REST API

FortiToken Cloud 24.2.a

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

June 6, 2024
FortiToken Cloud 24.2.a REST API

TABLE OF CONTENTS

Overview **5**
Introduction 6
Audience 7
Authentication 8
Generate new applications 9
Configuration 10
General web app 11
Management app 12
Access FortiToken Cloud RESTful API 13
Client ID/secret authentication 14
API request general information 15
Format of URL 16
Format of URL with parameters 17
Supported HTTP methods 18
HTTP header 19
API workflow 20
Resource API summary 21
General web applications 22
Management applications 23
General error response codes 24
API QuickStart Guide **25**
Web Applications 26
Create a general web application from the FTC Portal 27
Get the access token and create users in the web application 28
Management applications 38
Create a management application from the portal 39
Get access token and manage users from management applications 40
RESTful API References **43**
Web application log-in 44
POST 45
Realm 47
GET 48
User 50
GET 51
POST 54
PUT 57
DELETE 60
User base 61
GET 62
PUT 66
DELETE 70
User authentication 71
GET 72

POST 74
Sync user data to FTC 78
POST 79
Query asynchronous task status 81
GET 82
Token 84
GET 85
Temporary token 87
POST 88
DELETE 90
Balance 91
GET 92
Count resource 93
GET 94
Version 96
GET 97
User Quota Allocation per Realm 98
GET 99
Pagination 101
Realm 101
User 103
User base 105
Change log 109

Overview

- Introduction on page 6
- Audience on page 7
- Authentication on page 8
- Generate new applications on page 9
- Configuration on page 10
- Client ID/secret authentication on page 14
- Access FortiToken Cloud RESTful API on page 13
- API request general information on page 15
- Format of URL on page 16
- Format of URL with parameters on page 17
- Supported HTTP methods on page 18
- HTTP header on page 19
- API workflow on page 20
- Resource API summary on page 21
- General error response codes on page 24

Introduction

An API (Application Programming Interface) is a set of defined interfaces to accomplish a task, such as retrieving or modifying data. FortiToken Cloud provides a Representational State Transfer (REST) API for interaction with its system components.

This document introduces the supported FortiToken Cloud REST API methods for Web applications, and provides instructions on how to configure and use it.

Audience

General web applications and management applications.

General web applications have the same functions as Fortinet applications, including creating, editing, and authenticating users. They provides the flexibility to handle authentication in a customized manner.

Management applications provide an alternative to the FTC portal where one can access and manage all the users and tokens. Management applications manage user bases directly, in contrast to general web applications where only local user references are accessible.

Authentication

A client ID and secret combination is required to authenticate for FortiToken Cloud Rest API clients.

Generate new applications

To register for the FortiToken Cloud Rest API client and get the client ID/secret, the FortiToken Cloud customer admin must log in to the FortiToken Cloud portal.

To add a general web application, go to the *Applications>Web Applications* page, and click the *Add Web Application* button.

To add a management application, go to the *applications>Mgmt Application*s page, and click the *Add Mgmt Application* button.

The ID and secret will be displayed once on the portal or emailed to the new API client. The client will use the client ID and the secret for future API authentication.

Configuration

Each web application has its own name, client ID, and secret for authentication. There are unique properties to configure for different types of web application.

General web app

A general web application has its own adaptive auth profile and auth scope.

- An adaptive auth profile is a set of rules that are customized to filter and handle authentications based on their IP origins, EMS tags, and the like.
- The auth scope represents the user scope that this particular application can authenticate. If set to “Self”, this web app can only authenticate its local users; if set to “Realm”, it can authenticate users in the whole realm.

Management app

A management application has its own management scope.

The management scope of a management application can be set to customer level or realm level. If set to “Customer Level”, then this application has an admin permission to access all the realms under the account. If set to “Realm Level”, then it can only access the realms that are specified in the managed realms property.

Access FortiToken Cloud RESTful API

Any authenticated web application can access the FortiToken Cloud API.

Client ID/secret authentication

The following outlines the process in which a Web application client passes FortiToken Cloud authentication:

1. The API client calls the Login API with its client ID and secret to request an access token from FortiToken Cloud.
2. FortiToken Cloud recognizes the client ID and secret and responds with an access token.
3. The API client puts the access token into the API request authorization header.
4. FortiToken Cloud checks the API request authorization header to finalize the authentication.
5. The access token remains valid for an hour, after which the client must call the Login API again to renew the access token.

API request general information

FortiToken Cloud API is based on Hypertext Transfer Protocol (HTTP 1.1 and later). The message includes:

- HTTP method
- HTTP URL
- HTTP header and HTTP body

The supported HTTP body format is .JSON. Not all the aforementioned components are required for all HTTP methods, as indicated in the following table:

HTTP Method	RFC	Request Body	Response Body
GET	RFC 7231	No	Yes
POST	RFC 7231	Yes	Yes
PUT	RFC 7231	Yes	Yes
DELETE	RFC 7231	No	No

Format of URL

All the resource URLs are in the following format:

```
https://ftc.fortinet.com:[port]/[api/version/resource]
```

URL Component	Description
Port	The default service port for external API service is 9696.
API	REST API.
Version	The supported API version. Currently, it is “v1”.
Resource	The resource to be accessed.

Format of URL with parameters

URL?param1=***&param2=***&param3=***…

URL Component	Description
?	Denotes that there will be input fields to be passed in to the URL.
paramX	An input field which you are passing in to call the API. Supported input fields are given in RESTful API References on page 43.
***	The actual value of the input field you are passing in.
&	Denotes that there will be an additional input field to be passed in.

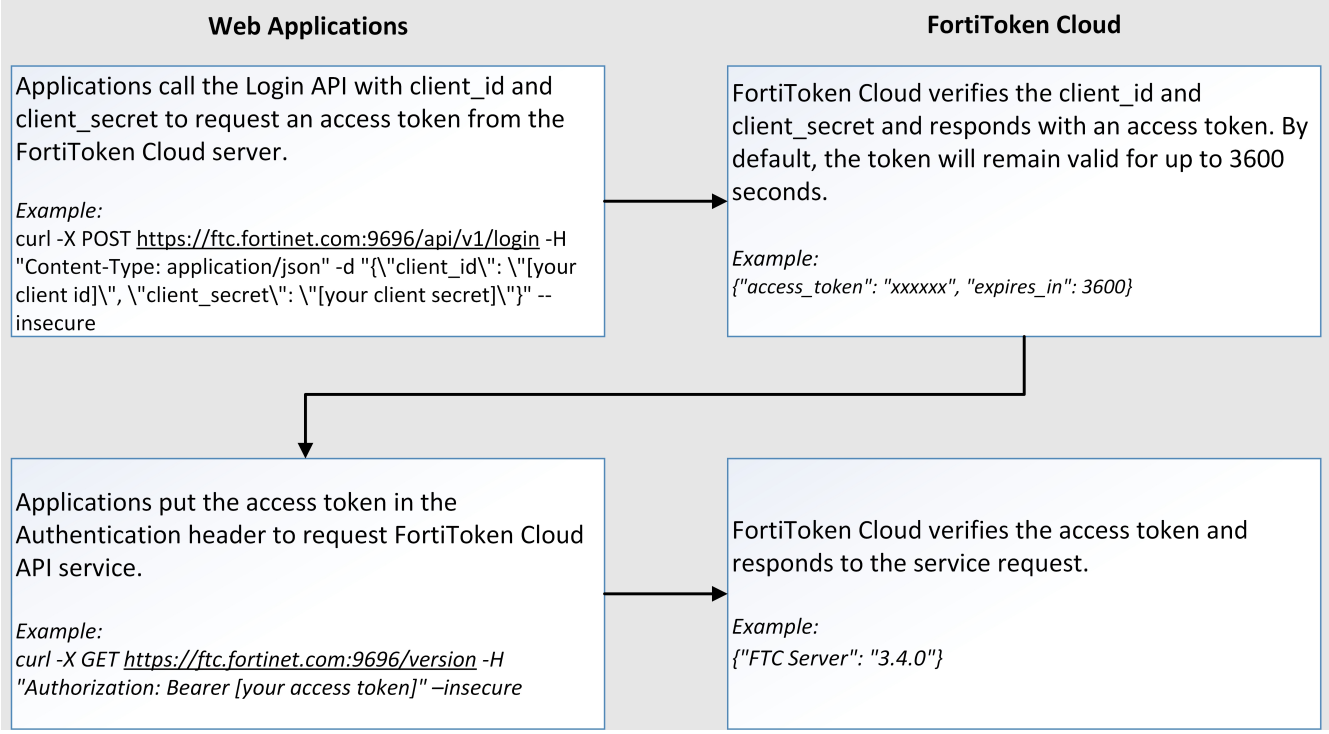
Supported HTTP methods

Method	Description
GET	Retrieves a specific resource with an ID or other fields as the filter.
POST	Creates a new resource on a given endpoint.
PUT	Updates an existing item specified with an ID.
DELETE	Removes an existing item specified with an ID.

HTTP header

- Content-type: `application/json`
- Authorization header: `"Authorization: Bearer [access_token]"`
- Access token: Bearer token, all supported APIs except the Login API need the authorization header.

API workflow



Resource API summary

The following is a summary of the supported APIs for general web applications and management applications. All supported APIs except the Login API must be accessed with the authorization header.

General web applications

Resource	Resource in URI	HTTP methods
Login	/api/v1/login	POST
Realm	/api/v1/realm	GET
User	/api/v1/user	GET, POST, PUT, DELETE
User Authentication	/api/v1/auth	GET, POST
Sync user to FTC	/api/v1/user_sync	POST
Task status	/api/v1/task	GET
Token	/api/v1/token	GET
Token Temp	/api/v1/token/temp	POST, DELETE
Quota	/api/v1/quota	GET
Count resource	/api/v1/count	GET
Balance	/api/v1/balance	GET
Version	/version	GET

Management applications

Resource	Resource in URI	HTTP methods
Login	/api/v1/login	POST
Realm	/api/v1/realm	GET
User	/api/v1/user	PUT
User Base	/api/v1/user/base	GET, PUT, DELETE
Task status	/api/v1/task	GET
Token	/api/v1/token	GET
Token Temp	/api/v1/token/temp	POST, DELETE
Quota	/api/v1/quota	GET
Count resource	/api/v1/count	GET
Balance	/api/v1/balance	GET
Version	/version	GET

General error response codes

The response codes listed here may be seen from any of the API calls and are not specific to a single one.

