



## IGEL Management Interface (IMI)

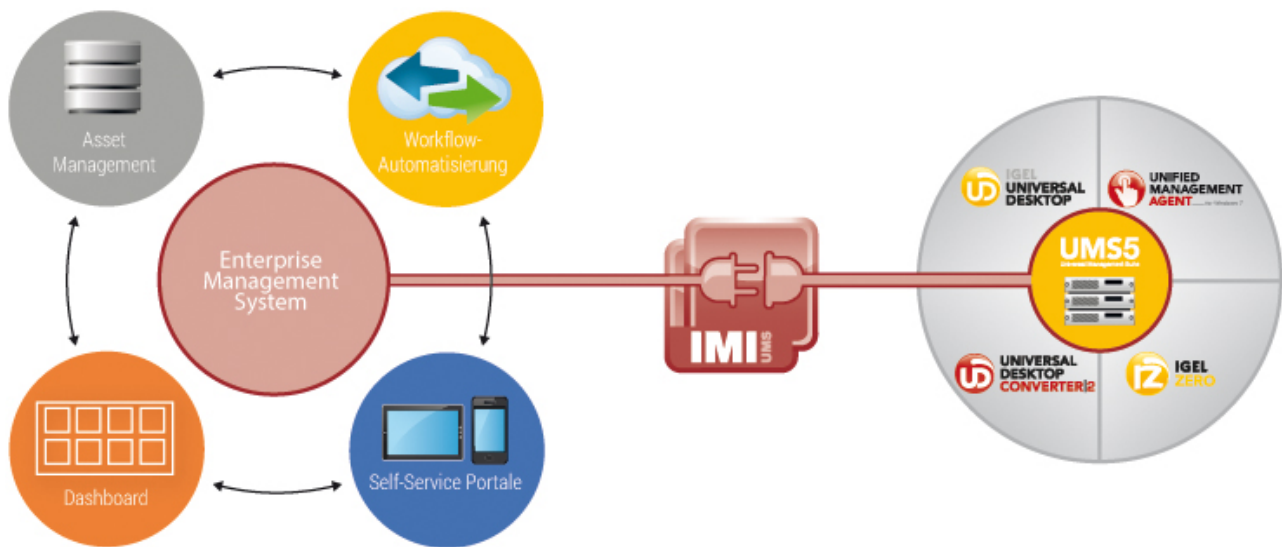


- [IMI Manual](#)(see page 3)
- [Powershell](#)(see page 35)
- [IMI API V2 Reference](#)(see page 87)
- [IMI How-Tos](#)(see page 244)



## IMI Manual

IGEL Management Interface (IMI) enables you to connect UMS to systems management tools. It is a programming interface that can create and delete thin clients, move them between directories, reboot them and much more. Its implementation as a REST API makes IMI agnostic of hardware platforms, operating systems and programming languages, thus ensuring maximum interoperability.



This document serves as an introduction to using IGEL Management Interface (IMI).

Detailed information about the requests can be found in the [IMI API V2 Reference](#)(see page 87).

- [Licensing](#)(see page 4)
- [REST Basics](#)(see page 5)
- [Prerequisites](#)(see page 8)
- [First Steps](#)(see page 9)
- [Creating, Updating and Deleting Resources](#)(see page 23)
- [Further Operations](#)(see page 30)



## Licensing

The licensing for IGEL Management Interface (IMI) depends on the UMS version you are using.

When Asset Inventory Tracker (AIT) has been licensed, you can use the resources [assetinfo](#) and [assethistory](#).

### UMS 5.09.100 or Older

For basic functionality, see Licensing IMI.

For Asset Inventory Tracker (AIT), see Licensing AIT.

### UMS 6.01.100 or Newer

IMI is part of the IGEL Workspace Edition (WE); no additional license is necessary. For further information, see Workspace Edition.

For Asset Inventory Tracker (AIT), licenses from the Enterprise Management Pack (EMP) are required. For further information, see Enterprise Management Pack.



## REST Basics

*IGEL Management Interface* uses *REST*, which stands for Representational State Transfer.

*REST* is an architectural style for client-server applications, mainly implemented in the HTTP(S) protocol. Therefore it can be used with all technologies that can send and receive HTTP requests.

*REST* establishes a typical pattern that helps programmers to understand the structure of individual APIs. Its most important concepts are [Ressources](#)(see page 6) and [HTTP Methods](#)(see page 7).



