
DuoSTATION MDM REST API

v4.8.46

Stefan Johansson

2024-11-09

Contents

Introduction.....	3
Technique	3
Authentication.....	3
Parameters	3
Manage groups.....	3
Get groups	3
Get specific group by id	3
Get specific group by name.....	4
Update a group.....	4
Add a group	4
Delete group.....	5
Manage users	5
Get users.....	5
Get user specified by id	5
Search for users	6
Update a user	6
Add a user.....	7
Delete a user.....	8
Manage devices.....	8
Get devices	8
Get specific device by id	8
Search for devices.....	9
Update a device.....	9
Manage tags	9
Get tags.....	9
Get specific tag by id	10
Update a tag	10
Add a tag.....	11
Delete a tag.....	11
Manage device tags.....	11
Get device tags	11
Get specific device tag.....	12
Search for device tags.....	12

Link a tag to a device	12
Unlink a tag from a device.....	13
Manage user tags	13
Get user tags.....	13
Get specific user tag	13
Search for user tags	14
Link a tag to a user.....	14
Unlink a tag from a user	15
Manage group tags.....	15
Get group tags	15
Get specific group tag.....	15
Link a tag to a group	16
Unlink a tag from a group.....	16
Manage apps	16
Get apps.....	16
Get specific app	17
Add or update app.....	18
Send commands to devices	19
Inventory device	19
Install app	19
Uninstall app.....	19
Lock screen	20
Wipe device	20
Clear passcode.....	20
Send message	21
Revisions	22
Support	22

Introduction

This documentation contains information how to use the DuoSTATION MDM REST API. This API can be used to send and get data to/from DuoSTATION MDM.

<url> in each request should be replaced with the domain name of your own site.

Technique

Authentication

The API uses Basic Authentication. User name and password is the same that can be used to log in to DuoSTATION MDM and must be included in each request.

Parameters

Text parameters in URLs must be URL encoded.

Data in DELETE-, PUT- and POST-commands must be structured as JSON objects.

200 OK will be returned for successful requests.

Returned data will be structured as a JSON object or an array of JSON objects.

Manage groups

Get groups

Descriptions

Get all available groups.

Request

GET `https://<url>/api/groups`

Parameters

-

In data

-

Response data

Array of groups.

Get specific group by id

Description

Get a group specified by id.

Request

GET `https://<url>/api/groups/<id>`

Parameters

<id> is the unique id (integer) for the requested group.

In data

-

Response data

Requested group.

Get specific group by name

Description

Get a group by the name of the group. For the moment it is possible to add several groups with the same name (which is not recommended). If more than one group matches the name an error will be returned.

Request

GET `https://<url>/api/groups?path=<path>`

Parameters

<path> is the full path for the group separated with / or \. The value must be URL encoded.

In data

-

Response data

Requested group.

Example

Get the group 'Company Inc' which is a subgroup to 'Customer' (Customers\Company Inc):

GET `https://domain.com/api/groups?path=Customers%5CCompany%20Inc`

Update a group

Description

Change the name of a group specified by id.

Request

PUT `https://<url>/api/groups/<id>`

Parameters

<id> is the unique id (integer) that identifies the group that will be updated.

In data

```
{ "name": "<name>" }
```

<name> is the new name of the group. Required.

Response data

-

Add a group

Description

Add a new group.

Request

POST `https://<url>/api/groups`

Parameters

-

In data

```
{ "name": "<name>", "parentId": parentId }
```

<name> is the name of the new group. Required.

<parentId> is the id (integer) of the group which the new group will have as parent group. If this value is 0, or omitted, the new group will be placed directly under the root in the group tree.

Response data

Id of the newly created group.

Delete group

Description

Delete a group specified by id. A group cannot be deleted if the group has subgroups or if other objects are linked to the group.

Request

DELETE `https://<url>/api/groups/<id>`

Parameters

<id> is the unique id (integer) that specifies which group to delete.