Error code	Description
400	Bad request.
401	Unauthorized.
403	Access denied.
404	Not found.
405	Method not allowed.
409	Conflict.

API QuickStart Guide

Web Applications

This section shows how:

1. Create a general web application from the Portal.
2. Get the access token and create users in the web application.

Create a general web application from the FTC Portal

Use the following steps to generate a client ID and secret from the FortiToken Cloud portal.

1. Log in to the portal, and select *Applications>Web Applications>Add Web Application.*
2. Name the web app, select a realm and auth scope, set up an optional adaptive auth profile, and click *Save.*
3. Note down the client ID and secret, and click *OK.*

Get the access token and create users in the web application

This section shows how to get the access token and use it to create users in:

- Python examples on page 29
- Curl examples on page 32
- Postman examples on page 34

Python examples

- Get access token on page 29
- Create user with access token on page 29
- Get a list of all users with access token on page 30
- Get user via id with access token on page 30
- FTM user authentication with access token on page 30
- FTM user OTP code authentication with access token on page 30
- Check user authentication status with access token on page 31
- Delete user with access token on page 31

Get access token

The following code example shows how to get an access token.

```
import requests


client_id = 'ec601e2a-73bbxxxxxxxx'
client_secret = 'mRR6ciFnKxxxxxxxxx'

payload = {
    'client_id': client_id,
    'client_secret': client_secret
}

resp = requests.request('post', 'https://ftc.fortinet.com:9696/api/v1/login', json=payload,
verify=False)

access_token = resp.json()['access_token']
```

Create user with access token

The following shows how to create a user with the access token.

```
import requests


headers = {
    'Authorization': 'Bearer eyJ0ex*******Q6to',
    'Content-Type': 'application/json'
}

payload = {
    'username': 'ftc_webapp_user',
    'email': '***@gmail.com'
}

resp = requests.request('post', 'https://ftc.fortinet.com:9696/api/v1/user',
headers=headers, json=payload, verify=False)
```

Get a list of all users with access token

```
import requests

headers = {
   'Authorization': 'Bearer eyJ0ex*******Q6to',
   'Content-Type': 'application/json'

payload = {}
resp = requests.request('get', 'https://ftc.fortinet.com:9696/api/v1/user', headers=headers,
json=payload, verify=False)
```

Get user via id with access token

```
import requests

headers = {
   'Authorization': 'Bearer eyJ0ex*******Q6to',
   'Content-Type': 'application/json'

payload = {}
resp = requests.request('get', 'https://ftc.fortinet.com:9696/api/v1/user/[input user
reference ID here]', headers=headers, json=payload, verify=False)
```

FTM user authentication with access token

```
import requests

headers = {
   'Authorization': 'Bearer eyJ0ex*******Q6to',
   'Content-Type': 'application/json'

payload = {
   "username":"ftc_webapp_user",
}
resp = requests.request('post', 'https://ftc.fortinet.com:9696/api/v1/auth',
headers=headers, json=payload, verify=False)
```

FTM user OTP code authentication with access token

You can also include the 'auth_method' field in the payload to specify whether the OTP code should be sent via FTM, SMS, email, etc..

```
import requests

headers = {
   'Authorization': 'Bearer eyJ0ex*******Q6to',
   'Content-Type': 'application/json'

payload = {
   "username":"ftc_webapp_user",
   "token":"123456"
}
```

```
resp = requests.request('post', 'https://ftc.fortinet.com:9696/api/v1/auth',
headers=headers, json=payload, verify=False)
```

Check user authentication status with access token

```
import requests

headers = {
   'Authorization': 'Bearer eyJ0ex*******Q6to',
   'Content-Type': 'application/json'

payload = {}
resp = requests.request('get', 'https://ftc.fortinet.com:9696/api/v1/auth/[input auth ID
here]', headers=headers, json=payload, verify=False)
```

Delete user with access token

```
import requests

headers = {
   'Authorization': 'Bearer eyJ0ex*******Q6to',
   'Content-Type': 'application/json'

payload = {}
resp = requests.request('delete', 'https://ftc.fortinet.com:9696/api/v1/auth/[input user
reference ID here]', headers=headers, json=payload, verify=False)
```

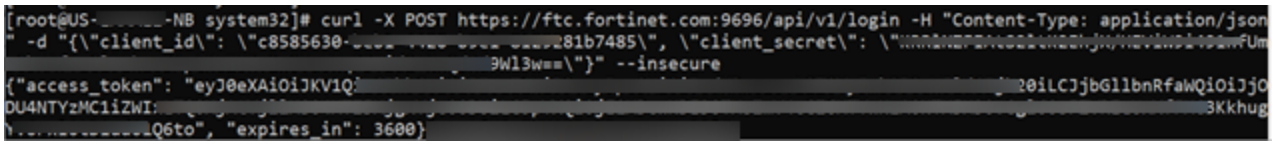
Curl examples

- Linux
- Windows Command Prompt

Linux

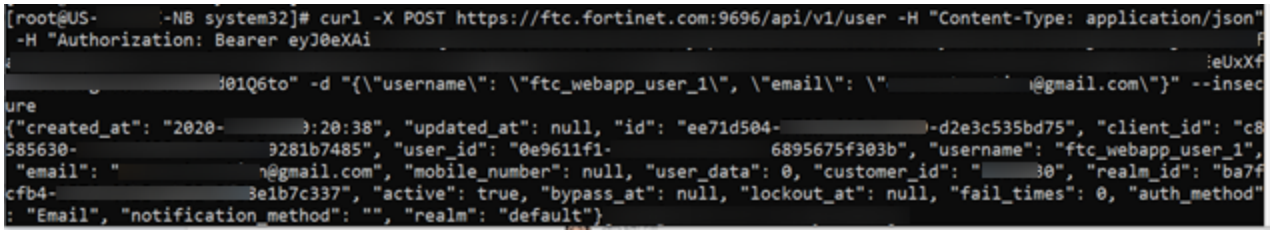
Get access token

```
curl -X POST https://ftc.fortinet.com:9696/api/v1/login -H "Content-Type: application/json"
-d "{\"client_id\": \"c8585630**************81b7485\", \"client_secret\":
\"************************************Wl3w==\"}" --insecure
```



Create user with access token

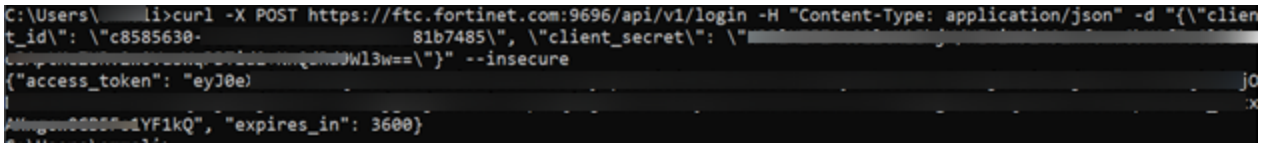
```
curl -X POST https://ftc.fortinet.com:9696/api/v1/user -H "Content-Type: application/json" -
H "Authorization: Bearer
eyJ0eXAi************************************************************01Q6to" -d "{\"username\":
\"ftc_webapp_user_1\", \"email\": \"***@gmail.com\"}" -insecure
```



Windows Command Prompt

Get access token

```
curl -X POST https://ftc.fortinet.com:9696/api/v1/login -H "Content-Type: application/json"
-d "{\"client_id\": \"c8585630-**************81b7485\", \"client_secret\":
\"************************************Wl3w==\"}" --insecure
```



Create user with access token

```
curl -X POST https://ftc.fortinet.com:9696/api/v1/user -H "Content-Type: application/json" -
H "Authorization: Bearer eyJ0e************************************1YF1kQ" -d "{\"username\":
```

```
\"ftc_webapp_user_2\", \"email\": \"***@gmail.com\"}" --insecure
```

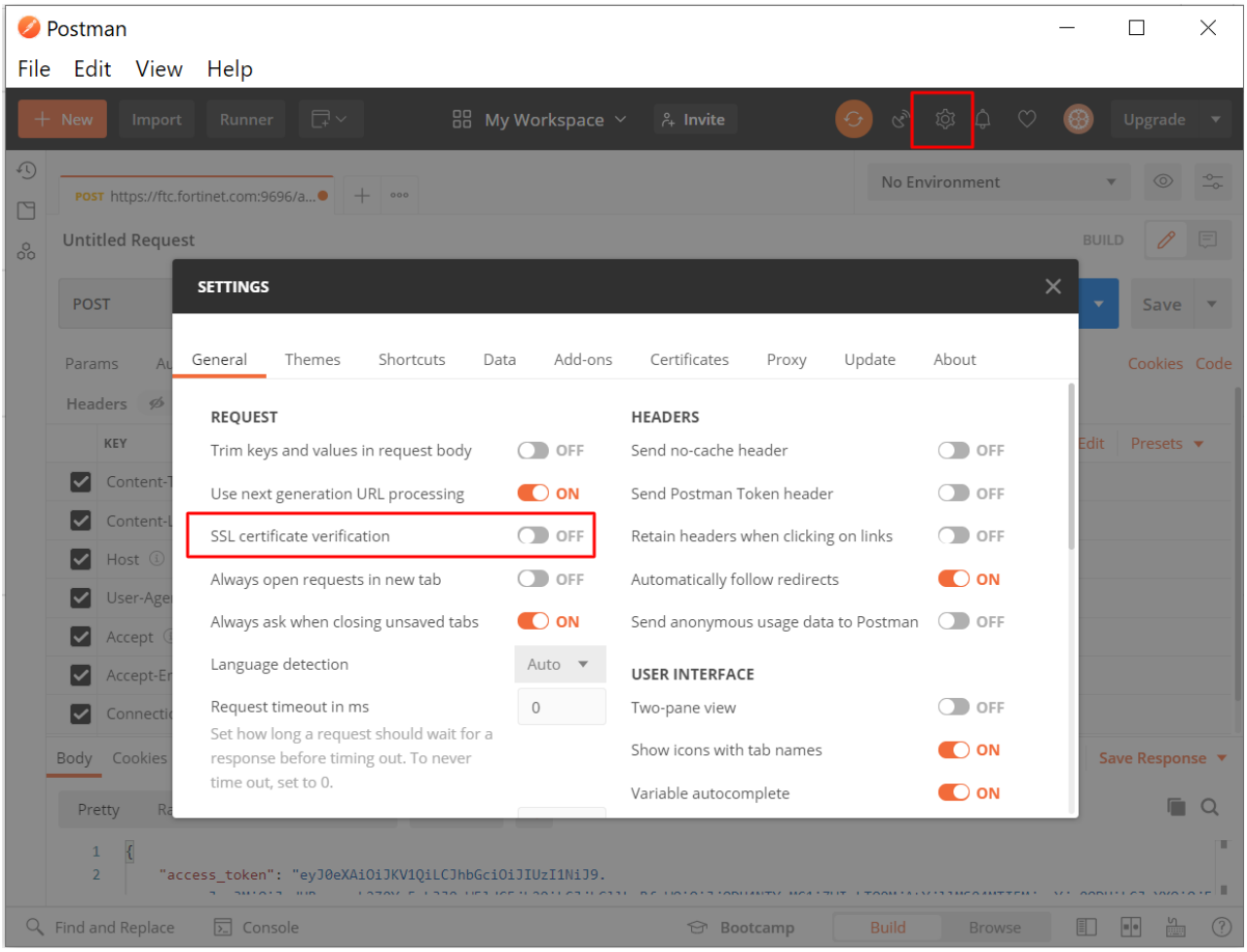
```
C:\Users\     i>curl -X POST https://ftc.fortinet.com:9696/api/v1/user -H "Content-Type: application/json" -H "Authoriza
tion: Bearer eyJ0e
\
      1YF1kQ" -d "{\"username\": \"ftc_webapp_user_2\", \"email\": \"          @gmail.com\"}" --insecure
{"created_at": "2020-      00:24:48", "updated_at": null, "id": "77efc181                    5b192f82", "client_id": "c8
585630-                  9281b7485", "user_id": "50cddc1                  f1f51e4c3183", "username": "ftc_webapp_user_2",
 "email": "          n@gmail.com", "mobile_number": null, "user_data": 0, "customer_id": "     0", "realm_id": "ba7f
c                    b7c337", "active": true, "bypass_at": null, "lockout_at": null, "fail_times": 0, "auth_method"
: "Email", "notification_method": "", "realm": "default"}
```

