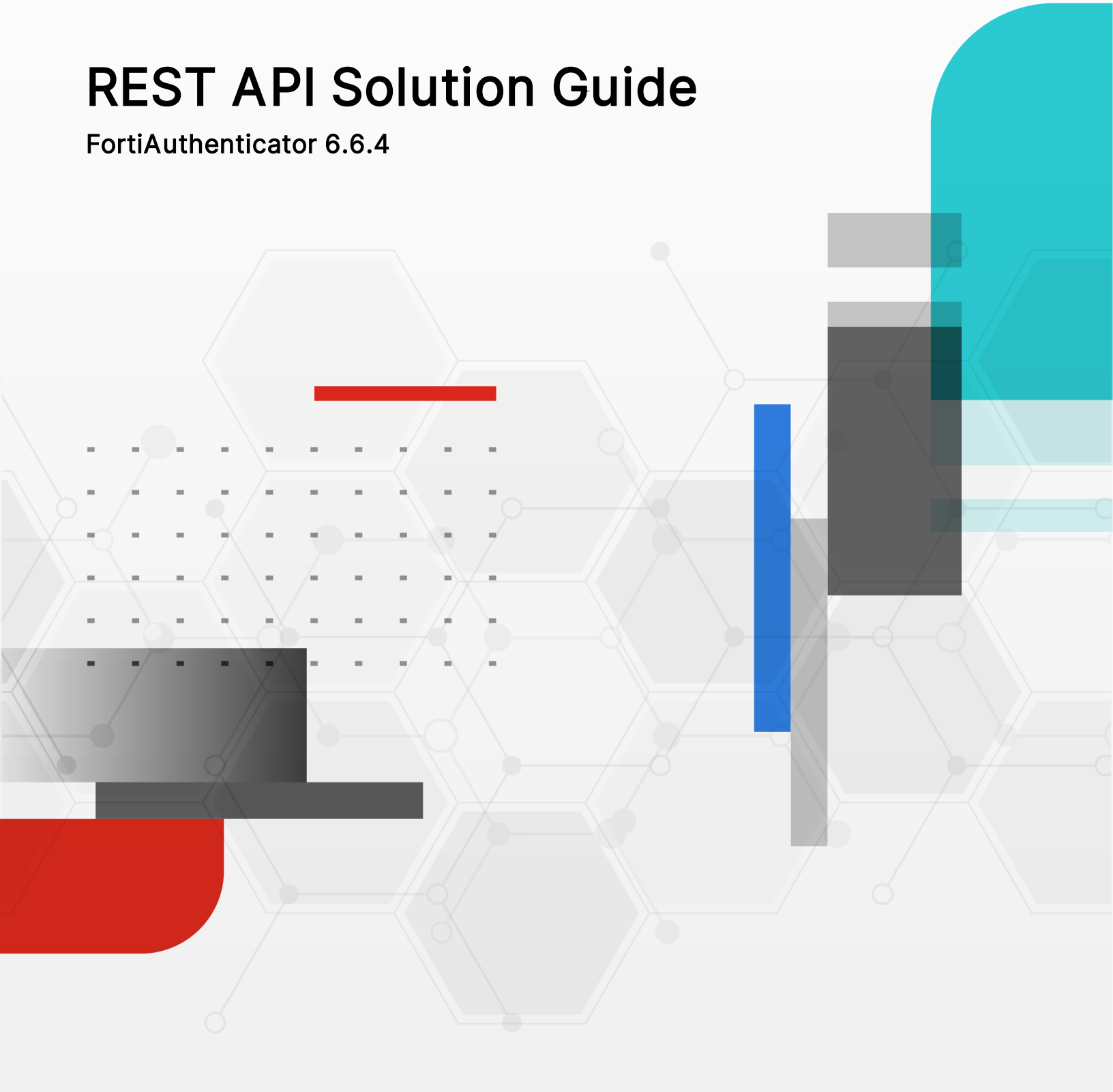
# REST API Solution Guide

FortiAuthenticator 6.6.4



**FORTINET DOCUMENT LIBRARY**
https://docs.fortinet.com

**FORTINET VIDEO LIBRARY**
https://video.fortinet.com

**FORTINET BLOG**
https://blog.fortinet.com

**CUSTOMER SERVICE & SUPPORT**
https://support.fortinet.com

**FORTINET TRAINING & CERTIFICATION PROGRAM**
https://www.fortinet.com/training-certification

**FORTINET TRAINING INSTITUTE**
https://training.fortinet.com

**FORTIGUARD LABS**
https://www.fortiguard.com

**END USER LICENSE AGREEMENT**
https://www.fortinet.com/doc/legal/EULA.pdf

**FEEDBACK**
Email: techdoc@fortinet.com

July 18, 2025
FortiAuthenticator 6.6.4 REST API Solution Guide
23-664-935247-20250718

# TABLE OF CONTENTS

# Change Log

| Date | Change Description |
|---|---|
| 2025-05-26 | Initial release. |
| 2025-06-02 | Updated User groups (/usergroups/) on page 23. |
| 2025-07-18 | Updated LDAP users (/ldapusers/) on page 49. |
| | |

# Introduction

This document introduces the FortiAuthenticator REST API and details how it can be configured and utilized.

# Software versions

The API described within this document is supported by FortiAuthenticator 6.6.4 and upwards.

# What's new in FortiAuthenticator

This section provides a summary of new endpoints and enhancements in the FortiAuthenticator REST API:



## FortiAuthenticator 6.6.4

No new endpoints have been introduced in the release of FortiAuthenticator 6.6.4.

## FortiAuthenticator 6.6.3

The following new endpoints have been released for FortiAuthenticator 6.6.3:

- Relying Party Logout (/oauth/logout/) on page 123: A new endpoint to log out end users in OAuth.

The following enhancements have been released for FortiAuthenticator 6.6.3:

- Authentication (/auth/) on page 70: A new `user_ip` field.
- Realm authentication (/realmauth/) on page 73: A new `user_ip` field.
- OAuth server token (/oauth/token/) on page 109: A new `user_ip` field.
- OIDC Authorization (/oauth/authorize/) on page 107: New `approval_prompt` and `prompt` fields.
- Local users (/localusers/) on page 36: A new `is_locked` field.
- LDAP users (/ldapusers/) on page 49: A new `is_locked` field.
- RADIUS users (/radiususers/) on page 53: A new `is_locked` field.

- FortiTokens (/fortitokens/) on page 31: A new `sub_type` field.

# FortiAuthenticator 6.6.2

The following enhancements have been released for FortiAuthenticator 6.6.2:

- RADIUS clients (/radiusclients/) on page 75: A new `require_message_authenticator` field.

# FortiAuthenticator 6.6.1

The following new endpoints have been released for FortiAuthenticator 6.6.1:

- Local users CSV file import and export (/csv/localusers/) on page 47: Import and export local users using CSV files.

# FortiAuthenticator 6.6.0

The following enhancements have been released for FortiAuthenticator 6.6.0:

- OIDC Authorization (/oauth/authorize/) on page 107: The following new fields are required in requests to the `/oauth/authorize/` endpoint:
  - `code_challenge_method`
  - `code_challenge`
- OAuth server token (/oauth/token/) on page 109: The following new fields have been introduced in the `/oauth/token/` endpoint:
  - `iam_account`
  - `iam_user`
  - `code_verifier`
  - `code`
- Local users (/localusers/) on page 36: The following two new fields have been introduced in the `/localusers/` endpoint:
  - `company`
  - `department`
- LDAP users (/ldapusers/) on page 49: The following two new fields have been introduced in the `/ldapusers/` endpoint:
  - `company`
  - `department`
- RADIUS users (/radiususers/) on page 53: The following two new fields have been introduced in the `/radiususers/` endpoint:
  - `company`
  - `department`

# The FortiAuthenticator API

An API (Application Programming Interface) is a set of defined interfaces to accomplish a task, such as retrieving or modifying data. FortiAuthenticator provides a Representational State Transfer (REST) API for interaction with components of the system. Programs communicate with the REST API over HTTP, the same protocol that your web browser uses to interact with web pages.

## Introduction to REST

The REST API is based on interactions with a web page; data is treated like a static web page:

- Add data by POSTing a web page
- Fetch data by GETing a web page
- Update data by PUTing a web page
- Partial updates supported by PATCHing a web page
- Delete data by DELETEing a web page

After receiving the request, the FortiAuthenticator API sends back an HTTP response code. These error codes are summarized in General API response codes on page 138.

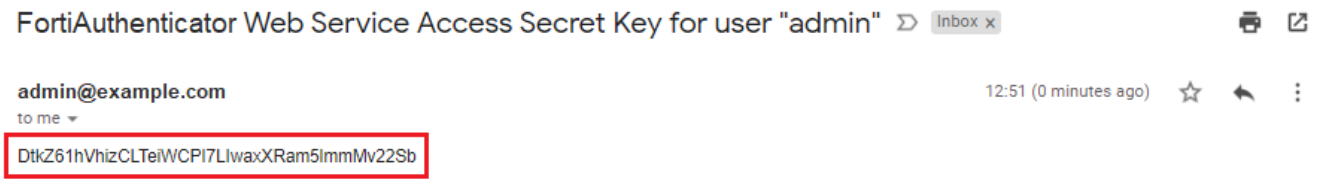
## Initializing the REST API

The FortiAuthenticator API is accessible without additional cost or licensing, however, the server is disabled by default and needs to be configured.

To access the API, a user must be granted administrator rights and web service access. A valid e-mail address is also required as the API challenge key will be emailed to the user.

To enable the API, create a new user or edit an existing one and specify the following:

1. Under **User Role**, select **Administrator**.
2. Enable **Web service access**.
3. Under **User Information**, enter a valid email address.
   Note: Ensure email routing is working beforehand as the API Key will be forwarded to this address.
4. Click **OK** to save the details.
   The API Web Service Access Key used to authenticate to the API is emailed to the user.
5. Make a note of the API Web Service Access Key.




Should the API Web Service Access Key be lost, access can be recovered by disabling the Web Service feature for the user, saving and then re-enabling the feature. A new key will be generated (and all code using it will need to be updated with the new credentials).

# Accessing the REST API

The FortiAuthenticator API can be accessed from most browsers (GET) however browser add-ons may be required for extended operations (e.g. PUT). More complicated, scripted queries can be made using utilities such as cURL and most scripting languages such as Perl or Python have built in libraries for interacting with RESTful APIs.

Example shown within this document will be demonstrated with the cross platform utility cURL.

All of the resource URLs are in this form:

https://[server_name]/api/[api_version]/[resource]/

where:

| | | |
|---|---|---|
| `server_name` | = | Name or IP of the FortiAuthenticator |
| `api_version` | = | API version to be used (currently v1) |
| `resource` | = | Resource or part of config to be viewed |
| `id` | = | Resource ID to view, edit, or delete |

# Filtering query results

Queries to the API can be to modify the query/response format or to filter the results. Below are some arguments that can be passed to the REST API URL. Please refer to the specific resource documentation to find out which of these filter operations are allowed.

| | | |
|---|---|---|
| `?format=[format_type]` | = | where format _type= xml or json (default) |

| | | |
|---|---|---|
| `?limit=[integer]` | = | where integer specifies number of records to return (default = 20) |
| `?offset=[integer]` | = | where integer specifies number of items in resource list to skip e.g. if there are 10 items, to return item #5 - #10 only, specify offset=4 |
| `?order_by=[field]` | = | order returned list by a known field name (e.g.?order_by=name) |

# Field filters

- exact: search for an exact match
  (e.g. to return items that has a name matching "John Doe", `name__exact=John Doe`)
- in: search for items that matches specific filter criteria
  (e.g. to return items that has a name matching "John" or "Bill", `?name__in=John&name__in=Bill`)

# View pages for large lists

By default, the API record query limit is set to 20, or can be set up to a maximum of 1000. This value is controlled by the `limit`, as shown in the table above. Note that this only determines how many records are returned and displayed per page.

REST API uses multiple pages when there are a large number of entries in the list. In order to get the following pages, use the `next` field from the response (see example below):

```
{"meta": {"limit": 1000, "next": null, "offset": 0, "previous": null, "total_count": 3}
```

When the response is the last page, `next` is set to `null`. Otherwise, set `next` to a URL that can be used in a subsequent REST API request to get the next page of records. For example:

```
{"meta": {"limit": 20, "next": "/api/v1/localusers/?offset=20&limit=20&format=json", "offset":
0, "previous": null, "total_count": 23}, "objects": [{ ...
```

# Supported API methods

All of the resource URLs are in this form: https://[server_name]/api/[api_version]/[resource]/. The current API version is v1.

To list all of the available resource endpoints, send a request to:

https://[server_name]/api/v1/?format=xml

To view schema, supported methods and available fields for each endpoint, append /schema/ to the endpoint URL. For example, to view schema for /auth/ API, perform a GET request to:

https://[server_name]/api/v1/auth/schema/?format=xml

In general, an endpoint may support the following methods, though not all methods are supported by all endpoints (see each endpoint's documentation for the list of allowed methods):

| Method | URL | Operation description | Success response code |
|---|---|---|---|
| `GET (list)` | `/[resource]/` | Retrieve a list of all resources for the endpoint | 200 OK |
| `GET (detail)` | `/[resource]/[id]/` | Retrieve a specific resource with ID id from the endpoint | 200 OK |
| `POST` | `/[resource]/` | Create a new resource on the given endpoint. The data being POST-ed must follow the same format as the data returned by the GET parameter | 201 CREATED |
| `PUT (list)` | `/[resource]/` | Update all of the resources for the given endpoint. Any existing items will be replaced with the new data. Data must follow the same format as the data returned by the GET parameter. | 204 NO CONTENT |
| `PUT (detail)` | `/[resource]/[id]/` | Update an existing item specified with ID id. Data must follow the same format as the data returned by the GET parameter. | 204 NO CONTENT |
| `PATCH (detail)` | `/[resource]/[id]/` | Update specific fields on an existing item with ID id | 202 ACCEPTED |
| `DELETE (list)` | `/[resource]/` | Delete all resources from an endpoint | 204 NO CONTENT |
| `DELETE (detail)` | `/[resource]/[id]/` | Delete an existing resource specified with ID id from an endpoint | 204 NO CONTENT |

# Supported data formats

Currently, JSON and XML are supported. To specify a format on the request:

For a GET request, there are two options:

- Use the GET format parameter (e.g. ?format=json or ?format=xml)
- Specify an Accept HTTP header with a correct mimetype (e.g. Accepts: application/json for JSON)

The GET format parameter takes precedence over the Accept header.

Browsers will usually default to requesting for an XML data type when format is not specified for a GET request.

# Resource Summary

Below are the main resources and the root record which can be accessed via the API:

| Resource | URL | Operation description | Supported methods |
|---|---|---|---|
| **Root** | `/` | Allows querying of available resources. | GET |
| **Local User Management** | `/localusers/` | Allows the creation, modification and deletion of user accounts. | GET, POST, PATCH |
| **Local Group Management** | `/usergroups/` | Allows the creation and deletion of user groups and specify users within that group. | GET, POST, PUT, DELETE |
| **LDAP Users** | `/ldapusers/` | Allows querying of LDAP user records and updating of specific fields. Allows triggering of out of band (email//SMS tokens to LDAP users. | GET, POST, PATCH, DELETE |
| **RADIUS users** | `/radiususers/` | Allows querying of RADIUS user records and update of specific fields. Allows triggering of out of band (email//SMS tokens to RADIUS users. | GET, POST, PATCH, DELETE |
| **Local Group Membership** | `/localgroup-memberships/` | Represents local user group membership resource (relationship between local user and local user group). | GET, POST, DELETE |
| **User Authentication** | `/auth/` | Allows validation of user authentication credentials. | POST |
| **FortiToken** | `/fortitokens/` | Allows provisioning of FortiTokens. | GET |
| **Push Authentication** | `/pushauth/` | Allows token code validation from a user's FortiToken Mobile app. | POST |
| **Push Authentication Response** | `/pushauthresp/` | Allows FortiToken Mobile devices to submit the response to a token code validation request triggered by a prior call to the `/pushauth/` endpoint. | POST |
| **SSO Group** | `/ssogroup/` | Enables remote configuration of the **Fortinet SSO Methods & Dynamic Policies > SSO > SSO Groups** table. | GET, POST, DELETE |

| Resource | URL | Operation description | Supported methods |
|---|---|---|---|
| **FortiGate Filter Group** | `/fgtgroupfilter/` | Enables remote configuration of the **Fortinet SSO Methods & Dynamic Policies > SSO > FortiGate Filtering** table. | GET, PUT |
| **SSO Authentication** | `/ssoauth/` | Adds/removes a user from the FSSO logged in users table. | POST |
| **Syslog Servers** | `/syslogservers/` | Allows creating, updating, editing, and deleting of syslog servers. | GET, POST, PATCH, DELETE |
| **Log Settings** | `/logsettings/` | Allows editing of log settings. | GET, POST, PATCH |
| **User Certificate Management** | `/usercerts/` | Allows renewing and revoking of user certificates. | GET, POST, PATCH |

# Authorization and Permissions

In most cases, once a user is authenticated by a method such as OAuth or Basic Authentication, the api will check if the user is authorized to use that endpoint based on the permissions they have been assigned by higher level administrators.

Permissions are contained within built-in admin profiles which are configured in **System > Administration > Admin Profiles**. Generally, for example, if an admin has the 'Can view local users' permission, they will be able to successfully perform a GET request to the '/localusers' endpoint. Similarly, if they do NOT have 'Can change local users' permission, any of their POST requests to the '/localusers' endpoint should fail. These profiles can be assigned to an admin by selecting an admin under **Authentication > User Management > Local / Remote Users**, and adding an admin profile, which contains the correct permission, to their list of applicable admin profiles.

If you want to give an admin only the permissions required to use an endpoint, without giving them the many permissions that go along with a built-in permission set, you can make a custom permission set with only the permissions required. This can be done by navigating to **System > Administration > Admin Profiles**, creating a custom permission set with permissions of your choice, and then applying that admin profile to your admin user.

For a summary of the authentication methods, permission sets, and permissions that each endpoint requires, see the Authorization and Permissions Table below.

| Resource Name | Base URL | Authentication Method | Applicable Built-in Permission Set | Required Permission code |
|---|---|---|---|---|
| auth | https://[server_name]/api/v1/auth/ | Webservice Basic | Webservice Authenticatio | Can use API to |

| Resource Name | Base URL | Authentication Method | Applicable Built-in Permission Set | Required Permission code |
|---|---|---|---|---|
| | | Authentication | n | authenticate |
| fabric | https://[server_name]/api/v1/fabric/ | OAuth Bearer Token Authentication | Widgets | Can read and access Fabric widgets |
| fabric (no version) | https://[server_name]/api/fabric | None | Webservice Authentication | Can authenticate FAC as fabric device |
| fgtgroupfilter | https://[server_name]/api/v1/fgtgroupfilter/ | Webservice Basic Authentication | SSO Settings | Can view / change FortiGate filter |
| fortiguardmessages | https://[server_name]/api/v1/fortiguardmessages/ | Webservice Basic Authentication | System Administration | Can view / change FortiGuard settings |
| fortitokenmobilelicenses | https://[server_name]/api/v1/fortitokenmobilelicenses/ | Webservice Basic Authentication | Users and Devices | Can view / change FortiToken |
| fortitokenmobileprovisioning | https://[server_name]/api/v1/fortitokenmobileprovisioning/ | Webservice Basic Authentication | System Administration | Can view / change FortiGuard settings |
| fortitokens | https://[server_name]/api/v1/fortitokens/ | Webservice Basic Authentication | Users and Devices | Can view / change FortiToken |
| ftpservers | https://[server_name]/api/v1/ftpservers/ | Webservice Basic Authentication | Maintenance | Can view / change FTP server |
| ldapusers | https://[server_name]/api/v1/ldapusers/ | Webservice Basic Authentication | Users and Devices | Can view / change remote LDAP user |
| licensing | https://[server_name]/api/v1/licensing/ | Webservice Basic Authentication | System Administration | Can import a new FAC license |
| localapiadmin | https://[server_name]/api/v1/localapiadmin/ | Webservice Basic | Administrators | Can view / change group |

| Resource Name | Base URL | Authentication Method | Applicable Built-in Permission Set | Required Permission code |
|---|---|---|---|---|
| | | Authentication | | |
| localgroup-memberships | https://[server_name]/api/v1/localgroup-memberships/ | Webservice Basic Authentication | Users and Devices | Can view / change user group |
| localusers | https://[server_name]/api/v1/localusers/ | Webservice Basic Authentication | Users and Devices | Can view / change local user |
| logsettings | https://[server_name]/api/v1/logsettings/ | Webservice Basic Authentication | Logs | Can view / change log settings |
| oauth | https://[server_name]/api/v1/oauth/ | None | None | None |
| passwordpolicies | https://[server_name]/api/v1/passwordpolicies/ | Webservice Basic Authentication | Account Policy | Can view / change Password policy |
| pushauth | https://[server_name]/api/v1/pushauth/ | Webservice Basic Authentication | None | None |
| pushauthresp | https://[server_name]/api/v1/pushauthresp/ | None | None | None |
| pushpoll | https://[server_name]/api/v1/pushpoll/ | None | None | None |
| radiususers | https://[server_name]/api/v1/radiususers/ | Webservice Basic Authentication | Users and Devices | Can view / change remote RADIUS user |
| realmauth | https://[server_name]/api/v1/realmauth/ | Webservice Basic Authentication | Webservice Authentication | Can use API to authenticate |
| scepreqs | https://[server_name]/api/v1/scepreqs/ | Webservice Basic Authentication | Certificate Management | Can view / change certificate enrollment request |
| recovery | https://[server_name]/api/v1/recovery/ | Webservice Basic Authentication | Maintenance | Can perform configuration backup |

| Resource Name | Base URL | Authentication Method | Applicable Built-in Permission Set | Required Permission code |
|---|---|---|---|---|
| scheduledbackupsettings | https://[server_name]/api/v1/scheduledbackupsettings/ | Webservice Basic Authentication | Maintenance | Can change scheduled configuration backup settings |
| smtpservers | https://[server_name]/api/v1/smtpservers/ | Webservice Basic Authentication | Messaging Configuration | Can view / change SMTP server |
| ssoauth | https://[server_name]/api/v1/ssoauth/ | Webservice Basic Authentication | Webservice Authentication | Can use API to authenticate |
| ssogroup | https://[server_name]/api/v1/ssogroup/ | Webservice Basic Authentication | SSO Settings | Can view / change SSO group |
| syslogservers | https://[server_name]/api/v1/syslogservers/ | Webservice Basic Authentication | SSO Settings | Can view / change syslog source |
| system | https://[server_name]/api/v1/system/ | Webservice Basic Authentication | System Administration | Can change system access settings |
| systeminfo | https://[server_name]/api/v1/systeminfo/ | Webservice Basic Authentication | Maintenance | Can view / change HA setting |
| radiusclients | https://[server_name]/api/v1/radiusclients/ | Webservice Basic Authentication | RADIUS Service | Can view / change RADIUS Clients |
| radiuspolicies | https://[server_name]/api/v1/radiuspolicies/ | Webservice Basic Authentication | RADIUS Service | Can view RADIUS Policies |
| radiuspolicyclient | https://[server_name]/api/v1/radiuspolicyclient/ | Webservice Basic Authentication | RADIUS Service | Can view / change RADIUS Policies/ Clients |

| Resource Name | Base URL | Authentication Method | Applicable Built-in Permission Set | Required Permission code |
|---|---|---|---|---|
| tacplusclients | https://[server_name]/api/v1/tacplusclients/ | Webservice Basic Authentication | TACACS+ Service | Can view / change TACACS+ Clients |
| tacpluspolicies | https://[server_name]/api/v1/tacpluspolicies/ | Webservice Basic Authentication | TACACS+ Service | Can view TACACS+ Policies |
| tacpluspolicyclient | https://[server_name]/api/v1/tacpluspolicyclient/ | Webservice Basic Authentication | TACACS+ Service | Can view / change TACACS+ Policies/Clients |
| transfertoken | https://[server_name]/api/v1/transfertoken/ | None | None | None |
| usercerts | https://[server_name]/api/v1/usercerts/ | Webservice Basic Authentication | Certificate Management | Can view / change user certificate |
| userfortitokenpolicy | https://[server_name]/api/v1/userfortitokenpolicy/ | Webservice Basic Authentication | Webservice Authentication | Can use API to authenticate |
| userlockoutpolicy | https://[server_name]/api/v1/userlockoutpolicy/ | Webservice Basic Authentication | Account Policy | Can view / change user lockout policy settings |

# Example API calls

For the purpose of these examples, cURL is being used to make the requests. cURL is more flexible than a browser alone, is cross platform and can be called from most scripts. It is not as flexible as native scripting languages but is a good clear example which can be used to understand how the API functions.

