



REST API Reference

FortiMail 7.2.0



FORTINET DOCUMENT LIBRARY

<https://docs.fortinet.com>

FORTINET VIDEO GUIDE

<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>

NSE INSTITUTE

<https://training.fortinet.com>

FORTIGUARD CENTER

<https://www.fortiguard.com>

END USER LICENSE AGREEMENT

<https://www.fortinet.com/doc/legal/EULA.pdf>

FEEDBACK

Email: techdoc@fortinet.com



May 10, 2022

FortiMail 7.2.0 REST API Reference

06-720-000000-20220510

TABLE OF CONTENTS

Change Log	5
Introduction	6
Enabling REST API support	7
Authentication	8
PKI authentication	9
To use PKI admin authentication	9
Password-based authentication	10
FortiMail REST API HTTP response codes	11
REST API for system level resources	12
REST API for domain level resources	13
REST API for administrative actions	14
REST API for file upload	15
Require file uploading authorization token	15
Upload local file with authorization token	15
Apply uploaded file on resource	16
Update firmware	16
Update configuration	16
Restore Block/Safe list	16
REST API for IP group management	19
Create a new IP group	19
Delete an IP group	19
Add a new member to an IP group	19
Delete a member from an IP group	20
Replace all members of an IP group	20
REST API for impersonation rule management	21
View rules list	21
Domain level	21
System level	21
View match rules entry	21
Domain level	21
System level	21
Add a new rule entry	21
Domain level	22
System level	22
Modify a rule entry	22
Domain level	22
System level	23
Delete a rule entry	23
Domain level	23
System level	23

System resource list and URLs	24
Example commands	31
Admin login with PKI certificate-based authentication	31
Admin login with password-based authentication	31
To get domain information	31
To get domain information with password-based authentication	31
To get domain information with certificate-based authentication	31
To create a domain association	32
Access control rule management	32
List access control rules	33
Create a new access control rule	33
Modify an existing access control rule	33
Delete an existing access control rule	33
Move existing access control rule	33
Safelist/blocklist management	34

Change Log

Date	Change Description
2022-05-10	Initial release.
2022-06-23	Bug fix.
2022-10-11	Changed UserMaillist to SenderListV2.
2023-05-25	Bug fix.

Introduction

This document provides the REST API information supported in FortiMail version 7.2.0 release. This document covers the FortiMail GUI supported REST API reference only. These APIs can be used to retrieve, create, update and delete configuration settings, to retrieve dynamic system statistics, and to perform basic administrative actions such as reboot and shut down.

When using the APIs, the following conventions are followed:

- `Http GET`: To retrieve all resources or particular resource
- `Http POST`: To create a new resource or perform certain administrative actions
- `Http PUT`: To update an existing resource
- `Http Delete`: To delete an existing resource

Enabling REST API support

By default, this feature is disabled on FortiMail. To enable it, use the following CLI command:

```
config system web-service
    set rest-api-status enable
end
```

Authentication

When making requests to FortiMail appliance using the REST API, you will need to pass the authentication. There are two authentication options you can use:

- PKI certificate-based authentication
- Local user password-based authentication

You also need the appropriate admin profile to access the FortiMail resources.

PKI authentication

For PKI certificate-based authentication, you must create two certificates using the same CA. One certificate will be used for FortiMail HTTP server and the other certificate will be associated with an admin user. Both the user certificates and private key has to be moved to the PC running the script. The CA certificate needs to be copied to the PC as well. Otherwise CURL will not be able to verify the FortiMail certificate.

To use PKI admin authentication

1. Enable PKI mode with the following CLI command:

```
config system global
    set pki-mode enable
end
```

2. Create a PKI user under *Domain & User > User > PKI User*.
3. Create an admin account using PKI authentication type under *System > Administrator > Administrator*.
4. Split the PKCS12 certificate into cert and key. They will be used when you log in to FortiMail.

```
openssl pkcs12 -in test.p12 -out test.pem -nokeys
openssl pkcs12 -in test.p12 -out test.key nocerts -nodes
```

Password-based authentication

To establish a valid authentication session, you must make a POST request to the FortiMail login handler with your admin username and password. The POST request should contain JSON data with "name" and "password" fields:

- URL: `http(s)://host_or_ip/api/v1/AdminLogin/`
- Method: POST
- JSON: `{"name": "admin", "password": "*****"}`

If login is successful, the response will contain the authentication token in the APSCookie cookie value. This cookie value must be included in any further requests.



The permissions for the administrative account you use will affect which objects and operations you'll have access to, so ensure the user has the permissions required for the actions you wish to perform.