Postman examples

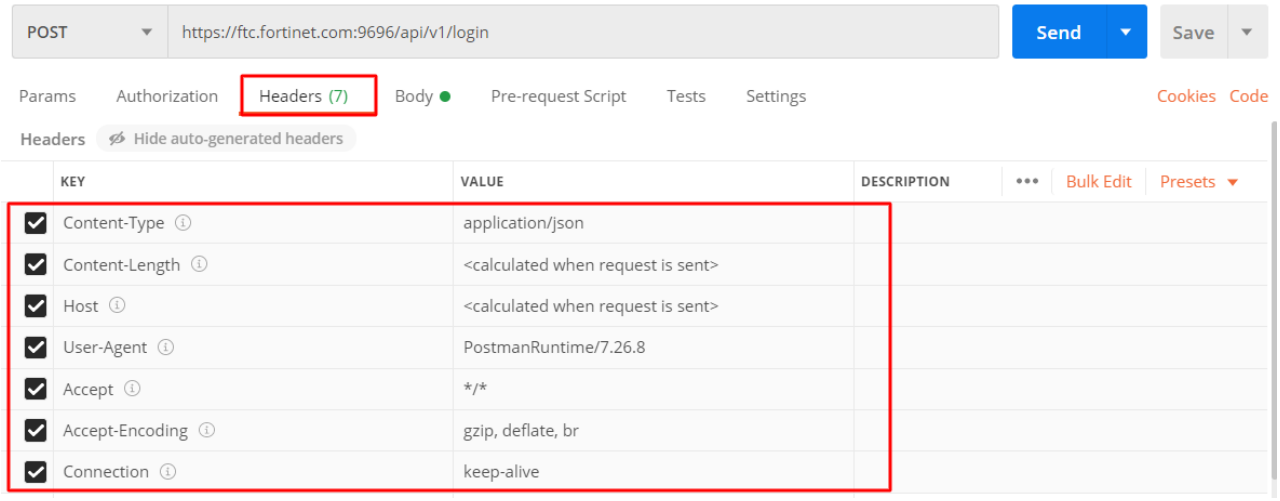
- Get access token
- Create user with access token

Get access token

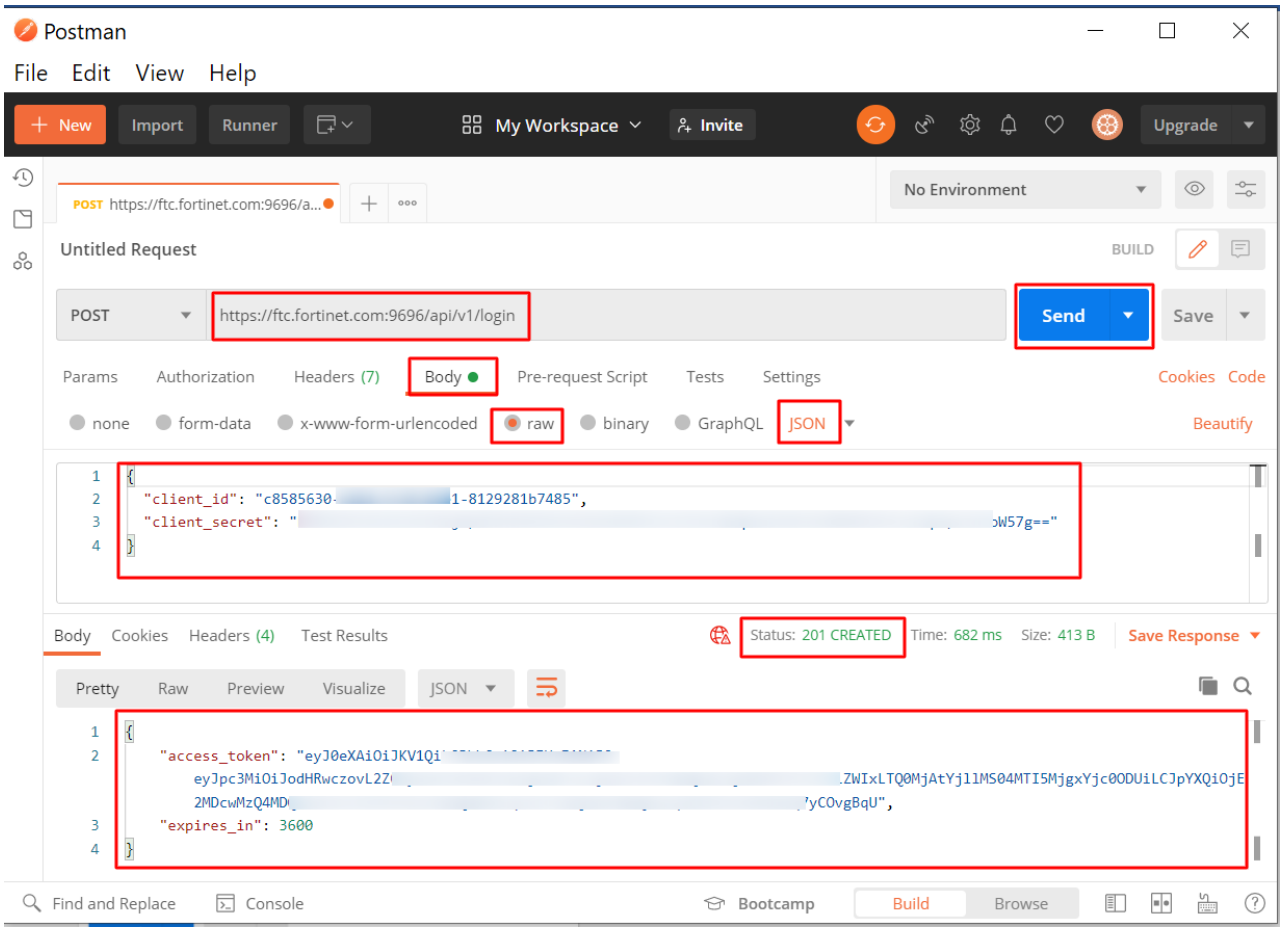
1. Update Postman setting, and turn off “SSL certificate verification”.



