

NetFortress

Complete API Reference

Every endpoint with verified authentication requirements and real response field structures

Appliance version 3.13.40 · 189 paths · 225 operations · second edition

Auth verified from the route source code · response fields from live sampling and schema expansion

1. How to use this reference

- Base URL: **https://YOUR-APPLIANCE/api/v1** or **http://LAN-IP:8080/api/v1**.
- **Authentication was verified per endpoint from the appliance source code** - not assumed. Three levels appear in this reference: **Public** (no token), **Token** (session or API token), and **Token - scope: X** (the token must carry the listed scope). A 401 means a bad/expired token; 403 means the token lacks the scope.
- **Response fields** are documented three ways, whichever is authoritative for the endpoint: expanded schema models ({field: type}, an asterisk * marks optional fields), live-sampled field shapes from a running appliance, or the returned field list for simple confirmation objects.
- API tokens (console: Users → API tab) use the same Bearer header as sessions. The websocket is the one exception: its token rides the query string.
- The live contract is always at **GET /openapi.json** (public) on the appliance itself.

2. Live-monitoring cookbook

Purpose	Call	Cadence	Notes & auth
Heartbeat / up-down	/api/v1/health	5–30 s	Version, backend, apply mode — public
Per-second live throughput	/api/v1/traffic/live	1 s	Server-side /proc/net/dev deltas: smooth Mbit/s graphs — token (audit:read)
Historical traffic	/api/v1/traffic	60 s	vnstat-backed history for trends — token (audit:read)
Live security events	/ws/events (websocket)	push stream	Token on query string; one JSON message per event — ?
Event history	/api/v1/events	60 s	Query/paginate recorded security events — token (audit:read)
Dashboard overview	/api/v1/system/overview	10–30 s	The numbers the appliance dashboard shows — token (deployment:read)
Device inventory	/api/v1/devices	60 s	Discovered devices: hostnames, IPs, MACs, groups — token (deployment:read)
Threat counters	/api/v1/threat/status	60 s	Threat-feed hit counters and settings — token (deployment:read)
Service states	/api/v1/system/services	30 s	Per-service status for health boards — token (deployment:read)
Integrity / tamper	/api/v1/system/integrity	300 s	Baseline integrity - shows tampered on file drift — token (deployment:read)
Licence state	/api/v1/licensing/status	300 s	Edition, expiry, days remaining, device counts — public

3. Complete endpoint catalogue

225 operations across 42 groups. Auth verified from source; * marks optional fields. Response field lists are authoritative for this build.

Health (1)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/health	Health Public - no token required Response (live-sampled): {status: str, version: str, firewall_backend: str, apply_mode: str, network_apply: str}

Authentication & Sessions (14)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/auth/status	Auth Status Public - no token required Response (live-sampled): {bootstrap_required: bool, authentication_enabled: bool}
POST	/api/v1/auth/bootstrap	Bootstrap Public - no token required Response (AuthResponse): {access_token*: string, token_type*: string, expires_in*: integer, user*: object, requires_2fa*: boolean, challenge*: string} Request body: {username: string, password: string}
POST	/api/v1/auth/login	Login Public - no token required Response (AuthResponse): {access_token*: string, token_type*: string, expires_in*: integer, user*: object, requires_2fa*: boolean, challenge*: string} Request body: {username: string, password: string}
POST	/api/v1/auth/login/2fa	Login Second Factor Public - no token required Response (AuthResponse): {access_token*: string, token_type*: string, expires_in*: integer, user*: object, requires_2fa*: boolean, challenge*: string} Request body: {challenge: string, code: string}
POST	/api/v1/auth/logout	Logout Public - no token required Response: object with fields ok
GET	/api/v1/auth/2fa/status	Two Factor Status Public - no token required Response (live-sampled): {enabled: bool}
POST	/api/v1/auth/2fa/setup	Two Factor Setup Public - no token required Response: object with fields secret, uri
POST	/api/v1/auth/2fa/enable	Two Factor Enable Public - no token required Response: object with fields enabled, recovery_codes Request body: {secret: string, code: string}
POST	/api/v1/auth/2fa/disable	Two Factor Disable Public - no token required Response: object with fields enabled Request body: {password: string, code: string}

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/auth/me	Me Public - no token required Response (live-sampled): {id: int, username: str, role: str, actor_type: str, scopes: ['str']}
GET	/api/v1/auth/tokens	List Tokens Token - scope: tokens:write Response: ['<empty>']
POST	/api/v1/auth/tokens	Create Token Token - scope: tokens:write Response: object with fields id, name, token, prefix, scopes, expires_at, warning Request body: {name: string, scopes: list[string], expires_at*: string (date-time) / null}
DELETE	/api/v1/auth/tokens/{token_id}	Revoke Token Token - scope: tokens:write Response: HTTP 204 - no body
POST	/api/v1/auth/support	Support Sign In Public - no token required Response (AuthResponse): {access_token*: string, token_type*: string, expires_in*: integer, user*: object, requires_2fa*: boolean, challenge*: string} Request body: {code: string}

Setup State (1)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/setup-state	Setup State Public - no token required Response (live-sampled): {pending: bool, redirect_url: NoneType}

System (27)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/system/interfaces	Interfaces Token - scope: deployment:read Response: ['str']
PUT	/api/v1/system/interfaces/labels	Save Interface Label Token - scope: deployment:write Response: object with fields interface, label Request body: {interface: string, label*: string}
GET	/api/v1/system/hostname	Get Hostname Token - scope: deployment:read Response (live-sampled): {hostname: str}
PUT	/api/v1/system/hostname	Save Hostname Token - scope: deployment:write Response: object with fields hostname, transaction Request body: {hostname: string}
GET	/api/v1/system/overview	Overview Token - scope: deployment:read Response (live-sampled): {hostname: str, kernel: str, uptime_seconds: int, load_average: ['str'], cpu_cores: int, memory: {total_bytes: int, available_bytes: int, used_percent: int}}
GET	/api/v1/system/enforcement	Get Enforcement Token - scope: deployment:read Response (live-sampled): {flags: ['str'], restart_pending: bool}