The following flags are used in the cURL query:

- **-kIgnore certificate errors** - This can be overcome with use of a valid certificate.
- **-vVerbose** - Increase the level of logging information (useful for debugging).
- **-uUser** - Login information in the format `USER[:PASSWORD]`.

When using PUT/POST with cURL on Windows, problems can be encountered with escaping of the required double quotes in the data content, leading to errors related to incomplete closed brackets. To avoid this, the code should be properly escaped (using \ before any double quotes) or the data text stored in a file and referenced using:

`-d @<filename>`

Alternatively, it is highly recommended that this is run on a Linux OS, where escaping of characters in cURL is more predictable.

# General API usage

## View available endpoint resources

### JSON query

- JSON specified via GET

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS"
      https://192.168.0.122/api/v1/?format=json
```

- JSON specified via Accept Header

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS"  -H 'Accept: application/json'
      https://192.168.0.122/api/v1/
```

### Response

```
< HTTP/1.1 200 OK< Date: Mon, 09 Jun 2014 10:51:23 GMT< Server: Apache< Vary: Accept-
      Language,Cookie< X-Frame-Options: SAMEORIGIN< Content-Language: en< Transfer-Encoding:
      chunked< Content-Type: application/json<* Connection #0 to host 192.168.0.122 left intact*
      Closing connection #0
```

```
{"auth": {"list_endpoint": "/api/v1/auth/", "schema": "/api/v1/auth/schema/"}, "fgtgroupfilter":
      {"list_endpoint": "/api/v1/fgtgroupfilter/", "schema": "/api/v1/fgtgroupfilter/schema/"},
      "fortitokens": {"list_endpoint": "/api/v1/fortitokens/", "schema":
      "/api/v1/fortitokens/schema/"}, "localusers": {"list_endpoint": "/api/v1/localusers/",
      "schema": "/api/v1/localusers/schema/"}, "ssoauth": {"list_endpoint": "/api/v1/ssoauth/",
      "schema": "/api/v1/ssoauth/schema/"}, "ssogroup": {"list_endpoint": "/api/v1/ssogroup/",
      "schema": "/api/v1/ssogroup/schema/"}, "usergroups": {"list_endpoint": "/api/v1/usergroups/",
      "schema": "/api/v1/usergroups/schema/"}}
```


### XML query

- XML specified via GET

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS"
      https://192.168.0.122/api/v1/?format=xml
```

- XML specified via Accept Header

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS" -H 'Accept: application/xml'
      https://192.168.0.122/api/v1/
```

### Response


```
< HTTP/1.1 200 OK
< Date: Mon, 09 Jun 2014 11:03:25 GMT
< Server: Apache
< Vary: Accept-Language,Cookie
< X-Frame-Options: SAMEORIGIN
< Content-Language: en
< Transfer-Encoding: chunked
< Content-Type: application/xml; charset=utf-8
<
<?xml version='1.0' encoding='utf-8'?>
* Connection #0 to host 192.168.0.122 left intact
* Closing connection #0
<response><fgtgroupfilter type="hash"><list_endpoint>/api/v1/fgtgroupfilter/</list_
      endpoint><schema>/api/v1/fgtgroupfilter/schema/</schema></fgtgroupfilter><localusers
      type="hash"><list_endpoint>/api/v1/localusers/</list_
      endpoint><schema>/api/v1/localusers/schema/</schema></localusers><usergroups
      type="hash"><list_endpoint>/api/v1/usergroups/</list_
      endpoint><schema>/api/v1/usergroups/schema/</schema></usergroups><auth type="hash"><list_
      endpoint>/api/v1/auth/</list_endpoint><schema>/api/v1/auth/schema/</schema></auth><fortitokens
      type="hash"><list_endpoint>/api/v1/fortitokens/</list_
      endpoint><schema>/api/v1/fortitokens/schema/</schema></fortitokens><ssogroup
      type="hash"><list_endpoint>/api/v1/ssogroup/</list_
      endpoint><schema>/api/v1/ssogroup/schema/</schema></ssogroup><ssoauth type="hash"><list_
      endpoint>/api/v1/ssoauth/</list_
      endpoint><schema>/api/v1/ssoauth/schema/</schema></ssoauth></response>
```


# User groups (/usergroups/)

URL: `https://[server_name]/api/[api_version]/usergroups/`

This endpoint represents the local user group resource.

In the FortiAuthenticator GUI, this resource corresponds to Authentication → User Groups.

This API is for use by third-party user provisioning systems.

# Supported fields

| Field | Description | Type | Required | Other restrictions |
|---|---|---|---|---|
| name | Group name | String | Yes | max length = 50 |
| users | List of local users in the group | List | No | List of local users URI |
| password_policy | Associated password policy | String | No | Default is set if not specified. |
| return_members | Boolean to disable return of members | boolean | No | Must be present as a query parameter. |

# Allowed methods

| Allowed methods | Resource URI | Action |
|---|---|---|
| GET | | Get all groups and associated users. |
| POST | | Create a new user. |
| PUT | | Replaces all of the resources for the group. This is done by removing all existing items first before creating the new items. Data must follow the same format as the data returned by the GET parameter. |
| PATCH | | Add users to a user group. |
| DELETE | | Delete a specified group. |

# Allowed filters

| Field | Filters |
|---|---|
| name | `exact` |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# View all user groups

### JSON query

- JSON specified via GET

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS"
      https://192.168.0.122/api/v1/usergroups/?format=xml
```

- JSON specified via Accept Header

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS" -H 'Accept: application/xml'
      https://192.168.0.122/api/v1/usergroups/
```

### Response

```
< HTTP/1.1 200 OK
< Date: Mon, 09 Jun 2014 11:46:34 GMT
< Server: Apache
< Vary: Accept,Accept-Language,Cookie
< X-Frame-Options: SAMEORIGIN
< Content-Language: en
< Cache-Control: no-cache
< Transfer-Encoding: chunked
< Content-Type: application/xml; charset=utf-8
<
<?xml version='1.0' encoding='utf-8'?>
* Connection #0 to host 192.168.0.122 left intact
* Closing connection #0
<response>

<objects type="list"><object><users type="list"/>
<idtype="integer">5</id><name>Test_Local_Engineers</name><resource_
      uri>/api/v1/usergroups/5/</resource_uri></object>

<object><users type="list"/>
<idtype="integer">4</id><name>Test_Local_Marketing</name><resource_
      uri>/api/v1/usergroups/4/</resource_uri></object>

<object><users type="list"><value>/api/v1/localusers/4/</value></users>
<idtype="integer">3</id><name>Test_Local</name><resource_uri>/api/v1/usergroups/3/</resource_
      uri></object></objects>

<meta type="hash"><next type="null"/><total_count type="integer">3</total_count><previous
      type="null"/><limit type="integer">20</limit><offset
      type="integer">0</offset></meta></response>
```

The response above has been reformatted with carriage returns to make the results more clear.

The response shows that there are 3 groups already configured (in RED).

- Test_Local_Engineers (in ID position 5)
- Test_Local_Marketing (in ID position 4)
- Test_Local (in ID position 3)

Test_Local_Engineers and Test_Local_Marketing groups do not contain any users, however, the Test_Local group contains 1 user, identified as local user with ID=4 (in **GREEN**). See the LocalUsers for identifying Usernames from user IDs.

The total number of configured and supported User Groups is also returned for troubleshooting purposes (in **GOLD**).

# Create a user group

### JSON query

- JSON specified via Accept Header

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS" -X POST -d '{"name":"Group999"}' -H
      'Content-Type: application/json' https://192.168.0.122/api/v1/usergroups/
```

### Response

```
< HTTP/1.1 201 CREATED
< Date: Mon, 09 Jun 2014 12:02:33 GMT
< Server: Apache
< Vary: Accept,Accept-Language,Cookie
< X-Frame-Options: SAMEORIGIN
< Content-Language: en
< Location: https://192.168.0.122/api/v1/usergroups/6/
< Content-Length: 0
< Content-Type: text/html; charset=utf-8
```

### Verify user group creation

Use API call documented in Allowed filters on page 24

| Field | Lookup expressions | Values |
|---|---|---|
| username | exact, iexact, contains, icontains, in | |
| first_name | exact, iexact, contains, icontains | |
| last_name | exact, iexact, contains, icontains | |
| email | exact, iexact, contains, icontains, in | |
| active | exact | |
| city | exact, iexact, contains, icontains | |
| state | exact, iexact, contains, icontains | |
| country | exact, iexact, contains, icontains | |
| token_type | | ftk, ftm, email, sms |
| token_serial | exact, iexact | |

# Third-party Integration: FortiToken Mobile provisioning

For integration with a third-party authentication server which needs to manage token validation, it is possible for the FortiAuthenticator to return FortiToken Mobile (FTM) seed during provisioning. However, certain conditions must be met:

- Seed may only be returned when creating a new local user via POST method and when provisioning an FTM to an existing user via PATCH method.
- A GET URL parameter (returnseed=1) needs to be specified to explicitly tell FortiAuthenticator to return an encrypted seed for the token (e.g. https://[server_name]/api/v1/localusers/2/?returnseed=1).
- A seed encryption passphrase must be specified in FortiGuard settings.

The seed is encrypted and returned as a PSKC XML file string according to RFC 6030. The key is derived from the configured passphrase using the PBKDF2 key derivation function (32 byte key length, 1000 iterations), encrypted with AES 256 CBC encryption, and signed with a SHA256 HMAC.

Whenever an FTM is provisioned, its activation code will be returned as well.

# List all local users above

```
< HTTP/1.1 200 OK
< Date: Mon, 09 Jun 2014 12:18:19 GMT
< Server: Apache
< Vary: Accept,Accept-Language,Cookie
< X-Frame-Options: SAMEORIGIN
< Content-Language: en
< Cache-Control: no-cache
< Transfer-Encoding: chunked
< Content-Type: application/xml; charset=utf-8
<
<?xml version='1.0' encoding='utf-8'?>
* Connection #0 to host 192.168.0.122 left intact
* Closing connection #0
<response><objects type="list"><object><users type="list"/><id type="integer">6</id>
      <name>Group999</name><resource_uri>/api/v1/usergroups/6/</resource_uri></object><object><users
      type="list"/><id type="integer">5</id><name>Test_Local_Engineers</name><resource_
      uri>/api/v1/usergroups/5/</resource_uri></object><object><users type="list"/><id
      type="integer">4</id><name>Test_Local_Marketing</name><resource_
      uri>/api/v1/usergroups/4/</resource_uri></object><object><users
      type="list"><value>/api/v1/localusers/4/</value></users><id type="integer">3</id><name>Test_
      Local</name><resource_uri>/api/v1/usergroups/3/</resource_uri></object></objects><meta
      type="hash"><next type="null"/><total_count type="integer">4</total_count><previous
      type="null"/><limit type="integer">20</limit><offset
      type="integer">0</offset></meta></response>
```

### Attempt to create a user group with the same name

```
< HTTP/1.1 400 BAD REQUEST
< Date: Mon, 09 Jun 2014 12:04:06 GMT
< Server: Apache
< Vary: Accept-Language,Cookie
< X-Frame-Options: SAMEORIGIN
< Content-Language: en
< Connection: close
< Transfer-Encoding: chunked
< Content-Type: application/json
<
* Closing connection #0
{"usergroups": {"name": ["A user group with that name already exists."]}}
```

# Add a user to a group

Note, the required users should be elucidated by querying the /localusers/ list as documented in the Local users (/localusers/) on page 36 section. In this example:

| | | |
|---|---|---|
| test_user | = | /api/v1/localusers/5/ |
| test_user2 | = | /api/v1/localusers/5/ |

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS" -X PATCH -d '{"users":
      ["/api/v1/localusers/5/","/api/v1/localusers/4/"]}' -H 'Content-Type: application/json'
      https://192.168.0.122/api/v1/usergroups/9/
```

This command is not additive i.e. adding a single user entry will not increment the list it will overwrite. Using `{"users":[ ]}` for example will clear the users list.

# Delete a user group

### JSON query

- JSON specified via Accept Header

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS" -X DELETE -H 'Content-Type:
      application/json' https://192.168.0.122/api/v1/usergroups/6/
```

### Response

```
< HTTP/1.1 204 NO CONTENT
< Date: Mon, 09 Jun 2014 12:25:18 GMT
< Server: Apache
< Vary: Accept,Accept-Language,Cookie
< X-Frame-Options: SAMEORIGIN
< Content-Language: en
< Content-Length: 0
< Content-Type: text/html; charset=utf-8
<
* Connection #0 to host 192.168.0.122 left intact
* Closing connection #0
```

Note that 204 NO CONTENT shows that the group has been successfully deleted. A subsequent listing confirms this as Group999 no longer exists:

```
< HTTP/1.1 200 OK
< Date: Mon, 09 Jun 2014 12:26:05 GMT
< Server: Apache
< Vary: Accept,Accept-Language,Cookie
< X-Frame-Options: SAMEORIGIN
< Content-Language: en
< Cache-Control: no-cache
< Transfer-Encoding: chunked
< Content-Type: application/xml; charset=utf-8
<
```

```
<?xml version='1.0' encoding='utf-8'?>
* Connection #0 to host 192.168.0.122 left intact
* Closing connection #0
<response><objects type="list"><object><users type="list"/><id type="integer">5</id><name>Test_
      Local_Engineers</name><resource_uri>/api/v1/usergroups/5/</resource_
      uri></object><object><users type="list"/><id type="integer">4</id><name>Test_Local_
      Marketing</name><resource_uri>/api/v1/usergroups/4/</resource_uri></object><object><users
      type="list"><value>/api/v1/localusers/4/</value></users><id type="integer">3</id><name>Test_
      Local</name><resource_uri>/api/v1/usergroups/3/</resource_uri></object></objects><meta
      type="hash"><next type="null"/><total_count type="integer">3</total_count><previous
      type="null"/><limit type="integer">20</limit><offset
      type="integer">0</offset></meta></response>[Carl@CentOS ~]$
```

The Delete command will delete the group even if the group contains users or if it is in use e.g. in a RADIUS Client configuration. Checks should be made prior to executing this command.

# View a specific user group

### JSON query

- JSON specified via GET

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS"
      “https://192.168.0.122/api/v1/usergroups/?format=json&name=/api/v1/usergroups/8/”
```

- JSON specified via Accept Header

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS" -H 'Accept:
      application/json'“https://192.168.0.122/api/v1/usergroups/?format=json&name=Group999”
```

The filter used in this situation is the group “name” not the URL or ID.

The URL requires additional quoting in this case otherwise the Unix CLI treats the “&” as an instruction to place the cURL command into the background.

Querying a non-existent group will return a successful 200 OK response with empty object data. This is by design as this is not necessarily an error situation.

### Response

```
< HTTP/1.1 200 OK
< Date: Tue, 10 Jun 2014 10:11:47 GMT
< Server: Apache
< Vary: Accept,Accept-Language,Cookie
< X-Frame-Options: SAMEORIGIN
```

```
< Content-Language: en
< Cache-Control: no-cache
< Transfer-Encoding: chunked
< Content-Type: application/json
<
* Connection #0 to host 192.168.0.122 left intact
* Closing connection #0
{"meta": {"limit": 20, "next": null, "offset": 0, "previous": null, "total_count": 1}, "objects":
      [{"id": 9, "name": "Group999", "resource_uri": "/api/v1/usergroups/9/", "users":
      ["/api/v1/localusers/5/"]}]}
```

# FortiTokens (/fortitokens/)

URL: `https://[server_name]/api/[api_version]/fortitokens/`

This endpoint represents the FortiToken resource. In the FortiAuthenticator GUI, this resource corresponds to **Authentication > User Management > FortiTokens**. This API is for use by third-party user provisioning systems to ascertain which tokens are available to be provisioned to a user.

## Supported fields

| Field | Display name | Type | Required | Read Only | Other restrictions |
|---|---|---|---|---|---|
| serial | Serial number | string | No | | |
| type | Type | string | No | | Either `ftk` or `ftm` |
| status | Status | string | No | | One of `new, available, pending, assigned` |
| locked | locked | boolean | No | | `true` or `false` |
| license | license | string | No | | The license under which the FortiToken was activated |
| last_used_at | Last used time | string | No | Yes | ISO-8601 formatted time in UTC. |
| sub_type | FortiToken Mobile sub-type | string | No | Yes | Only present if `type==ftm`.<br>Value can be one of `default`, `offline` or `third-party`. |

# Allowed methods

| HTTP Method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/fortitokens/ | Get all FortiTokens |
| DELETE | /api/v1/fortitokens/[id] | Delete one FortiToken |

# Allowed filters

| Field | Lookup expressions | Values |
|---|---|---|
| serial | exact, iexact | |
| type | | `ftk, ftm` |
| status | | `new, available, pending, assigned` |
| license | | A string, for example `FTMTRIALNOREGIST` |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# View all tokens

### JSON query

- JSON specified via GET

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS"
      https://192.168.0.122/api/v1/fortitokens/?format=json
```

### Response

```
< HTTP/1.1 200 OK
< Date: Mon, 09 Jun 2014 18:17:42 GMT
< Server: Apache
< Vary: Accept,Accept-Language,Cookie
< X-Frame-Options: SAMEORIGIN
< Content-Language: en
< Cache-Control: no-cache
< Transfer-Encoding: chunked
< Content-Type: application/json
<
* Connection #0 to host 192.168.0.122 left intact
* Closing connection #0
{"meta": {"limit": 20, "next": null, "offset": 0, "previous": null, "total_count": 2}, "objects":
      [{"resource_uri": "/api/v1/fortitokens/1/", "serial": "FTKMOB44142CCBF3", "status":
      "available", "type": "ftm"}, {"resource_uri": "/api/v1/fortitokens/2/", "serial":
      "FTKMOB4471BB94D1", "status": "available", "type": "ftm"}]}
```

# View subset of tokens using filters

This example shows how it is possible to obtain a list of specific tokens e.g. The first available FortiToken Mobile token.