2. Request Headers setting.

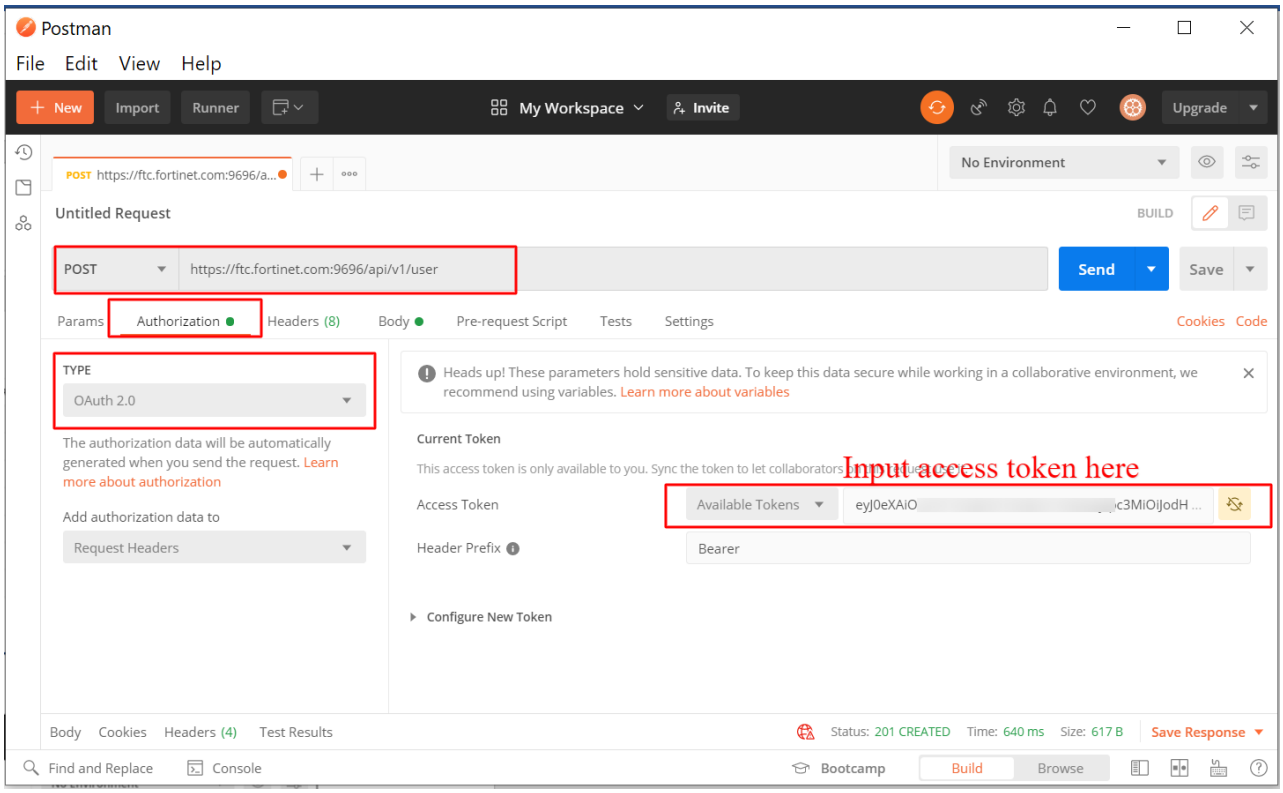


3. Request Body format and response.

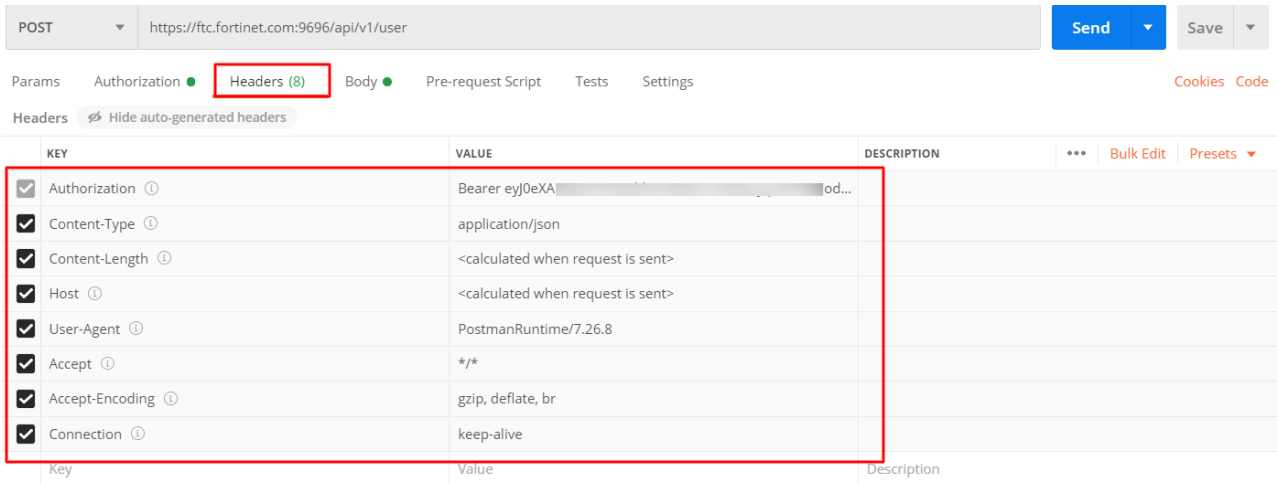


Create user with access token

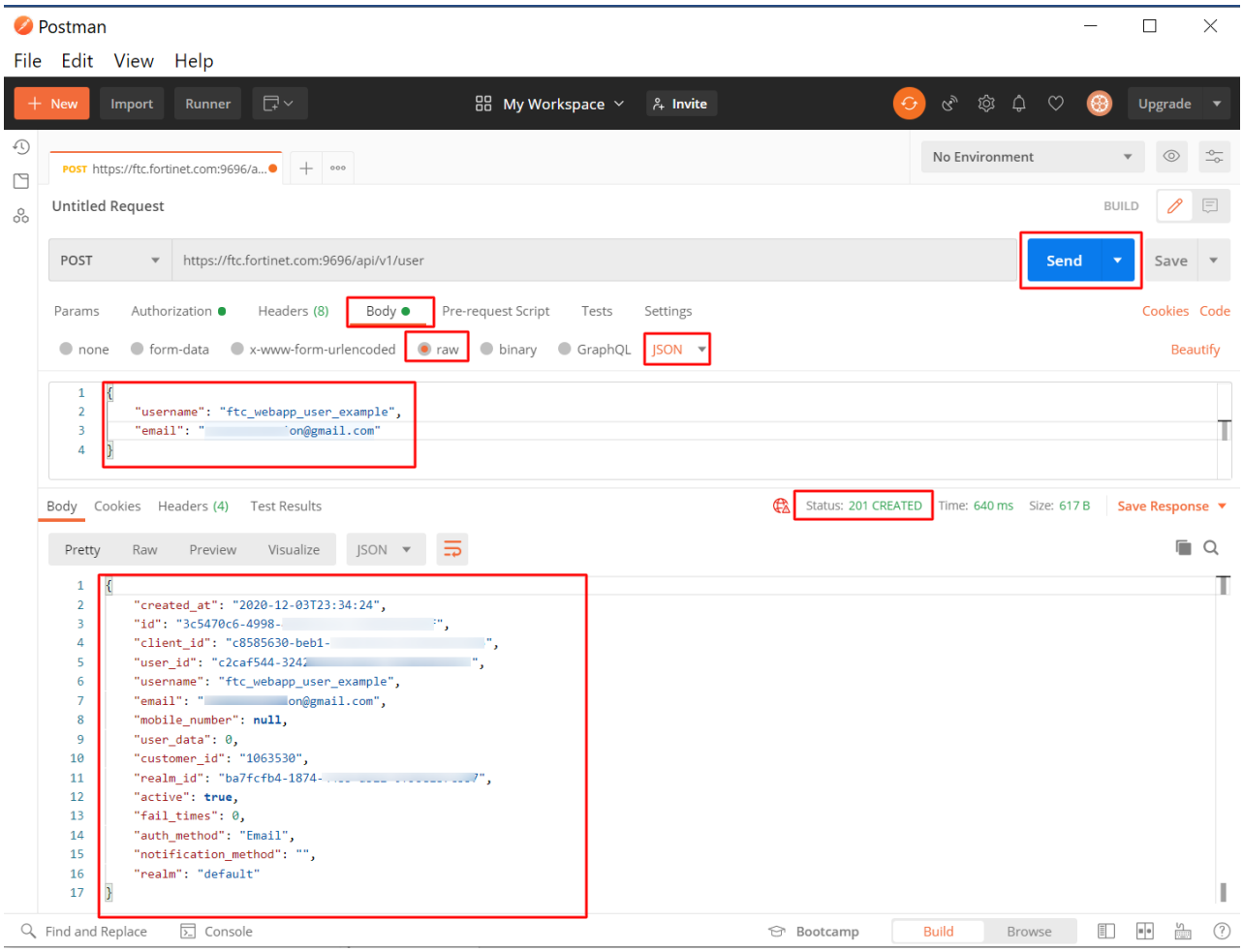
1. Input the access token for authorization.



2. Configure the request Headers.



3. Configure the request Body and response.



Management applications

1. Create a management application from the portal on page 39
2. Get access token and manage users from management applications on page 40

Create a management application from the portal

1. Log in to the Portal, and then select *Applications>Mgmt Applications>Add Mgmt Application.*
2. Name the application, select a management scope and a realm(s) if you have chosen Realm Level, and click *Save.*
3. Note down the client ID and secret, and click *OK.*

Get access token and manage users from management applications

This section shows how to get the access token and use it to create users in:

- Python examples on page 41
- Curl examples (Linux) on page 42

Python examples

- Get access token on page 41
- Get a list of all users with access token on page 41
- Assign new token for user with access token on page 41

Get access token

```
import requests


client_id = 'ec601e2a-73bbxxxxxxxx' client_secret = 'mRR6ciFnKxxxxxxxxx'

payload = {
'client_id': client_id, 'client_secret': client_secret
}

resp = requests.request('post', 'https://ftc.fortinet.com:9696/api/v1/login', json=payload,
verify=False)

access_token = resp.json()['access_token']
```

Get a list of all users with access token

```
import requests


headers = {
'Authorization': 'Bearer eyJ0ex*******Q6to', 'Content-Type': 'application/json'
}

payload = {}

resp = requests.request('get', 'https://ftc.fortinet.com:9696/api/v1/user/base',
headers=headers, json=payload, verify=False)
```

Assign new token for user with access token

```
import requests headers = {
‘Authorization’: ‘Bearer eyJ0ex*******Q6to’, ‘Content-Type’: ‘application/json’

payload = {
'change_token': true,
}
resp = requests.request('put', 'https://ftc.fortinet.com:9696/api/v1/user/[input user
reference ID here]', headers=headers, json=payload, verify=False)
```

Curl examples (Linux)

- Get access token on page 42
- Create user with access token on page 42

Get access token

```
curl -X POST https://ftc.fortinet.com:9696/api/v1/login -H "Content-Type: application/json"
-d "{\"client_id\": \"c8585630**************81b7485\", \"client_ secret\":
\"**********************************Wl3w==\"}" --insecure
```

Create user with access token

```
curl -X GET https://ftc.fortinet.com:9696/api/v1/user/base -H "Content-Type:
application/json" -H "Authorization: Bearer
eyJ0eXAi*************************************************************01Q6to" –insecure
```

RESTful API References

Web application log-in

POST

URI:

- /api/v1/login

Input fields:

Field Name	Type	Required	Description
client_id	string	Yes	The client_id.
client_secret	string	Yes	The client_secret.

Successful response fields:

Field name	Type	Description
access_token	string	The access token for web API client to make API request.
expires_in	integer	The remaining time (in seconds) that access token still valid.

Response codes

Code	Response content	Description
201 OK		The access token was generated successfully.
400 Bad Request	The client secret format is invalid.	The client secret in the input field is invalid.
401 Unauthorized	The client's login failed due to wrong ID or secret.	Wrong or expired client ID or client secret.
404 Not Found	The client does not exist.	The client is not in FortiToken Cloud database.

Example request body:

```
{
     'client_id': '12345678-abcd-efgh-1234-xxxxxxxxxxxx',
     'client_secret': 'xxxxxx'
}
```

Example of successful response body:

```
Return:
{
     'access_token': 'xxxxxx',
     'expires_in': 2
}
```

This API is only used for a client to get an access token to finalize authentication for the API service. By default, the access token expires in an hour (i.e., 3600 seconds). The client does not need to call the API when it has an active access token.

Realm

GET

URI:

- /api/v1/realm
- /api/v1/realm/[id]
- /api/v1/realm?name=***

Input fields:

Field Name	Type	Required	Description
name	string	No	The realm name.
id	string	No	The realm ID.
deleted	boolean	No	Get the deleted realm record.