	Endpoint	Purpose · auth · request/response fields
PUT	/api/v1/system/enforcement	Set Enforcement Token - scope: deployment:write Response: object with fields ok, flag, value, restart_scheduled, message Request body: {flag: string, value: integer}
PUT	/api/v1/system/enforcement/batch	Set Enforcement Batch Token - scope: deployment:write Response: object with fields ok, applied, restart_scheduled, message, ok, applied, restart_scheduled, message Request body: {updates: list[EnforcementUpdate]}
GET	/api/v1/system/integrity	Get Integrity Token - scope: deployment:read Response (live-sampled): {ok: bool, files: int, sha256: str}
GET	/api/v1/system/support	Get Support Token - scope: deployment:read Response (live-sampled): {active: bool}
POST	/api/v1/system/support/grant	Grant Support Token - scope: deployment:write Response: the created/updated object - see the matching GET field list in this section Request body: {duration_hours: integer, reason*: string}
POST	/api/v1/system/support/revoke	Revoke Support Token - scope: deployment:write Response: the created/updated object - see the matching GET field list in this section
GET	/api/v1/system/acme	Acme Settings Token - scope: deployment:read Response (live-sampled): {enabled: bool, email: str, zone: str, appliance_fqdn: str, block_fqdn: str, api_token: str, issued: bool, expires_at: str, certificate_names: [str], vendor: {hostname: str, block_hostname: str, active: bool}}
PUT	/api/v1/system/acme	Save Acme Settings Token - scope: deployment:write Response: object with fields enabled, appliance_fqdn, block_fqdn, api_token Request body: {enabled*: boolean, email*: string, zone*: string, appliance_fqdn*: string, block_fqdn*: string, api_token*: string}
POST	/api/v1/system/acme/request	Request Acme Certificate Token - scope: deployment:write Response: the created/updated object - see the matching GET field list in this section
POST	/api/v1/system/reboot	Reboot Appliance Token - scope: deployment:write Response: object with fields status, message
GET	/api/v1/system/services	Services Token - scope: deployment:read Response: [str]
GET	/api/v1/system/update-status	Get Update Status Token - scope: deployment:read Response (live-sampled): {mode: str, current_version: str, available: bool, version: str, notes: str}
POST	/api/v1/system/update-status	Check For Updates Now Token - scope: deployment:write Response: object with fields mode, current_version, available, version, notes
POST	/api/v1/system/update/apply	Apply Published Update Token - scope: deployment:write Response: the created/updated object - see the matching GET field list in this section

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/system/traffic	System Traffic Token - scope: deployment:read Response (live-sampled): {interface: str, rx_bytes: int, tx_bytes: int, ts: float}
GET	/api/v1/system/deep-check	System Deep Check Token - scope: deployment:read Response (live-sampled): {ran_at: str, summary: {ok: int, warn: int, fail: int}, checks: ['str']}
GET	/api/v1/system/docs/{filename}	Get Documentation Public - no token required Response: the created/updated object - see the matching GET field list in this section
GET	/api/v1/system/admin-port	Get Admin Port Token - scope: deployment:read Response (live-sampled): {port: int, reserved: ['str']}
PUT	/api/v1/system/admin-port	Set Admin Port Token - scope: deployment:write Response: object with fields port, previous_port, changed, message, port, previous_port, changed, message Request body: {port: integer}
GET	/api/v1/system/update-mode	Get Update Mode Token - scope: deployment:read Response (live-sampled): {mode: str}
PUT	/api/v1/system/update-mode	Set Update Mode Token - scope: deployment:write Response: object with fields mode, message Request body: {mode: string}

Traffic (2)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/traffic	Traffic Token - scope: audit:read Response (live-sampled): {available: bool, interfaces: ['<empty>']}
GET	/api/v1/traffic/live	Traffic Live Token - scope: audit:read Response (live-sampled): {window_seconds: int, interfaces: ['<empty>']}

Security Events (2)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/events	Security Events Token - scope: audit:read Response: ['str']
POST	/api/v1/events	Ingest Security Event Token - scope: events:write Response: the created/updated object - see the matching GET field list in this section Request body: {event_type: string, action: string, client_ip*: string / null, hostname*: string / null, destination*: string / null, category*: string / null, policy*: string / null, details*: object}

Devices (1)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/devices	Devices Token - scope: deployment:read Response: ['str']

Device Groups (4)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/device-groups	List Device Groups Token - scope: policies:read Response: ['<empty>']
POST	/api/v1/device-groups	Create Device Group Token - scope: policies:write Response: the created/updated object - see the matching GET field list in this section Request body: {name: string, description*: string}
POST	/api/v1/device-groups/{group_id}/members	Add Device Group Member Token - scope: policies:write Response: the created/updated object - see the matching GET field list in this section Request body: {target_type: string, target_value: string}
DELETE	/api/v1/device-groups/{group_id}	Delete Device Group Token - scope: policies:write Response: HTTP 204 - no body

Users (5)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/users	List Users Token - scope: users:read Response: ['str']
POST	/api/v1/users	Create User Account Token - scope: users:write Response: the created/updated object - see the matching GET field list in this section Request body: {username: string, password: string, role*: string, group_id*: integer / null}
PUT	/api/v1/users/{user_id}	Update User Account Token - scope: users:write Response: the created/updated object - see the matching GET field list in this section Request body: {role: string, enabled*: boolean, group_id*: integer / null}
PUT	/api/v1/users/{user_id}/password	Set User Password Token - scope: users:write Response: object with fields ok Request body: {new_password: string}
DELETE	/api/v1/users/{user_id}	Delete User Account Token - scope: users:write Response: HTTP 204 - no body

User Groups (4)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/user-groups	List User Groups Token - scope: users:read Response: ['str']

	Endpoint	Purpose · auth · request/response fields
POST	/api/v1/user-groups	Create User Group Token - scope: users:write Response: the created/updated object - see the matching GET field list in this section Request body: {name: string, description*: string, permissions: list[string]}
PUT	/api/v1/user-groups/{group_id}	Update User Group Token - scope: users:write Response: the created/updated object - see the matching GET field list in this section Request body: {name*: string / null, description*: string / null, permissions: list[string]}
DELETE	/api/v1/user-groups/{group_id}	Delete User Group Token - scope: users:write Response: HTTP 204 - no body

Permissions (1)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/permissions	Permissions Catalog Token - scope: users:read Response (live-sampled): {permissions: ['str']}