## Ressources

### URLs Represent Resources

A *REST* API makes resources available at specific URLs. You can find a list of all thin clients in the UMS REST API at:

- `https://[server]:8443/umsapi/v3/thinclients`

In a shorter notation, which assumes the base URL to be known:

- `/v3/thinclients`

In order to address an individual instance of a resource - a thin client, for example - you specify its ID in the URL:

- `/v3/thinclients/8`

### Further Examples of Resources

URL	Resource
<code>/v3/directories/tcdirectories</code>	A list of all thin client directories
<code>/v3/directories/tcdirectories/123</code>	The thin client directory with the ID 123
<code>/v3/firmwares</code>	A list of all firmwares in the UMS instance
<code>/v3/firmwares/7</code>	The firmware with the ID 7

Find a list of all available resources in the [IMI API Reference](#) (see page 87).



## HTTP Methods

You call HTTP methods for resources in order to manipulate them. The REST architectural style has a conventional meaning for each of the HTTP methods, which are also called verbs. The methods are listed below:

HTTP Method	
GET	Read information from a resource.
PUT	Create a new resource or update an existing resource.
POST	(Create a new resource *), send a command.
DELETE	Delete a resource.

\* There is a subtle semantic difference between PUT and POST, which is sometimes a matter of dispute. *IMI* API favors

- PUT for create and update actions and uses
- POST for logins and for sending commands to resources.



## Prerequisites

### IGEL Universal Management Suite (UMS)

UMS 5.02.100

### Networking

In order to use *IGEL Management Interface* you need to be able to reach the API host via the network and connect to its API port, TCP 8443 by default.

The base URL is

- `https://[server]:8443/umsapi/`

The resources for IMI version 2 are available at:

- `https://[server]:8443/umsapi/v2/`

❗ *IMI* uses HTTPS to ensure the integrity and confidentiality of the network traffic. It is good practice to use a valid server certificate with a verifiable signature. Most clients and libraries can be configured to work with self-signed or invalid certificates, but that should not be done in a production environment.





## First Steps

- [Client Applications and Libraries](#)(see page 10)
- [Getting the Server Status](#)(see page 12)
- [Authentication](#)(see page 13)
- [Listing all Thin Clients](#)(see page 14)
- [Getting Information on a Thin Client](#)(see page 16)
- [Getting All Details about a Thin Client](#)(see page 18)
- [Getting Thin Client Status](#)(see page 21)

