Admin API

Research & Discovery

Overview

The Admin API gives Kit and admin roles programmatic access to Duo. You need Owner-level access to use it. It supports creating, updating, and deleting users, phones, hardware tokens, admins, and integrations, and access to logs.

What should we work on?

Select a focus area

- Ability to add and manage AI agents
- Improve SSO adoption
- Improve Duo Directory adoption
- Work on the top customer complaints
- Something else

My initial suggestions

These are early-stage suggestions based on the research below, not yet validated against usage data.

- Solve the **rate limiting issue**, Kit needs to be able to generously use these APIs without running into errors.
- Add **more bulk CRUD** options, this would help scale adoption for any feature.
- Build a **setup boilerplate** - Write instructions for claude/shell script to run and set up via Duo directory. Include recommended policies and other settings, so Kits have a working sandbox to explore from.

Risks

> - These recommendations don't address the two highest-confidence use cases — SIEM ingestion and user lifecycle management — which are both read-heavy workflows. Kits may primarily use the API to pull data, not create or modify it, making bulk CRUD and rate limit improvements less impactful than assumed.

Gaps

Unsupported features:

Source: Sangeetha

- SSO named integrations — only generic integrations are available via the API
- Self-Service Portal (SSP)
- Provisioning
- Duo Directory / password reset
- MSP subaccount management via API
  - may need to use the Accounts API instead
- Adding and managing AI Agents (MCPs)

Customer reported Gaps:

Source: Claude internet scraping

- **Opaque rate limiting** — The API enforces multi-layer throttling (per-IP, per-key, per-key+IP) without publishing concrete per-endpoint limits, causing integrations to hit HTTP 429s well below the documented 100/min threshold. Multiple integrations sharing one tenant silently starve each other with no visibility.
- **Index-based pagination with race conditions** — Offset-based paging means records can be missed or duplicated if inserts or deletes happen mid-retrieval. The auth log v2 endpoint compounds this with a completely different pagination scheme that is poorly documented.
- **No bulk write operations** — The only bulk endpoint is Bulk Create Users, which fails the entire batch if one record is invalid. There is no bulk disable, bulk group change, or bulk policy assignment.
- **Admin Panel features with no API equivalent** — SSO/SAML/OIDC app configuration, full directory sync triggering, Trust Monitor events, and auth summary analytics are all UI-only or only partially accessible via the API.
- **Weak documentation** — HMAC signing instructions are ambiguous enough that developers following them verbatim get 401 errors. There are no official examples for HTTP POST requests and no PowerShell SDK.
- **SDKs lag the REST surface** — The Python client has 11+ open issues and doesn't expose several documented endpoints. The Java SDK is on v0.8.0 with similar gaps, and v3 endpoints broke the Python client as of mid-2025.
- **Fragile MSP/Accounts API multi-tenancy** — Child accounts on a different Duo deployment than the parent return HTTP 400 errors that can only be fixed by filing a support ticket. Cross-tenant queries are not possible without iterating per child account.
- **Limited log filtering and latency** — Auth logs cannot be filtered by account, bypass status, or service account, forcing client-side filtering. An undocumented ~2-minute consistency window means recent events return inconsistent results.
- **Version fragmentation and migration churn** — An ongoing v1→v2→v3 migration with inconsistent pagination semantics across versions, SDK clients that lag new handlers, and a hard CA bundle expiration cutoff in April 2026 create ongoing maintenance burden for customers.

- **Hard object caps return HTTP 500** — Exceeding one-to-many limits (e.g. 100 groups per user) returns a generic 500 Internal Server Error instead of a proper 4xx validation response, causing support tickets to be misdiagnosed.

What Do Kits Do With the Admin API?

Source: Claude synthesis. To be validated.

1. **SIEM Log Ingestion** — Pulling auth, admin, and telephony logs into tools like Splunk, Elastic, and Sumo Logic. *High confidence*
2. **User Lifecycle Management** — Syncing user creation and deletion with HR/IGA systems at onboarding and offboarding. *High confidence*
3. **SOAR Incident Response** — Disabling compromised accounts or removing group memberships via automated playbooks. *Medium confidence*
4. **Compliance Reporting & Audit Evidence** — Generating snapshots of bypass users, unenrolled devices, and admin activity for SOC 2, SOX, and HIPAA. *Medium confidence*
5. **MSP Multi-Tenant Management** — Onboarding clients, retrieving bypass users, and assigning policies across dozens of child accounts. *Medium confidence*
6. **Endpoint & Device Inventory** — Auditing which devices are accessing corporate resources and flagging out-of-policy endpoints. *Medium-low confidence*
7. **Group & Policy Management** — Mirroring AD/Azure group changes into Duo and assigning policies across apps at scale. *Medium-low confidence*
8. **App/Integration Onboarding** — Scripting new Duo integration creation as part of IaC pipelines when apps are deployed. *Low confidence*
9. **Hardware Token & Phone Fleet Management** — Bulk-importing tokens, associating them to users at provisioning, and cleaning up orphaned devices post-offboarding. *Low confidence*

Competitive analysis

Source: Claude synthesis.

- **Product Scope** — Okta covers full identity lifecycle (SSO, provisioning, governance, MFA); Duo is MFA and device trust only. *Okta broader*
- **Auth Model** — Okta uses industry-standard OAuth 2.0; Duo uses custom HMAC signing, which is error-prone for new developers. *Okta advantage*
- **Rate Limits** — Okta publishes per-endpoint limits with a real-time dashboard; Duo limits are undocumented and silent. *Okta advantage*
- **Webhooks** — Okta supports event-driven push notifications; Duo is polling only. *Okta advantage*
- **Bulk Operations** — Okta uses SCIM for robust bulk provisioning; Duo only supports bulk user creation. *Okta advantage*
- **SSO Config via API** — Okta has full CRUD for SAML/OIDC apps; Duo blocks SSO app modification entirely. *Major Duo gap*
- **Policy API** — Okta has deep policy coverage; Duo added Policy CRUD in 2023 but bulk-apply and SSO policies are still missing. *Okta advantage*
- **User Lifecycle** — Okta has reversible state transitions (staged → active → deactivated); Duo deletes are permanent. *Okta advantage*
- **Error Quality** — Okta returns structured 4xx errors; Duo returns HTTP 500 for validation failures. *Okta advantage*

- **MFA Management** — Duo has deeper MFA-specific API coverage (bypass codes, tokens, WebAuthn, TOTP). *Duo advantage*
- **Multi-Tenant/MSP** — Both have friction; Duo's cross-deployment bugs require support tickets. *Roughly equal*
- **Pricing** — Both include API access with paid plans. *Roughly equal*

To do

- [ ] Interview customers
- [ ] Look at Salesforce tickets
- [ ] Use the Duo admin api
- [ ] Set up an SSO integration using the Okta API
- [ ] How would I go about structuring what the team should investigate to assess level of effort?
- [ ] A strategy and broad business case