Okta vs Duo

Category-by-category breakdown

Category	Duo Admin API	Okta Management API	Edge
Product scope	MFA, device trust, and authentication management only	Full identity lifecycle: SSO, provisioning, governance, MFA, directory	**Okta** — broader surface by design
Authentication model	HMAC-SHA1 signed requests; custom Basic Auth header; per-integration key+secret	OAuth 2.0 scoped access tokens (recommended) or legacy SSWS API token Okta Developer	**Okta** — OAuth 2.0 is industry standard; Duo's HMAC signing is custom and error-prone for developers new to the platform
Rate limiting transparency	Limits not published per endpoint; multi-layer silent throttling (per-IP, per-key+IP, per-key); 429s frequently hit below documented thresholds	Bucket-based system with published per-endpoint quotas (e.g. 1,000 req/min for `/api/v1/users/*`); limits visible in the Admin Console rate limit dashboard Okta Developer	**Okta** — limits are documented and observable
Rate limit management	No per-key controls; workaround is one key per integration	Per-token and per-OAuth app rate limit allocation configurable in the Admin Console or via API — admins set each token's percentage of the org-wide bucket Okta Developer	**Okta** — granular control over how the quota is shared
Rate limit observability	No dashboard; 429 is the only signal	Real-time rate limit dashboard under Reports > Rate Limits; shows top offending APIs, usage trends, burst/warning/violation events; links to System Log for filtered events Okta Developer	**Okta** — significantly better

Burst tolerance	No burst allowance; hard limit	Burst rate limits allow temporary usage above the threshold for auth/authz endpoints when system capacity allows, to protect end-user experience during traffic spikes Okta Developer	**Okta**
Pagination	Integer offset-based; race conditions on concurrent writes; different scheme for auth logs vs. all other endpoints	Cursor-based pagination using opaque `before`/`after` parameters in the `Link` header; stable across concurrent writes Okta Developer	**Okta** — cursor pagination is structurally safer
Webhook / event push	No native webhook/push; polling only	Event hooks (webhooks) for push notifications when events occur — up to 25 active verified hooks per org, each subscribable to multiple event types Okta Developer	**Okta** — push vs. poll is a major operational difference for real-time use cases
Bulk operations	Bulk Create Users only (100/request, fail-entire-batch semantics); no bulk delete, disable, or modify	Native SCIM 2.0 support for bulk provisioning via apps; Okta itself doesn't expose a bulk REST endpoint but integrates with SCIM-compliant upstream systems for mass sync	**Okta** — SCIM-based bulk provisioning is more robust, though Okta's own REST API also lacks a true bulk endpoint
SSO / app config via API	Cannot modify Duo SSO applications via API; v1 endpoint explicitly blocked	Full Apps API CRUD — create, configure, and delete SAML/OIDC app integrations programmatically	**Okta** — this is a core gap for Duo
Policy management	Policy CRUD added 2023; bulk-apply still pending; cannot configure SSO-related policies via API	Full Policy API covering global session, authentication, password, and MFA enrollment policies; inline hooks can modify policy decisions in real-time	**Okta** — deeper policy API surface
User lifecycle	Create, read, update, delete users; soft-delete missing (API	Full lifecycle: staged → active → deactivated → deleted; deactivated users can be	**Okta** — reversible state machine vs. Duo's permanent API deletes

	deletes are permanent and immediate)	reactivated; full status transitions available	
Directory sync	Single-user sync trigger available; no full on-demand sync via API; scheduling via UI only	Full provisioning and deprovisioning via SCIM; push and import flows; event hooks for provisioning events added 2025	**Okta**
SDKs	Python, Java, Go, Node, Ruby, Perl, PHP, C# — but SDK coverage lags REST surface; no PowerShell	OpenAPI specification available; official SDKs for major languages; API reorganized into functional service groups in 2025 Okta Developer	**Okta** — OpenAPI spec enables auto-generated clients; Duo has no OpenAPI spec (though one was referenced as "coming soon" in a 2023 blog post)
Multi-tenant / MSP	Accounts API + Admin API; cross-deployment failures require support ticket; no cross-tenant queries	Okta Aerial for multi-org management; per-org API tokens; some cross-org federation via Org2Org	**Roughly equal** — both have MSP friction; Duo's cross-deployment bug is more painful; Okta Aerial has its own complexity
Error quality	Hard object cap violations return HTTP 500 (misleading); generic error messages	4xx errors returned with structured JSON: errorCode, errorSummary, errorId — all requests returning errors return appropriate 4xx or 5xx codes Okta Developer	**Okta** — 500s for validation failures is a significant Duo gap
Compliance / audit logging	Admin logs, auth logs, telephony logs via API; no filtering by account in MSP context	System Log API with rich filtering (SCIM filter syntax, date ranges, actor, target, event type); event types queryable via catalog	**Okta** — System Log API is significantly more filterable
Device / endpoint management	Endpoints API for trusted device inventory	Devices API for device identity and lifecycle management	**Roughly equal** — different depth; Duo stronger on MFA-specific device trust signals

Core MFA management	Deep: bypass codes, hardware tokens, phones, WebAuthn, TOTP — all CRUD via API	User Factors API covers MFA factor enrollment and management, but MFA is secondary to Okta's identity focus	**Duo** — richer MFA-specific API surface
Pricing / access	Included with all paid plans (Essentials, Advantage, Premier)	Management API access included; some advanced features (DynamicScale rate limit increases) are paid add-ons	**Roughly equal**