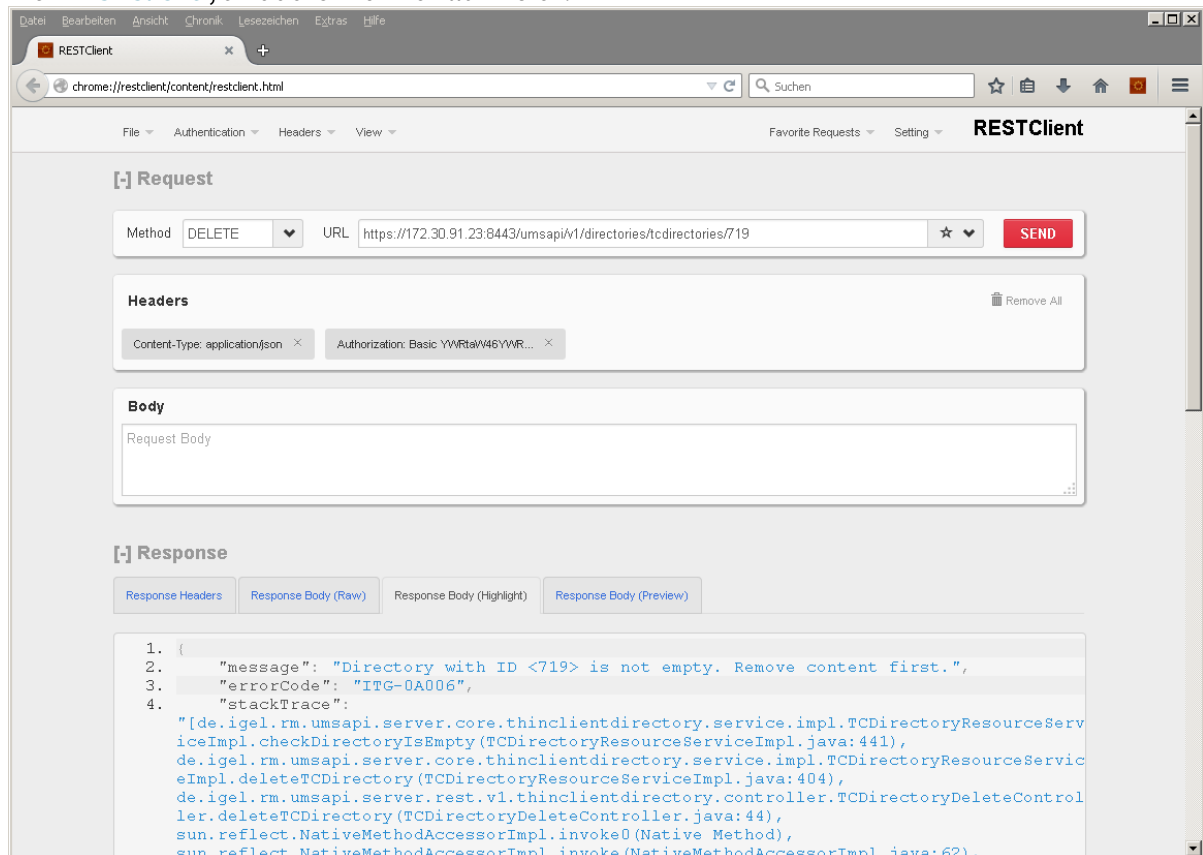


## Client Applications and Libraries

### Clients

The easiest way to try out the *IGEL Management Interface* is either

- with [RESTClient](http://restclient.net/)<sup>1</sup>, an add-on for Mozilla Firefox:



or

- with [cURL](http://curl.haxx.se/)<sup>2</sup>, a commandline network client:

<sup>1</sup> <http://restclient.net/>

<sup>2</sup> <http://curl.haxx.se/>

A screenshot of a terminal window with a blue title bar. The prompt is "Administrator@PM-MH-WIN7-UMS ~". The command entered is "\$ curl -ik --request GET https://localhost:8443/umsapi/v1/serverstatus". The output shows an HTTP 200 OK response with headers: "Server: Apache-Coyote/1.1", "Set-Cookie: JSESSIONID=A645FEC36AA9C0373F0D869CCDFC8C0B; Path=/umsapi/; Secure; Http Only", "Content-Type: application/json", "Transfer-Encoding: chunked", and "Date: Wed, 04 Mar 2015 12:44:19 GMT". The body is a JSON object containing server version, build number, activeMQ version, derby version, server UUID, server address, and a self-link. The prompt returns to "\$ " after the output.

```
Administrator@PM-MH-WIN7-UMS ~  
$ curl -ik --request GET https://localhost:8443/umsapi/v1/serverstatus  
HTTP/1.1 200 OK  
Server: Apache-Coyote/1.1  
Set-Cookie: JSESSIONID=A645FEC36AA9C0373F0D869CCDFC8C0B; Path=/umsapi/; Secure; Http  
Only  
Content-Type: application/json  
Transfer-Encoding: chunked  
Date: Wed, 04 Mar 2015 12:44:19 GMT  
  
{  
  "rmGuiServerVersion": "4.09.100.rc1",  
  "buildNumber": "21741",  
  "activeMQVersion": "5.6.0",  
  "derbyVersion": "10.8.3.0",  
  "serverUUID": "dd188671-e485-4679-8d2d-485be8a6275c",  
  "server": "172.30.91.23:8443",  
  "links": [{"rel": "self", "href": "https://localhost:8443/umsapi/v1/serverstatus"}]  
}  
Administrator@PM-MH-WIN7-UMS ~  
$
```

This guide uses cURL for examples, as the commandline makes all parameters visible in plain text. This need not stop you from using RESTClient if you are more comfortable with it. You can easily translate the commandline parameters into the fields of the RESTClient *RESTClient* GUI.

## Programming libraries

Most programming languages provide an HTTP and an SSL module, either in their standard library or as an extension, and a JSON library (for the API data format) as well.



## Getting the Server Status

► Send the endpoint `/v3/serverstatus` an HTTP GET request to retrieve some information about the server.