Licensing (10)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/licensing/status	License Status Public - no token required Response (LicenseStatus): {state: LicenseState, licensed: boolean, trial_available: boolean, edition: string, edition_label: string, license_type: string / null, features: list[string], restricted_features: list[string], expires_at: string (date-time) / null, grace_until: string (date-time) / null, updates_until: string (date-time) / null, days_remaining: integer / null, installation_id: string, last_verified_at: string (date-time) / null, device_limit*: integer, device_count*: integer, account_limit*: integer, account_count*: integer, free_tier_available*: boolean, message: string}
POST	/api/v1/licensing/trial	Start Trial Token - scope: licensing:write Response (LicenseStatus): {state: LicenseState, licensed: boolean, trial_available: boolean, edition: string, edition_label: string, license_type: string / null, features: list[string], restricted_features: list[string], expires_at: string (date-time) / null, grace_until: string (date-time) / null, updates_until: string (date-time) / null, days_remaining: integer / null, installation_id: string, last_verified_at: string (date-time) / null, device_limit*: integer, device_count*: integer, account_limit*: integer, account_count*: integer, free_tier_available*: boolean, message: string} Request body: ContactInfo / null
POST	/api/v1/licensing/activate	Activate License Token - scope: licensing:write Response (LicenseStatus): {state: LicenseState, licensed: boolean, trial_available: boolean, edition: string, edition_label: string, license_type: string / null, features: list[string], restricted_features: list[string], expires_at: string (date-time) / null, grace_until: string (date-time) / null, updates_until: string (date-time) / null, days_remaining: integer / null, installation_id: string, last_verified_at: string (date-time) / null, device_limit*: integer, device_count*: integer, account_limit*: integer, account_count*: integer, free_tier_available*: boolean, message: string} Request body: {license_key: string}

	Endpoint	Purpose · auth · request/response fields
POST	/api/v1/licensing/free-home-lite	Request Free Home Lite Token - scope: licensing:write Response (LicenseStatus): {state: LicenseState, licensed: boolean, trial_available: boolean, edition: string, edition_label: string, license_type: string / null, features: list[string], restricted_features: list[string], expires_at: string (date-time) / null, grace_until: string (date-time) / null, updates_until: string (date-time) / null, days_remaining: integer / null, installation_id: string, last_verified_at: string (date-time) / null, device_limit*: integer, device_count*: integer, account_limit*: integer, account_count*: integer, free_tier_available*: boolean, message: string} Request body: ContactInfo / null
POST	/api/v1/licensing/activate-offline	Activate License Offline Token - scope: licensing:write Response (LicenseStatus): {state: LicenseState, licensed: boolean, trial_available: boolean, edition: string, edition_label: string, license_type: string / null, features: list[string], restricted_features: list[string], expires_at: string (date-time) / null, grace_until: string (date-time) / null, updates_until: string (date-time) / null, days_remaining: integer / null, installation_id: string, last_verified_at: string (date-time) / null, device_limit*: integer, device_count*: integer, account_limit*: integer, account_count*: integer, free_tier_available*: boolean, message: string} Request body: {envelope: SignedLicenseEnvelope}
POST	/api/v1/licensing/deactivate	Deactivate License Token - scope: licensing:write Response: HTTP 204 - no body
GET	/api/v1/licensing/refresh	License Refresh Status Token - scope: licensing:read Response: the created/updated object - see the matching GET field list in this section
POST	/api/v1/licensing/refresh	Refresh License Token - scope: licensing:write Response (LicenseStatus): {state: LicenseState, licensed: boolean, trial_available: boolean, edition: string, edition_label: string, license_type: string / null, features: list[string], restricted_features: list[string], expires_at: string (date-time) / null, grace_until: string (date-time) / null, updates_until: string (date-time) / null, days_remaining: integer / null, installation_id: string, last_verified_at: string (date-time) / null, device_limit*: integer, device_count*: integer, account_limit*: integer, account_count*: integer, free_tier_available*: boolean, message: string}
GET	/api/v1/licensing/history	License History Token - scope: licensing:read Response: [str]
GET	/api/v1/licensing/features	Licensing Features Public - no token required Response (FeatureCatalog): {features: list[FeatureDefinition], editions: list[EditionDefinition]}

Firewall (19)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/firewall/zones	List Zones Token - scope: policies:read Response: [empty]
POST	/api/v1/firewall/zones	Create Zone Token - scope: policies:write Response: the created/updated object - see the matching GET field list in this section Request body: {name: string, interfaces*: list[string], networks*: list[string], default_action*: string}

	Endpoint	Purpose · auth · request/response fields
DELETE	/api/v1/firewall/zones/{item_id}	Delete Zone Token - scope: policies:write Response: HTTP 204 - no body
GET	/api/v1/firewall/port-forwards	List Port Forwards Token - scope: policies:read Response: ['<empty>']
POST	/api/v1/firewall/port-forwards	Create Port Forward Token - scope: policies:write Response: the created/updated object - see the matching GET field list in this section Request body: {name: string, protocol: string, external_port: integer, internal_ip: string, internal_port: integer, source_network*: string / null, enabled*: boolean}
DELETE	/api/v1/firewall/port-forwards/{item_id}	Delete Port Forward Token - scope: policies:write Response: HTTP 204 - no body
GET	/api/v1/firewall/engine	Firewall Engine Token - scope: deployment:read Response (live-sampled): {backend: str, legacy_binary: str, settings: {wan_interface: str, lan_interface: str, lan_network: str, management_network: str, masquerade: bool, ip_forward: bool, default_input: str, default_forward: str}, configured: bool, live_apply_enabled: bool}
PUT	/api/v1/firewall/engine/settings	Update Firewall Engine Token - scope: deployment:write Response: the created/updated object - see the matching GET field list in this section Request body: {wan_interface*: string, lan_interface*: string, lan_network*: string, management_network*: string, masquerade*: boolean, ip_forward*: boolean, default_input*: string, default_forward*: string, rollback_seconds*: integer}
GET	/api/v1/firewall/modules	Firewall Modules Token - scope: deployment:read Response: ['str']
PUT	/api/v1/firewall/modules/{module_key}	Update Firewall Module Token - scope: deployment:write Response: object with fields module, enabled, state, note Request body: {enabled: boolean}
POST	/api/v1/firewall/modules/ids_ips/rules/update	Ids Ips Rules Update Token - scope: deployment:write Response: the created/updated object - see the matching GET field list in this section
GET	/api/v1/firewall/modules/ids_ips/status	Ids Ips Status Token - scope: deployment:read Response (live-sampled): {ok: bool, supported: bool, platform: NoneType, installed: bool, running: bool, enabled_in_kernel: bool, version: str, home_networks: ['str'], queue: int, installed_at: float, ruleset: str, rules_count: int, updated_at: str, rules_auto_update: bool, returncode: int}
POST	/api/v1/firewall/modules/ids_ips/install	Ids Ips Install Token - scope: deployment:write Response: the created/updated object - see the matching GET field list in this section
POST	/api/v1/firewall/modules/ids_ips/uninstall	Ids Ips Uninstall Token - scope: deployment:write Response: the created/updated object - see the matching GET field list in this section
POST	/api/v1/firewall/iptables/render	Render Iptables Candidate Token - scope: deployment:write Response: the created/updated object - see the matching GET field list in this section