### JSON query

- JSON specified via GET

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS" -H 'Accept: application/json'
      "https://192.168.0.122/api/v1/fortitokens/?format=json&type=ftm&status=available&limit=1"
```

The URL requires additional quoting in this case otherwise the Unix CLI treats the "&" as an instruction to place the cURL command into the background.

### Response

```
< HTTP/1.1 200 OK
< Date: Mon, 09 Jun 2014 18:17:42 GMT
< Server: Apache
< Vary: Accept,Accept-Language,Cookie
< X-Frame-Options: SAMEORIGIN
< Content-Language: en
< Cache-Control: no-cache
< Transfer-Encoding: chunked
< Content-Type: application/json
<
* Connection #0 to host 192.168.0.122 left intact
* Closing connection #0
{"meta": {"limit": 1, "next":
      "/api/v1/fortitokens/?status=available&type=ftm&offset=1&limit=1&format=json", "offset": 0,
      "previous": null, "total_count": 2}, "objects": [{"resource_uri": "/api/v1/fortitokens/1/",
      "serial": "FTKMOB44142CCBF3", "status": "available", "type": "ftm"}]}
```

# Push authentication (/pushauth/)

URL: `https://[server_name]/api/[api_version]/pushauth/`

This endpoint is used to trigger a token code validation from a user's FTM app. The validation involves the use of a third-party's (e.g. Apple or Google) Push servers. This API is for use by third-party authentication system for verify login against FortiAuthenticator on their mobile devices.

In order to use the Push authentication feature, please ensure the FTM version is newer than 4.0.

If mobile devices and FortiAuthenticator are not in the same subnet, please configure the public IP/FQDN settings at **System > Administration > System Access** page to guarantee that FortiAuthenticator is reachable from FTM.

# Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| username | User Name | string | Yes | max length=50, unique |
| realm | Realm | string | No | One of the existing realm configured in FAC. Required if more than one user matches the username field. |
| user_ip | User IP | string | No | |
| timestamp | Timestamp | string | No | UTC format |
| account | User account in third-party system | string | No | |
| user_agent | The end-user's software agent that triggered the push request | string | No | |
| log_message | Log information | string | No | |

# Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| POST | /api/v1/pushauth/ | Create and send a push message. |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# Response codes

In addition to the general codes defined in General API response codes on page 138, a POST request to this resource can also result in the following return codes:

| Code | Response content | Description |
|---|---|---|
| 200 OK | | User is successfully authenticated on their mobile devices. |
| 401 Unauthorized | | User rejected the authentication request. |
| 404 Not Found | | The given username does not exist in the system or there is no FortiToken Mobile assigned to the given user. |

| Code | Response content | Description |
|---|---|---|
| 500 Internal Server Error | | Push server is refusing to send the push notification. |
| 503 Service Unavailable | | Push server is unreachable. |

# Push authentication response (/pushauthresp/)

URL: `https://[server_name]/api/[api_version]/pushauthresp/`

This endpoint is used by FortiToken Mobile devices to submit the response to a token code validation request triggered by a prior call to the /pushauth/ endpoint. This API is for use by FTM2 to send back the OTP for login verification.

## Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| session_id | Authentication session ID | string | Yes | unique |
| action | Requested action | string | Yes | Must be "validate" or "alert" |
| token_code | Security token code | string | Yes | Only required when "action" is "validate" |
| message | Alert message | string | Yes | Only required when "action" is "alert" |
| hmac | HMAC verification | string | Yes | Only required when "action" is "alert" |

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| `POST` | `/api/v1/pushauthresp/` | Validate the token code for the specified authentication session. |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

## Response codes

In addition to the general codes defined in General API response codes on page 138, a POST request to this resource can also result in the following return codes:

| Code | Response content | Description |
|---|---|---|
| 200 OK | | Valid credentials |
| 401 Unauthorized | | Invalid credentials |

# External IP/FQDN configuration (/system/external_ip_fqdn/)

URL: `https://[server_name]/api/[api_version]/system/external_ip_fqdn/`

This endpoint is used to set IP/FQDN exposing FortiAuthenticator to external internet.

## Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| value | External IP/FQDN | string | Yes | IP or FQDN, port number is optional and defaults to 443. |

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| `GET` | `/api/v1/system/external_ip_fqdn/` | Get current value of IP/FQDN settings. |
| `POST` | `/api/v1/system/external_ip_fqdn/` | Set a new value for IP/FQDN. |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# Local users (/localusers/)

URL: `https://[server_name]/api/[api_version]/localusers/`

This endpoint represents local user resource, namely a user account. This resource can be found in the FortiAuthenticator GUI under **Authentication > Local Users**. This API is for use by third-party provisioning systems.

# Supported fields

| Field | Display name | Type | Required | Read Only | Other restrictions |
|---|---|---|---|---|---|
| username | Username | string | Yes | Yes (unless when creating a new user) | max length = 253, contains only letters, numbers and @/./+/-/_ characters |
| address | Address | string | No | | max length = 80 |
| city | City | string | No | | max length = 40 |
| country | Country | string | No | | Must be a country code from ISO-3166 list |
| custom1 | Custom user field 1 | string | No | | max length = 255 |
| custom2 | Custom user field 2 | string | No | | max length = 255 |
| custom3 | Custom user field 3 | string | No | | max length = 255 |
| email | E-mail address | string | No | | Must be a valid e-mail address |
| first_name | First name | string | No | | max length = 30 |
| last_name | Last name | string | No | | max length = 30 |
| active | Account Status | boolean | No | | |
| reason | Disable reason | Interger | No | | Default is 0. One of 0 (manually disabled), 1 (account inactivity), 2 (too many failed attempts), 3 (account expiry), 4 (password expiry), 5 (FTM activation expiry), 6 (revoked token), 7 (usage limit exceeded), or 8 (pending administrator approval). |
| mobile_ number | Mobile number | string | No | | max length = 25, must follow international number format: +[country_code]- |

| Field | Display name | Type | Required | Read Only | Other restrictions |
|---|---|---|---|---|---|
| | | | | | [number] |
| phone_ number | Mobile number | string | No | | max length = 25 |
| state | State or province | string | No | | max length = 40 |
| user_groups | Local user groups that this user is a member of | list | No | | List of user groups URI. Read Only. See usergroups or localgroup-memberships to make relations. |
| token_auth | Token Auth | boolean | No | | Whether second factor authentication should be enabled. If 'true', token_ type is required. |
| token_type | Token Type | string | No | | One of ftk, ftm, ftc, email, sms, or dual. If email is chosen, email is required. If sms is chosen, mobile_number is required. Both are required if dual is selected. |
| token_serial | Token Serial | string | No | | If token_type is ftm, or ftk, and this is not present or blank, the next available token will be assigned. |
| ftm_act_ method | FTM Activation Delivery Method | string | No | | One of email or sms. If email is chosen, email is required. If sms is chosen, mobile_number is required. |
| ftk_only | Enable FortiToken-only authentication | boolean | No | | If set, token_auth must be true, and token_type must be either ftk or ftm. If this field is changed to false, email must be set to reset user's password and send a new random password. Mutually exclusive with password. |
| expires_at | Expiration time | string | No | | ISO-8601 formatted user |

| Field | Display name | Type | Required | Read Only | Other restrictions |
|---|---|---|---|---|---|
| | | | | | expiration time in UTC. Specified time should be formatted using ISO-8601 with a timezone offset. If timezone info is not set, time is always assumed to be in UTC. To remove an expiration time, set this field to an empty string. Time must be at least an hour in the future. |
| token_fas | Token from FortiAnalyzer | boolean | No | | True if token is issued from FortiAnalyzer. The default is false. |
| fido | FIDO | boolean | No | No | Default is disabled. |
| company | Company | string | No | | |
| department | Department | string | No | | |
| is_locked | Administrative account lock | boolean | No | | |

Additionally, when creating a new user, the following field is available:

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| password | Password | string | No | max length = 50 |
| change_password | Change password on next logon | boolean | No | |
| recovery_by_question | Allow password recovery with security question | boolean | No | |
| recovery_question | Password recovery security question | string | No | Required if recovery_by_question is true. |
| recovery_answer | Password recovery security answer | string | No | Required if recovery_by_question is true. |
| fido | FIDO | boolean | No | No |

# Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/localusers/ | Get all regular local users. |
| GET | /api/v1/localusers/[id]/ | Get a specific local user with ID. |
| POST | /api/v1/localusers/ | Create a new local user.<br>**Notes:**<br>• If password is specified, that password will be set.<br>• If password is not specified, email field becomes required, and a random password will be created and e-mailed to the new user. |
| POST | /api/v1/localusers/[id]/sendoobtoken/ | Send an out-of-band token code (email/SMS token) to a local user. |
| POST | /api/v1/localusers/[id]/verifyrecoveryanswer/ | Verify the recovery answer for a specific local user. Note: recovery_answer must be included. Returns status 202 if the supplied recovery_answer parameter is correct, or 404 if not correct. |
| PATCH | /api/v1/localusers/[id]/ | Update specified fields for a specific local user with ID. |
| DELETE | /api/v1/localusers/[id]/ | Delete a local user. |

# Allowed filters

| Field | Lookup expressions | Values |
|---|---|---|
| username | `exact, iexact, contains, icontains, in` | |

| Field | Lookup expressions | Values |
|---|---|---|
| abridged | Use `abridged=1` as query parameter with username to get reduced user objects which returns faster. | `active, address, city. country, custom1, custom2, custom3, email, first_name, last_name, id, mobile_number, phone_number, recovery_by_question, resource_uri, state, token_auth, token_type, user_groups, username` |
| first_name | `exact, iexact, contains, icontains` | |
| last_name | `exact, iexact, contains, icontains` | |
| email | `exact, iexact, contains, icontains, in` | |
| active | `exact` | |
| city | `exact, iexact, contains, icontains` | |
| state | `exact, iexact, contains, icontains` | |
| country | `exact, iexact, contains, icontains` | |
| token_type | | `ftk, ftm, ftc, email, sms` |
| token_serial | `exact, iexact` | |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# Third-party integration: FTM provisioning

For integration with a third-party authentication server which needs to manage token validation, it is possible for the FortiAuthenticator to return FTM seed during provisioning. However, certain conditions must be met:

- Seed may only be returned when creating a new local user via POST method and when provisioning an FTM to an existing user via PATCH method.
- A GET URL parameter (returnseed=1) needs to be specified to explicitly tell FortiAuthenticator to return an encrypted seed for the token (e.g. https://[server_name]/api/v1/localusers/2/?returnseed=1).
- A seed encryption passphrase must be specified in FortiGuard settings.

The seed is encrypted and returned as a PSKC XML file string according to RFC 6030. The key is derived from the configured passphrase using the PBKDF2 key derivation function (32 byte key length, 1000 iterations), encrypted with AES 256 CBC encryption, and signed with a SHA256 HMAC.

Whenever an FTM is provisioned, its activation code will be returned as well.

# List all local users

### JSON query

- JSON specified via GET

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS"
      https://192.168.0.122/api/v1/localusers/?format=xml
```

- JSON specified via Accept Header

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS" -H 'Accept: application/xml'
      https://192.168.0.122/api/v1/ localusers/
```

### Response

```
< HTTP/1.1 200 OK
< Date: Mon, 09 Jun 2014 20:14:23 GMT
< Server: Apache
< Vary: Accept,Accept-Language,Cookie
< X-Frame-Options: SAMEORIGIN
< Content-Language: en
< Cache-Control: no-cache
< Transfer-Encoding: chunked
< Content-Type: application/json
<
* Connection #0 to host 192.168.0.122 left intact
* Closing connection #0
{"meta": {"limit": 20, "next": null, "offset": 0, "previous": null, "total_count": 2}, "objects":
      [{"address": "", "city": "", "country": "", "custom1": "", "custom2": "", "custom3": "",
      "email": "", "first_name": "", "id": 5, "last_name": "", "mobile_number": "", "phone_number":
      "", "resource_uri": "/api/v1/localusers/5/", "state": "", "token_auth": false, "token_serial":
      "", "token_type": null, "user_groups": ["/api/v1/usergroups/9/", "/api/v1/usergroups/8/"],
      "username": "test_user2"}, {"address": "", "city": "", "country": "", "custom1": "",
      "custom2": "", "custom3": "", "email": "", "first_name": "", "id": 4, "last_name": "",
      "mobile_number": "", "phone_number": "", "resource_uri": "/api/v1/localusers/4/", "state": "",
      "token_auth": false, "token_serial": "", "token_type": null, "user_groups":
      ["/api/v1/usergroups/8/"], "username": "test_user"}]}
```

Here you will notice that there are 2 users defined “test_user” and “test_user2”. Note that admin users are not returned by the localusers query.

# Create local user

### JSON query

- JSON specified via Accept Header

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS" -X POST -d '{"username":"test_
      user3","password":"testpassword","email":"test_user3@example.com","mobile_number":"+44-
      1234567890"}' -H 'Content-Type: application/json' https://192.168.0.122/api/v1/localusers/
```

### Response

```
< HTTP/1.1 201 CREATED
< Date: Mon, 09 Jun 2014 20:29:20 GMT
```

```
< Server: Apache
< Vary: Accept,Accept-Language,Cookie
< X-Frame-Options: SAMEORIGIN
< Content-Language: en
< Location: https://192.168.0.122/api/v1/localusers/6/
< Content-Length: 0
< Content-Type: text/html; charset=utf-8
```

### Verify user creation

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS"
      https://192.168.0.122/api/v1/localusers/?format=json


< HTTP/1.1 200 OK
< Date: Mon, 09 Jun 2014 20:30:26 GMT
< Server: Apache
< Vary: Accept,Accept-Language,Cookie
< X-Frame-Options: SAMEORIGIN
< Content-Language: en
< Cache-Control: no-cache
< Transfer-Encoding: chunked
< Content-Type: application/json
<
{"meta": {"limit": 20, "next": null, "offset": 0, "previous": null, "total_count": 3}, "objects":
      [{"address": "", "city": "", "country": "", "custom1": "", "custom2": "", "custom3": "",
      "email": "", "first_name": "", "id": 5, "last_name": "", "mobile_number": "", "phone_number":
      "", "resource_uri": "/api/v1/localusers/5/", "state": "", "token_auth": false, "token_serial":
      "", "token_type": null, "user_groups": ["/api/v1/usergroups/9/", "/api/v1/usergroups/8/"],
      "username": "test_user2"}, {"address": "", "city": "", "country": "", "custom1": "",
      "custom2": "", "custom3": "", "email": "", "first_name": "", "id": 4, "last_name": "",
      "mobile_number": "", "phone_number": "", "resource_uri": "/api/v1/localusers/4/", "state": "",
      "token_auth": false, "token_serial": "", "token_type": null, "user_groups":
      ["/api/v1/usergroups/8/"], "username": "test_user"}, {"address": "", "city": "", "country":
      "", "custom1": "", "custom2": "", "custom3": "", "email": "test_user3@example.com", "first_
      name": "", "id": 6, "last_name": "", "mobile_number": "", "phone_number": "", "resource_uri":
      "/api/v1/localusers/6/", "state": "", "token_auth": false, "token_serial": "", "token_type":
      null, "user_groups": [], "username": "test_user3"}]}
```

# Modify local user

### JSON query

- JSON specified via Accept Header

Modify the newly created user “test_user3” aka User ID == 6 using the PATCH command.

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS" -X PATCH -d '
      {"custom1":"example","country":"GB"}' -H 'Content-Type: application/json'
      https://192.168.0.122/api/v1/localusers/6/
```

### Response

```
< HTTP/1.1 202 ACCEPTED
< Date: Mon, 09 Jun 2014 21:07:28 GMT
< Server: Apache
< Vary: Accept,Accept-Language,Cookie
```

```
< X-Frame-Options: SAMEORIGIN
< Content-Language: en
< Content-Length: 0
< Content-Type: text/html; charset=utf-8
```

# Delete local user

Send an HTTP DELETE to the resource with the user ID to delete a local user, in the following format:

```
https://<server-name>/api/v1/localusers/5/
```

A successful response will show in the following format:

```
HTTP/1.1 204 NO CONTENT
```

# Applying filters

## List specific local user

### JSON query

- JSON specified via GET

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS"
      “https://192.168.0.122/api/v1/localusers/?format=json&username=test_user3”
```

- JSON specified via Accept Header

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS" -H 'Accept: application/json'
      “https://192.168.0.122/api/v1/localusers/?username=test_user3”
```

### Response

```
< HTTP/1.1 200 OK
< Date: Tue, 10 Jun 2014 11:06:20 GMT
< Server: Apache
< Vary: Accept,Accept-Language,Cookie
< X-Frame-Options: SAMEORIGIN
< Content-Language: en
< Cache-Control: no-cache
< Transfer-Encoding: chunked
< Content-Type: application/json
<
* Connection #0 to host 192.168.0.122 left intact
* Closing connection #0
{"meta": {"limit": 20, "next": null, "offset": 0, "previous": null, "total_count": 1}, "objects":
      [{"address": "", "city": "", "country": "", "custom1": "example", "custom2": "", "custom3":
      "", "email": "test_user3@example.com", "first_name": "", "id": 6, "last_name": "", "mobile_
      number": "", "phone_number": "", "resource_uri": "/api/v1/localusers/6/", "state": "", "token_
      auth": false, "token_serial": "", "token_type": null, "user_groups": [], "username": "test_
      user3"}]}[
```

## View all users from Country=GB

### JSON query

- JSON specified via Accept Header

View all users from the country GB (Great Britain).

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS" -H 'Accept: application/json'
      "https://192.168.0.122/api/v1/localusers/?country=GB"
```

### Response

```
< HTTP/1.1 200 OK
< Date: Tue, 10 Jun 2014 11:14:39 GMT
< Server: Apache
< Vary: Accept,Accept-Language,Cookie
< X-Frame-Options: SAMEORIGIN
< Content-Language: en
< Cache-Control: no-cache
< Transfer-Encoding: chunked
< Content-Type: application/json
<
* Connection #0 to host 192.168.0.122 left intact
* Closing connection #0
{"meta": {"limit": 20, "next": null, "offset": 0, "previous": null, "total_count": 2}, "objects":
      [{"address": "", "city": "", "country": "GB", "custom1": "example", "custom2": "", "custom3":
      "", "email": "test_user3@example.com", "first_name": "", "id": 6, "last_name": "", "mobile_
      number": "", "phone_number": "", "resource_uri": "/api/v1/localusers/6/", "state": "", "token_
      auth": false, "token_serial": "", "token_type": null, "user_groups": [], "username": "test_
      user3"}, {"address": "", "city": "", "country": "GB", "custom1": "example", "custom2": "",
      "custom3": "", "email": "", "first_name": "", "id": 5, "last_name": "", "mobile_number": "",
      "phone_number": "", "resource_uri": "/api/v1/localusers/5/", "state": "", "token_auth": false,
      "token_serial": "", "token_type": null, "user_groups": ["/api/v1/usergroups/9/",
      "/api/v1/usergroups/8/"], "username": "test_user2"}]}[
```

# Add RADIUS attributes to local users

URL: `https://[server_name]/api/[api_version]/localusers/[id]/radiusattributes/`

This resource can only be used for RADIUS attribute of local users. All the fields are case-sensitive.

## Supported fields

| Field | Display name | Type | Required | Read only | Other restrictions |
|---|---|---|---|---|---|
| owner | owner | string | No | Yes | - |
| vendor | vendor | string | No | No | max length = 40, default = "Default" |

| Field | Display name | Type | Required | Read only | Other restrictions |
|---|---|---|---|---|---|
| attribute | RADIUS attribute | string | Yes | No | max length = 255 |
| attr_val | Attribute Value | Depends on RADIUS attribute | Yes | No | max length = 255 |

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/localusers/[id]/radiusattributes/ | Get all Radius Attributes for a specific Local User |
| GET | /api/v1/localusers/[id]/radiusattributes/[attribute_id]/ | Get a Radius Attribute for a specific Local User |
| POST | /api/v1/localusers/[id]/radiusattributes/ | Create a new Radius Attribute for a specific Local User |
| PUT | /api/v1/localusers/[id]/radiusattributes/ | Update all Radius Attributes that belong to a Local User |
| PATCH | /api/v1/localusers/[id]/radiusattributes/[attribute_id]/ | Update fields of a Radius Attribute |
| DELETE | /api/v1/localusers/[id]/radiusattributes/ | Delete all Radius Attributes from a specific Local User |
| DELETE | /api/v1/localusers/[id]/radiusattributes/[attribute_id]/ | Delete a Radius Attribute from a specific Local User |

## Allowed filters

| Field | Lookup expressions | Values |
|---|---|---|
| vendor | `exact` | - |
| attribute | `exact` | - |
| attr_value | `exact, iexact, contains, icontains, in` | - |

# Local API admin (/localapiadmin/)

URL: `https://[server_name]/api/[api_version]/localapiadmin/`

This endpoint represents local admin resource with access to API only.

## Supported fields

Same as the fields supported by Local users resource plus these additional ones:

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| trusted_hosts | Trusted subnet from which this admin is allowed to logon | list | No | List of IPv4/IPv6 subnets |
| password | Password | string | No | max length = 50 |

Additionally, randomly generated `api_key` would be returned as a field in response upon success. Please refer to examples for more details.

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/localapiadmin/[id]/ | Get a specific local admin with ID `id` |
| POST | /api/v1/localapiadmin | Create a new local admin with access to API endpoints |
| DELETE | /api/v1/localapiadmin/[id]/ | Delete a local admin |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# Local users CSV file import and export (/csv/localusers/)

URL : `https://[server_name]/api/v1/csv/localusers/`

This endpoint is used to import and export local users using CSV files.

# Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| csv | Local users CSV file | String | Yes | |
| missing_users | Action to take for existing user accounts missing from the CSV file | String | No | For CSV import only. Set to one of the following values:<br>• `keep`: Leave the existing user accounts unchanged.<br>• `disable`: Set the status of the existing user accounts to disable.<br>• `delete`: Delete the existing user accounts.<br>**Note**: The default when not specified is `keep`. |

# Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/csv/localusers/ | Export local users CSV file |
| POST | /api/v1/csv/localusers/ | Import local users CSV file |
| GET | /api/v1/csv/localusers/[task_id]/ | Check the status of the local user import |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# Example

JSON query

- Export

```
curl -k -X GET https://[FAC_IP]/api/v1/csv/localusers/
  -H 'Content-Type:application/json'
```

```
    -u "admin:[hash]"
    --output localusers.csv
```

**JSON query**

- Import

```
   curl -k -X POST https://[FAC_IP]/api/v1/csv/localusers/
     -H 'Content-Type:multipart/form-data'
     -u "admin:[hash]"
     -F 'csv=@localusers.csv'
     -F 'missing_users=keep'
```

**Response**