Successful response fields:

Field name	Type	Description
description	string	A brief description about the realm.
deleted_at	datetime	The deleted timestamp, if the realm is not deleted, the field is None.
name	string	The realm name.
is_default	boolean	If the realm is the default realm.
id	string	The realm ID.

Response codes:

Code	Response content	Description
200 OK		Realm(s) info are successfully returned.
404 Not Found	Realm ID was not found	Cannot find realm with Realm ID.

Example request body:

```
None
```

Example of successful response body:

```
Return:
[
      {
            'description': 'abc',
            'deleted_at': None,
            'name': 'abc',
```

```
            'is_default': False,
            'id': '12345678-abcd-efgh-1234-xxxxxxxxxxxx'
        }
]
```

The response body will be a single realm dictionary if the API input field is set to request a specific realm, for example, get realm by its ID. Otherwise, the response body will be a list of realm dictionaries.

User

When attempting to use GET, PUT and DELETE FTC API calls for the Management App application, be sure to use /user/base. For merging user aliases, Be sure to use the regular /user PUT API call.

GET

URI:

- /api/v1/user?username=***&realm_id=***
- /api/v1/user/[id]
- /api/v1/user

Input fields:

Field Name	Type	Required	Description
id	string	No	The user ID.
realm_id	string	No	The realm ID.
username	string	No	
email	string	No	The user's email address.
mobile_number	string	No	The user's mobile phone number.
active	boolean	No	Get active user.
bypass_at	datetime	No	The user's bypass date and time.
lockout_at	datetime	No	The user's lockout date and time.
user_data	integer	No	User type, which can be either of the following: • 0 for local user • 1 for remote LDAP user
auth_method	string	No	The authentication method, which can be one of the following: • 'FTM' • 'Email' • 'SMS' • 'FTK'
notification_ method	string	No	Supported notification method, which can be either of the following: • 'Email' • 'SMS'
brief	boolean	No	The API response only includes the mobile_number, username, email, vdom, realm, id, user_data if brief=True.
case_accent_ sensitive	boolean	No	Whether the username is case-sensitive.

Successful response fields:

Field name	Type	Description
customer_id	string	The customer ID.
client_id	string	The client ID.
user_id	string	The user ID.
notification_method	string	Supported notification method, which can be either of the following: • 'Email' • 'SMS'
auth_method	string	The authentication method, which can be one of the following: • 'FTM' • 'Email' • 'SMS' • 'FTK'
username	string	The user name.
id	string	The userref ID.
realm_id	string	The realm ID.
mobile_number	string	The user's mobile phone number.
email	string	The user's email address.
active	boolean	If user is active.
created_at	string	The timestamp when user was created.
updated_at	string	The timestamp when user was last updated.
bypass_at	string	The timestamp when user was bypassed
lockout_at	string	The timestamp when user was locked out
fail_times	integer	The failed attempts of user login validation
user_data	integer	User type, which can be either of the following: • 0 for local user • 1 for remote LDAP user
temp_token	boolean	Whether temporary token is enabled for the user.

Response codes:

Code	Response content	Description
200 OK		User(s) info is(are) successfully returned.
400 Bad Request	application was not found.	
402 Payment	No valid FTC subscription was found.	Please enable free trial or register licenses.

Code	Response content	Description
Required		
403 Forbidden	You are not authorized to access user.	
404 Not Found	• Cannot find related customer info • User does not exist.	

Example request body:

```
None
```

Example of successful response body:


```
[
   {
      "auth_method":"FTM",
      "client_id":"12345678-abcd-efgh-1234-xxxxxxxxxxxx",
      "created_at":"2020-02-02T02:22:22",
      "customer_id":"123456",
      "email":"abc@fortinet.com",
      "user_data":0,
      "id":"12345678-abcd-efgh-1234-xxxxxxxxxxxx",
      "lockout_at":"None",
      "mobile_number":"1123456789",
      "notification_method":"Email",
      "user_id":"12345678-abcd-efgh-1234-xxxxxxxxxxxx",
      "realm_id":"12345678-abcd-efgh-1234-xxxxxxxxxxxx",
      "username":"abc",
      "updated_at":"None",
      "active":true,
      "temp_token":false,
      "bypass_at":"None",
      "fail_times":0
   }
]
```

POST

URI:

- /api/v1/user

Input fields:

Field Name	Type	Required	Description
username	string	Yes	The user's username.
email	string	Yes	The user's email address.
mobile_number	string	No	The user's mobile phone number.
license	string	No	FTM token migration license number.
token_data	string	No	FTM token migration data.
token	string	No	The user token serial number.
activation_code	boolean	No	If skip token migration notification
user_data	integer	No	User type, which can be either of • 0 for local user • 1 for remote LDAP user
auth_method	string	No	The authentication method, which can be one of the following: • 'FTM' • 'Email' • 'SMS' • 'FTK'
notification_ method	string	No	Supported notification method, which can either of the following: • 'Email' • 'SMS'
brief	boolean	No	The API response only includes the mobile_number, username, email, vdom, realm, id, user_data if brief=True.
case_accent_ sensitive	boolean	No	Wheher the username is case-sensitive.

Successful response fields:

Field name	Type	Description
customer_id	string	The customer ID.

Field name	Type	Description
client_id	string	The client ID.
user_id	string	The user ID.
notification_method	string	Supported notification method, which can be either of the following: • 'Email' • 'SMS'
auth_method	string	The authentication method, which can be one of the following: • 'FTM' • 'Email' • 'SMS' • 'FTK'
username	string	The user's username.
id	string	The userref ID.
realm_id	string	The realm ID.
mobile_number	string	The user's mobile phone number.
email	string	The user's email address.
active	boolean	Whether the user is active.
created_at	string	The timestamp when user was created.
updated_at	string	The timestamp when user was last updated.
bypass_at	string	The timestamp when user was bypassed
lockout_at	string	The timestamp when user was locked out
fail_times	integer	The failed attempts of user login validation.
user_data	integer	User type, which can be either of the following: • 0 for local user • 1 for remote LDAP user
temp_token	boolean	Whether temporary token is enabled for the user.

Response codes:

Code	Response content	Description
201 OK		The user is successfully created.
400 Bad Request	• Cannot identify related customer info. • No client_id, serial number or vdom was specified. • application was not found • Trial customer not allow to use SMS service	

Code	Response content	Description
	• Mobile number is invalid • License is required for FTM token migration	
402 Payment Required	No valid FTC subscription was found.	Please enable free trial or register licenses.
403 Forbidden	• You are not authorized to create users. • License is not in migration.	
409 Conflict	application has been locked from adding user due to a pending task.	
424 Failed Dependency	application needs to be associated with a realm in FTC.	

Example request body:

```
Input:
{
      'username': 'abc',
      'email': 'abc@fortinet.com',
      'mobile_number': '+11234567890'
}
```

Example of successful response body:

```
{
   "auth_method":"FTM",
   "client_id":"12345678-abcd-efgh-1234-xxxxxxxxxxxx",
   "created_at":"2020-02-02T02:22:22",
   "customer_id":"123456",
   "email":"abc@fortinet.com",
   "user_data":0,
   "id":"12345678-abcd-efgh-1234-xxxxxxxxxxxx",
   "lockout_at": None,
   "mobile_number":"1123456789",
   "notification_method":"Email",
   "user_id":"12345678-abcd-efgh-1234-xxxxxxxxxxxx",
   "realm_id":"12345678-abcd-efgh-1234-xxxxxxxxxxxx",
   "username":"abc",
   "updated_at": None,
   "active":true,
   "temp_token":false,
   "bypass_at": None,
   "fail_times":0
}
```

PUT

URI:

- /api/v1/user/[id]

Input fields:

Field Name	Type	Required	Description
id	string	No	The user ID.
username	string	No	The user's username.
email	string	No	The user's email address, which can be up to 80 characters in length.
mobile_number	string	No	The user's mobile phone number.
activation_code	boolean	No	Whether to skip token migration notification.
dst_user_id	string	No	The base user's ID.
is_refid	boolean	No	Whether it is a userref ID.
change_token	boolean	No	Whether the user wants to change one-time password.
active	boolean	No	Whether the user is active.
bypass	boolean	No	Whether to bypass the user.
lockout	boolean	No	Whether to lock out the user.
user_data	integer	No	User type, which can be either of the following: • 0 for local user • 1 for remote LDAP user
auth_method	string	No	The authentication method, which can be one of the following: • ‘FTM’ • ‘Email’ • ‘SMS’ • ‘FTK’
notification_ method	string	No	Supported notification method, which can be either of the following: • ‘Email’ • ‘SMS’
brief	boolean	No	The API response only includes the mobile_number, username, email, vdom, realm, id, user_data if brief=True.
case_accent_ sensitive	boolean	No	Whether the username is case-sensitive.

Successful response fields:

Field name	Type	Description
customer_id	string	The customer ID.
client_id	string	The client ID.
user_id	string	The user ID.
notification_method	string	Supported notification method, which can be either of: • 'Email' • 'SMS'
auth_method	string	The authentication method, which can be one of the following: • 'FTM' • 'Email' • 'SMS' • 'FTK'
username	string	The user's username.
id	string	The userref ID.
realm_id	string	The realm ID.
mobile_number	string	The user's mobile phone number.
email	string	The user's email address.
active	boolean	Whether the user is active.
created_at	string	The timestamp when the user was created.
updated_at	string	The timestamp when the user was last updated.
bypass_at	string	The timestamp when the user was bypassed.
lockout_at	string	The timestamp when the user was locked out.
fail_times	integer	The failed attempts of user login validation.
temp_token	boolean	Whether temporary token is enabled for the user.
user_data	integer	User type, which can be either of • 0 for local user • 1 for remote LDAP user

Response codes:

Code	Response content	Description
202 OK		User info has been updated.
400 Bad Request	• Cannot identify related customer info. • Trial customer not allow to use SMS service	

Code	Response content	Description
	• Mobile number is invalid	
403 Forbidden	• Failed to modify user, locked user cannot bypass authentication	
404 Not Found	User does not exist.	
409 Conflict	application has been locked from adding user due to a pending task.	
424 Dependency	You cannot remove Alias from user when auto alias is enabled.	