	Endpoint	Purpose · auth · request/response fields
POST	/api/v1/firewall/iptables/candidates	Prepare Iptables Candidate Token - scope: deployment:write Response: object with fields transaction_id, sha256, expires_at, rollback_seconds, validation, next
POST	/api/v1/firewall/iptables/transactions/{transaction_id}/{action}	Manage Iptables Transaction Token - scope: deployment:write Response: the created/updated object - see the matching GET field list in this section
GET	/api/v1/firewall/iptables/transactions/{transaction_id}	Get Iptables Transaction Token - scope: deployment:read Response: the created/updated object - see the matching GET field list in this section
GET	/api/v1/firewall/iptables/live	Live Iptables Token - scope: deployment:read Response (live-sampled): {available: bool, rules: str, message: str, return_code: int}

Deployment (9)

	Endpoint	Purpose · auth · request/response fields
POST	/api/v1/deployment/validate	Validate Public - no token required Response (ValidationResult): {valid: boolean, messages: list[ValidationMessage]} Request body: {mode: DeploymentMode, wan_interface*: string / null, lan_interface*: string / null, trunk_interface*: string / null, management_interface: string, management_address: string, upstream_gateway*: string / null, block_page_ipv4*: string / null, block_page_hostname*: string / null, vlans*: list[VlanAssignment], enable_dhcp*: boolean, architecture_acknowledged*: boolean, cabling_acknowledged*: boolean, recovery_acknowledged*: boolean}
POST	/api/v1/deployment/plan	Plan Public - no token required Response (DeploymentPlan): {mode: DeploymentMode, title: string, full_enforcement: boolean, diagram: list[string], interface_assignments: object, generated_actions: list[string], safety_actions: list[string], limitations: list[string], validation: ValidationResult} Request body: {mode: DeploymentMode, wan_interface*: string / null, lan_interface*: string / null, trunk_interface*: string / null, management_interface: string, management_address: string, upstream_gateway*: string / null, block_page_ipv4*: string / null, block_page_hostname*: string / null, vlans*: list[VlanAssignment], enable_dhcp*: boolean, architecture_acknowledged*: boolean, cabling_acknowledged*: boolean, recovery_acknowledged*: boolean}
POST	/api/v1/deployment/render	Render Candidate Token - scope: deployment:write Response: object with fields transaction_id, sha256, nftables_valid, validation_message, rules_preview, network_plan, blocking_targets, security_controls, network_rules, service_configurations Request body: {mode: DeploymentMode, wan_interface*: string / null, lan_interface*: string / null, trunk_interface*: string / null, management_interface: string, management_address: string, upstream_gateway*: string / null, block_page_ipv4*: string / null, block_page_hostname*: string / null, vlans*: list[VlanAssignment], enable_dhcp*: boolean, architecture_acknowledged*: boolean, cabling_acknowledged*: boolean, recovery_acknowledged*: boolean}

	Endpoint	Purpose · auth · request/response fields
POST	/api/v1/deployment/apply	Apply Token - scope: deployment:write Response: object with fields status, message, transaction, plan Request body: {mode: DeploymentMode, wan_interface*: string / null, lan_interface*: string / null, trunk_interface*: string / null, management_interface: string, management_address: string, upstream_gateway*: string / null, block_page_ipv4*: string / null, block_page_hostname*: string / null, vlans*: list[VlanAssignment], enable_dhcp*: boolean, architecture_acknowledged*: boolean, cabling_acknowledged*: boolean, recovery_acknowledged*: boolean}
GET	/api/v1/deployment/transactions/{transaction_id}	Transaction Status Token - scope: deployment:read Response: the created/updated object - see the matching GET field list in this section
POST	/api/v1/deployment/transactions/{transaction_id}/activate	Activate Transaction Token - scope: deployment:write Response: object with fields transaction, health_checks, confirmation_required, rollback_seconds
POST	/api/v1/deployment/transactions/{transaction_id}/confirm	Confirm Transaction Token - scope: deployment:write Response: the created/updated object - see the matching GET field list in this section
POST	/api/v1/deployment/transactions/{transaction_id}/rollback	Rollback Transaction Token - scope: deployment:write Response: the created/updated object - see the matching GET field list in this section
GET	/api/v1/deployment/status	Deployment Status Token - scope: deployment:read Response (live-sampled): {deployed: bool, enforcement: {iptables: str, dns: str}}

Networking & Pools (12)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/network/pool-rules	List Pool Rules Token - scope: deployment:read Response: [{};]
POST	/api/v1/network/pool-rules	Create Pool Rule Token - scope: deployment:write Response: object with fields rule, applied Request body: {pool_a: integer, pool_b: integer, note*: string}
DELETE	/api/v1/network/pool-rules/{item_id}	Delete Pool Rule Token - scope: deployment:write Response: HTTP 204 - no body
GET	/api/v1/network/managed-interfaces	List Managed Interfaces Token - scope: deployment:read Response: [{};]
POST	/api/v1/network/managed-interfaces	Create Managed Interface Token - scope: deployment:write Response: object with fields interface, pool, applied, apply_detail Request body: {name: string, address: string, label*: string, vlan_id*: integer, pool_start*: string, pool_end*: string, enabled*: boolean}
PUT	/api/v1/network/managed-interfaces/{item_id}	Update Managed Interface Token - scope: deployment:write Response: object with fields interface, pool, applied, apply_detail Request body: {address: string, label*: string, vlan_id*: integer, pool_start*: string, pool_end*: string, enabled*: boolean}