With *cURL* the commandline looks like this:

```
curl \
--request GET \
https://[server]:8443/umsapi/v3/serverstatus
```

You will receive a response like the following:

```
200 OK
---
{
  "rmGuiServerVersion": "5.07.100.rc15",
  "buildNumber": "32287",
  "activeMQVersion": "5.6.0",
  "derbyVersion": "10.12.1.1",
  "serverUUID": "9fe719f1-c16e-4744-9ff1-b9c314ae151c",
  "server": "mhuber:8443",
  "links": []
}
```

## Response Headers

The server replied with the status code 200 in the HTTP header, which means that your request was successful.

You can find a list of status codes in the [IMI API Reference](#)(see page 87).

## JSON Response Body

Information about the server is given in the response body in the *JavaScript Object Notation (JSON)* format. It is enclosed in curly braces and contains key-value-pairs separated by a colon. The line separator is a comma. Your programming language probably offers a `JSON` module that will help you parse and process information in this format.



## Authentication

You must be authenticated in order to use *IGEL Management Interface*, otherwise the server will return the HTTP status 401 "Unauthorized". Only querying the server status is allowed without authentication.

IMI uses *Basic Authentication* ([RFC 2617](#)(see page 13)).

## Logging In

► Send an HTTP POST request to `/v2/login` to log in. Your HTTP client of choice will offer a way of sending a *Basic Authentication*- header, which is produced from the username/password combination.

This is the commandline for *cURL*:

```
curl \
--request POST \
--user '[Username]:[Passwort]' \
https://[server]:8443/umsapi/v3/login
```

The server replies, sending back a session ID:

```
200 OK
Set-Cookie: JSESSIONID=3FB2F3F6A089FE9029DFD6DAFEF146DC; Path=/umsapi/; Secure;
HttpOnly
---
{"message": "JSESSIONID=3FB2F3F6A089FE9029DFD6DAFEF146DC"}
```

## Staying Logged in

► You can maintain the session by sending the JSESSIONID in the Cookie header with every subsequent request:

```
Cookie: JSESSIONID=3FB2F3F6A089FE9029DFD6DAFEF146DC
```

Some clients will do this automatically for you, e.g. the [RESTClient](#) (see page 10)[Firefox add-on](#)(see page 10).



## Listing all Thin Clients

- Send a GET request to `/v3/thinclients` to list all thin clients.
- Don't forget to tell the server your JSESSIONID:

```
curl \
--request GET\
--header 'Cookie: JSESSIONID=3FB2F3F6A089FE9029DFD6DAFEF146DC' \
https://[server]:8443/umsapi/v3/thinclients
```

## Response

The response contains a large *JSON* document, listing all thin clients and their most important properties. The links make further parts of the API discoverable for a client that spiders the interface.

Look up the format for a thin client resource in the [IMI API Reference](#) (see page 87).

The last thin client entry contains `"movedToBin": true`, meaning that the thin client has been deleted and moved to the **Recycling Bin**. Objects in the **Recycling Bin** are listed, but you cannot update them or send them commands.

```
[
{
  "unitID": "00E0C54DCB8E",
  "mac": "00E0C54DCB8E",
  "firmwareID": "21",
  "lastIP": "172.30.91.43",
  "id": "27",
  "name": "Front Desk",
  "parentID": "-1",
  "movedToBin": false,
  "objectType": "tc",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v3/thinclients/27"
    },
    {
```



```
"rel": "Parent",
"href": "root"
},
{
  "rel": "Firmware",
  "href": "https://172.30.91.227:8443/umsapi/v3/firmwares/21"
}
],
{
  "unitID": "DC9C5207694E",
  "mac": "DC9C5207694E",
  "firmwareID": "13",
  "lastIP": "172.30.91.24",
  "id": "6888",
  "name": "UD3 M340C_Board",
  "parentID": "15592",
  "movedToBin": false,
  "objectType": "tc",
  "links": [...]
},
{
  "unitID": "00E0C5080834",
  "mac": "00E0C5080834",
  "firmwareID": "2",
  "lastIP": "172.30.91.132",
  "id": "6899",
  "name": "UD10",
  "parentID": "15592",
  "movedToBin": true,
  "objectType": "tc",
  "links": [...]
}
]
```



## Getting Information on a Thin Client

- Send a GET request to `/v3/thinclients/[id]` to get information about a thin client:

```
curl \
--request GET\
--header 'Cookie: JSESSIONID=3FB2F3F6A089FE9029DFD6DAFEF146DC' \
https://[server]:8443/umsapi/v3/thinclients/27
```

The thin client ID is passed to the API as a *Path Variable* in the URL.