In data

-

Response data

-

Manage users

Get users

Description

Get all available users.

Request

GET `https://<url>/api/users`

Parameters

-

In data

-

Response data

Array of users.

Get user specified by id

Description

Get a user specified by id.

Request

GET `https://<url>/api/users/<id>`

Parameters

<id> is the unique id (integer) of the requested user.

In data

–

Response data

Requested user.

Search for users

Description

Search for user(s) by email, phone number or account name.

Request

GET `https://<url>/api/users?email=<email>`

GET `https://<url>/api/users?phoneNumber=<phoneNumber>`

GET `https://<url>/api/users?account=<account>`

GET `https://<url>/api/users?groupId=<groupId>`

GET `https://<url>/api/users?tagId=<tagId>`

Parameters

<email> is the email address of the user(s) to search for.

<phoneNumber> is the phone number of the user(s) to search for.

<account> is the account name of the user(s) to search for.

<groupId> is the id of the group linked to the user(s) to search for.

<tagId> is the id of the tag linked to the user(s) to search for.

The parameter values must be URL encoded.

It is possible to combine the parameters.

In data

–

Response data

Array of users.

Example

Get user(s) with the email address stefan@programduon.se:

GET `https://domain.com/api/users?email=stefan%40programduon.se`

Update a user

Description

Change the name, email, phone number and/or group of a user specified by id. If the group of the user is changed, the devices of the user also will be moved. These devices will be reconfigured automatically.

Request

PUT `https://<url>/api/users/<id>`

Parameters

<id> is the unique id (integer) of the user that will be updated.

In data

```
{ "name": "<name>", "email": "<email>", "phoneNumber": "<phoneNumber>",  
  "groupId": <groupId>, "company": "<company>", "office": "<office>",  
  "department": "<department>", "position": "<position>" }
```

<name> is the new name of the user.

<email> is the new email address of the user.

<phoneNumber> is the new phone number of the user.

<groupId> is the id of the new group of the user.

<company> is the new company of the user.

<office> is the new office of the user.

<department> is the new department of the user.

<position> is the new position of the user.

Response data

-

Add a user**Description**

Add a user.

Request

POST `https://<url>/api/users`

Parameters

-

In data

```
{ "name": "<name>", "email": "<email>", "phoneNumber": "<phoneNumber>",  
  "accountGroup": "<accountGroup>": "accountName": "<accountName>",  
  "groupId": <groupId>, "company": "<company>", "office": "<office>",  
  "department": "<department>", "position": "<position>" }
```

<name> is the name of the new user. Required.

<email> is the email address of the new user.

<phoneNumber> is the phone number of the new user.

<accountGroup> is the account group of the new user.

<accountName> is the account name of the new user.

<groupId> is the id of the group of the new user.

<company> is the company of the user.

<office> is the office of the user.

<department> is the department of the user.

<position> is the position of the user.

Response data

Id of the newly added user.

Delete a user

Description

Delete a user specified by id. A user cannot be deleted if the user has linked devices.

Request

DELETE `https://<url>/api/users/<id>`

Parameters

<id> is the unique id (integer) of the user that will be deleted.

In data

-

Response data

-

Manage devices

Get devices

Description

Get all available devices.

Request

GET `https://<url>/api/devices`

Parameters

-

In data

-

Response data

Array of devices.

Get specific device by id

Description

Get a device specified by id.

Request

GET `https://<url>/api/devices/<id>`

Parameters

<id> is the unique id (GUID) of the requested device.

In data

-

Response data

Requested device.

Search for devices

Description

Search for device(s) by serial number.

Request

GET `https://<url>/api/devices?serialNumber=<serialNumber>`

GET `https://<url>/api/devices?groupId=<groupId>`

GET `https://<url>/api/devices?userId=<userId>`

GET `https://<url>/api/devices?accountName=<accountName>`

GET `https://<url>/api/devices?tagId=<tagId>`

Parameters

<serialNumber> is the serial number of the device(s) to search for. The value must be URL encoded.