Example request body:

```
{
   "change_token": True
}
```

Example of successful response body:

```
{
   "auth_method":"FTM",
   "client_id":"12345678-abcd-efgh-1234-xxxxxxxxxxxx",
   "created_at":"2020-02-02T02:22:22",
   "customer_id":"123456",
   "email":"abc@fortinet.com",
   "user_data":0,
   "id":"12345678-abcd-efgh-1234-xxxxxxxxxxxx",
   "lockout_at": None,
   "mobile_number":"1123456789",
   "notification_method":"Email",
   "user_id":"12345678-abcd-efgh-1234-xxxxxxxxxxxx",
   "realm_id":"12345678-abcd-efgh-1234-xxxxxxxxxxxx",
   "username":"abc",
   "updated_at": None,
   "active":true,
   "temp_token":false,
   "bypass_at": None,
   "fail_times":0
}
```

DELETE

URI:

- /api/v1/user/[id]

Input fields:

Field Name	Type	Required	Description
id	string	Yes	The user ID.
username	string	No	The username.
sn	string	No	The serial number of the device.
vdom	string	No	The VDOM of the application.

Successful response fields:

Field name	Type	Description

Response codes:

Code	Code content	Description
204 NO CONTENT		The user has been deleted.
400 Bad Request	Cannot identify related customer info.	
403 Forbidden	You are not allowed to access user.	
404 Not Found	User does not exist.	
409 Conflict	application has been locked. No user can be deleted due to the pending task.	

Example request body:

```
None
```

Example of successful response body:

```
None.
```

User base

When attempting to use GET, PUT and DELETE FTC API calls for a Management application, be sure to use /user/base. For merging user aliases, be sure to use the regular /user PUT API call.

GET

URI:

- /api/v1/user/base?username=***&realm_id=***
- /api/v1/user/base/[id]
- /api/v1/user/base

Input fields:

Field name	Type	Required	Description
id	string	No	The user ID.
realm_id	string	Yes	The realm ID.
username	string	No	
email	string	No	The user's email address.
mobile_number	string	No	The user's mobile phone number.
case_accent_sensitive	boolean	No	Whether the username is case-sensitive.

Successful response fields:

Field name	Type	Description
customer_id	string	The customer ID.
client_id	string	The client ID.
user_id	string	The user ID.
notification_method	string	Supported notification method, which can either of • 'Email' • 'SMS'
auth_method	string	The authentication method, which can be one of the following: • 'FTM' • 'Email' • 'SMS' • 'FTK'
username	string	The user name.
id	string	The user ID.

Field name	Type	Description
realm_id	string	The realm ID.
mobile_number	string	The user's mobile phone number.
email	string	The user's email address.
active	boolean	If the user is active.
created_at	string	The timestamp when the user was created.
updated_at	string	The timestamp when the user was last updated.
bypass_at	string	The timestamp when the user was bypassed.
lockout_at	string	The timestamp when the user was locked out.
fail_times	integer	The number of failed attempts of user login validation.
temp_token	boolean	If the user is enabled for temporary token.

Response codes:

Code	Response content	Description
200 OK		User info is successfully returned.
400 Bad Request	application is not found.	
402 Payment Required	No valid FTC subscription is found.	Please enable free trial or register the license(s).
403 Forbidden	You are not authorized to access the user.	
404 Not Found	• Cannot find the relevant customer information. • The user does not exist.	

Example request body:

```
{
   "id": "12345678-abcd-efgh-1234-xxxxxxxxxxxx"
}
```

Example of successful response body:

```
{
   "created_at": "2023-03-10T23:39:15",
   "updated_at": null,
   "id": "12345678-abcd-efgh-1234-xxxxxxxxxxxx",
```

```
"username": "abc",
"email": "abc@fortinet.com"
"mobile_number": "+1123456789",
"customer_id": "123456",
"realm_id": "12345678-abcd-efgh-1234-xxxxxxxxxxxx",
"active": true,
"bypass_at": null,
"lockout_at": null,
"temp_token": false,
"fail_times": 0,
"auth_method": "FTM",
"notification_method": "Email",
"sn": "abcd",
"expire_at": "2023-03-13T23:39:15",
"status": 0,
"refs": 1,
"usernames": "abc",
"refs_list": [
   {
      "name": "MyAuthClient",
      "sn": null,
      "cluster_id": null,
      "vdom": "root",
      "id": "12345678-abcd-efgh-1234-xxxxxxxxxxxx",
      "members": null,
      "username": "abc",
      "email": "abc@fortinet.com",
      "mobile_number": "+1123456789",
```

```
            "refid": "12345678-abcd-efgh-1234-xxxxxxxxxxxx"
        }
    ],
    "auth_history": {}
}
```

PUT

URI:

- /api/v1/user/base/[id]

Input fields:

Field name	Type	Required	Description
id	string	No	The user ID.
username	string	No	
email	string	No	The user's email address.
mobile_number	string	No	The user's mobile phone number.
activation_code	boolean	No	Whether skip token migration notification is enabled.
dst_user_id	string	No	The base user's ID.
change_token	boolean	No	Whether update user token is enabled.
active	boolean	No	Whether the user is active.
bypass	boolean	No	Whether bypass is enabled for the user.
lockout	boolean	No	If lookout is enabled for the user.
auth_method	string	No	The authentication method, which can be one of the following: • 'FTM' • 'Email' • 'SMS' • 'FTK'
notification_ method	string	No	Supported notification method, which can either of • 'Email' • 'SMS'

Successful response fields:

Field name	Type	Description
customer_id	string	The customer ID.
client_id	string	The client ID.
user_id	string	The user ID.
notification_method	string	Supported notification method, which can either of • 'Email' • 'SMS'
auth_method	string	The authentication method, which can be one of the following: • 'FTM' • 'Email' • 'SMS' • 'FTK'
username	string	The user name.
id	string	The userref ID.
realm_id	string	The realm ID.
mobile_number	string	The user's mobile phone number.
email	string	The user's email address.
active	boolean	Whether the user is active.
created_at	string	The timestamp when the user was created.
updated_at	string	The timestamp when the user was last updated.
bypass_at	string	The timestamp when the user was bypassed.
lockout_at	string	The timestamp when the user was locked out.
fail_times	integer	The number of failed attempts of user login validation.
temp_token	boolean	Whether temporary token is enabled for the user.

Response codes:

Code	Response Content	Description
202 OK		User info has been updated.
400 Bad Request	• Cannot identify related customer information. • Trial customer is not allowed to use SMS service. • The mobile number is invalid.	

Code	Response Content	Description
403 Forbidden	Failed to modify the user; the locked user cannot bypass authentication.	
404 Not Found	The user does not exist.	
409 Conflict	application has been locked from adding user due to a pending task.	
424 Dependency	You cannot remove the alias from the user when auto alias is enabled.	

Example request body:

```
{
   "id": "12345678-abcd-efgh-1234-xxxxxxxxxxxx"
   "auth_method": "Email"
   "change_token": true
   "email": "test@test.com"
}
```

Example of successful response body:

```
{
   "created_at": "2023-03-10T23:39:15",
   "updated_at": null,
   "id": "12345678-abcd-efgh-1234-xxxxxxxxxxxx",
   "username": "abc",
   "email": "abc@fortinet.com"
   "mobile_number": "+1123456789",
   "customer_id": "123456",
   "realm_id": "12345678-abcd-efgh-1234-xxxxxxxxxxxx",
   "active": true,
   "bypass_at": null,
   "lockout_at": null,
```

```
  "temp_token": false,

  "fail_times": 0,

  "auth_method": "FTM",

  "notification_method": "Email",

}
```

DELETE

URI:

- /api/v1/user/base/[id]

Input fields:

Field name	Type	Required	Description
id	string	Yes	The user ID.

Response codes:

Code	Response content	Description
204 NO CONTENT		The user has been deleted.
400 Bad Request	Cannot identify related customer info.	
403 Forbidden	You are not allowed to access user.	
404 Not Found	User does not exist.	
409 Conflict	application has been locked. No user can be deleted due to the pending task.	

Example request body:

```
None
```

Example of successful response body:

```
None
```

User authentication

GET

URI:

- /api/v1/auth/[id]
- api/v1/auth

Input fields:

Field Name	Type	Required	Description
id	string	No	The auth ID obtained from an auth POST request.
chunksize	string	No	The query size for a batch auth result query. The default chunksize is 500, and the valid range is from 100 to 1000.

Successful response fields:

Field name	Type	Description
status	string	The return status could be "waiting", “authenticated", or "rejected". • If the API request is made with an auth ID, the return status could be "waiting", “authenticated", or "rejected". • If the API request is made with an serial number (client id) for batch auth status checking, the status could be “authenticated" or "rejected". The API request returns an empty list if no push auth response is received.

Response codes:

Code	resppnse content	
200 OK		Auth info is successfully returned.
400 Bad Request	• No authid is specified. • SN is invalid.	

Example request body:

```
{
   "sn":"abcd"
}
```

Example of successful response body:

```
[
   {
      "authid":"12345678-abcd-efgh-1234-xxxxxxxxxxxx",
```

```
        "status":"authenticated"
    }
]
```

POST

Pre-authentication URI:

- /api/v1/auth/preview

Input fields:

Field Name	Type	Required	Description
realm	string	No	The realm name, which can be up to 80 characters in length.
realm_id	string	No	The realm ID.
username	string	Yes	The username, which can be up to 80 characters in length.
case_accent_sensitive	boolean	No	Whether the username is case-sensitive.
auth_method	string	No	The authentication method can be one of the following: • 'FTM' • 'Email' • 'SMS' • 'FTK'
user_ip	string	No	The user's IP address.
location	string	No	The user's location.

Successful response fields:

Field name	Type	Description
auth_method	string	The authentication method, which can be one of the following: • 'FTM' • 'Email' • 'SMS' • 'FTK'
action	string	The action for this authentication. Can be one of the following: • 'Bypass' • 'MFA' • 'Block'

Field name	Type	Description
temp_token	boolean	Whether there is a temp token available for the user. If there is a temp token, the auth_method in response will indicate the original auth method.
push_enabled	boolean	Whether push is enabled for this user. This field will be added if the action is 'MFA' and auth_method is 'FTM'.
message	boolean	Error message on why the action is 'Block'.

Response codes:

Code	Response content	Description
200 OK	Successful content as above	Regular preview auth request.
400 Bad Request	• application not found. • Realm not assigned. • User not found. • No token or token was not activated.	
402 Payment Required	• Account was disabled. • The remaining SMS under customer is not enough.	