The response contains the most important properties of the thin client with the ID 27:

### Response

```
{
  "unitID": "00E0C54DCB8E",
  "mac": "00E0C54DCB8E",
  "firmwareID": "21",
  "lastIP": "172.30.91.43",
  "id": "27",
  "name": "Front Desk",
  "parentID": "-1",
  "movedToBin": false,
  "objectType": "tc",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v3/thinclients/27"
    },
    {
      "rel": "Parent",
      "href": "root"
    },
    {
      "rel": "Firmware",
      "href": "https://172.30.91.227:8443/umsapi/v3/firmwares/21"
```





}

]

}



## Getting All Details about a Thin Client

- Use the `details` facet in order to get all details about a thin client:

```
curl \
--request GET\
--header 'Cookie: JSESSIONID=3FB2F3F6A089FE9029DFD6DAFEF146DC' \
https://[server]:8443/umsapi/v3/thinclients/27?facets=details
```

The response contains all the properties of the thin client with the ID 27:

### Response

```
{
  "unitID": "00E0C54DCB8E",
  "mac": "00E0C54DCB8E",
  "firmwareID": "21",
  "networkName": "D.Weinert",
  "site": "1. Stock",
  "department": "Product Management",
  "lastIP": "172.30.91.43",
  "costCenter": "",
  "comment": "",
  "assetID": "",
  "inServiceDate": "01.06.215",
  "serialNumber": "",
  "productId": "UD6-LX 51cps",
  "umsStructuralTag": "",
  "cpuSpeed": 2416,
  "cpuType": "Intel(R) Celeron(R) CPU J1900 @ 1.99GHz",
  "deviceType": "IGEL H830C",
  "deviceSerialNumber": "14D3D3C03B14470B9EM",
  "osType": "IGEL Linux V5 (Kernel Version 3.13.11-ckt20)",
  "flashSize": 1883,
  "memorySize": 1853,
  "networkSpeed": 1000,
```



```
"graphicsChipset0": "INTEL HD Graphics (Baytrail)",
"graphicsChipset1": "",
"monitorVendor1": "Samsung Electric Company",
"monitorModel1": "S24C650",
"monitorSerialnumber1": "H4MG404381",
"monitorSize1": 24,
"monitorNativeResolution1": "1920 x 1200",
"monitor1YearOfManufacture": "2015",
"monitor1WeekOfManufacture": "17",
"monitorVendor2": "Samsung Electric Company",
"monitorModel2": "S24C650",
"monitorSerialnumber2": "H4MG404389",
"monitorSize2": 24,
"monitorNativeResolution2": "1920 x 1200",
"monitor2YearOfManufacture": "2015",
"monitor2WeekOfManufacture": "17",
"biosVendor": "INSYDE Corp.",
"biosVersion": "H830C V:3.5.13-11282014",
"biosDate": "11/28/2014",
"totalUsagetime": "5798368000",
"totalUptime": "18513000",
"lastBoottime": "2015-09-29 08:30",
"id": "27",
"name": "Front Desk",
"parentID": "-1",
"movedToBin": false,
"objectType": "tc",
"links": [
{
"rel": "self",
"href": "https://172.30.91.227:8443/umsapi/v3/thinclients/27"
},
{
"rel": "Parent",
```



```
"href": "root"
```

```
},
```

```
{
```

```
"rel": "Firmware",
```

```
"href": "https://172.30.91.227:8443/umsapi/v3/firmwares/21"
```

```
}
```

```
]
```

```
}
```



## Getting Thin Client Status

- Use the `online` facet in order to get the online status of a thin client with a specific ID:

```
curl \
--request GET\
--header 'Cookie: JSESSIONID=3FB2F3F6A089FE9029DFD6DAFEF146DC' \
https://[server]:8443/umsapi/v3/thinclients/27?facets=online
```

The response contains, among others, the `online` property, which can have the values `true` and `false`:

### Response

```
{
  "unitID": "00E0C54DCB8E",
  "mac": "00E0C54DCB8E",
  "firmwareID": "21",
  "lastIP": "172.30.91.43",
  "online": false,
  "id": "27",
  "name": "Front Desk",
  "parentID": "-1",
  "movedToBin": false,
  "objectType": "tc",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v3/thinclients/27"
    },
    {
      "rel": "Parent",
      "href": "root"
    },
    {
      "rel": "Firmware",
      "href": "https://172.30.91.227:8443/umsapi/v3/firmwares/21"
```



}

]

}



## Creating, Updating and Deleting Resources

The thin client directories in UMS are a good resource for trying out the creation, update and deletion API calls.