	Endpoint	Purpose · auth · request/response fields
DELETE	/api/v1/network/managed-interfaces/{item_id}	Delete Managed Interface Token - scope: deployment:write Response: HTTP 204 - no body
GET	/api/v1/network/interface-states	List Interface States Token - scope: deployment:read Response: ['<empty>']
PUT	/api/v1/network/interfaces/{name}/state	Set Interface State Token - scope: deployment:write Response: object with fields name, enabled, applied Request body: {enabled: boolean}
GET	/api/v1/network/bonds	List Bonds Token - scope: deployment:read Response: ['<empty>']
POST	/api/v1/network/bonds	Create Bond Token - scope: deployment:write Response: object with fields bond, applied, apply_detail Request body: {name: string, mode: string, members: list[string], enabled*: boolean}
DELETE	/api/v1/network/bonds/{item_id}	Delete Bond Token - scope: deployment:write Response: HTTP 204 - no body

DHCP (12)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/dhcp/settings	Dhcp Settings Token - scope: deployment:read Response (DhcpSettings): {enabled*: boolean, interface*: string, subnet*: string, pool_start*: string, pool_end*: string, router*: string, dns_servers*: list[string], lease_seconds*: integer}
PUT	/api/v1/dhcp/settings	Save Dhcp Settings Token - scope: deployment:write Response (DhcpSettings): {enabled*: boolean, interface*: string, subnet*: string, pool_start*: string, pool_end*: string, router*: string, dns_servers*: list[string], lease_seconds*: integer} Request body: {enabled*: boolean, interface*: string, subnet*: string, pool_start*: string, pool_end*: string, router*: string, dns_servers*: list[string], lease_seconds*: integer}
GET	/api/v1/dhcp/leases	Dhcp Leases Token - scope: deployment:read Response (live-sampled): {active: int, leases: [<empty>]}
GET	/api/v1/dhcp/render	Dhcp Render Token - scope: deployment:read Response (live-sampled): {Dhcp4: {interfaces-config: dict, lease-database: dict, valid-lifetime: int, subnet4: list, loggers: list, control-socket: dict, authoritative: bool, echo-client-id: bool}}
GET	/api/v1/dhcp/pools	List Ip Pools Token - scope: deployment:read Response: ['str']
POST	/api/v1/dhcp/pools	Create Ip Pool Token - scope: deployment:write Response: object with fields pool, applied, apply_detail Request body: {name: string, vlan_id*: integer, parent_interface*: string, subnet: string, gateway*: string, pool_start: string, pool_end: string, enabled*: boolean, isolated*: boolean}

	Endpoint	Purpose · auth · request/response fields
PUT	/api/v1/dhcp/pools/{item_id}	Update Ip Pool Token - scope: deployment:write Response: object with fields pool, applied, apply_detail Request body: {name: string, vlan_id*: integer, parent_interface*: string, subnet: string, gateway*: string, pool_start: string, pool_end: string, enabled*: boolean, isolated*: boolean}
DELETE	/api/v1/dhcp/pools/{item_id}	Delete Ip Pool Token - scope: deployment:write Response: HTTP 204 - no body
POST	/api/v1/dhcp/devices/assign	Assign Device To Pool Token - scope: deployment:write Response: object with fields reservation, pool, applied, apply_detail Request body: {mac_address: string, pool_id: integer, ip_address*: string / null, hostname*: string}
GET	/api/v1/dhcp/reservations	List Reservations Token - scope: deployment:read Response: ['<empty>']
POST	/api/v1/dhcp/reservations	Create Reservation Token - scope: deployment:write Response: the created/updated object - see the matching GET field list in this section Request body: {mac_address: string, ip_address: string, hostname: string}
DELETE	/api/v1/dhcp/reservations/{item_id}	Delete Reservation Token - scope: deployment:write Response: HTTP 204 - no body

DNS (9)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/dns/settings	Dns Settings Token - scope: dns:read Response (DnsSettings): {enabled*: boolean, listen_addresses*: list[string], upstream_servers*: list[string], dnssec_validation*: boolean, block_response*: string, block_page_ipv4*: string, block_page_ipv6*: string / null, block_hostname*: string / null, block_doh*: boolean, block_dot*: boolean}
PUT	/api/v1/dns/settings	Save Dns Settings Token - scope: dns:write Response (DnsSettings): {enabled*: boolean, listen_addresses*: list[string], upstream_servers*: list[string], dnssec_validation*: boolean, block_response*: string, block_page_ipv4*: string, block_page_ipv6*: string / null, block_hostname*: string / null, block_doh*: boolean, block_dot*: boolean} Request body: {enabled*: boolean, listen_addresses*: list[string], upstream_servers*: list[string], dnssec_validation*: boolean, block_response*: string, block_page_ipv4*: string, block_page_ipv6*: string / null, block_hostname*: string / null, block_doh*: boolean, block_dot*: boolean}
GET	/api/v1/dns/render	Dns Render Token - scope: dns:read Response: non-JSON response (html/binary)
POST	/api/v1/dns/candidates	Prepare Dns Candidate Token - scope: dns:write Response: object with fields transaction_id, sha256, domain_count, validation, next

	Endpoint	Purpose · auth · request/response fields
POST	/api/v1/dns/transactions/{transaction_id}/{action}	Manage Dns Transaction Token - scope: dns:write Response: the created/updated object - see the matching GET field list in this section
GET	/api/v1/dns/status	Dns Status Token - scope: dns:read Response (live-sampled): {dns_service: str, installed: bool, blocked_domains: int, active: bool, transaction: {activated_at: float, domain_count: int, sha256: str, state: str, transaction_id: str}}
GET	/api/v1/dns/records	List Dns Records Token - scope: dns:read Response: ['str']
POST	/api/v1/dns/records	Create Dns Record Token - scope: dns:write Response: the created/updated object - see the matching GET field list in this section Request body: {name: string, record_type: string, value: string}
DELETE	/api/v1/dns/records/{item_id}	Delete Dns Record Token - scope: dns:write Response: HTTP 204 - no body

Domain Rules (3)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/domains	List Local Domains Token - scope: dns:read Response: ['<empty>']
POST	/api/v1/domains	Create Local Domain Token - scope: dns:write Response: the created/updated object - see the matching GET field list in this section Request body: {domain: string, action*: string, category*: string, expires_at*: string / null}
DELETE	/api/v1/domains/{domain_id}	Delete Local Domain Token - scope: dns:write Response: HTTP 204 - no body