```
{"message": "Importing local users from csv file. This may take a few moments to complete.",
"task_id": "08123f1a"}
```

**Check import status**

```
    curl -k -X GET https://[FAC_IP]/api/v1/csv/localusers/08123f1a/
      -H 'Content-Type:application/json'
      -u "admin:[hash]"
```

**Response**

```
{"status": "completed", "message": "Successfully inserted 2 users, updated 0 users, deleted 0
users - (total: 2 users)"}
```

# LDAP users (/ldapusers/)

**URL:** `https://[server_name]/api/[api_version]/ldapusers/`

This endpoint represents imported remote LDAP user resource. This can be found in the FortiAuthenticator GUI under **Authentication > Remote Auth. Servers > LDAP**.

## Supported fields

| Field | Display name | Type | Required | Read Only | Other restrictions |
|---|---|---|---|---|---|
| username | Username | string | Yes | Yes | |
| dn | Distinguished name | string | Yes | Yes | |
| server_name | Server name | string | No | Yes | |

| Field | Display name | Type | Required | Read Only | Other restrictions |
|---|---|---|---|---|---|
| server_address | Server address | string | No | Yes | |
| email | E-mail address | string | No | | Must be a valid e-mail address |
| first_name | First name | string | No | | max length = 30 |
| last_name | Last name | string | No | | max length = 30 |
| active | Account Status | boolean | No | | |
| reason | Disable reason | interger | No | | Default is 0. One of 0 (manually disabled), 1 (account inactivity), 2 (too many failed attempts), 3 (account expiry), 4 (password expiry), 5 (FTM activation expiry), 6 (revoked token), 7 (usage limit exceeded), or 8 (pending administrator approval).<br>**Note**: `reason==2` only applies to the PERMANENT user lockout case. |
| mobile_number | Mobile number | string | No | | max length = 25, must follow international number format: +[country_code]-[number] |
| token_auth | Token Auth | boolean | No | | Whether second factor authentication should be enabled. If true, token_type is required. |
| token_type | Token Type | string | No | | One of ftk, ftm, ftc, email, sms, or dual. If email is chosen, email is required. If SMS is chosen, mobile_number is required. |
| token_serial | Token Serial | string | No | | If token_type is ftm, or ftk, and this is not present or blank, the next available token will be assigned. |

| Field | Display name | Type | Required | Read Only | Other restrictions |
|---|---|---|---|---|---|
| ftm_act_method | FTM Activation Delivery Method | string | No | | One of email or sms. If email is chosen, email is required. If SMS is chosen, mobile_number is required. Both are required if dual is selected. |
| password | Password | string | No | Patching this attribute will effectively reset the password. | max length = 50 |
| recovery_by_question | Allow password recovery with security question | boolean | No | | |
| recovery_question | Password recovery security question | string | No | Required if recovery_by_question is true. | |
| recovery_answer | Password recovery security answer | string | Yes | Required if recovery_by_question is true. | |
| fido | FIDO | boolean | No | No | Default is disabled. |
| company | Company | string | No | | |
| department | Department | string | No | | |
| is_locked | Administrative account lock | boolean | No | | |

# Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/ldapusers/ | Get all non-admin LDAP users. |
| GET | /api/v1/ldapusers/[id]/ | Get a specific non-admin LDAP user. |
| POST | /api/v1/ldapusers/[id]/sendoobtoken/ | Send an out-of-band token code (email/SMS token) to an LDAP user. |
| POST | /api/v1/ldapusers/[id]/verifyrecoveryanswer/ | Verify the recovery answer for a specific LDAP user. Note: recovery_answer must be |

| HTTP method | Resource URI | Action |
|---|---|---|
| | | included. |
| PATCH | /api/v1/ldapusers/[id]/ | Update specified fields for a specific LDAP user with ID. |
| DELETE | /api/v1/ldapusers/[id]/ | Delete an LDAP user. |

# Allowed filters

| Field | Lookup expressions | Values |
|---|---|---|
| username | `exact, iexact, contains, icontains, in` | |
| dn | `exact, iexact, contains, icontains` | |
| first_name | `exact, iexact, contains, icontains, in` | |
| last_name | `exact, iexact, contains, icontains, in` | |
| email | `exact, iexact, contains, icontains, in` | |
| active | `exact` | |
| server_name | `exact, iexact, contains, icontains` | |
| server_address | `exact, iexact, contains, icontains` | |
| token_type | | ftk, ftm, ftc, email, sms |
| token_serial | `exact, iexact` | |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# Third-party integration: FTM provisioning

For integration with a third-party authentication server which needs to manage token validation, it is possible for the FortiAuthenticator to return FTM seed during provisioning. However, certain conditions must be met:

- Seed may only be returned when provisioning an FTM to an existing user via PATCH method.
- A GET URL parameter (returnseed=1) needs to be specified to explicitly tell FortiAuthenticator to return an encrypted seed for the token (e.g. https://[server_name]/api/v1/ldapusers/2/?returnseed=1).
- A seed encryption passphrase must be specified in FortiGuard settings.

The seed is encrypted and returned as a PSKC XML file string according to RFC 6030. The key is derived from the configured passphrase using the PBKDF2 key derivation function (32 byte key length, 1000 iterations), encrypted with AES 256 CBC encryption, and signed with a SHA256 HMAC.

Whenever an FTM is provisioned, its activation code will be returned as well.

# RADIUS users (/radiusers/)

URL: `https://[server_name]/api/v1/radiusers/`

This endpoint represents imported remote RADIUS user resource.

## Supported fields

| Field | Display name | Type | Required | Read Only | Other restrictions |
|---|---|---|---|---|---|
| username | Username | string | Yes | Yes | |
| server_name | Server name | string | Yes, if creating user | Yes | |
| server_address | Server address | string | Yes, if creating user | Yes | |
| email | E-mail address | string | No | | Must be a valid e-mail address |
| active | Account Status | boolean | No | | |
| reason | Disable reason | interger | No | | Default is 0. One of 0 (manually disabled), 1 (account inactivity), 2 (too many failed attempts), 3 (account expiry), 4 (password expiry), 5 (FTM activation expiry), 6 (revoked token), 7 (usage limit exceeded), or 8 (pending administrator approval). |
| mobile_number | Mobile number | string | No | | max length = 25, must follow international number format: +[country_code]-[number] |
| token_auth | Token Auth | boolean | No | | Whether second factor authentication should be enabled. If true, token_type is required. |

| Field | Display name | Type | Required | Read Only | Other restrictions |
|---|---|---|---|---|---|
| token_type | Token Type | string | No | | One of ftk, ftm, ftc, email, sms, or dual. If email is chosen, email is required. If SMS is chosen, mobile_number is required. Both are required if dual is selected. |
| token_serial | Token Serial | string | No | | If token_type is ftm, or ftk, and this is not present or blank, the next available token will be assigned. |
| ftm_act_method | FTM Activation Delivery Method | string | No | | One of email or sms. If email is chosen, email is required. If SMS is chosen, mobile_number is required. |
| fido | FIDO | boolean | No | No | Default is disabled. |
| company | Company | string | No | | |
| department | Department | string | No | | |
| is_locked | Administrative account lock | boolean | No | | |

# Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/radiususers/ | Get all non-admin RADIUS users |
| GET | /api/v1/radiususers/[id]/ | Get a specific non-admin RADIUS user |
| POST | /api/v1/radiususers/[id]/sendoobtoken/ | Create a new RADIUS user |
| POST | /api/v1/radiususers/[id]/sendoobtoken/ | Send an out-of-band token code (email/SMS token) to a RADIUS user |
| PATCH | /api/v1/radiususers/[id]/ | Update specified fields for a specific RADIUS user with ID id |
| DELETE | /api/v1/radiususers/[id]/ | Delete a RADIUS user |

## Allowed filters

| Field | Lookup expressions | Values |
|---|---|---|
| username | `exact, iexact, contains, icontains, in` | |
| email | `exact, iexact, contains, icontains, in` | |
| active | `exact` | |
| server_name | `exact, iexact, contains, icontains` | |
| server_address | `exact, iexact, contains, icontains` | |
| token_type | | ftk, ftm, ftc, email, sms |
| token_serial | `exact, iexact` | |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

## Third-party integration: FTM provisioning

This resource allows for FTM provisioning in the same manner specified above for remote LDAP users.

# Local user group memberships (/localgroup-memberships/)

URL:`https://[server_name]/api/[api_version]/localgroup-memberships/`

This endpoint represents local user group membership resource (relationship between local user and local user group).

## Supported fields

| Field | Description | Type | Required | Read-only | Other restrictions |
|---|---|---|---|---|---|
| group | Group | string | Yes | | Local user group URI |
| user | Member of the group | string | Yes | | Local user URI |
| group_name | Member of the group | string | No | Yes | |

| Field | Description | Type | Required | Read-only | Other restrictions |
|---|---|---|---|---|---|
| username | Member username | string | No | Yes | |

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/localgroup-memberships/ | Get all local group memberships |
| GET | /api/v1/localgroup-memberships/[id]/ | Get a specific local group membership |
| POST | /api/v1/localgroup-memberships/ | Create a new local group membership |
| DELETE | /api/v1/localgroup-memberships/[id] | Delete a local group membership |

## Allowed filters

| Field | Filters | Description |
|---|---|---|
| group | exact, in | Accepts group ID (e.g. group=15) |
| user | exact, in | Accepts user ID |
| group_name | exact, iexact, contains, icontains, in | |
| username | exact, iexact, contains, icontains, in | |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# SNMP general setting (/snmpgeneral/)

URL : `https://[server_name]/api/v1/snmpgeneral/`

This end-point is used to set the general SNMP settings and trap thresholds.

## Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| contact | SNMP contact | String | No | |

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| description | SNMP description | String | No | |
| location | SNMP location | String | No | |
| users | Provisioned users nearly full threshold | Integer | No | Percentage (0-100) |
| groups | Provisioned usergroups nearly full threshold | Integer | No | Percentage (0-100) |
| radius_clients | Provisioned RADIUS clients nearly full threshold | Integer | No | Percentage (0-100) |
| tacplus_clients | Provisioned TACACS+ clients nearly full threshold | Integer | No | Percentage (0-100) |
| auth_events | High authentication events rate threshold | Integer | No | |
| auth_failures | High authentication failure rate threshold | Integer | No | |
| cpu | High CPU utilization threshold | Integer | No | Percentage (0-100) |
| memory | High memory utilization threshold | Integer | No | Percentage (0-100) |
| disk | High disk utilization threshold | Integer | No | Percentage (0-100) |

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/snmpgeneral/ | Retrieve SNMP general settings. |
| PATCH | /api/v1/snmpgeneral/ | Update SNMP general settings. |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# SNMP communities (/snmp/)

URL : `https://[server_name]/api/v1/snmp/`

This endpoint is used to configure SNMP v1/v2 communities.

# Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| name | SNMP community name | String | Yes | |
| cpu | Enable high CPU usage trap | Boolean | No | Default is disabled |
| memory | Enable high memory usage trap | Boolean | No | Default is disabled |
| disk | Enable high disk usage trap | Boolean | No | Default is disabled |
| interface_ip | Enable interface IP change trap | Boolean | No | Default is disabled |
| users | Enable provisioned users exceed threshold trap | Boolean | No | Default is disabled |
| groups | Enable provisioned usergroups exceed threshold trap | Boolean | No | Default is disabled |
| radius_clients | Enable provisioned RADIUS clients exceed threshold trap | Boolean | No | Default is disabled |
| tacplus_clients | Enable provisioned TACACS+ clients exceed threshold trap | Boolean | No | Default is disabled |
| auth_events | Enable high authentication events rate trap | Boolean | No | Default is disabled |
| auth_failures | Enable high authentication failures rate trap | Boolean | No | Default is disabled |
| user_lockout | Enable user lockout trap | Boolean | No | Default is disabled |
| ha_status | Enable HA status change trap | Boolean | No | Default is disabled |
| ha_sync | Enable low HA synchronization activity trap | Boolean | No | Default is disabled |
| raid | Enable RAID status change trap | Boolean | No | Default is disabled |

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/snmp/ | Retrieve all SNMP communities. |
| GET | /api/v1/snmp/[id]/ | Retrieve SNMP community with ID id. |
| POST | /api/v1/snmp/ | Create new SNMP community. |
| PATCH | /api/v1/snmp/[id]/ | Update SNMP community with ID id. |
| DELETE | /api/v1/snmp/[id]/ | Delete SNMP community with ID id. |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# SNMP community hosts (/snmp/[id]/hosts/)

URL : `https://[server_name]/api/v1/snmp/[id]/hosts/`

This endpoint is used to configure the allowed hosts of SNMP v1/v2 communities.

## Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| address | IP or subnet of the allowed host(s) | String | Yes | |
| query | Allow host(s) to query the SNMP community | Boolean | No | Default is disabled. |
| trap | Enable sending traps to the host | Boolean | No | Only available to single IP host (always disabled for subnets). |

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/snmp/[id]/hosts/ | Retrieve all hosts for SNMP community with ID id. |

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/snmp/[id]/hosts/[hid]/ | Retrieve host with ID hid for SNMP community with ID id. |
| POST | /api/v1/snmp/[id]/hosts/ | Create new host for SNMP community with ID id. |
| PATCH | /api/v1/snmp/[id]/hosts/[hid]/ | Update host with ID hid for SNMP community with ID id. |
| DELETE | /api/v1/snmp/[id]/hosts/[hid]/ | Delete host with ID hid for SNMP community with ID id. |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# SSO/Remote groups (/ssogroup/)

URL: `https://[server_name]/api/[api_version]/ssogroup/`

This can be found in the FortiAuthenticator GUI under **Fortinet SSO Methods > SSO > SSO Groups**.

## Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| name | Name | string | Yes | max length=50, unique |

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/ssogroup/ | Get all SSO groups |
| GET | /api/v1/ssogroup/[id]/ | Get an SSO group with ID id |
| POST | /api/v1/ssogroup/ | Create a new SSO group |
| DELETE | /api/v1/ssogroup/ | Delete all SSO groups |
| DELETE | /api/v1/ssogroup/[id]/ | Delete an SSO group with ID id |

# Allowed filters

| Field | Lookup expressions |
|---|---|
| name | exact, in |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# View SSO group configuration

### JSON query

- JSON specified via GET

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS"
      https://192.168.0.122/api/v1/ssogroup/?format=json
```

- JSON specified via Accept Header

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS" -H 'Accept: application/json'
      https://192.168.0.122/api/v1/ssogroup/
```

### Response

```
< HTTP/1.1 200 OK
< Date: Tue, 10 Jun 2014 11:48:08 GMT
< Server: Apache
< Vary: Accept,Accept-Language,Cookie
< X-Frame-Options: SAMEORIGIN
< Content-Language: en
< Cache-Control: no-cache
< Transfer-Encoding: chunked
< Content-Type: application/json
<
* Connection #0 to host 192.168.0.122 left intact
* Closing connection #0
{"meta": {"limit": 20, "next": null, "offset": 0, "previous": null, "total_count": 1}, "objects":
      [{"id": 1, "name": "Test_Group1", "resource_uri": "/api/v1/ssogroup/1/"}]}{"meta": {"limit":
      20, "next": null, "offset": 0, "previous": null, "total_count": 1}, "objects": [{"id": 1,
      "name": "Test_Group1", "resource_uri": "/api/v1/ssogroup/1/"}]}
```

### JSON query

- JSON specified via GET

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS"
      https://192.168.0.122/api/v1/ssogroup/?format=json
```

- JSON specified via Accept Header

```
curl -k -v -u "zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS" -H 'Accept: application/json'
      https://192.168.0.122/api/v1/ssogroup/
```

### Response

```
< HTTP/1.1 200 OK
< Date: Tue, 10 Jun 2014 11:48:08 GMT
< Server: Apache
< Vary: Accept,Accept-Language,Cookie
< X-Frame-Options: SAMEORIGIN
< Content-Language: en
< Cache-Control: no-cache
< Transfer-Encoding: chunked
< Content-Type: application/json
<
* Connection #0 to host 192.168.0.122 left intact
* Closing connection #0
{"meta": {"limit": 20, "next": null, "offset": 0, "previous": null, "total_count": 1}, "objects":
      [{"id": 1, "name": "Test_Group1", "resource_uri": "/api/v1/ssogroup/1/"}]}{"meta": {"limit":
      20, "next": null, "offset": 0, "previous": null, "total_count": 1}, "objects": [{"id": 1,
      "name": "Test_Group1", "resource_uri": "/api/v1/ssogroup/1/"}]}
```

# Create SSO group

### JSON query

- JSON specified via POST

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS" -X POST -d '{"name":"Test_Group2"}' -
      H 'Content-Type: application/json' https://192.168.0.122/api/v1/ssogroup/
```

### XML query

- JSON specified via Accept Header

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS" -X POST -d '<object><name>Test_
      Group2</name></object>' -H 'Content-Type: application/xml'
      https://192.168.0.122/api/v1/ssogroup/
```

### Response

```
< HTTP/1.1 201 CREATED
< Date: Tue, 10 Jun 2014 11:51:31 GMT
< Server: Apache
< Vary: Accept,Accept-Language,Cookie
< X-Frame-Options: SAMEORIGIN
< Content-Language: en
< Location: https://192.168.0.122/api/v1/ssogroup/3/
< Content-Length: 0
< Content-Type: text/html; charset=utf-8
```

Successful 201 CREATED response code. See General API response codes on page 138.

# Filter lookup expressions

| Expression | Description |
|---|---|
| `exact` | search for an exact match (e.g. name__exact=John Doe, would return user with name "John Doe", but not "john doe") |
| `iexact` | search for a case-insensitive exact match (e.g. name__iexact=john doe, would return user with name "John Doe") |
| `contains` | search for an item that contains a specific keyword |
| `icontains` | same as above, but case-insensitive |
| `in` | search for items that matches specific filter criteria (e.g. to return items that has a name matching "John" or "Bill", ?name__in=John&name__in=Bill) |
| `startswith` | search for items that starts with a text |
| `istartswith` | same as above, but case-insensitive |

See General API response codes on page 138 for full details.

# Delete SSO group

**Query**

- Specified via POST

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS" -X DELETE
     https://192.168.0.122/api/v1/ssogroup/3/
```

**Response**

```
< HTTP/1.1 204 NO CONTENT
< Date: Tue, 10 Jun 2014 11:53:52 GMT
< Server: Apache
< Vary: Accept,Accept-Language,Cookie
< X-Frame-Options: SAMEORIGIN
< Content-Language: en
< Content-Length: 0
< Content-Type: text/html; charset=utf-8
```

204 NO CONTENT is a successful result. Verify by querying /ssogroup/ to verify group 3 has been deleted.

# FortiGate group filter (/fgtgroupfilter/)

**URL:** `https://[server_name]/api/[api_version]/fgtgroupfilter/`

This can be found in the FortiAuthenticator GUI under **Fortinet SSO Methods > SSO > FortiGate Filtering**.

# Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| shortname | Name | string | Yes | max length=32, unique |
| nasname | NAS name/IP | string | Yes | max length=128, unique |

# Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/fgtgroupfilter/ | Get all FortiGate Group Filters. |
| GET | /api/v1/fgtgroupfilter/[id]/ | Get a specific FortiGate Group Filter with ID `id`. |
| PUT | /api/v1/fgtgroupfilter/[id]/ | Update an existing FortiGate Group Filter specified with ID `id`. |

# Allowed filters

| Field | Filters |
|---|---|
| shortname | `exact, iexact, contains, icontains, in` |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# View FortiGate group filter configuration

### JSON query

- JSON specified via GET

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS"
     https://192.168.0.122/api/v1/fgtgroupfilter/?format=json
```

- JSON specified via Accept Header

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS" -H 'Accept: application/json'
     https://192.168.0.122/api/v1/fgtgroupfilter/
```

### Response

```
< HTTP/1.1 200 OK
< Date: Tue, 10 Jun 2014 13:49:24 GMT
< Server: Apache
< Vary: Accept,Accept-Language,Cookie
```

```
< X-Frame-Options: SAMEORIGIN
< Content-Language: en
< Cache-Control: no-cache
< Transfer-Encoding: chunked
< Content-Type: application/json
<
* Connection #0 to host 192.168.0.122 left intact
* Closing connection #0
{"meta": {"limit": 20, "next": null, "offset": 0, "previous": null, "total_count": 1}, "objects":
      [{"address": "1.1.1.1", "id": 1, "name": "GroupFilter_Test1", "nasname": "1.1.1.1", "resource_
      uri": "/api/v1/fgtgroupfilter/1/", "shortname": "GroupFilter_Test1", "sso_groups": [
```

# Add FortiGate group filter configuration

Note that POST is not an allowed method so FGTGroup filters cannot be created via the API, however once created via the GUI, they can be modified. See below.

# Modify FortiGate group filter configuration

**JSON query**

- JSON specified via Accept Header

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS" -X PUT -d '{"shortname":"GroupFilter_
      Test1","nasname":"2.2.2.2", "sso_groups": []}' -H 'Content-Type: application/json'
      https://192.168.0.122/api/v1/fgtgroupfilter/1/
```

**Response**

```
< HTTP/1.1 204 NO CONTENT
< Date: Mon, 16 Jun 2014 16:35:16 GMT
< Server: Apache
< Vary: Accept,Accept-Language,Cookie
< X-Frame-Options: SAMEORIGIN
< Content-Language: en
< Content-Length: 0
< Content-Type: text/html; charset=utf-8
```

# SSO authentication (/ssoauth/)

URL: `https://[server_name]/api/[api_version]/ssoauth/`

This endpoint represents the Fortinet SSO Authentication. This resource can be found in the FortiAuthenticator GUI under **Fortinet SSO Methods > SSO**. This API is for use by third-party authentication systems for dynamic transparent user Single Sign-on to a Fortinet protected network.