## Listing all Thin Client Directories

- List all thin client directories:

```
curl \
--request GET\
--header 'Cookie: JSESSIONID=3FB2F3F6A089FE9029DFD6DAFEF146DC' \
https://[server]:8443/umsapi/v3/directories/tcdirectories/
```

## Response

The response contains a list of thin client directories in *JSON* format. The ID of the root thin client directory is always -1. Find a detailed description of the **Thin Client Directory** resource in the [IMI API Reference](#) (see page 87).

- Compare the *JSON* response to the thin client directory tree you see in your *UMS console*.



IMI provides a flat view of these directories, subdirectories can only be recognized via their `parentID`. In order to better see the nesting, use the `children` facet .

```
[
{
  "id": "15592",
  "name": "Pool",
  "parentID": "-1",
  "movedToBin": false,
  "objectType": "tcdirectory",
  "links": [
    {
      "rel": "self",
```





```
"href": "https://172.30.91.227:8443/umsapi/v3/directories/tcdirectories/15592"
},
{
  "rel": "Parent",
  "href": "root"
}
],
},
{
  "id": "76863",
  "name": "New Subdirectory",
  "parentID": "76462",
  "movedToBin": false,
  "objectType": "tcdirectory",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v3/directories/tcdirectories/76863"
    },
    {
      "rel": "Parent",
      "href": "https://172.30.91.227:8443/umsapi/v3/directories/tcdirectories/76462"
    }
  ],
},
{
  "id": "76462",
  "name": "New Directory",
  "parentID": "-1",
  "movedToBin": false,
  "objectType": "tcdirectory",
  "links": [
    {
      "rel": "self",
```



```
"href": "https://172.30.91.227:8443/umsapi/v3/directories/tcdirectories/76462"
},
{
  "rel": "Parent",
  "href": "root"
}
]
}
```



## Creating a Thin Client Directory

► Send an HTTP PUT request to the resource `/v3/directories/tcdirectories` to create a new thin client directory.

This request must contain *JSON* data in its body specifying a name for the new directory. The `parentID` is optional and defaults to the root thin client directory:

```
{  
  "name": "My Directory",  
  "parentID": "-1"  
}
```

*cURL* accepts *JSON* data as the `--data` parameter. This request also specifies `Content-type` header to tell *IMI* that the data has the type `application/json`.

► Use it whenever you send *JSON* data:

```
curl \  
--request PUT \  
--header 'Cookie: JSESSIONID=3FB2F3F6A089FE9029DFD6DAFEF146DC' \  
--header "Content-type: application/json" \  
--data '{"name": "My Directory", "parentID": "-1"}' \  
https://[server]:8443/umsapi/v3/directories/tcdirectories
```

## Response

*IMI* replies with a success message and data about the newly created directory:

```
{  
  "message": "Directory successfully inserted.",  
  "id": "77118",  
  "name": "My Directory",  
  "parentID": "-1"  
}
```



## Updating a Thin Client Directory

► Send an HTTP PUT request to `/v3/directories/tcdirectories/[id]` to update the properties of a thin client directory. The directory ID is passed as a variable in the URL, a new directory name as *JSON* data in the request body. Do not forget the `Content-type` header.

```
curl \
--request PUT \
--header 'Cookie: JSESSIONID=3FB2F3F6A089FE9029DFD6DAFEF146DC' \
--header "Content-type: application/json" \
--data '{"name":"My Wonderful Directory"}' \
https://[server]:8443/umsapi/v3/directories/tcdirectories/77118
```

## Response

► IMI replies with a success message:

```
200 OK
---
{
  "message": "Updated directory successfully."
}
```



## Deleting a Thin Client Directory

► Send an HTTP DELETE request to `/v3/directories/tcdirectories/[id]` to delete a thin client directory. The ID of the directory is passed as a path variable in the URL.

```
curl \
--request DELETE \
--header 'Cookie: JSESSIONID=3FB2F3F6A089FE9029DFD6DAFEF146DC' \
https://[server]:8443/umsapi/v3/directories/tcdirectories/77118
```

## Response

► IMI replies with a brief success message:

Code Example:

```
<code> 200 OK</code>
```

Code Example:

```
<code>---</code>
```

Code Example:

```
<code> {"message":"Deletion successful."} </code>
```

A thin client directory has to be empty to be removed. If you try to delete a non-empty directory, IMI will respond with an error message:

```
400 Bad Request
---
{
  "message": "Directory with ID <77118> is not empty. Remove content first.",
  "errorCode": "ITG-0A006",
  "time": "2016-04-06T13:29:59.362",
  "stackTrace": "[de.igel.rm.umsapi.server.services.thinclientdirectory.
[...]]
}
```

Learn more about error codes and messages in the [IMI API Reference](#)(see page 87).