<groupId> is the id of the group linked to the device(s) to search for.

<userId> is the id of the user linked to the device(s) to search for.

<accountName> is the account name of the user linked to the device(s) to search for.

<tagId> is the id of the tag linked to the device(s) to search for.

The parameter values must be URL encoded.

It is possible to combine the parameters.

In data

-

Response data

Array of devices.

Update a device

Description

Change linked user of a device specified by id. The device will be reconfigured if needed.

Request

PUT `https://<url>/api/devices/<id>`

Parameters

<id> is the unique id (GUID) of the device that will be updated.

In data

```
{ "userId": <userId> }
```

<userId> is the unique id (integer) of the user. Required.

Response data

-

Manage tags

Get tags

Description

Get all available tags.

Request

GET `https://<url>/api/tags`

Parameters

-

In data

-

Response data

Array of tags.

Get specific tag by id

Description

Get tag specified by id.

Request

GET `https://<url>/api/tags/<id>`

Parameters

<id> is the unique id (integer) of the requested tag.

In data

-

Response data

Requested tag.

Update a tag

Description

Update name, description and/or group of a tag specified by id. If the group of the tag is updated, the tag will be moved. Devices that are affected by this, will be automatically reconfigured.

Request

PUT `https://<url>/api/tags/<id>`

Parameters

<id> is the unique id (integer) of the tag that will be updated.

In data

```
{ "name": "<name>", "description": "<description>", "groupId": <groupId>
}
```

<name> is the new name of the tag.

<description> is the new description of the tag.

<groupId> is the id of the new group of the tag.

Response data

-

Add a tag

Description

Add a tag.

Request

POST `https://<url>/api/tags`

Parameters

-

In data

```
{ "name": "<name>", "description": "<description>", "groupId": <groupId>
}
```

<name> is the name of the new tag. Required.

<description> is the description of the new tag.

<groupId> is the id of the group of the new tag.

Response data

Id of the newly created tag.

Delete a tag

Description

Delete a tag specified by id. A tag cannot be deleted if the tag has linked configurations.

Request

DELETE `https://<url>/api/tags/<id>`

Parameters

<id> is the unique id (integer) of the tag that will be deleted.

In data

-

Response data

-

Manage device tags

Get device tags

Description

Get all available device tags.

Request

GET `https://<url>/api/devicetags`

Parameters

-

In data

-

Response data

Array of device tags.

Get specific device tag

Description

Get a device tag specified by id.

Request

GET `https://<url>/api/devicetags/<id>`

Parameters

<id> is the unique id (integer) for requested device tag.

In data

–

Response data

Requested device tag.

Search for device tags

Description

Search for tags linked to a specific device by unique id or serial number.

Request

GET `https://<url>/api/devicetags?deviceId=<deviceId>`

GET `https://<url>/api/devicetags?serialNumber=<serialNumber>`

Parameters

<deviceId> is the unique id (GUID) of the device.

<serialNumber> is the serial number of the device.

The parameter values must be URL encoded.

It is possible to combine these parameters.

In data

–

Response data

Array of device tags.

Example

Get all tags that are linked to the device with serial number SN123456ABC:

GET `https://domain.com/api/devicetags?serialNumber=SN123456ABC`

Link a tag to a device

Description

Add a tag to a device. The device will be reconfigured if needed.

Request

POST `https://<url>/api/devicetags`

Parameters

-

In data

```
{ "deviceId": "<deviceId>", "tagId": <tagId> }
```

<deviceId> is the unique id (GUID) of the device. Required.

<tagId> is the unique id (integer) of the tag. Required.

Response data

Id of the newly created device tag.

Unlink a tag from a device

Description

Remove a tag from a device. The device will be reconfigured if needed.

Request

DELETE `https://<url>/api/devicetags/<id>`

Parameters

<id> is the unique id (integer) of the device tag that will be deleted.

In data

-

Response data

-

Manage user tags

Get user tags

Description

Get all available user tags.