Before attempting to authenticate, additional configuration is required under **Fortinet SSO Methods > Portal Services > SSO Web Service** to select which user directory is to be used for group embellishment.

# Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| event | Event type | integer/string | Yes | 1=Logon<br>0=Logoff |
| username | User's username | string | Yes | max length=253 |
| user_ip | User's workstation IP (Calling-Station-Id) | IPv4 | Yes | |
| user_ipv6 | User's workstation IPv6 (Calling-Station-Id) | IPv6 | No | One of 'user_ip' or 'user_ipv6' is required |
| user_groups | Groups to send (Fortinet-Group-Name) | string | No | max length=253, list of groups must be separated with "+" character (group name cannot contain a "+" character) |

For local users, the user must be part of a local group for successful SSO logon.
External users must have a group passed in via the user_groups field for logon/logoff.

# Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| POST | api/v1/ssoauth/ | Logon/logoff users to/from FSSO |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# Response codes

In addition to the general codes defined in General API response codes on page 138, a POST request to this resource can result in the following return codes:

| Code | Response content | Description |
|---|---|---|
| **200 OK** | | FSSO login/logout request has been successfully sent to FSSO (but this doesn't mean that user has been logged-on/off, as the request is done asynchronously and is queued on FSSO side. Factors such as configuration and user not rxisting in LDAP may cause the entry to not populate FSSO). |
| **404 Not Found** | SSO web service is disabled | SSO web service has not been enabled so it can't be used in REST API |
| **500 Internal Server Error** | | Failed to send logon/logoff request to FSSO |

# FSSO user login

### JSON query

- JSON specified via Accept Header

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS" -d '
      {"event":"1","username":"cwindsor","user_ip":"10.1.73.175"}' -H "Content-Type:
      application/json" https://192.168.0.122/api/v1/ssoauth/
```

### Response

```
< HTTP/1.1 200 OK
< Date: Fri, 20 Sep 2013 08:27:27 GMT
< Server: Apache
< Vary: Accept,Accept-Language,Cookie
< Content-Language: en
< Set-Cookie: sessionid=6q6m6ne4v7p76qclajitlf2q7202f7g6; httponly; Path=/
< Content-Length: 0
< Content-Type: text/html; charset=utf-8
<
* Connection #0 to host 192.168.0.122 left intact
* Closing connection #0
```

Verify login on FortiAuthenticator from Monitor > SSO > SSO Sessions.

# Overwrite FSSO user login with different user

Note that if a login event is received with the same IP address but with a different username, the existing entry will be overwritten.

### JSON query

- JSON specified via Accept Header

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS" -d '
      {"event":"1","username":"atano","user_ip":"10.1.73.175"}' -H "Content-Type: application/json"
      https://192.168.0.122/api/v1/ssoauth/
```

### Response

```
< HTTP/1.1 200 OK
< Date: Fri, 20 Sep 2013 08:32:21 GMT
< Server: Apache
< Vary: Accept,Accept-Language,Cookie
< Content-Language: en
< Set-Cookie: sessionid=g062qqmsj6nr0hk5khd2q7202e4v36m; httponly; Path=/
< Content-Length: 0
< Content-Type: text/html; charset=utf-8
<
* Connection #0 to host 192.168.0.122 left intact
* Closing connection #0
```

Verify login on FortiAuthenticator from Monitor > SSO > SSO Sessions.

# Logout FSSO user

### JSON query

- JSON specified via Accept Header

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS" -d '
      {"event":"0","username":"atano","user_ip":"10.1.73.175"}' -H "Content-Type: application/json"
      https://192.168.0.122/api/v1/ssoauth/
```

### Response

```
< HTTP/1.1 200 OK
< Date: Fri, 20 Sep 2013 08:34:09 GMT
< Server: Apache
< Vary: Accept,Accept-Language,Cookie
< Content-Language: en
< Set-Cookie: sessionid=2q de4v36msj6g05khm6nr02q72q02hk; httponly; Path=/
< Content-Length: 0
< Content-Type: text/html; charset=utf-8
* Connection #0 to host 192.168.0.122 left intact
* Closing connection #0
```

Verify logout on FortiAuthenticator from Monitor > SSO > SSO Sessions.

# Logging

Note that SSO Login requests are logged regardless of whether the user details can be inserted into FSSO. For example logs may exist for SSO Logon for a user but an entry not appear in the monitor because when an LDAP lookup for group info was performed, no user existed.

# SSO filtering objects (/fgtgroupfilter/[id]/ssofilterobjects/)

URL:`https://[server_name]/api/v1/fgtgroupfilter/[id]/ssofilterobjects/`

This resource can only be used alongside the FortiGate filter resource above.

# Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| name | Object name / DN | string | Yes | max length=255, unique for each FortiGate filter |
| obj_type | Object Type | string | Yes | One of user, group (default), user container, group container, user and group container |

# Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/fgtgroupfilter/[id]/ssofilterobjects/ | Get all SSO filtering objects for a specific FortiGate filter. |
| GET | /api/v1/fgtgroupfilter/[id]/ssofilterobjects/[filter_id]/ | Get an SSO filtering object for a specific FortiGate filter. |
| POST | /api/v1/fgtgroupfilter/[id]/ssofilterobjects/ | Create a new SSO filtering object for a specific FortiGate filter. |
| PUT | /api/v1/fgtgroupfilter/[id]/ssofilterobjects/ | Update all SSO filtering objects that belongs to a FortiGate filter. |
| PATCH | /api/v1/fgtgroupfilter/[id]/ssofilterobjects/[filter_id]/ | Update fields of an SSO filtering object. |
| DELETE | /api/v1/fgtgroupfilter/[id]/ssofilterobjects/ | Delete all SSO filtering objects from a specific FortiGate filter. |
| DELETE | /api/v1/fgtgroupfilter/[id]/ssofilterobjects/[filter_id]/ | Delete an SSO filtering object. |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

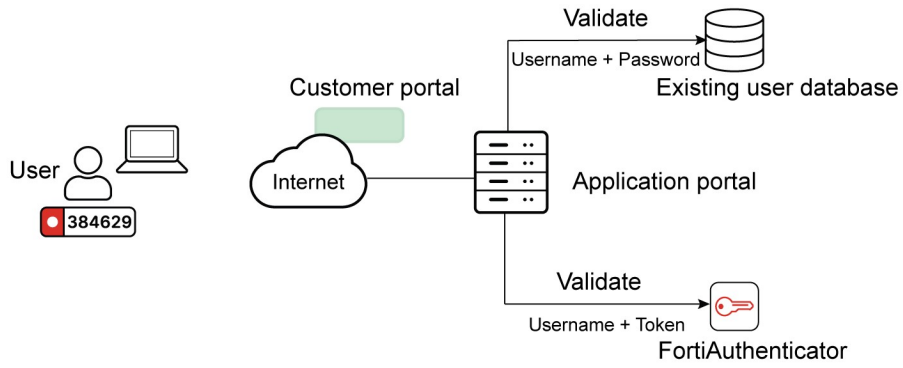
# Authentication (/auth/)

URL: `https://[server_name]/api/[api_version]/auth/`

This authentication API is for validation of user credentials. Either the password, token or both can be validated. This is useful for adding an additional factor authentication (e.g. token) to web portals where the first factor as already being validated locally e.g. via LDAP and RADIUS user credentials, or local DB or a proprietary, unsupported authentication method as is common in the banking industry.

This API is for the validation of local user password and token passcode or remote user passcode only. Validation of remote (LDAP) user password is not supported. This is by design as most systems have an established mechanism for authentication via e.g. LDAP or some other proprietary mechanism as shown below.

User lockout policies can be configured under **Authentication > User Account Policies > Lockouts**. The policies will be applied as configured.




To authenticate a user, you need to POST to `https://[server_name]/api/1/auth/` with the following key-value pair (in JSON format, but XML also possible):

```
{"username": "<username>", "token_code": "<token_code>", "password": "<password>"}
```

with `"token_code"` and `"password"` being optional fields i.e. you can just validate the token only or the password only. If password and token are specified, the password will be validated first before token code.

# Behavior of the API

- Either `password` or `token_code` needs to be specified.
- If both are specified, `password` will be validated first, then `token_code`.

- If both are specified, it is acceptable to concatenate both the user's password and token code in as the password value and provide an empty string as the `token_code` value.
- If only one is specified (either `password`or `token_code`), only that credential will be validated.
- If a user doesn't have two-factor authentication configured, validation for that user with any `token_code` will fail.
- If a user is configured with only FortiToken authentication (password-based authentication is disabled), specifying any password will fail.

Before being able to validate an email token or SMS token, a token code needs to be triggered and sent to the user.

Please refer to either Local users (/localusers/), LDAP users (/ldapusers/) or RADIUS users (/radiusers/) documentation on how to send the token code.

# Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| username | Username | string | Yes | |
| password | Password | string | No | |
| token_code | Security token code | string | No | Supported token authentication: FortiToken, FortiToken Cloud, email token, SMS token. |
| user_ip | User IP address | string | No | |

# Allowed methods

| Type | Allowed methods | Action |
|---|---|---|
| List | POST | Validate user's credentials. |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# Response codes

In addition to the general codes defined in General API response codes on page 138, a POST request to this resource can result in the following return codes:

| Code | Response content | Description |
|---|---|---|
| 200 OK | | User is successfully authenticated. |
| 401 Unauthorized | User authentication failed | Credential is incorrect. |

| Code | Response content | Description |
|---|---|---|
| 401 Unauthorized | Account is disabled | User account is currently disabled. |
| 401 Unauthorized | No token configured | User does not have token-based authentication configured. |
| 401 Unauthorized | Token is out of sync | The security token requires synchronization. |
| 404 Not Found | User does not exist | The given username does not exist in the system. |

# Validate a user password

### Query

- JSON specified via Accept Header

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS" -d '
      {"username":"testuser","password":"testpass"}' -H "Content-Type: application/json"
      https://192.168.0.122/api/v1/auth/
```

### Response

```
< HTTP/1.1 200 OK
< Date: Fri, 14 Sep 2012 15:38:57 GMT
< Server: Apache
< Vary: Cookie
< Set-Cookie: sessionid=6b17c5bbb86419a94f6979a05bd84139; httponly; Path=/
< Content-Length: 0
< Content-Type: text/html; charset=utf-8
```

# Validate a users token code

### Query

- JSON specified via Content-Type Header

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS" -d '{"username":"testuser","token_
      code":"893753"}' -H "Content-Type: application/json" https://192.168.0.122/api/v1/auth/
```

### Response

```
< HTTP/1.1 200 OK
< Date: Fri, 14 Sep 2012 15:47:22 GMT
< Server: Apache
< Vary: Cookie
< Set-Cookie: sessionid=f15beeab159a4bf2d0402a05db40d6ae; httponly; Path=/
< Content-Length: 0
< Content-Type: text/html; charset=utf-8
```

# Error states

**Response (incorrect password)**

```
HTTP/1.1 401 UNAUTHORIZED
Date: Thu, 13 Sep 2012 13:57:24 GMT
Server: Apache
Vary: Cookie
Set-Cookie: sessionid=abe8bac6fc50caf5eadf1e57f0c60e3e; httponly; Path=/
Content-Length: 26
Content-Type: text/html; charset=utf-8
```

**Response (incorrect token code)**

```
HTTP/1.1 401 UNAUTHORIZED
Date: Thu, 13 Sep 2012 13:55:18 GMT
Server: Apache
Vary: Cookie
Set-Cookie: sessionid=e95090804ee0e3b8903618138b38a5c8; httponly; Path=/
Content-Length: 26
Content-Type: text/html; charset=utf-8
```

**Response (incorrect username)**

```
HTTP/1.1 404 NOT FOUND
Date: Thu, 13 Sep 2012 13:58:54 GMT
Server: Apache
Vary: Cookie
Set-Cookie: sessionid=3b353061d9141567c02bb0d057b18284; httponly; Path=/
Content-Length: 19
Content-Type: text/html; charset=utf-8
```

# Realm authentication (/realmauth/)

URL: `https://[server_name]/api/[api_version]/realmauth/`

This end-point is used to validate local, LDAP and RADIUS user credentials based on realm.

User lockout policy can be changed under **Authentication > User Account Policies > Lockouts**. The policy will be applied as configured.

# Behavior of the API

- Either password or `token_code` needs to be specified.
- If both are specified, password will be validated first, then `token_code`.
- If only one is specified (either password or `token_code`), only that credential will be validated.

- If a user doesn't have two-factor authentication configured, validation for that user with any `token_code` will fail.
- If a user is configured with only FortiToken authentication (password-based authentication is disabled), specifying any password will fail.

Before being able to validate an email token or SMS token, a token code needs to be sent to the user first. Please refer to either **/localusers**, **/ldapusers** or **/radiususers** documentation on how to send the token code.

# Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| username | Username | string | Yes | |
| realm | Realm | string | Yes | |
| password | Password | string | No | |
| token_code | Security token code | string | No | Supported token authentication: FortiToken, FortiToken Cloud, email token, SMS token. |
| user_ip | User IP address | string | No | |

# Allowed Methods

| HTTP Method | Resource URI | Action |
|---|---|---|
| POST | /api/v1/realmauth/ | Validate user's credentials. |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# Response codes

In addition to the general codes defined in General API response codes on page 138, a POST request to this resource can result in the following return codes:

| Code | Response content | Description |
|---|---|---|
| 200 OK | | User is successfully authenticated. |
| 202 OK | | User authenticated and password change required. |

| Code | Response content | Description |
|---|---|---|
| 401 Unauthorized | User authentication failed | Credential is incorrect. |
| 401 Unauthorized | Account is disabled | User account is currently disabled. |
| 401 Unauthorized | No token configured | User does not have token-based authentication configured. |
| 401 Unauthorized | Token is out of sync | The security token requires synchronization. |
| 404 Not Found | User does not exist | The given username does not exist in the system. |

# RADIUS clients (/radiusclients/)

URL: `https://[server_name]/api/v1/radiusclients/`

This endpoint represents RADIUS client resource.

## Supported fields

| Field | Display name | Type | Required | Read Only | Other restrictions |
|---|---|---|---|---|---|
| name | RADIUS client name | string | Yes | No | max length = 32 |
| address | RADIUS IP address or subnet | string | Yes | No | max length = 128 |
| secret | RADIUS client secret | string | Yes | No | max length = 63 |
| accounting_usage | Accept RADIUS accounting messages for usage enforcement | boolean | No | No | |
| disconnect | Support RADIUS Disconnect messages | boolean | No | No | |

| Field | Display name | Type | Required | Read Only | Other restrictions |
|---|---|---|---|---|---|
| require_message_authenticator | Require `Message-Authenticator` RADIUS attribute. | boolean | No | No | |

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/radiusclients/ | Get all RADIUS clients. |
| GET | /api/v1/radiusclients/[id]/ | Get a specific RADIUS client with ID. |
| POST | /api/v1/radiusclients/ | Create a new RADIUS client. |
| PUT | /api/v1/radiusclients/[id]/ | Update an existing RADIUS client specified with ID. |
| DELETE | /api/v1/radiusclients/[id]/ | Delete the RADIUS client corresponding to ID. |

## Allowed filters

| Field | Lookup expressions | Values |
|---|---|---|
| name | `exact` | |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# RADIUS policies (/radiuspolicies/)

URL: `https://[server_name]/api/v1/radiuspolicies/`

This endpoint represents RADIUS policy resource.

## Supported fields

| Field | Display name | Type | Required | Read Only | Other restrictions |
|---|---|---|---|---|---|
| name | RADIUS policy name | string | Yes | Yes | |

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/radiuspolicies/ | Get all RADIUS policies. |
| GET | /api/v1/radiuspolicies/[id]/ | Get a specific RADIUS policy with ID. |

## Allowed filters

| Field | Lookup expressions | Values |
|---|---|---|
| name | `exact` | |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# RADIUS Policy/ Client Associations (/radiuspolicyclient/)

URL: `https://[server_name]/api/v1/radiuspolicyclient/`

This endpoint represents a resource for RADIUS policy/ client associations.

## Supported fields

| Field | Display name | Type | Required | Read Only | Other restrictions |
|---|---|---|---|---|---|
| policy | RADIUS policy URI | string | No | No | |

| Field | Display name | Type | Required | Read Only | Other restrictions |
|---|---|---|---|---|---|
| client | RADIUS client URI | string | No | No | |
| policy_name | RADIUS policy name | string | No | No | |
| client_name | RADIUS client name | string | No | No | |

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/radiuspolicyclient/ | Get all RADIUS policy/ client associations. |
| GET | /api/v1/radiuspolicyclient/[id]/ | Get a specific RADIUS policy/ client association with ID. |
| POST | /api/v1/radiuspolicyclient/ | Create a new RADIUS policy/ client association. One of policy or policy_name fields must be specified. One of client or client_name fields must be specified. |
| DELETE | /api/v1/radiuspolicyclient/[id]/ | Delete the RADIUS policy/ client association corresponding to ID. |

## Allowed filters

| Field | Lookup expressions | Values |
|---|---|---|
| policy_name | `exact` | |
| policy_client | `exact` | |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# FortiGuard messaging (/fortiguardmessages/)

URL: `https://[server_name]/api/[api_version]/fortiguardmessages/`

This endpoint is used to query FortiGuard Messaging (SMS) license status including the number of messages in the licenses and the number of messages in total available for use. It's also used to activate a FortiGuard Messaging (SMS) license.

# Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| license | FortiGuard Messaging license | string | Yes | Only valid in POST method. |
| total_sms | Total number of SMS messages in the license | integer | No | Read-only field returned by the GET method. |
| used_sms | Number of SMS messages that have been used | integer | No | Read-only field returned by the GET method. |
| available_sms | Number of SMS messages available for future use | integer | No | Read-only field returned by the GET method. |

# Allowed methods

| HTTP method | Resource URI | Action | Note |
|---|---|---|---|
| GET | /api/v1/fortiguardmessages/ | Get FortiGaurd Messaging statistics. | Returns total_sms, used_sms, and available_sms fields. |
| POST | /api/v1/fortiguardmessages/ | Activate FortiGuard Messaging license. | Requires license field. |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

**Examples**

```
Get fortiguardmessages stats:

curl -k -v \
   -u "webadmin:[hash]" \
   https://[FAC_IP]/api/v1/fortiguardmessages/

Response : {"available_sms": ###, "total_sms": ###, "used_sms": ###}

Activate a license:

curl -k -v -X POST \
   https://[FAC_IP]/api/v1/fortiguardmessages/ \
```

```
    -H 'Content-Type: application/json' \
    -u "webadmin:[hash]" \
    -d '{"license": "####-####-####-####-####"}'

Note: Used valid licenses will return a success response, but will not add SMS
```

# FTM licenses (/fortitokenmobilelicenses/)

URL: `https://[server_name]/api/[api_version]/fortitokenmobilelicenses/`

This endpoint is used to query FTM token license status inlcuding the number of FTM tokens in the licenses and the total number of available tokens for use. It's also used to activate an FTM token license.

## Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| license | FortiToken Mobile license | string | Yes | Only valid in POST method. |
| total_ftm | Total number of FortiToken Mobiles on the FortiAuthenticator | integer | No | Read-only field returned by the GET method. |
| used_ftm | Number of FortiToken Mobiles assigned to users | integer | No | Read-only field returned by the GET method. |
| available_ftm | Number of FortiToken Mobiles available for future assignments | integer | No | Read-only field returned by the GET method. |

## Allowed methods

| HTTP method | Resource URI | Action | Notes |
|---|---|---|---|
| GET | /api/v1/fortitokenmobilelicenses/ | Get licensed FortiToken Mobiles statistics. | Returns total_ftm, used_ftm, and available_ftm fields. |

| HTTP method | Resource URI | Action | Notes |
|---|---|---|---|
| POST | /api/v1/fortitokenmobilelicenses/ | Activate a FortiToken Mobile license. | Requires license field. |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

**Examples**

```
curl -k -v \-u "webadmin:[hash]" \ https://[FAC_IP]/api/v1/fortitokenmobilelicenses/

Response : {"available_ftm": 30, "total_ftm": 32, "used_ftm": 2}

curl -k -v -X POST \
   https://[FAC_IP]/api/v1/fortitokenmobilelicenses/ \
   -H 'Content-Type: application/json' \
   -u "webadmin:[hash]" \
   -d '{"license": "####-####-####-####-####"}'

Response : {
"license": "####-####-####-####-####",
"messages": {
      "success": "Successfully imported 10 FortiTokens"
   }
}
```

# Email servers (/smtpservers/)

**URL:** `https://[server_name]/api/[api_version]/smtpservers/`

This endpoint is used to set up email servers and senders.

## Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| name | SMTP server name | string | Yes | Must be unique. |
| address | SMTP server IP address or SMTP server | string | Yes | Must be unique. |
| port | SMTP server port | integer | No | Default is 25 when not specified. |

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| sender_name | Sender name for email "from" field | string | No | |
| sender_email | Sender email for email "from" field | string | Yes | |
| secure | Secured communication method | string | No | Either "none" for no encryption or "starttls" for STARTTLS encryption. Default is "none" when not specified. |
| authentication | Use authentication | boolean | No | Default is disabled when not specified. When disabled, authentication name and password are set to None. |
| authentication_name | Authentication username | string | No | Required if "authentication" is enabled. |
| authentication_password | Authentication password | string | No | Required if "authentication" is enabled. |
| default | Default SMTP server | boolean | No | Default is "disable". Only one SMTP server can be the default. Setting to "enable" for an SMTP server will cause this server to be the default. Setting to false will cause the server with ID 1 to become the default. |

# Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/smtpservers/ | Get all SMTP servers. |
| GET | /api/v1/smtpservers/[id]/ | Get a specific SMTP server with ID. |
| POST | /api/v1/smtpservers/ | Create a new SMTP server. |
| PATCH | /api/v1/smtpservers/[id]/ | Update specified fields of an existing SMTP server with ID. |
| DELETE | /api/v1/smtpservers/[id]/ | Delete an SMTP server. |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

**Examples**