## Further Operations

- [Moving a Thin Client Directory](#)(see page 31)
- [Sending a Command to Thin Clients](#)(see page 32)
- [Debugging Requests](#)(see page 34)



## Moving a Thin Client Directory

► In order to move thin clients or directories send a PUT request to the target directory and append `?operation=move`. The request body must contain a list of API Objects in *JSON* format, representing the thin clients and directories to be moved. An API Object has an ID and a type.

```
curl \
--request PUT \
--header 'Cookie: JSESSIONID=3FB2F3F6A089FE9029DFD6DAFEF146DC' \
--header "Content-type: application/json" \
--data '[ { "id":"77123", "type":"tcdirectory"},\
{ "id":"1234", "type":"tcdirectory" } ]' \
https://[server]:8443/umsapi/v3/directories/tcdirectories/15592?operation=move
```

## Response

The following response contains an error message for one directory and a success message for the other:

```
200 OK
---
[
{
  "id": "1234",
  "results": "does_not_exist"
},
{
  "id": "77123",
  "results": "successful"
}
]
```



## Sending a Command to Thin Clients

► Send a POST request to `/v3/thinclients?command=[command]` to send a command to one or more thin clients. A command can be one of the following:

- `reboot`
- `shutdown`
- `wakeup`
- `settings2tc`
- `tcreset2facdefs`

The request body contains a list of API Objects representing the thin clients addressed.

To send the `reboot` command to two thin clients, send the following request to IMI:

```
curl \
--request POST \
JSESSIONID=3FB2F3F6A089FE9029DFD6DAFEF146DC' \
--header "Content-type: application/json" \
--data ' [{"id": "27", "type": "tc"}, {"id": "72014", "type": "tc"} ]' \
https://[server]:8443/umsapi/v3/thinclients?command=reboot
```

## Request

IMI replies with a JSON document containing a result for each thin client addressed:

```
{
  "CommandExecList": [
    {
      "execID": "ID-PM-MH-WIN7-UMS-54530-1456839861871-5-0",
      "id": "72014",
      "mac": "00E0C561EEED",
      "exectime": "1456845240566",
      "message": "OK",
      "state": "SUCCESS"
    },
    {
      "execID": "ID-PM-MH-WIN7-UMS-54530-1456839861871-5-0",
      "id": "27",
      "mac": "00E0C54DCB8E",
```





```
"exectime": "1456845240560",
```

```
"message": "OK",
```

```
"state": "SUCCESS"
```

```
}
```

```
]
```

```
}
```