Example request body:

```
{
   "username":"abc",
   "realm":"root",
   "location":"US",
   "user_ip":"10.160.x.x"
}
```

Example of successful response body:

```
{
   "auth_method":"FTM",
   "action":"MFA",
   "push_enabled":True
}
```

Authentication URI:

• /api/v1/auth

Input fields:

Field Name	Type	Required	Description
realm	string	No	The realm name, which can be up to 80 characters in length.
realm_id	string	No	The realm ID.
username	string	Yes	The username, which can be up to 80 characters in length.
token	string	No	The one-time password
case_accent_ sensitive	boolean	No	Whether the username is case-sensitive.
auth_method	string	No	The authentication method can be one of the following: • 'FTM' • 'Email' • 'SMS' • 'FTK'
user_ip	string	No	The user's IP address.
location	string	No	The user's location.
account	string	No	The account name of the application server.
user_agent	string	No	The user agent.
token_selection	string	No	Indicates which token/auth_method to use, original token or temporary token: • 'regular'—user regular token only • 'temp'—use temp token only • 'all'—try all the tokens

Successful response fields:

Field name	Type	Description
authid	string	The auth ID

Response codes:

Code	Response content	Description
200 OK		Regular auth request, request with a token code.
201 OK		Push authentication, auth method is FTM and request without token code.

Code	Response content	Description
202 OK		Client is requesting FTC to send an OTP to a user based on the user's settings in FTC (either email or SMS), no token code.
400 Bad Request	• No token was assigned to or activated by user. • The application version doesn't support authentication. • Failed to send verification code to user (by SMS) due to not enough balance • User does not exist.	
402 Payment Required	The remaining SMS under the customer is not enough.	
403 Forbidden	Rejected authentication for user.	The customer auth profile does not allow this specific action to use authentication service. Could be one of the following reasons: • Location IP is forbidden. • User is disabled/locked out/ bypassed • OTP verification failed for user

Example request body:

```
{
   "token":"xxxxxx",
   "username":"abc",
   "realm":"root",
   "user_agent":"Mobile Safari",
   "location":"US",
   "user_ip":"10.160.x.x",
   "account":"Fortinet"
}
```

Example of successful response body:

```
{
   "authid":"12345678-abcd-efgh-1234-xxxxxxxxxxxx"
}
```

Sync user data to FTC

POST

URI:

- /api/v1/user_sync

Input fields:

Field Name	Type	Required	Description
realm	string	No	The realm name.
users	list	No	A user list, which includes user IDs and actions.
sn	string	Yes	The device's serial number.
license	string	No	The FTM migration license number.

Successful response fields:

Field name	Type	Description
task_id	string	The user sync task ID

Response codes:

Code	Response content	Description
201 OK		User sync task is created.
401 Unauthorized	License was associated with other device.	The request license number is not available.
403 Forbidden	License is not in migration.	The request license number is not available.
404 not found	License was not found.	The request license number is not available.

Example request body:

```
{
   "realm":"default",
"sn":"abcd",
   "users":[
      {
         "username":"abc",
         "action":"delete",
         "id":"12345678-abcd-efgh-1234-xxxxxxxxxxxx"
      },
```

```
        {
           "username":"def",
           "action":"create",
           "id":"12345678-abcd-efgh-1234-xxxxxxxxxxxx"
        },
        {
           "username":"ghi",
           "action":"modify",
           "id":"12345678-abcd-efgh-1234-xxxxxxxxxxxx"
        }
    ]
}
```

Example of successful response body:

```
{
   "task_id":"12345678-abcd-efgh-1234-xxxxxxxxxxxx"
}
```

Query asynchronous task status

GET

URI:

- /api/v1/task/[id]

Input fields:

Field Name	Type	Required	Description
id	string	Yes	The task ID.

Successful response fields:

Field name	Type	Description
msg	dict	Task details
status	string	Task status

Response codes:

Code	Response content	Description
200 OK		Task status/details has been returned.
404 Not Found	Get task failed.	

Example request body:

```
None
```

Example of successful response body:

```
User sync task example
{
   "msg":{
      "create":{
         "success":1,
         "failure":0
      },
      "delete":{
         "success":1,
         "failure":0
      },
      "modify":{
         "success":1,
         "failure":0
      }
   },
```

```
    "status":"complete"
}
```

Token

GET

URI:

- /api/v1/token

Input fields:

Field name	Type	Required	Description
available	boolean	No	Get all available hard tokens.
soft_token	boolean	No	Get soft tokens.
status	string	No	Applicable status can be one of the following: • disable • enable • transfer • others **Note:** The status field applies only when you set soft_token=True
user_id	string	No	The user ID.
temp_token	boolean.	No	Get temporary token
token_sn	string	No	The serial number of the token.

Successful response fields:

Field name	Response content	Description
sn	string	The device's serial number.
algorithm	string	The token encrypt algorithm.
user_id	string	The user ID.
username	string	The username.
realm_id	string	The realm ID.
realm_name	string	The realm name.
platform	string	The platform to which the token is sent. It could be ‘iOS’, ‘Android’, ‘Windows’, or ‘unknown’.
registration_id	string	The registration ID that associates the token with a specific device.

Response codes:

Code	Response	Description
200 OK		Token info has been returned.
404 Not Found	No token matches the filter	

Example request body:

```
{
   "soft_token":"true",
   "status":"enabled"
}
```

Example of successful response body:

```
[
   {
      "sn":"abcdef",
      "algorithm":"TOTP",
      "user_id":"12345678-abcd-efgh-1234-xxxxxxxxxxxx",
      "username":"abc",
      "realm_id":"12345678-abcd-efgh-1234-xxxxxxxxxxxx",
      "realm_name":" root",
      "platform":"iOS",
      "registration_id":"abcd1234efgh5678"
   }
]
```

Temporary token

POST

Temporary Token generation URI:

- /api/v1/token/temp

Input fields:

Field name	Type	Required	Description
user_id	string	Yes	The user ID
auth_method	string	No	The authentication method, which can be either of the following: • 'Email' • 'SMS'
expired_at	string	No	The expiration date and time for the temporary token. If left as Null, the token won't expire on its own.

Successful response fields:

Field name	Type	Description
sn	string	The temporary token serial number.
expired_at	string	The expiration date and time for the temporary token.
auth_method	string	The authentication method, which can be either of the following: • 'Email' • 'SMS'
user_id	string	The user ID.
username	string	The username.
realm_id	string	The realm ID.
realm_name	string	The realm name.

Response codes:

Code	Response content	Description
201 OK		A temporary token has been generated.
400 Bad Request	• You cannot add temporary token for user.	• User auth method must be FTM/FTK to add a temporary token.

Code	Response content	Description
	• You cannot set temporary token's auth_method to 'SMS'.	• Cannot set auth method to SMS due to the user's invalid mobile number. • Expiration date must be set as a valid date.
404 Not Found	User does not exist.	

Example request body:

```
{
   "user_id":"12345678-abcd-efgh-1234-xxxxxxxxxxxx",
   "auth_method":"Email"
}
```

Example of successful response body:

```
{
   "sn":"TMP8D04X81LH12ZP",
   "expired_at": None,
   "auth_method":"Email",
   "user_id":"12345678-abcd-efgh-1234-xxxxxxxxxxxx",
   "username":"abc",
   "realm_id":"12345678-abcd-efgh-1234-xxxxxxxxxxxx",
   "realm_name":"FG101ETK18001260-root"
}
```

DELETE

Temporary token delete URI:

- /api/v1/token/temp/<id>

Input fields:

Field name	Type	Required	Description
id	string	Yes	The user ID.

Successful response fields:

Field name	Type	Description

Response codes:

Code	Response content	Description
204 No Content		The temporary token has been deleted.
404 Not Found		The uer does not exist.

Example request body:

```
{
   "id":"abcdefgh"
}
```

Example of successful response body:

```
None
```

Balance

GET

URI:

- /api/v1/balance

Input fields:

Field name	Type	Required	Description

Successful response fields:

Code	Rsponse conent	Description
balance	float	Total user quota.
trial_status	integer	Customer trial status: • 0 No license • 1 In trial • 2 In licensed
customer_id	string	Customer ID.
expiration	string	The end date of subscription.
unit	string	Users.

Response codes:

Code	Response content	Description
200 OK		Customer balance info has been returned.
404 Not Found	Cannot find related customer	

Example request body:

```
None
```

Example of successful response body:

```
{
   "balance":12000,
   "trial_status":2,
   "customer_id":"123456"
}
```

Count resource

GET

URI:

- /api/v1/count?resource=***

Input fields:

Field Name	Type	Required	Description
resource	string	Yes	Resources such as • user • realm • device • client • SMS • realm_quota
realm_id	string	No	The realm ID.

Successful response fields:

Field name	Type	Description
[specific resource]	integer	The remaining available resource count.
quota	integer	The total allowed resource count.

Response codes:

Code	Response content	Description
200 OK		Resource count has been returned.
400 Bad Request	• Cannot find customer info. • The requested resource does not exist .	
403 Forbidden	• Realm ID required for getting realm quota info. • Only global admin can get the device count.	

Example request body:

```
None
```

Example of successful response body:

```
{
   "user": 120,
   "quota": 1200000
}
```

Version

GET

URI:

- /version

Input fields:

Field name	Type	Required	Description

Successful response fields:

Field name	Type	Description
FTC server	string	The FortiToken Cloud server version number.

Response codes:

Code	Response content	Description
200 OK		The account balance info has been returned.

Example request body:

```
None
```

Example of successful response body:

```
{
   " FTC server":"2.5.0"
}
```

User Quota Allocation per Realm

GET

URI:

- /api/v1/quota

Input fields:

Field name	Type	Required	Description
realm_id	string	No	The realm ID.
brief	boolean	No	Whether to get user quota details per realm.

Successful response fields:

Field name	Type	Description
usage	integer	The user count.
total_quota	integer	The total user quota.
trial_status	integer	Customer trial status, which can be one of the following: • 0=No license • 1=In trial • 2=In licensed
end_date	string	The latest license expiration date.
realms	list	The realm quota allocation details, for each item (e.g., realm_id, quota, usage, etc.)

Response codes:

Code	Response content	Description
200 OK		User quota info has been successfully returned.
404 Not Found	• Cannot identify related customer info. • Cannot identify realm info.	