```
Get all servers: curl -k -v \ -u "webadmin:[hash]" \ https://[FAC_IP]/api/v1/smtpservers/

Post a server:

curl -k -X POST \
   https://[FAC_IP]/api/v1/smtpservers/ \
   -H 'Content-Type: application/json' \
   -u "webadmin"[hash]" \
   -d '{
      "address": "mail.server-test.com",
      "authentication" true,
      "authentication_name": "username",
      "authentication_password": "supersecretpassword",
      "default": true,
      "name": "fortitest25",
      "secure": "starttls",
      "sender_email": "email@fortinet.com",
      "sender_name": "email_sender name",
}'
```

# User lockout policy (/userlockoutpolicy/)

**URL:** `https://[server_name]/api/[api_version]/userlockoutpolicy/`

This endpoint is used to query and edit user account lockout policy settings including the maximum number of failed login attempts, specify the lockout period, and enable inactive user lockouts.

## Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| failed_login_lockout | Lockout user accounts after too many failed login attempts. | boolean | Yes | Either set to "true" or "false", enabling or disabling the login lockout (respectively). |
| failed_login_lockout_max_attempts | Maximum number of failed login attempts allowed before locking out the user account. | integer | No | Default is set to 3 if not specified. Must be set between 1-20. |
| failed_login_lockout_permanent | Permanency of user account lockout after too many failed login attempts. | boolean | No | Default is "false" if not specified. |

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| | | | | Set to "true" to permanently lockout the user account. Set to "false" to only lockout the user account for a period of time. When set to "true", then later changed to "false", the lockout period is set to its default. |
| failed_login_lockout_period | Period of time (in seconds) the user account is lockout after reaching the maximum number of failed login attempts. | integer | No | Default is 60 if not specified. Must be set between 60-86400. Only effective when "failed_login_lockout_permanent" is set to "false". |
| inactivity_lockout | Lockout user accounts that inactive for a specified period of time. | boolean | No | Default is "false" if not specified.<br>Set to "true" to disable when inactive for the time period specified by "inactivty_lockout_period". Set to "false" to never disable user accounts for inactivity. |
| inactivity_lockout_period | Inactivity period (in days) after which a user account is locked out. | integer | No | Default is 90 if not specified. Must be set between 1-1825. Only effective when "inactivty_lockout" is set to "true". |

# Allowed methods

| HTTP method | Resource URI | Action | Note |
|---|---|---|---|
| GET | /api/v1/userlockoutpolicy/ | Get user lockout settings. | |
| POST | /api/v1/userlockoutpolicy/ | Set user lockout fields. | Defaults are used if unspecified. |
| PATCH | /api/v1/userlockoutpolicy/ | Updated the specified user lockout fields. | Previously saved settings are used in unspecified. |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

### Examples

```
Get userlockout policy:

curl -k -v \
   -u "webadmin:[hash]" \
   https://[FAC_IP]/api/v1/userlockoutpolicy/

Response: {
   "failed_login_lockout": true
   "failed_login_lockout_max_attempts": 5,
   "failed_login_lockout_period": 60,
   "failed_login_lockout_permanent": false,
   "inactivity_lockout": true,
   "inactivity_lockout_period": 1600
}

Patch a server:

curl -k -X PATCH \
   https://[FAC_IP]/api/v1/userlockoutpolicy/ \
   -H 'Content-Type: application/json' \
   -u "webadmin"[hash]" \
   -d '{
      "failed_login_lockout_permanent": true
   }'

Response: {
   "failed_login_lockout": true
   "failed_login_lockout_max_attempts": 5,
   "failed_login_lockout_period": 0,
   "failed_login_lockout_permanent": true,
   "inactivity_lockout": true,
   "inactivity_lockout_period": 1600
}
```

# System Information (/systeminfo/)

URL: `https://[server_name]/api/v1/systeminfo/`

This REST API queries the FortiAuthenticator system information.

# Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| sn | Serial number | string | Yes | Read-only field returned by the GET method. |
| ha_sn | Serial number of other member in redundant HA cluster pair | string | No | Read-only field returned by the GET method. Only available in HA cluster member mode. |
| firmware | Current firmware version | string | Yes | Read-only field returned by the GET method. e.g. "FACVM v6.4.0-build0888 (GA)" |
| cpu | Current CPU usage percentage (0-100) | interger | Yes | Read-only field returned by the GET method. |
| memory | Current memory usage percentage (0-100) | interger | Yes | Read-only field returned by the GET method. |
| memory_ usage_detail | Current memory usage details (total, used) in KB | interger | Yes | Read-only field returned by the GET method. |
| disk | Current disk usage percentage (0-100) | interger | Yes | Read-only field returned by the GET method. |
| disk_usage_ detail | Current disk usage details (total, used) in KB | interger | Yes | Read-only field returned by the GET method. |
| users_usage_ detail | Current users usage details (max, used) | integer | Yes | Read-only field returned by the GET method. |
| groups_usage_ detail | Current groups usage details (max, used) | integer | Yes | Read-only field returned by the GET method. |
| ftk_usage_ detail | Current Hardware FortiTokens usage details (populated, used) | integer | Yes | Read-only field returned by the GET method. |

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| ftm_usage_detail | Current FortiTokens Mobile usage details (populated, used) | integer | Yes | Read-only field returned by the GET method. |
| fsso_usage_detail | Current FSSO usage details (max, used) | integer | Yes | Read-only field returned by the GET method. |
| ssoma_usage_detail | Current SSO Mobility Agent usage details (max, used) | integer | Yes | Read-only field returned by the GET method. |

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/systeminfo/ | Get system information |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# Upgrade firmware (/upgrade/)

URL: `https://[server_name]/api/v1/upgrade/`

This endpoint is used to upgrade the FortiAuthenticator firmware or lad a debug kit.

## Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| url | URL where to download the firmware image from (e.g. https://fileserver.fortinet.com/facbuilds/0958/image.out) | string | Yes | |

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| downgrade | downgrade | boolean | No | Must be set to 'true' for a firmware downgrade to be accepted.<br>This acts as a protection against downgrades because of erroneous firmware image file since downgrades will factory reset the configuration.<br>Default is 'false' when not specified. |

# Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| POST | /api/v1/upgrade/ | Upgrade to the firmware in the specified file. |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# Response codes

In addition to the general codes defined in General API response codes on page 138, a POST request to this resource can also result in the following return codes:

| Code | Response content | Description |
|---|---|---|
| 200 OK | | Valid firmware image. Proceeding with the upgrade process. |
| 401 Unauthorized | | Invalid firmware image (firmware image |

| Code | Response content | Description |
|---|---|---|
| | | wrong model, firmware image version is lower than current firmware and "downgrade==false"). |

## Example

```
curl -k -X GET https://[FAC_IP]/api/v1/upgrade/
                  -H 'content-type: multipart/form-data'
                  -u 'admin:[hash]'
                  -F 'url=https://fileserver.fortinet.com/facbuilds/0958/image.out'
                  -F 'downgrade=false'
Response:
200 OK - Firmware upgrade process has been started. FortiAuthenticator will be restarted.
```

# Syslog servers (/syslogservers/)

URL: `https://[server_name]/api/[api_version]/syslogservers/`

This endpoint is used to create, update, edit, and delete syslog servers. This resource can be found in the FortiAuthenticator GUI under **Logging > Log Config > Syslog Servers**.

## Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| name | Syslog server name | string | Yes | |
| address | Syslog server IP address or syslog server name | string | Yes | |
| port | Syslog server port | integer | Yes | Default is set to 514 if not specified. |
| level | Level of logs to record | string | Yes | Default is set to "information" if not specified. Either "emergency", "alert", "critical", "error", "warning", "notice", "information", "debug". |
| facility | Facility or category of logs | string | Yes | Default is set to "user" when not specified. Either "kern", "user", "mail", |

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| | | | | "daemon", "auth", "syslog", "lpr", "news", "uucp", "cron", "authpriv", "ftp", "ntp", "audit", "alert", "clock", "local0", "local1", "local2", "local3", "local4", "local5", "local6", "local7". |

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/syslogservers/ | Get all syslog servers. |
| GET | /api/v1/syslogservers/[id]/ | Get a specific syslog server with ID. |
| POST | /api/v1/syslogservers/ | Create a new syslog server. |
| PATCH | /api/v1/syslogservers/[id]/ | Update specified fields of an existing syslog server with ID. |
| DELETE | /api/v1/syslogservers/[id]/ | Delete a syslog server. |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# Log settings (/logsettings/)

URL: `https://[server_name]/api/[api_version]/logsettings/`

This endpoint is used to edit the settings for logs. This resource can be found in the FortiAuthenticator GUI under **Logging > Log Config > Log Settings**.

## Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| delete_enable | Enable log auto-deletion | boolean | Yes | Default is set to "false" if not specified. |
| delete_age_n | Auto-delete logs older than number | integer | No | |

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| delete_age_mult | Auo-delete logs older than multiplier | integer | No | Default is set to "months". Either "days", "weeks", or "months". |
| faz_enable | Enable sending logs to FortiManager/FortiAnalyzer | boolean | No | Default is set to "false" if not specified. |
| faz_server | IP Address or FQDN of the FortiManager/FortiAnalyzer | string | No | Must be a valid FQDN or IPv4 address. |
| backup_enable | Enable remote backup | boolean | Yes | Default is set to "false" if not specified. |
| backup_frequency | How often the configuration is backup up | string | If backup_enable is true | Either "hourly", "daily", "weekly", or "monthly". |
| backup_directory | Directory on the FTP server in which to store the configuration | string | No | |
| backup_time | Time when the configuration is going to be backed up | string | If backup_frequency isn't daily | Default is set to "00:00:00". Must be in 24 hour time format. Accepted formats are "23:59" or "23:59:59". |
| backup_ftp | Name of the FTP server | string | No | Must be the name of an FTP server already known by the FortiAuthenticator. |
| remote_syslog_enable | Enable sending logs to remote syslog servers | boolean | Yes | Default is set to "false" if not specified. |
| remote_syslog_servers | Names of syslog servers | string | No | Must be names of syslog servers known by the FortiAuthenticator, separated by commas. For example, "server1,server2,server3". |

# Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/logsettings/ | Get log settings. |

| HTTP method | Resource URI | Action |
|---|---|---|
| POST | /api/v1/logsettings/ | Set log fields. |
| PATCH | /api/v1/logsettings/ | Update the specified log settings fields. |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# User certificate management (/usercerts/)

URL: `https://[server_name]/api/[api_version]/usercerts/`

This endpoint is used to renew and revoke user certificates.

## Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| cert_id | Certificate ID of the certificate to renew | string | Yes, if renewing user certificate | |
| status | User certificate status | string | Yes, if revoking or un-revoking user certificate | Either "active", "pending", "expired", or "revoked". |
| revocation_reason | Revocation reason | string | Yes, if revoking user certificate | Either "Unspecified", "Key Compromise", "CA Compromise", "Affiliation Changed", "Superseded", "Cessation Of Operation", or "Certificate Hold" |
| csr | Certificate signing request | CSR file | Yes, if renewing user certificate | Subject in the CSR must match the subject of the certificate specified by cert_id. |
| expiry | Number of days until new certificate expires | integer | Yes, if renewing user certificate | |
| revoke_old | Revoke previous certificate upon successful renewal | boolean | No | Default is set to "false" if not specified. |

# Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/usercerts/ | Get all user certificates. |
| GET | /api/v1/usercerts/pem/ | Get all user certificates in PEM format. |
| GET | /api/v1/usercerts/[id]/ | Get a specific user certificate with ID. |
| POST | /api/v1/usercerts/renew/ | Renew a user certificate. Requires 'cert_id', 'csr', and 'expiry'. |
| PATCH | /api/v1/usercerts/[id] | Revoke a user certificate with ID.<br>To revoke a user certificate, set the status field to "revoked" and the revocation_reason to one of the revocation reasons. |
| PATCH | /api/v1/usercerts/[id] | Un-revoke a user certificate with ID.<br>If a user certificate was revoked with revocation_reason set to "Certificate Hold", it can be un-revoked by setting the status field to active. |

# Allowed filtering

| Field | Lookup Expressions | Description |
|---|---|---|
| subject | exact, iexact, contains, icontains | |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

**Example**

```
Get user certificates:
curl -k -v \
-u "[webadmin]:[hash]" \
https://[FAC_IP]/api/v1/usercerts/

Response:
{
"id": 1,
"cert_id": "user_cert",
"expiry": "2019-08-15T01:02:07+00:00",
"issuer": "issuer_cert | C=CA, ST=BC, L=Burnaby, O=Fortinet, OU=RD, CN=test,
      emailAddress=####@####.com",
"revocation_reason": null,
"serial": "0122A3",
"status": "Active",
```

```
"subject": "/C=CA/ST=BC/L=Burnaby/O=o/OU=RD/CN=test"
}

Get user certificates in PEM format:
curl -k -v \
-u "[webadmin]:[hash]" \
https://[FAC_IP]/api/v1/usercerts/pem/

Response:
{
"cert_id": "user_cert",
"certificate": "-----BEGIN CERTIFICATE-----\n####################################\n-----END
      CERTIFICATE-----\n"
}

Renew a user certificate:
curl -k -X POST \
https://[FAC_IP]/api/v1/usercerts/renew/ \
-H 'content-type: multipart/form-data' \
-u '[webadmin]:[hash]' \
-F 'cert_id=user_cert' \
-F 'csr=@/path/to/csr/*.csr' \
-F 'expiry=[Number of days until new certificate expires]' \
-F 'revoke_old=[true/false; optional]'

Response:
{
"cert_id": "new_user_cert",
"certificate": "-----BEGIN CERTIFICATE-----\n####################################\n-----END
      CERTIFICATE-----\n"
}

Revoke a user certificate:
curl -k -v \
-X PATCH \
-H 'Content-Type: application/json' \
-u '[webadmin]:[hash]' \
-d '{"status":"revoked", "revocation_reason":"Certificate Hold"}' \
https://[FAC_IP]/api/v1/usercerts/1/

Response:
{
"cert_id": "user_cert",
"expiry": "2019-08-15T01:02:07+00:00",
"id": 1,
"issuer": "issuer_cert | C=CA, ST=BC, L=Burnaby, O=Fortinet, OU=RD, CN=test,
      emailAddress=####@####.com",
"revocation_reason": "Certificate Hold",
"serial": "0122A3",
"status": "Revoked",
"subject": "/C=CA/ST=BC/L=Burnaby/O=o/OU=RD/CN=test"
}

Un-revoke a user certificate:
curl -k -v \
-X PATCH \
-H 'Content-Type: application/json' \
```

```
-u '[webadmin]:[hash]' \
-d '{"status":"active"}' \
https://[FAC_IP]/api/v1/usercerts/1/

Response:
{
"cert_id": "user_cert",
"expiry": "2019-08-15T01:02:07+00:00",
"id": 1,
"issuer": "issuer_cert | C=CA, ST=BC, L=Burnaby, O=Fortinet, OU=RD, CN=test,
      emailAddress=####@####.com",
"revocation_reason": null,
"serial": "0122A3",
"status": "Active",
"subject": "/C=CA/ST=BC/L=Burnaby/O=o/OU=RD/CN=test"
}
```

# SCEP Enrollment Requests Management (/scepregs/)

URL: `https://[server_name]/api/v1/scepreqs/`

This end-point is used to manage SCEP enrollment requests.

## Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| status | Status of the SCEP enrollment request. | string | N/A | One of "Approved", "Pending", or "Rejected" |
| type | Type of the SCEP enrollment request. | string | N/A | One of "Wildcard" or "Regular" |
| subject | Subject of the SCEP enrollment request. | string | N/A | |
| issuer | Subject of the issuer for the SCEP enrollment request. | string | N/A | |

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/scepreqs/ | Get SCEP enrollment requests. |
| GET | /api/v1/scepreqs/[id]/ | Get SCEP enrollment requests. |
| DELETE | /api/v1/scepreqs/[id]/ | Delete a SCEP enrollment request. |

## Allowed filtering

| Field | Lookup Expressions | Description |
|---|---|---|
| subject | exact, iexact, contains, icontains | |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# FTP servers (/ftpservers/)

URL: `https://[server_name]/api/[api_version]/ftpservers/`

This endpoint is used to create, update, edit, and delete FTP servers.

## Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| name | Name of the FTP server | string | Yes | |
| address | Domain name or IP address of the FTP server | string | Yes | |
| port | Port of the syslog server | integer | Yes | Default is set to "21" |
| username | The username used to login to the FTP server | string | No | |
| password | The password required to login to the FTP server | string | No | |

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| conn_type | The type of connection | string | Yes | Either "ftp" or "sftp". The default is "ftp" |
| anonymous | Whether the connection is anonymous or not | boolean | No | Read-only field. |

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/ftpservers/ | Get all FTP servers |
| GET | /api/v1/ftpservers/[id]/ | Get a specific FTP server with id |
| POST | /api/v1/ftpservers/ | Create a new FTP server |
| PATCH | /api/v1/ftpservers/[id]/ | Update specified fields of an existing FTP server with id |
| DELETE | /api/v1/ftpservers/[id]/ | Delete an FTP server |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# Licensing (/licensing/)

URL: https://[server_name]/api/[api_version]/licensing/

This endpoint is used to update the FortiAuthenticator license.

## Supported fields

| Field | Description | Type | Required | Other restrictions |
|---|---|---|---|---|
| license | The .lic file | file | Yes | Must be a valid .lic file |

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| POST | /api/v1/licensing/ | Update the FortiAuthenticator license with a license file |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

**Examples**

```
curl -k -X POST \
   https://[FAC_IP]/api/v1/licensing/ \
   -H 'content-type: multipart/form-data' \
   -u "webadmin"[hash]" \
   -F 'license=@/path/to/license/VM-00000000.lic'
Response:
{
   "expiry_date": "2019-06-14T00:00:00+00:00",
   "ip": "192.168.60.908",
   "license": "FAC-VM###############.lic",
   "license_hash": "############################",
   "success_message": "Reboot process for license update of VM is started. Please wait for the FAC
        to restart before making new requests."
}
```

# FortiToken Mobile provisioning settings (/fortitokenmobileprovisioning/)

URL: https://[server_name]/api/[api_version]/fortitokenmobileprovisioning/

This endpoint is used to edit the FortiToken Mobile provisioning settings under **System > Administration > FortiGuard**.

## Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| server_address | The server which provisions the FortiTokens | string | No | The default is "fortitokenmobile.fortinet.com" if not specified. |
| server_port | The server port number | integer | No | The default is 433 if not specified. |

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| act_timeout | The activation timeout in hours | integer | No | Must be a number between 1 and 168. The default is 1 if not specified. |
| token_size | The size of the token | integer | No | Either 6 or 8. The default is 6 if not specified. |
| token_algo | The type of token algorithm | string | No | Either "totp" or "hotp". The default is "totp" if not specified. |
| time_step | The time step | integer | No | Either 30 or 60. The default is 60 if not specified. |
| require_pin | The setting for whether or not to require a PIN, or to enforce a mandatory PIN | string | No | Either "require", "not_requre", or "enforce". The default is "require" if not specified. |
| pin_length | The pin length | integer | No | Either 4, 6, or 8. The default is 4 if not specified. |

## Allowed methods

| HTTP method | Resource URI | Action | Note |
|---|---|---|---|
| GET | /api/v1/fortitokenmobileprovisioning/ | Get FortiToken Mobile provisioning settings. | |
| POST | /api/v1/fortitokenmobileprovisioning/ | Set FortiToken Mobile provisioning settings. | Defaults are used if fields are unspecified. |
| PATCH | /api/v1/fortitokenmobileprovisioning/ | Update the specified FortiToken Mobile provisioning settings fields. | Previously saved settings are used if unspecified. |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# Scheduled backup settings (/scheduledbackupsettings/)

URL: https://[server_name]/api/[api_version]/scheduledbackupsettings/

This endpoint is used to edit the settings for automatically backing up the FortiAuthenticator device's configuration file.

# Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| frequency | How often the configuration file is backed up | string | Yes, if enabled is true | Either "hourly", "daily", "weekly", or "monthly". |
| directory | The directory on the FTP server in which to store the configuration file | string | No | |
| time | The time when the configuration file is to be backed up | string | Yes, if frequency isn't daily | Must be in 24 hour time format. For example, either "23:59" or "23:59:59" are accepted. The default is midnight if not specified. |
| ftp | The primary FTP server | string | Yes, if enabled is true | Must be the name of an FTP server already configured on the FortiAuthenticator device. |
| ftp_2 | The secondary FTP server | string | No | Must be the name of an FTP server already configured on the FortiAuthenticator device. |
| enabled | Whether or not the configuration file will be scheduled to backup | boolean | No | Either "true", "false", "1", or "0". The values can be strings or primitive booleans. |

# Allowed methods

| HTTP method | Resource URI | Action | Note |
|---|---|---|---|
| GET | /api/v1/scheduledbackupsettings/ | Get scheduled backup settings. | |
| POST | /api/v1/scheduledbackupsettings/ | Set scheduled backup settings. | Defaults are used if fields are unspecified. |
| PATCH | /api/v1/scheduledbackupsettings/ | Update the specified scheduled backup fields. | Previously saved settings are used if |

| HTTP method | Resource URI | Action | Note |
|---|---|---|---|
| | | | fields are unspecified. |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# Fabric integration endpoints (/fabric/)



# Fabric authenticate (/fabric/authenticate)

URL: `https://[server_name]/api/fabric/authenticate`

This endpoint is used to deliver an access_token to FortiOS to integrate the FortiAuthenticator as a Fortinet Security Fabric device. Currently, these tokens do not expire, as long as the access token expiry of the default FortiOS fabric application remains at zero.

## Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| username | Administrator password | string | Yes, unless refreshing token | User should not require multi-factor authentication, and must have Widget read/write permissions or full permissions. |
| password | Administrator password | string | Yes, unless refreshing token | |
| grant_type | OAuth grant type | string | If refreshing token | |
| refresh_token | OAuth refresh token | string | If refreshing token | |

**Note**: Currently, FortiOS is not configured to refresh the token, so the token does not expire. Therefore, the refresh tokens that result from the application are set to zero.

# Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| POST | /api/fabric/authenticate | Get token, or refresh token |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

**Note**:

- If the user requires multi-factor authentication, this is bypassed when issuing an OAuth token. FortiOS does not yet prompt for additional challenges after the username and password.
- If your username is in email address format, and your Username/Realm format under **Authentication > Self-Service Portal > Access Control** uses the '@' symbol, ensure that you specify the realm. E.g. user@name.com@realm
- If authenticating multiple FortiOS devices with the Security Fabric endpoint, copy and paste the token from the first authentication onto subsequent devices. Authenticating will generate a new token.

# Response codes

In addition to the general codes defined in General API response codes on page 138, a POST request to this resource can also result in the following return codes:

| Code | Response content | Description |
|---|---|---|
| 200 OK | | Valid credentials |
| 401 Unauthorized | | Invalid credentials, or user improperly configured. |

**Example**

```
Get token:
curl -k -v -X POST \
https://[FortiAuthenticator_IP]/api/fabric/authenticate \
-H 'Content-Type: application/json' \
-d '{
   "username": "tfadmin",
   "password": "12345678"
}'

Response:
{
   "access_token": "shrWNdu1xJRUgpcUi2bhYRX1Sl8pXe",
   "expires_in": 0,
   "message": "successfully authenticated",
```

```
   "refresh_token": "tU85BMdOoV3pktSSiLaABJN7ySiADZ",
   "scope": "read",
   "success": "true",
   "token_type": "Bearer"
}

Refresh a token (for future reference):
curl -k -v -X POST \
https://[FortiAuthenticator_IP]/api/fabric/authenticate \
-H 'Content-Type: application/json' \
-d '{
   "grant_type": "refresh_token",
   "refresh_token": "Y53b5XCLUdjkHVH49ZSheYQjafn6EV"
}'

Response:
{
   "access_token": "fzMK69MdyA0vRJXh2CWnuHRcpuQrpL",
   "expires_in": 0,
   "message": "Token has been refreshed successfully",
   "refresh_token": "UqCV1xEPSoq4vSLE0YgXAkF2zzMGO5",
   "scope": "read",
   "success": "true",
   "token_type": "Bearer"
}
```

# Fabric device status (/fabric/device/status)

URL: `https://[server_name]/api/fabric/device/status`

This endpoint is used to retrieve the FortiAuthenticator status for FortiOS fabric display. It requires a valid Bearer token in the Authorization header.

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/fabric/device/status | Get FortiAuthenticator statistics and status |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

## Response codes

In addition to the general codes defined in General API response codes on page 138, a POST request to this resource can also result in the following return codes:

| Code | Response content | Description |
|---|---|---|
| 200 OK | FortiAuthenticator information | |
| 401 Unauthorized | | Invalid Bearer token |

**Example**

```
Get FortiAuthenticator information:
curl -k -v -X GET \
https://[FortiAuthenticator_IP]/api/fabric/device/status \
-H 'Authorization: Bearer shrWNdu1xJRUgpcUi2bhYRX1Sl8pXe'

Response:
{
   "build": {
      "number": xxx,
      "release_life_cycle": "dev"
   },
   "device_type": "fortiauthenticator",
   "host_name": "FortiAuthenticator",
   "model": "FACVM",
   "serial_number": "FAC-VM0000000000",
   "supported_api_versions": [
      "v1"
   ],
   "version": {
      "major": x,
      "minor": x,
      "patch": x
   }
}
```

# Fabric widget (/fabric/widget)

URL: `https://[server_name]/api/v1/fabric/widget`

This endpoint is used to retrieve a list of available fabric widgets that the FortiAuthenticator can provide. It requires a valid Bearer token in the Authentication header.

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/fabric/widget | Get a list of available widgets and the visualization types that they support. |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

## Response codes

In addition to the general codes defined in General API response codes on page 138, a POST request to this resource can also result in the following return codes:

| Code | Response content | Description |
|---|---|---|
| 200 OK | Widget information | |
| 401 Unauthorized | | Invalid Bearer token |

**Example**

```
Get Widget Info:
curl -k -v -X GET \
https://[FortiAuthenticator_IP]/api/v1/fabric/device/widget \
-H 'Authorization: Bearer shrWNdu1xJRUgpcUi2bhYRX1Sl8pXe'

Response:
{
   "data": [
      {
         "id": "sysinfo",
         "lang_key": "sysinfo",
         "supported_visualization_types": [
            "key-value-pair"
         ]
      },
   ],
   "meta": {
      "language": {
         "en": {
            "sysinfo": "System Information",
         }
      }
   }
}
```

# Fabric widget detail by visualization type (/fabric/widget/id)

**URL:** `https://[server_name]/api/v1/fabric/widget/(id)?visualization_type=(type)`

This endpoint is used to retrieve individual widget data for FortiOS to display. Widgets are obtained by their string ID, and the visualization type is a query parameter. It requires a valid Bearer token in the Authorization header.

# Supported fields

| Field | Display name | Type | Required |
|---|---|---|---|
| id | The string identifier of the widget | string | Yes |
| type | The string identifier for the visualization type | string | Yes |

# Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/fabric/widget/(id)?visualization_type=(type) | Get a list of available widgets and the visualization types that they support. |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# Response codes

In addition to the general codes defined in General API response codes on page 138, a POST request to this resource can also result in the following return codes:

| Code | Response content | Description |
|---|---|---|
| 200 OK | Widget detail information | |
| 401 Unauthorized | | Invalid Bearer token |

**Example**

```
Get Widget Detail Info:
curl -k -v -X GET \
https://[FortiAuthenticator_IP]/api/v1/fabric/widget/sysinfo?visualization_type=key-value-pair \
-H 'Authorization: Bearer shrWNdu1xJRUgpcUi2bhYRX1Sl8pXe'

Response:
{
   "data": [
      {
         "lang_key": "hostname",
         "value": "FortiAuthenticator"
      },
      ...
   ],
   "meta": {
      "language": {
         "en": {
```

```
            "devicefqdn": "Device FQDN",
            ...
         }
      },
      "polling": false,
      "polling_interval_min": 0,
      "visualization_type": "key-value-pair"
   }
}
```

# OAuth server endpoints (/oauth/)



# OIDC Authorization (/oauth/authorize/)

URL: `https://[server_name]/api/v1/oauth/authorize/`

The Authorization Code flow is best used in web and mobile apps. This is the flow used for third party integration, the user authorizes your partner to access its products in your APIs and get the authorization code. This code is needed along with *client_id* and *client_secret* to get the access token.

This endpoint will redirect to the access portal configured in the *Relying Party*.

## Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| response_type | response_type which should be set to code for now | string | Yes | |

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| client_id | client_id that is registered in FortiAuthenticator | string | Yes | |
| redirect_uri | Redirect URL after successful or failed authentication | string | Yes | |
| scope | Requested scopes | string | No | |
| code_challenge_method | Code Verifier hashing algorithm | string | If grant_type is **Authorization code with PKCE** | only 'S256' accepted |
| code_challenge | Base64 URL encoding of the SHA256 hash of the code_verifier | string | If grant_type is **Authorization code with PKCE** | |
| approval_prompt | Controls whether to show the approval page | string | No | Either 'auto' or 'force'. See approval_prompt on page 108. |
| prompt | Controls user authentication and consent flow | string | No | Either 'none' or 'login' or 'consent'. See prompt on page 109. |

## approval_prompt

| Value | Behavior |
|---|---|
| auto | Skips the consent screen if the user has previously granted the requested scopes. |
| force | Always prompts the user for consent, even if they have previously granted the requested scopes. |

### prompt

| Value | Behavior |
|---|---|
| none | Should not display any authentication or consent user interface pages. Fails if user interaction is required. |
| consent | Always prompts the user for consent. |
| login | Should prompt the end-user for authentication. |

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/oauth/authorize | Redirect to portal for authentication. |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

## Response codes

| Code | Response content | Description |
|---|---|---|
| 302 redirect | Portal for authentication | Successfully redirected to the portal. |
| 302 redirect | Callback URL with error | In case of error client will be redirected to redirect_uri with error and error_description. |

**Example**

```
https://fac3.org/api/v1/oauth/authorize/?response_type=code&client_
      id=hlcOZPMGhIjNgU4sZu90nmr4O6q9vsSSdTcUIubM&redirect_uri=https://some_url/noexist/callback

Response:
redirect to the portal specified in the policy of relying party configuration
```

# OAuth server token (/oauth/token/)

**URL:** `https://[server_name]/api/v1/oauth/token/`

This end-point is used to verify a user's identity and upon confirming that identity, issue a token which allows access to resources protected by Bearer Token. Tokens are issued per application and user, and applications are configurable in the GUI. So long as the access token expiry of the application in question isn't zero, then these tokens can expire, and also they can be refreshed. This endpoint can also be used to refresh a previously issued token.

# Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| username | User username | string | If grant_type is password | |
| iam_account | IAM account name or alias | string | If grant_type is password and username is not specified | Ignored if username is specified. |
| iam_user | IAM user | string | If grant_type is password and username is not specified | Ignored if username is specified. |
| password | User password | string | If grant_type is password | |
| realm | User realm | string | If grant_type is password, and user is not local | The default realm is the realm selected as the default under **System > Administration > System Access**. If you are authenticating a user from the default realm, you do not need to specify a realm. |
| refresh_token | Token used to refresh access_ token | string | If grant_type is refresh_token | |
| grant_type | OAuth grant type | string | Yes | |
| client_id | String ID of client or application | string | Yes | |
| client_secret | Hash client secret | string | If application client_ type is 'confidential' | |

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| challenge | The type of multi-factor authentication challenge | string | If responding to multi-factor authentication challenge with challenge response | Can be 'otp', 'radius', etc. Reuse the challenge you received from the token endpoint. |
| challenge_response | String code challenge response | string | If responding to challenge | |
| method | The method of challenge response | string | Yes | Required if responding with an OTP challenge. Can be 'ftm', 'ftm-push', 'ftk', 'sms', 'email', or 'dual'. |
| session | OAuth grant type | string | If responding with an OTP challenge with ftm-push method | |
| redirect_uri | Redirect URL callback | string | | |
| scope | Requested scopes | string | No | This parameter is ignored if the request is not for password-based grant type |
| code_verifier | High-entropy cryptographic random STRING<br>See RFC7636 Section-4.1 | string | If grant_type is **Authorization code with PKCE** | The code_challenge is an SHA256 transformation of the code_verifier |
| code | Single-use authorization code | string | If grant_type is **Authorization code** or **Authorization code with PKCE** | |

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| user_ip | User IP address | string | No | When omitted, the source IP address of the HTTP request (or the IP address in the forwarding HTTP header) is assumed |

# Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| POST | /api/v1/oauth/token/ | Get token, or refresh token. |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# Response

If the `openid` is specified as scope which is the default one at the moment, JWT token is returned as `id_token`. This token consists of three parts: Header, Payload, and keys (separated by dots).

# Response codes

In addition to the general codes defined in General API response codes on page 138, a POST request to this resource can also result in the following return codes:

| Code | Response content | Description |
|---|---|---|
| 200 OK | | Valid credentials |
| 401 Unauthorized | | Invalid credentials, or user improperly configured |
| 406 Not Acceptable | Challenge, method, status, and optional session | Initial credentials are valid, but the user requires more information. Send additional information. |

**Example**

```
Get token (Password-based)

curl -k -v -X POST \
```

```
https://[FAC_IP]/api/v1/oauth/token/ \
-H 'Content-Type: application/json' \
-d '{
    "username": "luser1",
    "password": "12345678",
    "client_id": "client_id",
    "client_secret": "client_secret", -> in case of confidential
    "grant_type": "password"
}'

Response:
{
    "access_token": "shrWNdu1xJRUgpcUi2bhYRX1Sl8pXe",
    "expires_in": 0,
    "message": "successfully authenticated",
    "refresh_token": "tU85BMdOoV3pktSSiLaABJN7ySiADZ",
    "scope": "read",
    "status": "success",
    "token_type": "Bearer"
}
```

**Get token (Authorization code)**

```
curl -k -v -X POST \
https://[FAC_IP]/api/v1/oauth/token/ \
-H 'Content-Type: application/json' \
-d '{
    "username": "luser1",
    "password": "12345678",
    "client_id": "client_id",
    "client_secret": "client_secret",
    "code": "04yjZyQNtsRKe9yzFBnmj9gf4wCdsY"
    "grant_type": "authorization_code"
}'

Response:
{
    "access_token": "nCKp5jGTfPGuk2Rv98chYUrBYqlhxZ",
    "expires_in": 3600,
    "token_type": "Bearer",
    "scope": "openid",
    "refresh_token": "5HibwqzdhCbWD3jEPP89FZJ5uJYxhd",
    "id_token":
"eyJ0eXAiOiAiSldUIiwgImFsZyI6ICJSUzI1NiIsICJraWQiOiAiZUE5VGxDTmE2cWo5OTRHeUg0VGtRZ1F0UV9QT0NFc3JTa
zlGTFF0b0dOVSJ9.eyJhdWQiOiAiaGxjT1pQTUdoSWpOZ1U0c1p1OTBubXI0TzZxOXZzU1NkVGNVSXViTSIsICJpYXQiOiAxNj
M1ODg1MDUyLCAiYXRfaGFzaCI6ICJtSHpPdUlYWm9qdVNNZURGeV9CS1hBIiwgInN1YiI6ICIyIiwgImlzcyI6ICIvYXBpL3Yx
L29hdXRoIiwgImV4cCI6IDE2MzU5MjEwNTIsICJqdGkiOiAiNjljNjEyNzktNTBlMi00YjQ3LWFhMGYtOTk0YWVhNjBiMmU0In
0.Rd64aeiPT7Tn2xsis3Vbdnu0Zaec0GA_K0VDoNOHBPhYgnMS6rx7tRJhL7A_i3ocOnGVUU7ufP4LVIUuhQx-p_LoDkOD_
MoR5PYUNg3M77zEXjdnrRJDiGB9DapwDKFNY_1hyjLbaBl9336kzoSqBxHqrkGDhiZQJvycH1epvZ_CsW0M-
VcVBeLtCZKWmFcraklIZKfpg4RaCxTs_fe07eBZL_Iok-Tlpx9R5PIxjAwBiUOoZiUyNmYF8BFQXe7eo2ikZ_
WsdvE0sfYFzmdukov4_BdytLXt8Hvqqi2CNw9AFOvszhCLweife_2vBNpiS_vL6r2LpZ7-
FCwGsqwTWuqyhuj4LtRodgqnmswrZ6yDEpxNIWCoE7m4u77GG9se7YU8CWllwr__8EZFEvrBB6_
```

```
YEhAxofHZaBXFXO2aXNW37CTCidLNuXbGlQp2HaJGSQnM7MDbHVbiJ3QyKc0aUWaYYqP5msW9A91Z2-
E7PTNp7UptsHnIiVzlfdFYyRskwym8j31WmlwCdS7BvxxgBCSbbanWznIKKiFcgdDOAyN-F6ePOSs8cCbAJoZ_
hQSPxYd3ChLBTSL38ZPUTzPxZH7zqG0gLX_
uSNkIMBSrqhnLDZGKogCZhuOUNY38UBwNrasAcwEObhuhMuHKYu5GkQZE79bTrcuXeHuxbGGRWQQ"
}

Refresh a token

curl -k -v -X POST \
https://[FAC_IP]/api/v1/oauth/token/ \
-H 'Content-Type: application/json' \
-d '{
    "client_id": "client_id",
    "client_secret": "client_secret", -> in case of confidential
    "grant_type": "refresh_token",
    "refresh_token": "tU85BMdOoV3pktSSiLaABJN7ySiADZ"
}'

Response:
{
    "access_token": "fzMK69MdyA0vRJXh2CWnuHRcpuQrpL",
    "expires_in": 0,
    "message": "Token has been refreshed successfully",
    "refresh_token": "UqCV1xEPSoq4vSLE0YgXAkF2zzMGO5",
    "scope": "read",
    "status": "success",
    "token_type": "Bearer"
}

Get a token with FTM push

curl -k -v -X POST \
https://[FAC_IP]/api/v1/oauth/token/ \
-H 'Content-Type: application/json' \
-d '{
    "username": "luser1",
    "password": "12345678",
    "client_id": "client_id",
    "client_secret": "client_secret", -> in case of confidential
    "grant_type": "password"
}'

Response:
{
    "challenge": "otp",
    "method": "ftm-push",
    "session": "480dccc0f6bf4ed69ba484320ef92781",
    "status": "pending"
}

Check for FTM-PUSH approval
```

```
curl -k -v -X GET \
'https://[FAC_IP]/api/v1/pushpoll/?s=480dccc0f6bf4ed69ba484320ef92781' \
-H 'Content-Type: application/json' \

Response if status is 'pending':
{
    "challenge": "otp",
    "method": "ftm-push",
    "session": "480dccc0f6bf4ed69ba484320ef92781",
    "status": "pending"
}

Response if status is 'success' (The push request was approved):
{
    "challenge": "otp",
    "challenge_response": "3njPWHp6LgXtRwwXabEN",
    "method": "ftm-push",
    "session": "480dccc0f6bf4ed69ba484320ef92781",
    "status": "success"
}

Use the successful push session code to get a token

curl -k -v -X POST \
https://[FAC_IP]/api/v1/oauth/token/ \
-H 'Content-Type: application/json' \
-d '{
    "username": "luser1",
    "password": "12345678",
    "client_id": "client_id",
    "grant_type": "password",
    "challenge": "otp",
    "challenge_response": "3njPWHp6LgXtRwwXabEN",
    "method": "ftm-push",
    "session": "480dccc0f6bf4ed69ba484320ef92781"
}'

Response:
{
    "access_token": "c1t2I989RnZCn7xFNsDGLtGShdeSL6",
    "expires_in": 36000,
    "refresh_token": "nP0Fq74huju4gDLCR5jXHSxerDAXD3",
    "scope": "read",
    "status": "success",
    "token_type": "Bearer"
}

If you opt to manually enter the token code, change the method from ftm-push to ftm

curl -k -v -X POST \
https://[FAC_IP]/api/v1/oauth/token/ \
-H 'Content-Type: application/json' \
```

```
-d '{
    "username": "luser1",
    "password": "12345678",
    "client_id": "client_id",
    "grant_type": "password",
    "challenge": "otp",
    "challenge_response": "12345678",
    "method": "ftm"
}'

Response:
{
    "access_token": "c1t2I989RnZCn7xFNsDGLtGShdeSL6",
    "expires_in": 36000,
    "refresh_token": "nP0Fq74huju4gDLCR5jXHSxerDAXD3",
    "scope": "read",
    "status": "success",
    "token_type": "Bearer"
}

If you opt to manually enter the token code, change the method from ftm-push to ftm:
curl -k -v -X POST \
https://[FAC_IP]/api/v1/oauth/token/ \
-H 'Content-Type: application/json' \
-d '{
    "username": "luser1",
    "password": "12345678",
    "client_id": "client_id",
    "grant_type": "password",
    "challenge": "otp",
    "challenge_response": "12345678",
    "method": "ftm"
}'

Response:
{
    "access_token": "c1t2I989RnZCn7xFNsDGLtGShdeSL6",
    "expires_in": 36000,
    "refresh_token": "nP0Fq74huju4gDLCR5jXHSxerDAXD3",
    "scope": "read",
    "status": "success",
    "token_type": "Bearer"
}
```

# OAuth server revoke token (/oauth/revoke_token/)

URL: `https://[server_name]/api/v1/oauth/revoke_token/`

This end-point is used to revoke or otherwise delete an an oauth access token entry from the database in the event that the authorized client wishes to revoke that token.

# Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| client_id | String ID of client or application. | string | Yes | |
| client_secret | Hash client secret. | string | Only if application client_type is **confidential**. | |
| token | Access Token to revoke. | string | Yes | |

# Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| POST | /api/v1/oauth/revoke_token/ | Revoke specified token. |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# Response codes

In addition to the general codes defined in General API response codes on page 138, a POST request to this resource can also result in the following return codes:

| Code | Response content | Description |
|---|---|---|
| 200 OK | | Valid credentials. |
| 400 BAD REQUEST | | If you specify the correct client_id and client_secret, but you enter an empty token. |
| 401 UNAUTHORIZED | | If you do not specify the correct client_id and client_secret. |

**Example**

**Revoke a Token**

```
curl -k -v -X POST \
```

```
https://[FAC_IP]/api/v1/oauth/revoke_token/ \
-H 'Content-Type: application/json' \
-d '{
"client_id": "fcare",
"token": "zGSaz2yqfjco7qWLQW2ctZXlhbRRJ"
}'
```

# OAuth server verify token (/oauth/verify_token/)

URL: `https://[server_name]/api/v1/oauth/verify_token/?client_id=<client_id>`

This is an endpoint for verifying an access token, to determine whether or not it is valid. Returns a HTTP 200 OK response if the token is valid. Username is returned upon success. You do not need to specify a client secret as a parameter for confidential applications.

## Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| client_id | String ID of client or application | string | Yes | Must be present as a query parameter |

## Allowed methods

| HTTP method | Resource URI | Action | Note |
|---|---|---|---|
| GET | /api/v1/oauth/verify_token/ | Verify specified token | The access token must be placed in the Authorization header of the request in this format: 'Authorization: Bearer [ACCESS_TOKEN]' |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

## Response codes

In addition to the general codes defined in General API response codes on page 138, a POST request to this resource can also result in the following return codes:

| Code | Response content | Description |
|---|---|---|
| 200 OK | Username is returned upon success | Token was successfully verified.<br>Includes `expires_in` field that gives number of seconds until expiry of the verified access token. |
| 401 | | Unauthorized because token is not valid. |

### Example

**Verify a Token:**

```
curl -k -v -X GET \
https://[FAC_IP]/api/v1/oauth/verify_token/ \
-H 'Content-Type: application/json' \
-H 'Authorization: Bearer Ua3tkmlDtePw7EQIXb1a2oGNkw4Li'

Response:
200 OK
```

# OIDC Userinfo (/oauth/userinfo/)

URL: `https://[server_name]/api/v1/oauth/userinfo/`

The UserInfo endpoint is supplied as part of the OIDC service, and is used to retrieve more information about the user than was supplied in the ID token when the user logged in to the OIDC client.

## Allowed methods

| HTTP method | Resource URI | Action | Note |
|---|---|---|---|
| GET | /api/v1/oauth/userinfo/ | Verify specified token and returns claims. | The access token must be placed in the Authorization header of the request in this format: 'Authorization: Bearer [ACCESS_TOKEN]' |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

## Response codes

| Code | Response content | Description |
|---|---|---|
| 200 OK | Claims are returned upon success. | Token was successfully verified. |
| 401 | | Unauthorized because the access token provided is expired, revoked, malformed, or invalid for other reasons. |

**Example**

```
curl -k -v -X GET \
https://[FAC_IP]/api/v1/oauth/userinfo/ \
-H 'Content-Type: application/json' \
-H 'Authorization: Bearer Ua3tkmlDtePw7EQIXb1a2oGNkw4Li'
Response:
{
    "sub": "2",
    "email": "test@test.com",
    "username" : "test",
    "groups" : "SW,HW",
}
```

# OIDC Keys (/oauth/.well-known/keys/)

**URL:** `https://[server_name]/api/v1/oauth/.well-known/keys/`

This endpoint provides details of the key used to sign the JWTs generated for ID tokens, so that clients are able to verify them.

**Example**

```
curl -k -v -X GET https://[FAC_IP]/api/v1/oauth/.well-known/keys/

Response:
{
    "keys": [
        {
        "alg": "RS256",
        "use": "sig",
        "kid": "eA9TlCNa6qj994GyH4TkQgQtQ_POCEsrSk9FLQtoGNU",
        "kty": "RSA",
        "n": "u_j5r-
YE1h2yYWjAxfZaVZ82kLJtrRmRiSgZIkTAYLuYoW60eOMrPmEIyo4EaoMTlT1sEqiS8VSa6hqOzsvw8esQ_
SWtYmhT4TplgHFnF0dR-
xitSVYY113ixNpPWiOxyANT2ArwEvnHAYHXzMXIiirSRfr5WmM28huuKvoIOeDeLt9ezARsCpwbQPMlq-p896QWba_
HAOhWL7YISwxtUf2VmDc5pKjqfYPuSw0LOIxvvINwEesIkbilJqx4QhEwnG3g0p2gsm7dx7zwUjuJyNjzLElrB5ZlAfxamkEp9
qHpOeYIOiTdLmD8wr8T7r-QGpdNYeJsLA9ycIO-bSGxk9n6eNXFlF5FoAbiTNJzvrIPbfCulGSMfE3PSO1Xro1YB_
QTKmjPlPyt2Ae7gBBN2yTGqbY0SjUk81Ca-DqorbjgOXYXMkxfjCg22qv5Yq_CgX7WZM6CV4IE_
B6R0wEJQLk9ccWblOpZ3eFafR7CbtXOjJuNQzBaMwTwtsbcQlOxW_kPuYj0gNgLvmC02C-
VE1VTi1ttnXsMwgO3qHgTj2mU11taFKjvf88__xIV4dt87AMiVJxRKVfH3JzknRlfW5VINYqpOut8JctPIRPCLEhhEGuRaIzz-
FbTPQpJY3ULrJ6D_v7pd5H3kF_89a-VMiodH_e52W6Wd4FnIE_IiVM",
        "e": "AQAB"
        }
    ]
}
```

# OIDC Connect Discovery Info

**URL:** `https://[server_name]/api/v1/oauth/.well-known/oauth-authorization-server/`

**URL:** `https://[server_name]/api/v1/oauth/.well-known/openid-configuration/`

This endpoint provides auto discovery information to OIDC clients, telling them the JWT issuer to use, the location of the JWKs to verify JWTs with, the token and user info endpoints to query, and other details.

**Example**

```
curl -k -v -X GET https://[FAC_IP]/api/v1/oauth/.well-known/oauth-authorization-server/

Response:
{
    "issuer": "/api/v1/oauth",
    "authorization_endpoint": "/api/v1/oauth/api/v1/oauth/authorize/",
    "token_endpoint": "/api/v1/oauth/api/v1/oauth/token/",
    "userinfo_endpoint": "/api/v1/oauth/api/v1/oauth/userinfo/",
    "jwks_uri": "/api/v1/oauth/api/v1/oauth/.well-known/keys/",
    "response_types_supported": [
        "code",
        "token",
        "id_token",
        "id_token token",
        "code token",
        "code id_token",
        "code id_token token"
    ],
    "subject_types_supported": [
        "public"
    ],
    "id_token_signing_alg_values_supported": [
        "RS256"
    ],
    "token_endpoint_auth_methods_supported": [
        "client_secret_post",
        "client_secret_basic"
    ]
}
```

# Relying Party Logout (/oauth/logout/)

URL: `https://[server_name]/api/v1/oauth/logout/`

The endpoint allows a Relying Party to request logging out the end user and revoking access tokens, refresh tokens, and ID tokens.

If the user is logged in, all corresponding tokens will be revoked.

## Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| id_token_hint | ID token of the user | String | No | Either the user should be logged in or this field should be specified |
| client_id | Client identifier of the RP | String | No | Will be verified if specified |
| post_logout_redirect_uri | Post logout callback URI | String | No | Will be verified if specified |
| state | State used by RP that is sent to the post logout callback | String | No | Will not be verified.<br>Only sent back to the callback |

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/oauth/logout | Request a logout. |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

## Response codes

| Code | Response content | Description |
|---|---|---|
| 200 OK | **Logout Success Page** configurable in the Replacement Messages of the Relying Party | Logout is successful without `post_logout_redirect_uri`. |
| 400 Bad Request | **400 Bad Request** configurable in the System Replacement Messages | Bad Request, invalid ID token, expired ID token, missing ID token. |
| 302 redirect | | Logout is successfully redirected to the `post_logout_redirect_uri`. |

**Example**

Request a logout on a logged in user:

```
curl -k -v -X GET \
'https://[FAC_IP]/api/v1/oauth/logout/?id_token_hint=[ID token]'
curl -k -v -X GET \
'https://[FAC_IP]/api/v1/oauth/logout/?id_token_hint=[ID token]&client_id=[Client ID]'
curl -k -v -X GET \
'https://[FAC_IP]/api/v1/oauth/logout/?id_token_hint=[ID token]&client_id=[Client ID]&post_logout_
      redirect_uri=[Callback URL]'
```

# Push authentication status polling (/pushpoll/)

URL :`https://[server_name]/api/v1/pushpoll/`

Endpoint for querying the status of an FTM-push session.

An FTM-push means that a push notification has been sent to the user's FortiToken Manager mobile application.

This endpoint is meant to see whether or not the user has approved the push notification on their phone, and provide a way for an API user to prove that it has been accepted.

## Supported fields

| Fields | Display Name | Type | Required | Other Restrictions |
|---|---|---|---|---|
| s | Session Id | String | Yes | |

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/pushpoll/?s=<session_id> | Poll for status of push session |

## Response codes

In addition to the general codes defined in General API response codes on page 138, a POST request to this resource can also result in the following return codes:

| Code | Response content | Description |
|---|---|---|
| 200 OK | | Session is valid. |
| 400 Bad Request | | Bad Request, invalid session. |

# MAC devices (/macdevices/)

URL: `https://[server_name]/api/v1/macdevices/`

This endpoint represents the MAC device resource.

## Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| name | MAC device name | String | Yes | Max length = 50. |
| address | MAC address | String | Yes | Max length = 25. Must be a valid MAC address. Acceptable formats are `11:22:33:44:aa:bb` or `11223344aabb`. Vendor wildcards are also supported. Acceptable formats are `11:22:33` or `112233` or `11:22:33:xx:xx:xx` or `112233xxxxxx`. Letters in MAC addresses can be upper or lower case. |
| description | Description | String | No | Max length = 255. |

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/macdevices/ | Get all MAC devices. |
| GET | /api/v1/macdevices/[id]/ | Get specific MAC device with ID. |
| POST | /api/v1/macdevices/ | Create a new MAC device. |
| PATCH | /api/v1/macdevices/[id]/ | Update specific fields of specific MAC device with ID. |
| DELETE | /api/v1/macdevices/[id]/ | Delete MAC device with ID. |

## Allowed filters

| Field | Lookup expressions | Values |
|---|---|---|
| name | `exact, iexact, contains, icontains, startswith, istartswith` | |
| address | `exact, iexact, contains, icontains, startswith, istartswith` | |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# MAC groups (/macgroups/)

URL: `https://[server_name]/api/v1/macgroups/`

This endpoint represents MAC device group resource.

## Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| name | MAC group name | String | Yes | Max length = 50. |
| devices | MAC address | list | No | List of MAC devices URI. Read only. |

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/macgroups/ | Get all MAC device groups. |
| GET | /api/v1/macgroups/[id]/ | Get MAC device group with ID. |
| POST | /api/v1/macgroups/ | Create a new MAC device group. |
| DELETE | /api/v1/macgroups/[id]/ | Delete MAC device with ID. |

## Allowed filters

| Field | Lookup expressions | Values |
|---|---|---|
| name | `exact` | |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# MAC device group associations (/macgroup-memberships/)

URL: `https://[server_name]/api/v1/macgroup-memberships/`

This endpoint represents MAC device group memberships resource (relationship between MAC devices and MAC device groups).

## Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| group | MAC group | String | Yes | MAC device group URI. |
| device | MAC device | String | Yes | MAC device URI. |
| group_name | MAC group name | String | No | |
| device_name | MAC device name | String | No | |

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/macgroup-memberships/ | Get all MAC device group memberships. |
| GET | /api/v1/macgroup-memberships/[id]/ | Get MAC device group membership with ID. |
| POST | /api/v1/macgroup-memberships/ | Create a new MAC device group membership. |
| DELETE | /api/v1/macgroup-memberships/[id]/ | Delete MAC device with ID. |

## Allowed filters

| Field | Lookup expressions | Values |
|---|---|---|
| group_name | `exact` | |
| device_name | `exact` | |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# TACACS+ clients (/tacplusclients/)

URL: `https://[server_name]/api/v1/tacplusclients/`

This endpoint represents TACACS+ Client resource.

## Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| name | TACACS+ client name | String | Yes | Max length = 32. |
| address | TACACS+ IP address or subnet | String | Yes | Max length = 47. |
| secret | TACACS+ client secret | String | Yes | Max length = 63. |

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/tacplusclients/ | Get all Get all TACACS+ clients. |
| GET | /api/v1/tacplusclients/[id]/ | Get a specific TACACS+ client with ID. |
| POST | /api/v1/tacplusclients/ | Create a new TACACS+ client. |
| PUT | /api/v1/tacplusclients/[id]/ | Update an existing TACACS+ client specified with ID. |
| DELETE | /api/v1/tacplusclients/[id]/ | Delete the TACACS+ client corresponding to ID. |

## Allowed filters

| Field | Lookup expressions | Values |
|---|---|---|
| name | `exact` | |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# TACACS+ policies (/tacpluspolicies/)

URL: `https://[server_name]/api/v1/tacpluspolicies/`

This endpoint represents TACACS+ Policy resource.

## Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| name | TACACS+ policy name | String | No | Read Only. |

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/tacpluspolicies/ | Get all TACACS+ policies. |

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/tacpluspolicies/[id]/ | Get a specific TACACS+ policy with ID. |

## Allowed filters

| Field | Lookup expressions | Values |
|---|---|---|
| name | exact | |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# TACACS+ policy client association (/tacpluspolicyclient/)

URL: `https://[server_name]/api/v1/tacpluspolicyclient/`

This endpoint represents a resource for TACACS+ Policy/Client associations.

## Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| policy | TACACS+ policy URI | String | Yes | |
| client | TACACS+ client URI | String | Yes | |
| policy_name | TACACS+ policy name | String | No | Read only. |
| client_name | TACACS+ client name | String | No | Read only. |

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/tacpluspolicyclient/ | Get all TACACS+ policy/client associations |
| GET | /api/v1/tacpluspolicyclient/[id]/ | Get a specific TACACS+ policy/client association with ID |
| POST | /api/v1/tacpluspolicyclient/ | Create a new TACACS+ policy/client association. |
| DELETE | /api/v1/tacpluspolicyclient/[id]/ | Delete the TACACS+ policy/client association corresponding to ID |

## Allowed filters

| Field | Lookup expressions | Values |
|---|---|---|
| policy_name | `exact` | |
| client_name | `exact` | |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# Backup and restore (/recovery/)

URL: `https://[server_name]/api/v1/recovery/`

This endpoint is used to backup FAC configuration and restore from a configuration file.

## Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| file | Backup config file to restore from | String | Yes | |
| key | encryption/decryption key | String | No | |

# Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/recovery/ | Download config backup file. |
| POST | /api/v1/recovery/ | Restore from a config backup file. |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# Example

### JSON query

- Backup

```
curl -k -X GET https://[FAC_IP]/api/v1/recovery/
   -H 'Content-Type:application/json'
   -u "admin:[hash]"
   --output backup.conf
   -d '{"key":"XXXXXX"}'
```

### JSON query

- Restore

```
curl -k -X POST https://[FAC_IP]/api/v1/recovery/
   -H 'content-type: multipart/form-data'
   -u "admin:[hash]"
   -F 'file=@backup.conf'
   -F 'key=decryption_key'
```

### Response

< System configuration restore process has been started. FortiAuthenticator will be restarted to apply the new configuration

# IAM accounts (/iamaccounts/)

**URL:** `https://[server_name]/api/v1/iamaccounts/`

This end-point represents IAM account resource.

## Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| name | name | String | Yes | Unique. |
| alias | alias | String | Yes | Unique. |

## Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/iamaccounts/ | Get all IAM accounts. |
| GET | /api/v1/iamaccounts/[id]/ | Get a specific IAM account with ID. |
| POST | /api/v1/iamaccounts/ | Create a new IAM account. |
| PATCH | /api/v1/iamaccounts/[id]/ | Update specified fields for a specific IAM account with ID. |
| DELETE | /api/v1/iamaccounts/[id]/ | Delete a IAM account. |

## Allowed filters

| Field | Lookup expressions | Values |
|---|---|---|
| id | `exact, in` | |
| name | `exact` | |
| alias | `exact` | |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# IAM users (/iamusers/)

URL: `https://[server_name]/api/v1/iamusers/`

This end-point represents IAM user resource.

# Supported fields

| Field | Display name | Type | Required | Other restrictions |
|---|---|---|---|---|
| username | username | String | Yes | Unique within its IAM account. |
| account | account | Interger/String | Yes | Must be IAM account ID. |
| is_admin | is_admin | Boolean | No | true or false |
| localuser | localuser | Interger/String | Yes | Must be local user id and must be null if ldapuser is set. |
| ldapuser | ldapuser | Interger/String | Yes | Must be remote LDAP user id and must be null if localuser is set. |

# Allowed methods

| HTTP method | Resource URI | Action |
|---|---|---|
| GET | /api/v1/iamusers/ | Get all IAM users. |
| GET | /api/v1/iamusers/[id]/ | Get a specific IAM user with ID. |
| POST | /api/v1/iamusers/ | Create a new IAM user. |
| PATCH | /api/v1/iamusers/[id]/ | Update specific fields for a specific IAM user with ID. |
| DELETE | /api/v1/iamusers/[id]/ | Delete an IAM user. |

# Allowed filters

| Field | Lookup expressions | Values |
|---|---|---|
| id | `exact, in` | |
| username | `exact` | |
| account | `exact` | Account ID |
| localuser | `exact` | Local user ID |
| ldapuser | `exact` | LDAP user ID |

To modify query and response formats or filter results using the REST API, see Filtering query results on page 13.

# Advanced filtering

Results of the API calls can be controlled in several ways. Below are some arguments that can be passed to the REST API URL. Please refer to the specific resource documentation to find out which of these filter operations are allowed.

# General filters

General filters can be applied to most resources.

## Limits

limit: Limit number of items returned.

To search for the first entry in a resource:

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS"
      “https://192.168.0.122/api/v1/localusers/?format=json&limit=1”
```

The URL requires additional quoting in this case otherwise the Unix CLI treats the “&” as a instruction to place the cURL command into the background.

**Response**

```
< HTTP/1.1 200 OK
< Date: Tue, 10 Jun 2014 09:43:33 GMT
< Server: Apache
< Vary: Accept,Accept-Language,Cookie
< X-Frame-Options: SAMEORIGIN
< Content-Language: en
< Cache-Control: no-cache
< Transfer-Encoding: chunked
< Content-Type: application/json
<
* Connection #0 to host 192.168.0.122 left intact
* Closing connection #0
{"meta": {"limit": 1, "next": "/api/v1/localusers/?offset=1&limit=1&format=json", "offset": 0,
      "previous": null, "total_count": 3}, "objects": [{"address": "", "city": "", "country": "",
      "custom1": "", "custom2": "", "custom3": "", "email": "", "first_name": "", "id": 5, "last_
      name": "", "mobile_number": "", "phone_number": "", "resource_uri": "/api/v1/localusers/5/",
      "state": "", "token_auth": false, "token_serial": "", "token_type": null, "user_groups":
      ["/api/v1/usergroups/9/", "/api/v1/usergroups/8/"], "username": "test_user2"}]}
```

Only the first user in the list is returned. Note that this excludes admin users which are never returned by this query hence the reason why this user ID is > 5.

# Offset

**offset**: Specify an offset for the returned items (zero-based). E.g. if there are 10 items, to return item #5 - #10 only, specify offset=4:

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS"
      “https://192.168.0.122/api/v1/localusers/?format=json&offset=4”
```

# Order

**order_by**: Order returned list by a known field name (e.g. ?order_by=<field name>):

```
curl -k -v -u "admin:zeyDZXmP6GbKcerqdWWEYNTnH2TaOCz5HTp2dAVS"
      “https://192.168.0.122/api/v1/localusers/?format=json&order_by=username”
```

**Response**

```
< HTTP/1.1 200 OK
< Date: Tue, 10 Jun 2014 16:41:23 GMT
< Server: Apache
< Vary: Accept,Accept-Language,Cookie
< X-Frame-Options: SAMEORIGIN
< Content-Language: en
< Cache-Control: no-cache
< Transfer-Encoding: chunked
< Content-Type: application/json
<
{"meta": {"limit": 20, "next": null, "offset": 0, "previous": null, "total_count": 3}, "objects":
      [{"address": "", "city": "", "country": "", "custom1": "", "custom2": "", "custom3": "",
      "email": "", "first_name": "", "id": 4, "last_name": "", "mobile_number": "", "phone_number":
      "", "resource_uri": "/api/v1/localusers/4/", "state": "", "token_auth": false, "token_serial":
      "", "token_type": null, "user_groups": ["/api/v1/usergroups/8/"], "username": "test_user"},
      {"address": "", "city": "", "country": "GB", "custom1": "example", "custom2": "", "custom3":
      "", "email": "", "first_name": "", "id": 5, "last_name": "", "mobile_number": "", "phone_
      number": "", "resource_uri": "/api/v1/localusers/5/", "state": "", "token_auth": false,
      "token_serial": "", "token_type": null, "user_groups": ["/api/v1/usergroups/9/",
      "/api/v1/usergroups/8/"], "username": "test_user2"}, {"address": "", "city": "", "country":
      "GB", "custom1": "example", "custom2": "", "custom3": "", "email": "test_user3@example.com",
      "first_name": "", "id": 6, "last_name": "", "mobile_number": "", "phone_number": "",
      "resource_uri": "/api/v1/localusers/6/", "state": "", "token_auth": false, "token_serial": "",
      "token_type": null, "user_groups": [], "username": "test_user3"}]}
```

# Filter lookup expressions

| Expression | Description |
|---|---|
| `exact` | Search for an exact match (e.g. `name__exact=John Doe`, would return user with name "John Doe", but not "john doe") |
| `iexact` | Search for a case-insensitive exact match (e.g. `name__iexact=john doe`, would return user with name "John Doe") |

| Expression | Description |
|---|---|
| `contains` | Search for an item that contains a specific keyword |
| `icontains` | Same as above, but case-insensitive |
| `in` | Search for items that matches specific filter criteria (e.g. to return items that has a name matching "John" or "Bill", `?name__in=John&name__in=Bill`) |
| `startswith` | Search for items that starts with a text |
| `istartswith` | Same as above, but case-insensitive |

# General API response codes

| Code | Description |
|---|---|
| **200 OK** | The request was successfully completed. |
| **201 Created** | The request successfully created a new resource and the response body does not contain the newly created resource. |
| **202 Accepted** | The server fulfilled the request and the response body contains the newly updated resource. |
| **204 No Content** | The server fulfilled the request, but does not need to return a response message body. |
| **400 Bad Request** | The request could not be processed because it contains missing or invalid information (i.e. the data in the request does not validate). |
| **401 Not Authorized** | The supplied credential is incorrect. |
| **403 Forbidden** | Permission is denied to perform an operation. |
| **500 Internal Server Error** | The server encountered an unexpected condition which prevented it from fulfilling the request. |

www.fortinet.com