## Debugging Requests

### Error Codes

To debug errors when communicating with IMI, observe

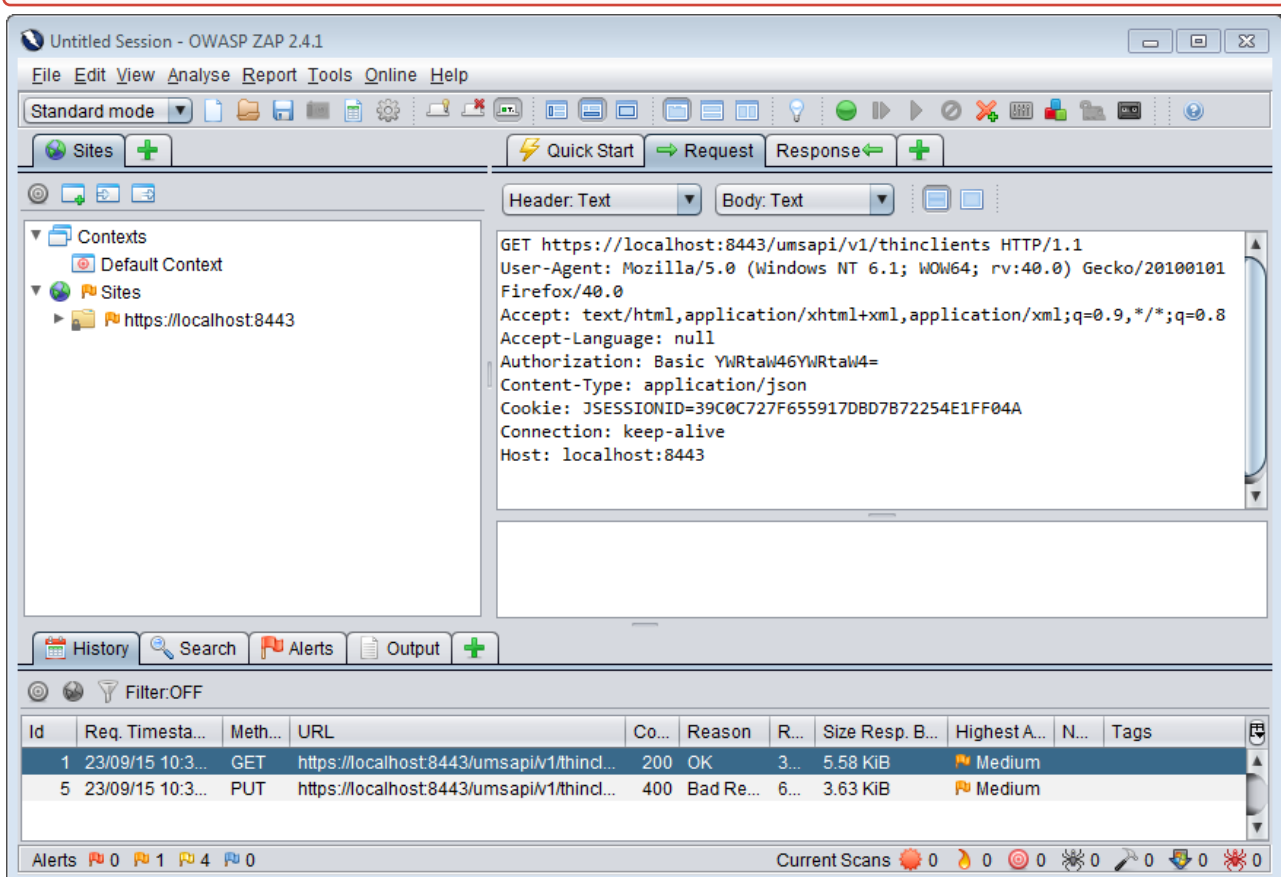
- the HTTP status in the response headers
- the messages and error codes in the response bodies.

### HTTP-Proxy

You may use an HTTP proxy in order to see what requests your scripts or programs produce. A proxy acts as a man-in-the-middle between client and IMI, letting you view and optionally edit requests.

An Open Source program for this task is [OWASP ZAP](https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project)<sup>3</sup>.

⚠ Use a proxy only for debugging your own HTTP requests. Viewing other users' network traffic is usually illegal.



<sup>3</sup> [https://www.owasp.org/index.php/OWASP\\_Zed\\_Attack\\_Proxy\\_Project](https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project)



## Powershell

- [Introduction](#)(see page 36)
- [Command Reference](#)(see page 40)



## Introduction

*IMI Cmdlets* is a collection of *Windows PowerShell* commandlets for use with the *IGEL Management Interface (IMI)*. The commandlets are installed as binary `*.dll` files and can be used in *PowerShell* scripts. They simplify frequent actions such as rebooting a thin client or assigning a profile via *IMI*, hiding the implementation details of the underlying REST API.

 Microsoft .NET version 4.5 or greater is required for IMI Cmdlets 1.04.100 for TLSv1.2

- 
- [Installation](#)(see page 37)
  - [Loading the IMI Cmdlets Snap-in](#)(see page 38)
  - [Authentication](#)(see page 39)



## Installation

### Prerequisites

- One of the following *Microsoft* operating systems:
  - *Windows 7*
  - *Windows Server 2008 or 2008 R2*
  - *Windows 8 or higher*
  - *Windows Server 2012 or 2012 R2*
- *Windows Management Framework 4* (which already includes *PowerShell 4*)
- *Microsoft .Net Framework 4* or newer

**i** IGEL Management Interface (IMI) PowerShell Cmdlets 1.03.100 or lower do not support UMS 5.08.100 or higher.  
UMS 5.08.100 or higher uses TLS 1.2, which is supported as of PowerShell Cmdlets 1.04.100 only.

### Installing

1. Download the *IMI* