU NEED	YOUR ACTION	SUPPORTED?
Call queued	contact external ID, campaign ID, call ID	Mark contact as in progress in CRM	—
Call started	call ID, caller ID used, start time	Open CRM activity	—
Voicemail or machine detected	call ID, detection result	Count attempt, schedule retry per rules	—
Handoff to human	call ID, destination, reason	Screen-pop trader record on the retention desk	—
Call ended	call ID, duration, end reason, cost	Close activity, trigger outcome fetch	—
Recording ready	call ID, recording URL, expiry	Attach to CRM record per retention policy	—
Transcript ready	call ID, transcript, language	Store, run QA sampling	—

5 Outcomes and write-back to the CRM

The outcome is the one thing your sales and retention teams read. It has to arrive in a shape your CRM can act on without a human translating it.

- ☐ **Confirm each completed call returns a structured outcome rather than a free-text summary alone.**

Interested, callback requested, not interested, wrong number, do not call, deposit intent. Your CRM workflows key off values, not paragraphs.

- ☐ **Check that you can define your own outcome labels or map the vendor's to yours.**

Your CRM already has a disposition field with values your team knows. The integration should fill it, not add a second one.

- ☐ **Confirm callback requests come back with the requested date, time and time zone as fields.**

A trader who says call me Thursday after 6 should appear in Thursday's queue automatically. That needs structured data.

- ☐ **Check that a do-not-call outcome on a call updates the contact's suppression flag by itself.**

If it doesn't, write the rule on your side and test it before the second batch. Nobody should be dialled twice after asking not to be.

- ☐ **Confirm the summary, the sentiment or intent fields and any extracted answers (reason for going dormant, objection raised) are available per call.**

Reason for leaving is the most useful field a reactivation campaign produces. Store it on the account as well as on the call.

- ☐ **Check that outcomes can be fetched by call ID, by contact ID and by campaign with a date range.**

Real-time delivery fails sometimes. A pull endpoint is how you reconcile the gaps every morning.

- ☐ **Confirm the recording URL and transcript are retrievable later by call ID, and how long the vendor keeps them.**

Your retention period for call recordings is a compliance decision. Write it down and confirm the vendor's default doesn't outlive it.

- ☐ **Map every outcome value to a CRM field and a follow-up owner before the first batch.**

Outcomes mapped: _____ of _____

6 Webhooks and automation path

Webhooks push events to you as they happen. Polling works for reconciliation, but the screen-pop and the same-minute CRM update depend on push.

- ☐ **Confirm webhooks exist for the events in the table above and that you can subscribe per event type.**

Subscribing to everything and filtering on your side works, but it doubles the traffic your endpoint handles.

- ☐ **Check that each webhook is signed (an HMAC header or similar) and that the signing secret can be rotated.**

Without a signature anyone who guesses your endpoint can inject a fake deposit intent outcome into your CRM.

- ☐ **Confirm the retry policy for failed deliveries: how many attempts, over how long, and whether you can replay from the dashboard or API.**

Your endpoint will be down at some point. What matters is whether the events queue or vanish.

- ☐ **Check for a delivery log you can read, with request body, response code and timestamp.**

When the CRM shows no outcome for a call, this log settles whether the vendor sent it or your side dropped it.

- ☐ **Confirm the payload includes an event ID so your handler can ignore duplicates.**

Retries produce duplicates by design. An idempotency key on your side stops a callback being created twice.

- ☐ **Check whether the vendor offers a no-code automation path to your CRM alongside raw webhooks.**

Topcalls reaches 764+ tools through its Integrations catalog, which covers HubSpot, Salesforce and Pipedrive without an engineer. Raw webhooks are still the right choice for MT4/MT5 write-back.

- ☐ **Test a webhook end to end with a single call before the campaign starts and record the round-trip time from call end to CRM update.**

Call end to CRM update: _____ seconds

7 Rate limits, errors and operations

The boring section that decides whether your nightly sync runs for a year without anyone touching it.

- ☐ **Record the documented rate limits per endpoint and per key, and how the API tells you when you've hit one.**

A 429 response with a retry-after header is the standard. Anything else means writing your own backoff logic against guesswork.

- ☐ **Confirm error responses are structured, with a code and a message per failed item in a batch.**

One bad phone number in a batch of 500 should not fail the other 499.

- ☐ **Check for a sandbox or test mode that accepts real requests without dialling real traders or billing minutes.**

Your integration tests will run every deploy. They can't ring a customer.

- ☐ **Confirm a status page or health endpoint exists and that incidents are announced somewhere your on-call can subscribe to.**

Whatever your vendor states, ask where the outages are posted.

☐ **Check that usage and cost are readable by API per campaign and per call.**

Cost per reactivated trader is your KPI. With a single all-inclusive per-minute rate like Topcalls' \$0.35/min, cost per call is minutes times rate, and you can compute it from the call-ended event.

☐ **Confirm you can export or delete a contact's data by API to answer a data subject request.**

GDPR access and erasure requests arrive with a deadline. Your compliance officer needs to know the vendor side can be cleared without a support ticket.

☐ **Read the SDKs and code samples in the language your team uses, and note what's missing.**

No SDK is fine if the REST reference is complete. A thin SDK with undocumented behaviour is worse than none.

☐ **Sign off the checklist with the integration owner and the compliance officer before the pilot batch.**

Signed: _____ / _____ Date: _____

How to use this checklist

1. Open the vendor's API reference next to this checklist and tick each item as supported, partly supported or missing. Partly supported gets a note on what your team would build around it.
2. Fill the call-events table first. It tells you fastest whether the API was designed for outbound campaigns or bolted on to a dashboard product.
3. Count the missing items that your CRM manager, your compliance officer and your on-call engineer each care about. Three separate lists, three separate conversations with the vendor.
4. Run the single-call test and the webhook round-trip test before you negotiate anything. Vendors quote what the docs say; the test shows what the API does.
5. Keep the signed copy with the vendor contract. When something breaks in month four, this is the record of what was promised.

Book a 30-minute call with a Topcalls co-founder

We'll map your CRM and MT4/MT5 flow to the Topcalls API and webhooks, size the dormant book and scope a pilot batch with outcomes written back to your records.

topcalls.ai/book