Block Page (11)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/block-page/client-info	Block Page Client Info Public - no token required Response (live-sampled): {client_ip: str, hostname: str, policy: str, requests_enabled: bool}
GET	/api/v1/block-page/logo	Block Page Logo Public - no token required Response: non-JSON response (html/binary)
POST	/api/v1/block-page/logo	Upload Block Page Logo Token - scope: policies:write Response: object with fields uploaded, size, url
DELETE	/api/v1/block-page/logo	Delete Block Page Logo Token - scope: policies:write Response: HTTP 204 - no body
GET	/api/v1/block-page/portal-managed	Block Page Portal Managed Token - scope: policies:read Response (live-sampled): {portal_managed: bool}

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/block-page/settings	Block Page Settings Token - scope: policies:read Response (BlockPageSettings): {title*: string, organization*: string, message*: string, support_contact*: string, accent_color*: string, background_color*: string, footer_text*: string, policy_url*: string, show_logo*: boolean, show_client_ip*: boolean, show_hostname*: boolean, show_policy*: boolean, allow_access_requests*: boolean, https_block*: boolean, custom_html*: string, custom_html_active*: boolean, redirect_url*: string}
PUT	/api/v1/block-page/settings	Save Block Page Token - scope: policies:write Response (BlockPageSettings): {title*: string, organization*: string, message*: string, support_contact*: string, accent_color*: string, background_color*: string, footer_text*: string, policy_url*: string, show_logo*: boolean, show_client_ip*: boolean, show_hostname*: boolean, show_policy*: boolean, allow_access_requests*: boolean, https_block*: boolean, custom_html*: string, custom_html_active*: boolean, redirect_url*: string} Request body: {title*: string, organization*: string, message*: string, support_contact*: string, accent_color*: string, background_color*: string, footer_text*: string, policy_url*: string, show_logo*: boolean, show_client_ip*: boolean, show_hostname*: boolean, show_policy*: boolean, allow_access_requests*: boolean, https_block*: boolean, custom_html*: string, custom_html_active*: boolean, redirect_url*: string}
GET	/api/v1/block-page/ca	Block Page Ca Public - no token required Response: non-JSON response (html/binary)
POST	/api/v1/block-page/custom	Upload Block Page Custom Token - scope: policies:write Response: object with fields uploaded, size, active, message
DELETE	/api/v1/block-page/custom	Delete Block Page Custom Token - scope: policies:write Response: HTTP 204 - no body
GET	/api/v1/block-page/preview	Block Page Preview Public - no token required Response: non-JSON response (html/binary)

Blocklists (7)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/blocklists	List Blocklists Token - scope: dns:read Response: ['str']
POST	/api/v1/blocklists	Create Blocklist Token - scope: dns:write Response: the created/updated object - see the matching GET field list in this section Request body: {name: string, url: string, format*: string, enabled*: boolean, update_hours*: integer, category*: string}
POST	/api/v1/blocklists/stevenblack	Create Stevenblack Token - scope: dns:write Response: the created/updated object - see the matching GET field list in this section
POST	/api/v1/blocklists/category-feeds	Register Category Feeds Token - scope: dns:write Response: object with fields registered, skipped

	Endpoint	Purpose · auth · request/response fields
POST	/api/v1/blocklists/{source_id}/update	Refresh Blocklist Token - scope: dns:write Response: the created/updated object - see the matching GET field list in this section
DELETE	/api/v1/blocklists/{source_id}	Delete Blocklist Token - scope: dns:write Response: HTTP 204 - no body
POST	/api/v1/blocklists/update-due	Refresh Due Blocklists Token - scope: dns:write Response: the created/updated object - see the matching GET field list in this section

Web Categories (6)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/categories	Categories Token - scope: policies:read Response: ['str']
POST	/api/v1/categories	Create Category Group Token - scope: policies:write Response: the created/updated object - see the matching GET field list in this section Request body: {name: string, description*: string}
DELETE	/api/v1/categories/{category}	Delete Category Group Token - scope: policies:write Response: HTTP 204 - no body
POST	/api/v1/categories/{category}/members	Add Group Member Token - scope: policies:write Response: object with fields group, application Request body: {application: string}
DELETE	/api/v1/categories/{category}/members/{application}	Remove Group Member Token - scope: policies:write Response: HTTP 204 - no body
PUT	/api/v1/categories/{category}	Save Category Token - scope: policies:write Response: object with fields name, applications, feed_source_id, enabled, block_quic, name, applications, feed_source_id, enabled, block_quic, custom Request body: {category: string, enabled*: boolean, block_quic*: boolean}

Content Filter (1)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/content-filter/compiled	Content Filter Compiled Token - scope: dns:read Response (live-sampled): {blocked: ['str'], allowed: [<empty>], blocked_count: int}

Application Policies (7)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/applications	Applications Token - scope: policies:read Response: ['str']

	Endpoint	Purpose · auth · request/response fields
POST	/api/v1/applications/{application}/domains	Add Application Domains Token - scope: policies:write Response: the created/updated object - see the matching GET field list in this section Request body: {domains: list[string]}
DELETE	/api/v1/applications/{application}/domains/{domain}	Remove Application Domain Token - scope: policies:write Response: object with fields name, domains
POST	/api/v1/applications	Create Custom Application Token - scope: policies:write Response: the created/updated object - see the matching GET field list in this section Request body: {name: string, category: string, domains: list[string]}
DELETE	/api/v1/applications/{application}	Delete Custom Application Token - scope: policies:write Response: HTTP 204 - no body
PUT	/api/v1/applications/{application}	Save Application Token - scope: policies:write Response: the created/updated object - see the matching GET field list in this section Request body: {application: string, enabled*: boolean, block_quic*: boolean}
POST	/api/v1/applications/discover	Discover Service Domains Token - scope: policies:write Response: object with fields domain, found Request body: {domain: string}