FortiMail REST API HTTP response codes

FortiMail REST APIs use well-defined HTTP status codes to indicate query results to the API. Following are some of the HTTP status codes used:

HTTP response code	Description
200 - OK	API request successful.
400 - Bad Request	Bad request.
403 - Forbidden	Request is missing authentication token or administrator is missing access profile permissions.
404 - Not Found	Unable to find the specified resource.
405 - Method Not Allowed	Specified HTTP method is not allowed for this resource.
500	Internal server error.

REST API for system level resources

FortiMail supports retrieval and modification of system level CMDB configuration settings as well as system level statistics. The API can be accessed using the following URL:

```
http(s)://host_ip/api/v1/resource_name/resource_id/sub_resource_name/sub_resource_id/
```

where:

- **resource_name**: Specifies the type of resource to query (such as SysInterface), required.
- **resource_id**: Unique ID of the resource as specified by resource_name (such as port1), optional. If not present, returns entire list of resources.
- **sub_resource_name**: Some resources may have sub / child resources, use this to query sub resources, optional.
- **sub_resource_id**: Unique ID of the sub resource as specified by sub_resource_name, optional. If not present, returns entire list of sub resources.

Examples:

- `.../api/v1/SysInterface/`: Returns list of network interfaces
- `.../api/v1/SysInterface/port1/`: Return details of network interface 'port1'
- `.../api/v1/SysGlobal/`: Returns details of global settings (only one instance)
- `.../api/v1/ProfSession/inbound/ ProfSessionSenderWhitelist/`: Returns sender safelist/safelist of session profile 'inbound'

For a full list of system level resources, refer to the Supported Resources List.

REST API for domain level resources

FortiMail also supports retrieval and modification of domain level CMDB configuration settings. The API can be accessed using the following URL:

```
http(s)://host_ip/api/v1/domain/domain_name/resource_name/resource_id/sub_resource_name/sub_resource_id/
```

It is very similar to the URL for system level resources, only two new tokens are added:

- **domain:** Required keyword, use to perform domain level queries
- **domain_name:** FQDN name of the domain to query (such as fortinet.com)

Examples:

- `.../api/v1/domain/abc.com/ProfAntispam/`: Returns list of antispam profiles for domain 'abc.com'
- `.../api/v1/domain/abc.com/PolicyRcpt/`: Returns list of recipient based policies for domain 'abc.com'
- `.../api/v1/domain/abc.com/PolicyRecipient/1/`: Returns details of recipient based policy '1' for domain 'abc.com'
- `.../api/v1/domain/abc.com/UserPreference/`: Returns list of entries in user preferences for domain 'abc.com'

For a full list of domain level resources, refer to the Supported Resources List.

REST API for administrative actions

Apart from resources, FortiMail REST API supports basic administrative actions such as restarting /shutting down a device. Use the following URL to send action request:

- URL: `http(s)://host_ip/api/v1/SysStatusCommand/`
- Method: `POST`
- JSON: `{"action": action_value}`

Where `action_value` is one of the following integers:

- 1:** Restart
- 2:** Shut down
- 3:** Reload

REST API for file upload

There are three steps to upload a file to FortiMail through the REST API:

1. Require file uploading authorization token
2. Upload local file with authorization token
3. Apply uploaded file on resource

URL	HTTP method	Summary
/resource/	POST	Require authorization token. "Resource" is the name of the resource on which the uploaded file will apply to. The request must set the "reqAction" field to 22.
/uploadfile?token=tokenStr	POST	Upload file with authorization token.
/resource/	PUT	Apply uploaded file on related resource.

Require file uploading authorization token

Example:

```
curl -v -H "Content-Type: application/json" -X POST -d '{"reqAction":22}' http://host_or_ip/api/v1/CentralRestoreFirmware --cookie cookie.txt
```

The "reqAction" field should be set to 22, which requires FortiMail to generate the file uploading authorization token. "CentralRestoreFirmware" is the ID of the resource on which the uploaded file will apply to. For resources that support sub resources, extra sub resource IDs should be added, which can always be set to "0". For example, resource ID "SenderListV2" supports sub resources, with an example URL of http://host_or_ip/api/v1/SenderListV2/0

Response:

```
{"token": "tokenStr"}
```

The "token" value is generated by FortiMail. This is used in the next step.

Upload local file with authorization token

Example:

```
curl -F 'testfile=@/tmp/file.diff' http://host_or_ip/api/v1/uploadfile?token=tokenStr--cookie cookie.txt
```

In this example, “testfile” is a self-assigned ID to indicate the uploading action, “/tmp/file.diff” is the full path of the local file you want to upload, and the “token” field should be set to the value provided in the previous step.