Request

GET `https://<url>/api/usertags`

Parameters

-

In data

-

Response data

Array of user tags.

Get specific user tag

Description

Get a user tag specified by id.

Request

GET `https://<url>/api/usertags/<id>`

Parameters

<id> is the unique id (integer) of the requested user tag.

In data

-

Response data

Requested user tag.

Search for user tags

Description

Search for tags linked to a specific user by unique id or email address.

Request

GET `https://<url>/api/usertags?userId=<userId>`

GET `https://<url>/api/usertags?email=<email>`

Parameters

<userId> is the unique id (integer) of the user.

<email> is the email address of the user.

The parameter values must be URL encoded.

It is possible to combine these parameters.

In data

-

Response data

Array of user tags.

Example

Get all tags that are linked to the user with email address stefan@programduon.se:

GET `https://domain.com/api/usertags?email=stefan%40programduon.se`

Link a tag to a user

Description

Add a tag to a user. If the user has linked device, these will be reconfigured if needed.

Request

POST `https://<url>/api/usertags`

Parameters

-

In data

```
{ "userId": "<userId>", "tagId": <tagId> }
```

<userId> is the unique id (integer) of the user. Required.

<tagId> is the unique id (integer) of the tag. Required.

Response data

Id of the newly created user tag.

Unlink a tag from a user

Description

Remove a tag from user. If the user has linked devices, these will be reconfigured if needed.

Request

DELETE `https://<url>/api/usertags/<id>`

Parameters

<id> is the unique id (integer) of the user tag that will be deleted.

In data

-

Response data

-

Manage group tags

Get group tags

Description

Get all available group tags.

Request

GET `https://<url>/api/grouptags`

Parameters

-

In data

-

Response data

Array of group tags.

Get specific group tag

Description

Get a group tag specified by id.

Request**GET** `https://<url>/api/grouptags/<id>`**Parameters**

<id> is the unique id (integer) of the requested group tag.

In data

-

Response data

Requested group tag.

Link a tag to a group

Description

Add a tag to a group. If the group has linked device, these will be reconfigured if needed.

Request**POST** `https://<url>/api/grouptags`**Parameters**

-

In data

```
{ "groupId": "<groupId>", "tagId": <tagId> }
```

<groupId> is the unique id (integer) of the group. 0 for 'All'. Required.

<tagId> is the unique id (integer) of the tag. Required.

Response data

Id of the newly created group tag.

Unlink a tag from a group

Description

Remove a tag from group. If the group has linked devices, these will be reconfigured if needed.

Request**DELETE** `https://<url>/api/grouptags/<id>`**Parameters**

<id> is the unique id (integer) of the group tag that will be deleted.

In data

-

Response data

-

Manage apps

Get apps

Description

Get all available apps.

Request

GET `https://<url>/api/apps`

Parameters

-

In data

-

Response data

Array of apps.

Get specific app

Description

Get an app specified by id.

Request

GET `https://<url>/api/apps/<id>`

Parameters

<id> is the unique id (integer) for the requested app.

In data

-

Response data

Requested app.

Add or update app**Description**

Add or update an app. If the app already exists, the app will be updated with the provided values. The API matches the app by using the parameters: packageId, platform and store.

Request

POST `https://<url>/api/apps`

Parameters

-

In data

```
{ "name": "<name>", "description": "<description>", "company":  
"<company>", "packageId": "<packageId>": "versionText": "<versionText>",  
"versionNumber": <versionNumber>, "platform": <platform>, "size": <size>,  
"store": <store>, "storeId": "<storeId>", "imageData": "<imageData>",  
"imageUrl": "<imageUrl>" }
```

<name> the name of the app. Required.

<description> a description of the app.

<company> the app company. Required.

<packageId> the app identifier. Required.

<versionText> app version as text. Must be same version that the app reports when installed. Required.

<versionNumber> app version as integer. Used mainly for Android apps.