Example request body:

```
None
```

Example of successful response body:


```
{
   "usage":10,
   "total_quota":240,
   "trial_status":2,
   "end_date":"2030-11-18",
   "realms":[
      [
         "12345678-abcd-efgh-1234-xxxxxxxxxxxx",
         10,
         1
      ]
   ]
}
```

Pagination

These are a series of pagination GET APIs that help with retrieving a large amount of data by paginating the results if you find that the regular GET APIs run too slowly. For instance:

```
api/v2/user?limit=200&page=MDAwMzZjMjctNjIxMi00YmUxLWIwMDgtOTMyNzNjZTcwMDdl
```

Where ‘limit’ specifies the number of entries per page (FTC supports 20, 100, 200, 500, and 1000 entries per page), and ‘page’ is the cursor for the next page. If ‘limit’ is not provided, FTC will default it to 500 per page; if ‘page’ is not provided, FTC will retrieve the very first page.

To maintain the current API response body format, all pagination information is included in the HTTP headers of the response as below:

```
link: <https://ftc.fortinet.com:8686/api/v2/user?limit=20&page=LWIwMD>; rel="next", <
     https://ftc.fortinet.com:8686/api/v2/user?limit=20&page=zZjMjctNjIx>; rel="previous
```

The following is an example of how you can retrieve the HTTP header in Python:

```
import requests
headers = {
   ‘Authorization’: ‘Bearer eyJ0ex*******Q6to’,
   ‘Content-Type’: ‘application/json’

}

payload = {}
resp = requests.request('get', 'https://ftc.fortinet.com:9696/api/v2/user?limit=200',
headers=headers, json=payload, verify=False)
link_header = resp.headers.get('Link')
```

You can then use the contents of ‘link_header’ to retrieve the next or the preceding cursor.

Realm

GET

URI:

- `/api/v2/realm?limit=***`
- `/api/v2/realm?limit=***&page=***`

Input fields:

Field Name	Type	Required	Description
limit	integer	No	The number of realms per page.
page	string	No	The cursor of the paginated page to retrieve.

Field Name	Type	Required	Description
name	string	No	The name of the realm.
id	string	No	The realm ID.
deleted	boolean	No	Get the deleted realm record.

Successful response fields:

Field name	Type	Description
description	string	A brief description of the realm.
deleted_at	datetime	The deleted timestamp. If the realm is not deleted, the field is None.
name	string	The name of the realm.
is_default	boolean	If the realm is the default realm.
id	string	The realm ID.

Response codes:

Code	Response content	Description
200 OK		Realm(s) info is successfully returned.
404 Not Found	Realm ID was not found	Cannot find the realm with the Realm ID.

Example request body:

```
None
```

Example of successful response body:

```
[
   {
      "description":"abc",
      "deleted_at":"None",
      "name":"abc",
      "is_default":false,
      "id":"12345678-abcd-efgh-1234-xxxxxxxxxxxx"
   }
]
```

User

GET

URI:

- `/api/v2/user?limit=***`
- `/api/v2/user?limit=***&page=***`

Input fields:

Field Name	Type	Required	Description
`limit`	integer	No	The number of users per page.
`page`	string	No	The cursor of the paginated page to retrieve.
`id`	string	No	The user ID.
`realm_id`	string	No	The realm ID.
`username`	string	No	The username of the end-user.
`email`	string	No	The end-user's email address.
`mobile_number`	string	No	The end-user's mobile phone number.
`active`	boolean	No	Get active user.
`bypass_at`	datetime	No	The end-user's bypass date and time.
`lockout_at`	datetime	No	The user's lockout date and time.
`user_data`	integer	No	User type, which can be either of the following: • 0 for local user. • 1 for remote LDAP user.
`auth_method`	string	No	The end-user's authentication method, which can be one of the following: • `FTM` • `Email` • `SMS` • `FTK`
`notification_method`	string	No	The configured notification method, which can be either of the following: • `Email` • `SMS`
`brief`	boolean	No	The API response only includes the mobile_number, username, email, vdom, realm, id, user_data if brief=True.
`case_accent_sensitive`	boolean	No	Whether the username is case-sensitive.

Successful response fields:

Field name	Type	Description
`customer_id`	string	The customer ID.
`client_id`	string	The client ID.
`user_id`	string	The user ID.
`notification_method`	string	The configured notification method, which can be either of the following: • `Email` • `SMS`
`auth_method`	string	The end-user's authentication method, which can be one of the following: • `FTM` • `Email` • `SMS` • `FTK`
`username`	string	The user name.
`id`	string	The `userref` ID.
`mobile_number`	string	The end-user's mobile phone number.
`email`	string	The end-user's email address.
`realm_id`	string	The realm ID.
`active`	boolean	If the end-user is active.
`created_at`	string	The timestamp when the end-user was created.
`updated_at`	string	The timestamp when the end-user was last updated.
`bypass_at`	string	The timestamp when the end-user was bypassed for MFA authentication.
`lockout_at`	string	The timestamp when the end-user was locked out.
`fail_times`	integer	The number of failed attempts the end-user made when trying to log in.
`user_data`	integer	The typw of the end-user, which can be either of the following: • 0 for local user • 1 for remote LDAP user
`temp_token`	boolean	Whether temporary token is enabled for the user.

Response codes:

Code	Response content	Description
200 OK		The end-user(s) info is successfully returned.
400 Bad Request	Auth Client was not found.	
402 Payment Required	No valid FTC subscription was found.	Please enable your free trial or register your purchased license.
403 Forbidden	You are not authorized to access user.	
404 Not Found	• Cannot find the related customer info. • User does not exist.	

Example request body:

None

Example of successful response body:


```
[
   {
      "auth_method":"FTM",
      "client_id":"12345678-abcd-efgh-1234-xxxxxxxxxxxx",
      "created_at":"2020-02-02T02:22:22",
      "customer_id":"123456",
      "email":"abc@fortinet.com",
      "user_data":0,
      "id":"12345678-abcd-efgh-1234-xxxxxxxxxxxx",
      "lockout_at":"None",
      "mobile_number":"1123456789",
      "notification_method":"Email",
      "user_id":"12345678-abcd-efgh-1234-xxxxxxxxxxxx",
      "realm_id":"12345678-abcd-efgh-1234-xxxxxxxxxxxx",
      "username":"abc",
      "updated_at":"None",
      "active":true,
      "temp_token":false,
      "bypass_at":"None",
      "fail_times":0
   }
]
```


User base

GET

URI:

- `/api/v2/user/base?limit=***`
- `/api/v2/user/base?limit=***&page=***`

Input fields:

Field name	Type	Required	Description
`limit`	integer	No	The number of users per page.
`page`	string	No	The cursor of the paginated page to retrieve.
`id`	string	No	The user ID.
`realm_id`	string	Yes	The realm ID.
`username`	string	No	The end-user's username.
`email`	string	No	The end-user's email address.
`mobile_number`	string	No	The end-user's mobile phone number.
`case_accent_sensitive`	boolean	No	Whether the username is case-sensitive.

Successful response fields:

Field name	Type	Description
`customer_id`	string	The customer ID.
`client_id`	string	The client ID.
`user_id`	string	The user ID.
`notification_method`	string	Supported notification method, which can either of • 'Email' • 'SMS'
`auth_method`	string	The authentication method, which can be one of the following: • 'FTM' • 'Email' • 'SMS' • 'FTK'
`username`	string	The user name.
`id`	string	The user ID.
`realm_id`	string	The realm ID.
`mobile_number`	string	The user's mobile phone number.

Field name	Type	Description
email	string	The user's email address.
active	boolean	If the user is active.
created_at	string	The timestamp when the user was created.
updated_at	string	The timestamp when the user was last updated.
bypass_at	string	The timestamp when the user was bypassed.
lockout_at	string	The timestamp when the user was locked out.
fail_times	integer	The number of failed attempts of user login validation.
temp_token	boolean	Whether temporary token is enabled for the end-user.

Response codes:

Code	Response content	Description
200 OK		User info is successfully returned.
400 Bad Request	Auth Client is not found.	
402 Payment Required	No valid FTC subscription is found.	Please enable free trial or register the license(s).
403 Forbidden	You are not authorized to access the user.	
404 Not Found	• Cannot find the relevant customer information. • The user does not exist.	

Example request body:

```
{
   "id": "12345678-abcd-efgh-1234-xxxxxxxxxxxx"
}
```

Example of successful response body:

```
{
   "created_at": "2023-03-10T23:39:15",
   "updated_at": null,
   "id": "12345678-abcd-efgh-1234-xxxxxxxxxxxx",
   "username": "abc",
   "email": "abc@fortinet.com"
   "mobile_number": "+1123456789",
   "customer_id": "123456",
   "realm_id": "12345678-abcd-efgh-1234-xxxxxxxxxxxx",
```

```
    "active": true,
    "bypass_at": null,
    "lockout_at": null,
    "temp_token": false,
    "fail_times": 0,
    "auth_method": "FTM",
    "notification_method": "Email",
    "sn": "abcd",
    "expire_at": "2023-03-13T23:39:15",
    "status": 0,
    "refs": 1,
    "usernames": "abc",
    "refs_list": [
       {
          "name": "MyAuthClient",
          "sn": null,
          "cluster_id": null,
          "vdom": "root",
          "id": "12345678-abcd-efgh-1234-xxxxxxxxxxxx",
          "members": null,
          "username": "abc",
          "email": "abc@fortinet.com",
          "mobile_number": "+1123456789",
          "refid": "12345678-abcd-efgh-1234-xxxxxxxxxxxx"
       }
    ],
  "auth_history": {}
}
```

Change log

Date	Description
June 6, 2024	Initial release.

www.fortinet.com

Copyright© 2024 Fortinet, Inc. All rights reserved. Fortinet®, FortiGate®, FortiCare® and FortiGuard®, and certain other marks are registered trademarks of Fortinet, Inc., and other Fortinet names herein may also be registered and/or common law trademarks of Fortinet. All other product or company names may be trademarks of their respective owners. Performance and other metrics contained herein were attained in internal lab tests under ideal conditions, and actual performance and other results may vary. Network variables, different network environments and other conditions may affect performance results. Nothing herein represents any binding commitment by Fortinet, and Fortinet disclaims all warranties, whether express or implied, except to the extent Fortinet enters a binding written contract, signed by Fortinet's Chief Legal Officer, with a purchaser that expressly warrants that the identified product will perform according to certain expressly-identified performance metrics and, in such event, only the specific performance metrics expressly identified in such binding written contract shall be binding on Fortinet. For absolute clarity, any such warranty will be limited to performance in the same ideal conditions as in Fortinet's internal lab tests. Fortinet disclaims in full any covenants, representations, and guarantees pursuant hereto, whether express or implied. Fortinet reserves the right to change, modify, transfer, or otherwise revise this publication without notice, and the most current version of the publication shall be applicable.