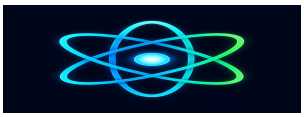

OKTA API ACCESS GUIDE

Okta API Access Setup

Configure the read-only Okta API Services app Atomation uses to inspect posture, capture evidence, and run assessment scans.

Field	Value
Owner	Customer Okta admin
Audience	IAM, security, audit
Use case	Assessment data plane
Source page	/docs/okta-api-access/
PDF filename	atomation-okta-api-access-guide.pdf
Status	Read-only setup guide
Screenshot policy	Insert approved Okta Admin Console or Atomation workspace captures before final distribution.

This document is an original Atomation guide. Vendor documentation is used as reference material only.

Contents

Before You Start

Atomation uses a customer-created Okta API Services app with read-only scopes. The customer registers the Atomation tenant JWKS URL in Okta. Atomation keeps the private key encrypted and never asks for customer admin passwords.

Need	Owner	Notes
Okta administrator	Customer	Can create API Services apps.
Scope grant ability	Customer	In many orgs, a Super Admin must grant Okta API scopes.
Tenant JWKS URL	Atomation	Tenant-specific public key URL.
Client ID	Customer	Copied from Okta after app creation.

Okta service apps need both scope grants and admin role assignment. Scopes allow token claims; roles control what the app may read.

Security Model for Review

This connection is designed for assessment visibility, not operational administration. It should be reviewed as a scoped service app with explicit customer approval, known owner, and documented verification state.

Control	Expectation	Evidence
Credential model	Private key remains with Atomation; Okta fetches the public key from the tenant JWKS URL.	Client Credentials screen and Atomation verification result.
Scope model	Grant only read scopes required for assessment evidence.	Okta API Scopes tab and Atomation scope check.
Role model	Assign an approved admin role that gives sufficient read visibility.	Admin Roles tab and successful read-call verification.
Change boundary	The assessment connection should not use write scopes.	No write scopes present during verification.

Setup Steps

Step 1

Create the API Services app

In Okta Admin Console, go to Applications -> Applications, click Create App Integration, choose API Services, and name the app using the generated Atomation service app name.

SCREENSHOT PLACEHOLDER

Create App Integration

Okta Admin Console screen for selecting the API Services app integration type.

SCREENSHOT PLACEHOLDER

API Services app type

API Services selected as the sign-in method.

Step 2

Register the Atomation JWKS URL

Under Client Credentials, select Public key / Private key, choose Use a URL to fetch keys dynamically, paste the tenant JWKS URL, save the app, and copy the Client ID.

SCREENSHOT PLACEHOLDER

Public key URL

Client Credentials screen showing dynamic public key URL configuration.

SCREENSHOT PLACEHOLDER

Client ID

Client ID copied from the saved Okta app.

Step 3

Assign the service app admin role

On the Admin Roles tab, assign a role that gives the service app the read visibility needed for the assessment. Use the least-privilege role model the customer approves.

SCREENSHOT PLACEHOLDER

Admin Roles tab

App role assignment for approved read visibility.

Step 4

Grant read-only Okta API scopes

On the Okta API Scopes tab, grant the required read scopes. Do not grant write scopes. Require DPoP when the Okta org and Atomation connection both support it.

SCREENSHOT PLACEHOLDER

Okta API Scopes

Required read-only scopes granted to the API Services app.

Step 5

Verify in Atomation

Paste the Client ID into Atomation and click Verify Connection. Atomation checks token acquisition, scope usability, role visibility, and last verified timestamp.

SCREENSHOT PLACEHOLDER

Atomation verification

Token acquired, scopes usable, role sufficient, and last verified timestamp.

Required Read-Only Scopes

Scope	Purpose
okta.users.read	Users and lifecycle state.
okta.groups.read	Groups and group membership.
okta.apps.read	Applications, assignments, and app configuration.
okta.policies.read	Sign-on, authenticator, password, and app policy posture.
okta.logs.read	System Log evidence for review windows.
okta.networkZones.read	Network zone posture.
okta.trustedOrigins.read	Trusted origin posture.
okta.roles.read	Admin role assignments and custom roles.
okta.authorizationServers.read	Authorization server posture.
okta.idps.read	Identity provider and routing posture.

Verification and Common Errors

Error	Likely cause	Fix
invalid_client	Client ID mismatch or JWKS URL not saved.	Re-check Client ID and Client Credentials settings.
Scope denied	Scope not granted.	Grant the missing read scope.
403 on read call	Admin role assignment missing or too narrow.	Assign an approved role with sufficient visibility.
DPoP failure	DPoP setting mismatch.	Confirm DPoP support and setting.

- Okta service app OAuth guide: https://developer.okta.com/docs/guides/implement-oauth-for-okta-serviceapp/main/
- Okta OAuth API setup guide: https://developer.okta.com/docs/guides/set-up-oauth-api/main/