Proxy (3)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/proxy/settings	Proxy Settings Token - scope: proxy:read Response (ProxySettings): {enabled*: boolean, listen_port*: integer, cache_enabled*: boolean, cache_size_mb*: integer, memory_cache_mb*: integer, max_object_mb*: integer, tls_inspection*: boolean}
PUT	/api/v1/proxy/settings	Save Proxy Settings Token - scope: proxy:write Response (ProxySettings): {enabled*: boolean, listen_port*: integer, cache_enabled*: boolean, cache_size_mb*: integer, memory_cache_mb*: integer, max_object_mb*: integer, tls_inspection*: boolean} Request body: {enabled*: boolean, listen_port*: integer, cache_enabled*: boolean, cache_size_mb*: integer, memory_cache_mb*: integer, max_object_mb*: integer, tls_inspection*: boolean}
GET	/api/v1/proxy/render	Proxy Render Token - scope: proxy:read Response: non-JSON response (html/binary)

Proxy Cache (2)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/cache/status	Get Cache Status Token - scope: proxy:read Response (live-sampled): {cache_directory: str, directory_present: bool, disk_total_bytes: int, disk_used_bytes: int, disk_free_bytes: int, squidclient_available: bool}

	Endpoint	Purpose · auth · request/response fields
POST	/api/v1/cache/purge	Purge Cache Token - scope: proxy:write Response: the created/updated object - see the matching GET field list in this section Request body: {url: string}

Blocking Policies (3)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/policies	List Policies Token - scope: policies:read Response: ['<empty>']
POST	/api/v1/policies	Create Policy Token - scope: policies:write Response: the created/updated object - see the matching GET field list in this section Request body: {name: string, target_type: string, target_value: string, enabled*: boolean, reason*: string, schedule_id*: integer / null}
DELETE	/api/v1/policies/{policy_id}	Delete Policy Token - scope: policies:write Response: HTTP 204 - no body

Policy Compiler (1)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/policy-compiler	Compiled Policies Token - scope: policies:read Response (live-sampled): {ipv4: ['<empty>'], ipv6: ['<empty>'], mac: ['<empty>'], unresolved_hostnames: ['<empty>']}

Schedules (3)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/schedules	Schedules Token - scope: policies:read Response: ['<empty>']
POST	/api/v1/schedules	Create Schedule Token - scope: policies:write Response: the created/updated object - see the matching GET field list in this section Request body: {name: string, timezone*: string, days*: list[integer], start_time*: string, end_time*: string, enabled*: boolean}
DELETE	/api/v1/schedules/{schedule_id}	Delete Schedule Token - scope: policies:write Response: HTTP 204 - no body

GeoIP (4)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/geoip/settings	Get Geoip Settings Token - scope: deployment:read Response (live-sampled): {enabled: bool, mode: str, countries: ['str'], sets_built: bool, feed: {updated_at: str, mode: str, countries: list, cidr_count: int, ok: bool, message: str}, ready: bool}

	Endpoint	Purpose · auth · request/response fields
PUT	/api/v1/geoip/settings	Put Geoip Settings Token - scope: deployment:write Response: the created/updated object - see the matching GET field list in this section Request body: {enabled*: boolean, mode*: string, countries*: list[string]}
POST	/api/v1/geoip/update	Geoip Update Now Token - scope: deployment:write Response: the created/updated object - see the matching GET field list in this section
GET	/api/v1/geoip/status	Get Geoip Status Token - scope: deployment:read Response (live-sampled): {enabled: bool, mode: str, countries: ['str'], sets_built: bool, feed: {updated_at: str, mode: str, countries: list, cidr_count: int, ok: bool, message: str}, ready: bool}

Threat Feeds (3)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/threat/settings	Threat Settings Token - scope: deployment:read Response (ThreatSettings): {feodo*: boolean, spamhaus*: boolean, dshield*: boolean, tor*: boolean}
PUT	/api/v1/threat/settings	Save Threat Settings Token - scope: deployment:write Response (ThreatSettings): {feodo*: boolean, spamhaus*: boolean, dshield*: boolean, tor*: boolean} Request body: {feodo*: boolean, spamhaus*: boolean, dshield*: boolean, tor*: boolean}
GET	/api/v1/threat/status	Threat Status Token - scope: deployment:read Response (live-sampled): {updated_at: str, feeds: {dshield: int, feodo: int, spamhaus: int}, entries: int, ok: bool, message: str}

Portal (5)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/portal/settings	Portal Settings Token - scope: policies:read Response (PortalSettings): {enabled*: boolean, title*: string, organization*: string, message*: string, support_contact*: string}
PUT	/api/v1/portal/settings	Save Portal Settings Token - scope: policies:write Response (PortalSettings): {enabled*: boolean, title*: string, organization*: string, message*: string, support_contact*: string} Request body: {enabled*: boolean, title*: string, organization*: string, message*: string, support_contact*: string}
GET	/api/v1/portal/accepted	Portal Accepted Token - scope: deployment:read Response: ['str']
DELETE	/api/v1/portal/accepted/{item_id}	Revoke Portal Acceptance Token - scope: policies:write Response: HTTP 204 - no body
POST	/api/v1/portal/accept	Portal Accept Public - no token required Response: object with fields ok, message, mac_address, ok, message, ok, message

Access Requests (6)

	Endpoint	Purpose · auth · request/response fields
POST	/api/v1/access-requests	Create Access Request Public - no token required Response: the created/updated object - see the matching GET field list in this section Request body: {requested_host: string, client_ip*: string, device_name*: string / null, reason: string}
GET	/api/v1/access-requests	Access Requests Token - scope: policies:read Response: ['<empty>']
GET	/api/v1/access-requests/status	Access Request Status Public - no token required Response: object with fields status, reviewed_by, review_note, status, status
DELETE	/api/v1/access-requests/reviewed	Clear Reviewed Access Requests Token - scope: policies:write Response: object with fields removed, pending
POST	/api/v1/access-requests/{request_id}/review	Review Access Request Token - scope: policies:write Response: the created/updated object - see the matching GET field list in this section Request body: {decision: string, note*: string, create_allow_rule*: boolean, duration*: string}
POST	/api/v1/access-requests/{request_id}/reblock	Reblock Access Request Token - scope: policies:write Response: object with fields ok, removed_allow_rules, removed_device_rules, request_id