Response:

```
{"id": "uploadedFileId"}
```

The “id” value is generated by FortiMail. This is used in the next step.

Apply uploaded file on resource

Update firmware

The resource used for updating the firmware is “CentralRestoreFirmware”.

URL	HTTP method	Summary
/CentralRestoreFirmware/	PUT	Set the “filename” field to the “id” value from the previous step.

Example:

```
curl -v -H "Content-Type: application/json" -X PUT -d '{"filename" : "uploadedFileId"}'  
http://host_or_ip/api/v1/CentralRestoreFirmware --cookie cookie.txt
```

Update configuration

The resource used for updating the configuration is “CentralRestoreConfig”.

URL	HTTP method	Summary
/CentralRestoreConfig/	PUT	Set the “filename” field to the “id” value from the previous step.

Example:

```
curl -v -H "Content-Type: application/json" -X PUT -d '  
{ "source": 0, "filename": "uploadedFileId", "check_platform": 1 }' http://host_or_  
ip/api/v1/CentralRestoreConfig --cookie cookie.txt
```

Restore Block/Safe list

System, domain, and personal block/safe lists can be restored. The resource used for these actions is “SenderListV2”.

Because “SenderListV2” supports sub resources, the URL used in the first step (Require file uploading authorization token) should have the format, http://host_or_api/api/v1/SenderListV2/0.

URL	HTTP method	Summary
/SenderListV2/system	PUT	System block list restore. Set the “extraParam” field to “blocklist”, and set the “restore” field to “id”.
/SenderListV2/system	PUT	System safe list restore. Set the “extraParam” field to “safelist”, and set the “restore” field to “id”.
/SenderListV2/domainname	PUT	Domain block list restore. Set the “extraParam” field to “blocklist”, and set the “restore” field to “id”. “domainname” is the domain on which the list will restore.
/SenderListV2/domainname	PUT	Domain safe list restore. Set the “extraParam” field to “safelist”, and set the “restore” field to “id”. “domainname” is the domain on which the list will restore.
/SenderListV2/useraddress	PUT	Personal block list restore. Set the “extraParam” field to “blocklist”, and set the “restore” field to “id”. “domainname” is the domain on which the list will restore.
/SenderListV2/useraddress	PUT	Personal safe list restore. Set the “extraParam” field to “safelist”, and set the “restore” field to “id”. “domainname” is the domain on which the list will restore.

Example for system block list restore:

```
curl -v -H "Content-Type: application/json" -X PUT -d '{"restore":"uploadedFileId",
  "extraParam":"blocklist"}' http://host_or_ip/api/v1/SenderListV2/system --cookie
  cookie.txt
```

Example for system safe list restore:

```
curl -v -H "Content-Type: application/json" -X PUT -d '{"restore":"uploadedFileId",
  "extraParam":"safelist"}' http://host_or_ip/api/v1/SenderListV2/system --cookie
  cookie.txt
```

Example for domain (1.ca) block list restore:

```
curl -v -H "Content-Type: application/json" -X PUT -d '{"restore":"uploadedFileId",
  "extraParam":"blocklist"}' http://host_or_ip/api/v1/SenderListV2/1.ca --cookie cookie.txt
```

Example for domain (1.ca) safe list restore:

```
curl -v -H "Content-Type: application/json" -X PUT -d '{"restore":"uploadedFileId",
  "extraParam":"safelist"}' http://host_or_ip/api/v1/SenderListV2/1.ca --cookie cookie.tx
```

Example for personal (u1@1.ca) block list restore:

```
curl -v -H "Content-Type: application/json" -X PUT -d '{"restore":"uploadedFileId",
  "extraParam":"blocklist"}' http://host_or_ip/api/v1/SenderListV2/u1@1.ca --cookie
  cookie.txt
```

Example for personal (u1@1.ca) safe list restore:

```
curl -v -H "Content-Type: application/json" -X PUT -d '{"restore":"uploadedFileId",  
  "extraParam":"safelist"}' http://host_or_ip/api/v1/SenderListV2/u1@1.ca --cookie  
  cookie.txt
```

REST API for IP group management

IP groups and the members of each group can be created, deleted, and managed through the REST API.



REST API only supports IP range values. The IP/netmask format is not accepted.

Create a new IP group

URL	HTTP method
<code>http(s)://ip_or_hostname/api/v1/ProfIp_address_group/new_group_name</code>	POST

Delete an IP group

URL	HTTP method
<code>http(s)://ip_or_hostname/api/v1/ProfIp_address_group/group_to_delete</code>	DELETE

Add a new member to an IP group

Note that members can only be added when expressed as an IP range.

URL	HTTP method
<code>http://ip_or_hostname/api/v1/profip_address_group/group_name/ProfIp_address_groupIpAddressGroupMember/member_ip_range</code>	POST

Example:

In the following example, the IP range 172.20.200.11-172.20.200.30 is added to the IP group named group3:

```
http://ip_or_hostname/api/v1/ProfIp_address_group/group3/ProfIp_address_groupIpAddressGroupMember/172.20.200.11-172.20.200.30
```

Delete a member from an IP group

URL	HTTP method
<code>http://ip_or_hostname/api/v1/ProfIp_address_group/group_name/ProfIp_address_groupIpAddressGroupMember/member_ip_range</code>	DELETE

Replace all members of an IP group

URL	HTTP method
<code>http://ip_or_hostname/api/v1/ProfIp_address_group/group_name/ProfIp_address_groupIpAddressGroupMember/</code>	PUT or POST

JSON data:

```
{ "reqAction":10, // set to 10 to replace all existing members
  "reqObjCount":n, // size of the new member list
  "mkey_0":"ip_range_1", // first new member
  "mkey_1":"ip_range_2", // second new member
  ...
  "mkey_[n-1]":"ip_range_n" // nth new member
}
```

Example:

The following URL and will clear the member list for the IP group group3, then add two new members:

```
http://ip_or_hostname/api/v1/ProfIp_address_group/group3/ProfIp_address_groupIpAddressGroupMember/
```

JSON data:

```
{ "reqAction":10,
  "reqObjCount":2,
  "mkey_0":"172.20.110.10-172.20.110.20",
  "mkey_1":"172.20.130.10-172.20.130.30"
}
```

REST API for impersonation rule management

Impersonation rules lists and entries of each list can be created, deleted, and managed through the REST API.

View rules list

Domain level

```
curl -b cookie.txt http://<ip >/api/v1/domain/<domain> /ProfImpersonation/<profile_  
name>/ProfImpersonationEntry
```

System level

```
curl -b cookie.txt https://<ip>/api/v1/ProfImpersonation/<profile_name>/ProfImpersonationEntry
```

View match rules entry

Domain level

```
curl -b cookie.txt http://<ip >/api/v1/domain/<domain> /ProfImpersonation/<profile_  
name>/ProfImpersonationEntry/<mkey>
```

System level

```
curl -b cookie.txt http://<ip >/api/v1/ProfImpersonation/<profile_  
name>/ProfImpersonationEntry/<mkey>
```

Add a new rule entry



For display_name_type, enter 0 for Wildcard or 1 for Regex.

Domain level

URL	HTTP method
<code>http://<ip>/api/v1/domain/<domain>/ProfImpersonation/<profile_name>/ProfImpersonationEntry/<mkey></code>	POST

Example:

```
curl -b cookie.txt -H 'content-type: application/json' -X POST -d '{"display_name":"hello world","display_name_type":0,"email_address":"hello@world.com"}'
http://<ip>/api/v1/domain/<domain>/ProfImpersonation/<profile_name>/ProfImpersonationEntry/<mkey>
```

System level

URL	HTTP method
<code>http://<ip>/api/v1/ProfImpersonation/<profile_name>/ProfImpersonationEntry/<mkey></code>	POST

Example:

```
curl -b cookie.txt -H 'content-type: application/json' -X POST -d '{"display_name":"hello world","display_name_type":0,"email_address":"hello@world.com"}'
http://<ip>/api/v1/ProfImpersonation/<profile_name>/ProfImpersonationEntry/<mkey>
```

Modify a rule entry



For display_name_type, enter 0 for Wildcard or 1 for Regex.

Domain level

URL	HTTP method
<code>http://<ip>/api/v1/domain/<domain>/ProfImpersonation/<profile_name>/ProfImpersonationEntry/<mkey></code>	PUT

Example:

```
curl -b cookie.txt -H 'content-type: application/json' -X PUT -d '{"display_name":"new hello world"}' http://<ip>/api/v1/domain/<domain>/ProfImpersonation/<profile_name>/ProfImpersonationEntry/<mkey>
```

System level

URL	HTTP method
<code>http://<ip>/api/v1/ProfImpersonation/<profile_name>/ProfImpersonationEntry/<mkey></code>	PUT

Example:

```
curl -b cookie.txt -H 'content-type: application/json' -X PUT -d '{"display_name":"new hello world"' http://<ip>/api/v1/ProfImpersonation/<profile_name>/ProfImpersonationEntry/<mkey>
```

Delete a rule entry

Domain level

URL	HTTP method
<code>http://<ip>/api/v1/domain/<domain>/ProfImpersonation/<profile_name>/ProfImpersonationEntry/<mkey></code>	DELETE

Example:

```
curl -b cookie.txt -H 'content-type: application/json' -X DELETE http://<ip>/api/v1/domain/<domain>/ProfImpersonation/<profile_name>/ProfImpersonationEntry/<mkey>
```

System level

URL	HTTP method
<code>http://<ip>/api/v1/ ProfImpersonation/<profile_name>/ProfImpersonationEntry/<mkey></code>	DELETE

Example:

```
curl -b cookie.txt -H 'content-type: application/json' -X DELETE http://<ip>/api/v1/ProfImpersonation/<profile_name>/ProfImpersonationEntry/<mkey>
```

System resource list and URLs



Resources marked with * also apply to domain level REST APIs.

URL	HTTP method	Summary
/AddressBook/ *	GET, POST, PUT, DELETE	Contacts
/AddressbookGroup/ *	GET, POST, PUT, DELETE	Contact groups
/ArchAccount/	GET, POST, PUT, DELETE	Archive accounts
/ArchExempt/	GET, POST, PUT, DELETE	Archive exempt policy
/ArchJournalSource/	GET, POST, PUT, DELETE	Archive journaling source
/ArchPolicy/	GET, POST, PUT, DELETE	Archive policy
/AsBounceverifyKey/	GET, POST, PUT, DELETE	Bounce verification keys
/AsDeepheader/	GET, PUT	Deep header analysis settings
/AsGreylist/	GET	Greylist
/AsGreylistyAutoexempt/	GET	Auto exempt greylist
/AsMsisdnReputationAuto_blocklist/	GET	Endpoint reputation auto blocklist
/AsMsisdnReputationblocklist/	GET, DELETE	Endpoint reputation blocklist
/AsMsisdnReputationExempt/	GET, DELETE	Endpoint reputation exempt list
/AsSenderReputation/	GET	Sender reputation list
/AsSpamreport/	GET, PUT	Quarantine / spam report settings
/AsUrl_fgas_exempt_list/	GET, POST, PUT, DELETE	URL exempt list
/CalResource/ *	GET, POST, PUT, DELETE	
/CalendarServer/	GET, PUT	Calendar server settings

URL	HTTP method	Summary
/CentralBackupConfig/	GET, PUT	Central backup configuration
/CentralConfigList/	GET, DELETE	Central backup list
/ContentScanRules/	GET, POST, PUT, DELETE	DLP content scan rules
/ContentScanRulesConditions/	GET, POST, PUT, DELETE	DLP content scan rule conditions
/ContentScanRulesExceptions/	GET, POST, PUT, DELETE	DLP content scan rule exceptions
/domain/	GET, POST, PUT, DELETE	Protected domain settings
/DomainSettingSenderAddrRateCtrlExempt/	GET, POST, PUT, DELETE	Sender rate control exempt list for specified domain settings
/DomainSpamReportRcpt/	GET, PUT	Domain level quarantine / spam report settings
/FilePattern/ *	GET, POST, PUT, DELETE	File patterns / filters
/FileSignature/	GET, POST, PUT, DELETE	File signatures for AV scan
/Fingerprint_doc/	GET, DELETE	Fingerprint document list
/LogAltMMailto/	GET, POST, DELETE	Alert email accounts
/LogAltMSetting/	GET, PUT	Alert email settings
/LogReport_config/	GET, POST, PUT, DELETE	Log report configurations
/LogReportFile/	GET, DELETE	Log report files
/LogSetLocal/	GET, PUT	Local log settings
/MailSetStrgNfs/	GET, PUT	Mail storage settings
/MailSetStrgRemote_storage_ibe/	GET, PUT	Centralized IBE storage settings
/MailSetStrgServer/	GET, PUT	Centralized quarantine storage settings
/MailSetSystemquarantine/	GET, PUT	System quarantine settings
/MailSetSystemquarantineFolder/	GET, POST, PUT, DELETE	System quarantine folders
/PolicyIp/	GET, POST, PUT, DELETE	IP policies
/PolicyRecipient/ *	GET, POST, PUT, DELETE	Recipient policies

URL	HTTP method	Summary
/ProfAntispam/ *	GET, POST, PUT, DELETE	AntiSpam profiles
/ProfAntispam_action/ *	GET, POST, PUT, DELETE	AntiSpam action profiles
/ProfAntispamBannedwords/ *	GET, POST, PUT, DELETE	AntiSpam profile banned words
/ProfAntispamDnsblServer/ *	GET, POST, PUT, DELETE	AntiSpam profile DNSBL servers
/ProfAntispamSurbServer/ *	GET, POST, PUT, DELETE	AntiSpam profile SURBL servers
/ProfAntispamWhitelistwords/ *	GET, POST, PUT, DELETE	AntiSpam profile safelist words
/ProfAntivirus/ *	GET, POST, PUT, DELETE	AntiVirus profiles
/ProfAntivirus_action/ *	GET, POST, PUT, DELETE	AntiVirus action profiles
/ProfAuthImap/ *	GET, POST, PUT, DELETE	IMAP authentication profiles
/ProfAuthPop3/ *	GET, POST, PUT, DELETE	POP3 authentication profiles
/ProfAuthRadius/ *	GET, POST, PUT, DELETE	RADIUS authentication profiles
/ProfAuthSmtp/ *	GET, POST, PUT, DELETE	SMTP authentication profiles
/ProfCertificate_binding/	GET, POST, PUT, DELETE	Certificate binding profiles
/ProfContent/ *	GET, POST, PUT, DELETE	Content profiles
/ProfContent_action/ *	GET, POST, PUT, DELETE	Content action profiles
/ProfContentAttachment/ *	GET, POST, PUT, DELETE	Content profile attachment scan rules
/ProfContentMonitor/ *	GET, POST, PUT, DELETE	Content monitor profile
/ProfDictionary/	GET, POST, PUT, DELETE	Dictionary profiles

URL	HTTP method	Summary
/ProfDictionary_group/	GET, POST, PUT, DELETE	Dictionary group
/ProfDictionaryDictionaryItem/	GET, POST, PUT, DELETE	Dictionary profile dictionary entries
/ProfDlp/	GET, POST, PUT, DELETE	DLP profiles
/ProfDlpContentScan/	GET, POST, PUT, DELETE	DLP profile content scan settings
/ProfEmail_address_group/	GET, POST, PUT, DELETE	Email address groups
/ProfEncryption/	GET, POST, PUT, DELETE	Encryption profiles
/ProfIp_address_group/	GET, POST, PUT, DELETE	IP address groups
/ProfIp_pool/	GET, POST, PUT, DELETE	IP pools
/ProfLdap/	GET, POST, PUT, DELETE	LDAP profiles
/ProfMisc/ *	GET, POST, PUT, DELETE	Resource profiles
/ProfNotification/	GET, POST, PUT, DELETE	Notification profiles
/ProfSessionRecipientWhitelist/	GET, POST, PUT, DELETE	Session profile recipient safelist
/ProfSessionRemovedHeader/	GET, POST, PUT, DELETE	Session profile removed headers
/ProfSessionSenderBlacklist/	GET, POST, PUT, DELETE	Session profile sender blacklist
/ProfSessionSenderWhitelist/	GET, POST, PUT, DELETE	Session profile sender safelist
/ProfTls/	GET, POST, PUT, DELETE	TLS profiles
/ProfUri_filter/	GET, POST, PUT, DELETE	URI filter profiles
/RaidSystem/	GET	Raid system status
/RaidSystemArray/	GET	Raid array information

URL	HTTP method	Summary
/RaidSystemDisk/	GET	Raid disk information
/SemailDbDomain/	GET, DELETE	IBE domains
/SemailDbUser/	GET, DELETE	IBE users
/Sensitive_dataCompliance/	GET	DLP standard compliance data
/Sensitive_dataFingerprint/	GET, POST, PUT, DELETE	DLP fingerprint data
/Sensitive_dataFingerprint_source/	GET, POST, PUT, DELETE	DLP fingerprint source
/Sensitive_dataFingerprintDocument/	GET, DELETE	DLP fingerprint documents
/SysAccprofile/	GET, POST, PUT, DELETE	Admin access profiles
/SysAdmin/	GET, POST, PUT, DELETE	System administrators
/SysAntispam/	GET, PUT	System AntiSpam settings
/SysAppearance/	GET, PUT	System appearance settings
/SysAutoupdate/	GET, PUT	FortiGuard AntiVirus auto update settings
/SysBackup_restore/	GET, PUT	Mail data auto backup settings
/SysBurstRestore/	PUT	Restore mail data from backup
/SysDateSetting/	GET, PUT	System date
/SysDdns/	GET, POST, PUT, DELETE	DDNS servers
/SysDisclaimer/	GET, PUT	System disclaimer settings
/SysDisclaimer_exclude/	GET, POST, PUT, DELETE	Disclaimer exclusion list
/SysDns/	GET, PUT	System DNS server settings
/SysEncryptionIbe/	GET, PUT	IBE encryption settings
/SysEncryptionIbe_auth/	GET, POST, PUT, DELETE	IBE user authentication list
/SysFortiguard/	GET, PUT	FortiGuard AntiSpam settings
/SysFortisandbox/	GET, PUT	FortiSandbox settings
/SysGlobal/	GET, PUT	System global settings
/SysHa/	GET, PUT	HA settings
/SysHaInterface/	GET, PUT	HA interface settings

URL	HTTP method	Summary
/SysHaService/	GET, PUT	HA service monitor settings
/SysHaStatus/	GET	HA status
/SysInterface/	GET, POST, PUT, DELETE	Network interface list
/SysLink_monitor/	GET, PUT	Link monitor settings
/SysLink_monitorInterface/	GET, PUT	Link monitor interface settings
/SysMailserver/	GET, PUT	Mail server settings
/SysRemote_mail_server/	GET, POST, PUT, DELETE	Remote email servers
/SysRoute/	GET, POST, PUT, DELETE	Network routing list
/SysSched_backup/	GET, PUT	System configuration scheduled backup settings
/SysScheduledLocalBackup/	GET, PUT	Scheduled local backup list
/SysSnmpCommunity/	GET, POST, PUT, DELETE	SNMP communities
/SysSnmpCommunityHost/	GET, POST, PUT, DELETE	Hosts for a SNMP community
/SysSnmpSnmpv3_user/	GET, POST, PUT, DELETE	SNMP users
/SysSnmpSnmpv3_userHost/	GET, POST, PUT, DELETE	Notification hosts for a SNMP user
/SysSnmpSysinfo/	GET, PUT	SNMP system information settings
/SysSnmpThreshold/	GET, PUT	SNMP threshold settings
/SysStatisticSummary/	GET	Mail statistics summary
/SysStatusCommand/	POST	Restart / Shut down / Reload system command
/SysStatusLicinfo/	GET	System AS / AV license status
/SysStatusLicinfoLicenses/	GET	Feature license list
/SysStatusSysinfo/	GET	System status information
/SysStatusUsage/	GET	System resource usage
/SysTimeManual/	GET, PUT	System time & zone settings
/SysTimeNtp/	GET, PUT	System NTP server settings

URL	HTTP method	Summary
/SysWccpSettings/	GET, PUT	FortiGate WCCP settings
/UserAlias/ *	GET, POST, PUT, DELETE	User aliases
/UserMail/ *	GET, POST, PUT, DELETE	Mail users
/UserMap/ *	GET, POST, PUT, DELETE	Address maps
/UserPki/	GET, POST, PUT, DELETE	PKI users
/UserUser_group/ *	GET, POST, PUT, DELETE	User groups

Example commands

This section contains various example commands that you may encounter.

Admin login with PKI certificate-based authentication

```
curl -v -c cookie.txt --cert test.pem --key test.key --cacert CA.cer -X POST -k https://ip_or_host/api/v1/AdminLogin
```

If login is successful, the cookies will be saved to cookie.txt, which will be used in below commands.

Admin login with password-based authentication

```
curl -v -H "Content-Type: application/json" -X POST -d '{"name":"admin","password":"*****"}' https://ip_or_host/api/v1/AdminLogin -c cookie.txt
```

If login is successful, the cookies will be saved to cookie.txt, which will be used in below commands.

If "post_login_banner:true" is present in the json response, issue the second request as below, and use newcookie.txt, instead of cookie.txt, in below commands.

```
curl -k -v --cookie aaaa.txt -H "Content-Type: application/json" -X POST -d '{"name": "admin", "password": "*****", "reqAction":7}' https://ip_or_host/api/v1/AdminLogin [^] -c newcookie.txt
```

To get domain information

If you log in to FortiMail with a PKI certificate, you must use both the cookie and certificate together to run the command sessions. If you log in with user name and password, you only need to use the cookie to run the command sessions. For example:

To get domain information with password-based authentication

```
curl -k -v --cookie cookie.txt https://ip_or_host/api/v1/Domain
```

To get domain information with certificate-based authentication

```
curl -k -v --cookie cookie.txt --cacert CA.cer https://ip_or_host/api/v1/Domain
```

To create a domain association

```
curl -v -H "Content-Type: application/json" -X POST -k -d '{  
  "domain": "example.com"}' -b cookie.txt http://ip_or_  
host/api/v1/MailSetDomainAssociation/example_domain_association.cn
```

Access control rule management

Supported values for 'action' attribute of ACL. If not set, the default action is reject:

- 1: safe-relay
- 2: relay
- 3: reject
- 4: discard
- 5: safe

Supported values for 'recipient-pattern-type' and 'sender-pattern-type' attributes of ACL:

- 0: default / wild-card
- 1: regular expression
- 2: email group
- 3: ldap group

Supported values for 'authenticated' attribute of ACL:

- 0: any
- 1: authenticated
- 2: not-authenticated

Supported values for 'sender-ip-type' attribute of ACL:

- 0: ip mask
- 1: ip group

Supported values for 'sortingDirection' attribute of applicable JSON requests (i.e. ACL rule):

- 1: enabled/sorting
- 2: disabled/no sorting

Supported values for 'reqAction' attribute of all JSON requests:

- 1: GET

2: CREATE
3: DELETE
5: UPDATET
14: MOVE



If reqAction is present in JSON, it takes precedence over HTTP method header (i.e. HTTP GET/POST/PUT/DELETE).

List access control rules

To list ACL rules in original order:

```
curl -v -b cookie.txt -v -H "Content-Type: application/json" -X PUT -d '{"reqAction":1,"sortingDirection":2}' http://ip_or_host/api/v1/MailSetAccessRule
```

To list ACL rules in descending order:

```
curl -v -b cookie.txt -v -H "Content-Type: application/json" -X PUT -d '{"reqAction":1,"sortingDirection":1}' http://ip_or_host/api/v1/MailSetAccessRule
```

Create a new access control rule

```
curl -v -H "Content-Type: application/json" -X POST -d '{"status":true,"sender_pattern":"*@example.com","sender_ip_mask":"192.168. 1.1/32", "action":2}' -b cookie.txt http://ip_or_host/api/v1/MailSetAccessRule/0 --No mkey is required
```

Modify an existing access control rule

```
curl -v -H "Content-Type: application/json" -X PUT -d '{"action":3}' -b cookie.txt http://ip_or_host/api/v1/MailSetAccessRule/1 --"1" is the mkey --Set access rule "1" action to "Reject"
```

Delete an existing access control rule

```
curl -v -H "Content-Type: application/json" -X DELETE -b cookie.txt http://ip_or_host/api/v1/MailSetAccessRule/3 --Delete Access Control Rule "3"
```

Move existing access control rule

You can move a rule up one place, down one place, before another rule, or after another rule. To move a rule to the top or bottom, you can find the first. or last rule ID (mkey) and move the rule before the first rule or after the last rule.

reqAction: 14 -- required, only one value: 14 means to move

moveAction : up -- required, 4 values: up/down/before/after

mmkey: 3 – required, ID of the ACL rule to be moved

refMkey: 2 – required, reference ID of the ACL rule when moving before/after this ID

To move rule "3" up one place:

```
curl -v -H "Content-Type: application/json" -X PUT -d '{
  "reqAction": "14", "mmkey": 3, "moveAction": "up"}' -b cookie.txt http://ip_or_
host/api/v1/MailSetAccessRule
```

To move rule "3" after rule "2":

```
curl -v -H "Content-Type: application/json" -X PUT -d '{
  "reqAction": "14", "mmkey": 3, "moveAction": "after", "refMkey": 2}' -b cookie.txt http://ip_
or_host/api/v1/MailSetAccessRule
```

Safelist/blocklist management

To get the system level safelist:

```
curl -v -b cookie -H "Content-Type: application/json" -X POST -d '{
  "reqAction": "1", "extraParam": "safelist"}' http://IP_Host/api/v1/SenderListV2/system/
```

To create a new system level safelist entry:

```
curl -v -b cookie -H "Content-Type: application/json" -X POST -d '{"extraParam": "safelist",
  "listitems": "ip_or_email"}' http://IP_Host/api/v1/SenderListV2/system/
```

To create a system level safelist entry with a comment:

```
curl -v -k -b cookie -H "Content-Type: application/json" -X POST -d '{"extraParam": "safelist",
  "listitems": "*@test.com: This is a test comment"}' https://ip_or_
hostname/api/v1/SenderListV2/system/
```

To delete a system level safelist entry:

```
curl -v -b cookie -H "Content-Type: application/json" -X POST -d '{"reqAction": 3,
  "extraParam": "safelist", "listitems": "ip_or_email"}' http://IP_
Host/api/v1/SenderListV2/system/
```



www.fortinet.com

Copyright© 2023 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 General Counsel, 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.