<platform> the app platform. 0 = iOS, 1 = Android, 2 = macOS, 3 = tvOS. Required.

<size> size of the app in bytes. Required.

<store> the app store. 0 = Appstore, 1 = Play, 2 = In house. Required.

<storeId> the app id/url. If store = 0, storeId must contain the Appstore id. If store = 1, storeId must contain same value as packageId. If store = 2 and platform = 2, storeId must be an URL to a .pkg file or .zip file containing an .app or .pkg file. Required.

<imageData> a base64-encoded PNG-image (preferable 64x64 pixels). Required.

<imageUrl> an url to a PNG-image (preferable 64x64 pixels). Required.

<imageData> and <imageUrl> cannot be combined. Only one of the values can be included.

Return data

Id of the added/updated app.

Send commands to devices

Inventory device

Description

Send an inventory command to the device specified by id.

Request

POST `https://<url>/api/devices/<id>/inventorydevice`

Parameters

<id> is the unique id (GUID) of the device that will be inventoried.

In data

-

Response data

-

Install app

Description

Install an app on a device specified by id. For Apple devices it is also possible to choose if a VPP should be used or not.

Request

POST `https://<url>/api/devices/<id>/installapp`

Parameters

<id> is the unique id (GUID) of the device.

In data

```
{ "appId": <appId>, "vpp": <vpp> }
```

<appId> is the unique id (integer) of the app to install. Required.

<vpp> true/false if a VPP license should be used or not.

Response data

-

Uninstall app

Description

Uninstall an app from a device specified by id.

Request

POST `https://<url>/api/devices/<id>/removeapp`

Parameters

<id> is the unique id (GUID) of the device.

In data

```
{ "identifier": "<identifier>" }
```

<identifier> is the identifier (com.xxxx.yyy) of the app to be uninstalled. Required.

Response data

-

Lock screen**Description**

Lock the screen of the device specified by id.

Request

POST `https://<url>/api/devices/<id>/lockdevice`

Parameters

<id> is the unique id (GUID) of the device which screen will be locked.

In data

-

Response data

-

Wipe device**Description**

Delete all data on the device and reset all settings to factory default.

Request

POST `https://<url>/api/devices/<id>/wipedevice`

Parameters

<id> is the unique id (GUID) of the device that will be wiped.

In data

-

Response data

-

Clear passcode**Description**

Clear the password/passcode that is used for the lock screen.

Request

POST `https://<url>/api/devices/<id>/clearpasscode`

Parameters

<id> is the unique id (GUID) of the device.

In data

-

Response data

-

Send message

Description

Send a message to a device specified by id.

Request

POST `https://<url>/api/devices/<id>/sendmessage`

Parameters

<id> is the unique id (GUID) of the device.

In data

```
{ "message": "<message>" }
```

<message> contains the text to send. Required.

Response data

-

Revisions

Date	MDM-version	Comments
2017-08-28	4.7.4	First version of this document.
2017-12-20	4.8.6	Added support to update email address and phone number of users. Fixed a bug that occurred when trying to create a subgroup to the root node.
2018-03-21	4.8.7	Bug fixes and optimizations. Fixed some minor errors in the documentation.
2018-04-29	4.8.8	Added support to manage tags via this API.
2018-10-16	4.8.16	Added support to send message to a device.
2021-03-08	4.8.27	Possibility to use more fields when adding and updating users.
2021-06-15	4.8.28	Bug fixes and optimizations.
2022-06-04	4.8.33	Added possibility to filter on tag or group when retrieving lists of users and devices. More information is returned for devices and for users.
2022-11-02	4.8.35	Added possibility to add or update an app by using this API.
2023-09-19	4.8.39	Added possibility to filter on user when retrieving lists of devices.
2024-11-09	4.8.46	GET /usertags/ also returns user name. GET /devicestags/ also returns serial number. Implemented support to manage group tags.

Support

If support of this API is needed, contact us on:

support@duostation.com

or

+46 31 7096066