WireGuard VPN (9)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/wireguard/settings	Wireguard Settings Token - scope: deployment:read Response (WireguardSettings): {enabled*: boolean, listen_port*: integer, tunnel_subnet*: string, endpoint_host*: string, allow_client_internet*: boolean}
PUT	/api/v1/wireguard/settings	Save Wireguard Settings Token - scope: deployment:write Response (WireguardSettings): {enabled*: boolean, listen_port*: integer, tunnel_subnet*: string, endpoint_host*: string, allow_client_internet*: boolean} Request body: {enabled*: boolean, listen_port*: integer, tunnel_subnet*: string, endpoint_host*: string, allow_client_internet*: boolean}
GET	/api/v1/wireguard/public-key	Wireguard Public Key Token - scope: deployment:read Response (live-sampled): {public_key: str, listen_port: int}
GET	/api/v1/wireguard/peers	Wireguard Peers Token - scope: deployment:read Response: ['str']
POST	/api/v1/wireguard/peers	Create Wireguard Peer Token - scope: deployment:write Response: the created/updated object - see the matching GET field list in this section Request body: {kind*: string, name: string, public_key*: string, preshared_key*: string, allowed_ips*: string, endpoint_host*: string, endpoint_port*: integer, keepalive*: integer, enabled*: boolean}

	Endpoint	Purpose · auth · request/response fields
PUT	/api/v1/wireguard/peers/{item_id}	Update Wireguard Peer Token - scope: deployment:write Response: object / null Request body: {kind*: string, name: string, public_key*: string, preshared_key*: string, allowed_ips*: string, endpoint_host*: string, endpoint_port*: integer, keepalive*: integer, enabled*: boolean}
DELETE	/api/v1/wireguard/peers/{item_id}	Delete Wireguard Peer Token - scope: deployment:write Response: HTTP 204 - no body
GET	/api/v1/wireguard/peers/{item_id}/config	Wireguard Peer Config Token - scope: deployment:read Response: the created/updated object - see the matching GET field list in this section
GET	/api/v1/wireguard/status	Wireguard Status Token - scope: deployment:read Response (live-sampled): {running: bool, peers: {qEHPEni9I6+HVGZUTTPF9PRvmUJuEFBK1OmCTj0afxE=: dict, GioCxm1vbj6WDmsTNc+nvvZGG3wpWmr/xCYIE4cJWY=: dict, ZJWBbh3ivktXgaqDUel5rF8athUfam5g5/ODacV2OCc=: dict}, public_key: str}

Reports (2)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/reports/summary	Report Summary Token - scope: audit:read Response (live-sampled): {generated_at: str, event_count: int, blocked_count: int, top_categories: [str], top_destinations: [str]}
GET	/api/v1/reports/events.csv	Report Csv Token - scope: audit:read Response: non-JSON response (html/binary)

Audit (1)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/audit	Audit Log Token - scope: audit:read Response: [str]

Retention (3)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/retention/settings	Retention Settings Token - scope: audit:read Response (RetentionSettings): {security_event_days*: integer, audit_days*: integer, access_request_days*: integer}
PUT	/api/v1/retention/settings	Save Retention Token - scope: audit:write Response (RetentionSettings): {security_event_days*: integer, audit_days*: integer, access_request_days*: integer} Request body: {security_event_days*: integer, audit_days*: integer, access_request_days*: integer}

	Endpoint	Purpose · auth · request/response fields
POST	/api/v1/retention/purge	Purge Retention Token - scope: audit:write Response: the created/updated object - see the matching GET field list in this section

Backup (2)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/backup/export	Export Backup Token - scope: deployment:read Response: non-JSON response (html/binary)
POST	/api/v1/backup/validate	Backup Validate Token - scope: deployment:write Response: the created/updated object - see the matching GET field list in this section

Restore (2)

	Endpoint	Purpose · auth · request/response fields
POST	/api/v1/restore/stage	Restore Stage Token - scope: deployment:write Response: the created/updated object - see the matching GET field list in this section
POST	/api/v1/restore/apply	Restore Apply Token - scope: deployment:write Response: the created/updated object - see the matching GET field list in this section Request body: {candidate_id: string, confirmation: string}

Operations (1)

	Endpoint	Purpose · auth · request/response fields
POST	/api/v1/operations/backup	Operations Backup Now Token - scope: deployment:write Response: object with fields saved, size, download, pending_updates

Updates (1)

	Endpoint	Purpose · auth · request/response fields
POST	/api/v1/updates/stage	Update Stage Token - scope: deployment:write Response: the created/updated object - see the matching GET field list in this section Request body: {manifest: object, signature: string, package_base64: string}

Support Access (1)

	Endpoint	Purpose · auth · request/response fields
GET	/api/v1/support/bundle	Download Support Bundle Token - scope: audit:read Response: non-JSON response (html/binary)

Other (5)

	Endpoint	Purpose · auth · request/response fields
GET	/	Public - no token required Response: non-JSON response (html/binary)

	Endpoint	Purpose · auth · request/response fields
GET	/enroll	Public - no token required Response: non-JSON response (html/binary)
GET	/enroll/windows	Public - no token required Response: non-JSON response (html/binary)
GET	/enroll/apple	Public - no token required Response: non-JSON response (html/binary)
WS	/ws/events	Token - scope: audit:read Response: the created/updated object - see the matching GET field list in this section

4. The websocket event stream

- Connect to **ws://APPLIANCE/ws/events?token=TOKEN** (or wss:// with the vendor certificate).
- Browsers and TV apps cannot set Authorization headers on a websocket, so the token rides the query string.
- The token may be a session token or an API token and must carry the **audit:read** scope.
- Every recorded security event is pushed as one JSON message: **{type: "security_event", event: {...}}**.
- Without a valid token the socket answers {"type": "error", "message": "authentication required (audit:read)"} and closes.
- Reconnect with backoff; the stream replays nothing - missed events remain available via GET /api/v1/events.

5. API scopes

deployment:read/write	Network, DHCP, recovery, backup and update transactions
policies:read/write	Groups, zones, NAT, schedules, applications and access requests
dns:read/write	DNS records, domain rules, blocklists and scheduled refresh
proxy:read/write	Proxy/cache configuration and purge
events:write	Ingest normalized enforcement events
audit:read/write	Audit, reports, retention, the events stream and support resources
licensing:read/write	Status, trial and activation lifecycle
users:read/write	Group-managed accounts, user groups, API tokens and permissions
tokens:write	Create and revoke API tokens
*	All scopes - administrator sessions inherently have full access

Generated from appliance 3.13.40: authentication extracted from route source code, response fields from live sampling of a running appliance and expanded schema models. /openapi.json remains the authoritative live contract.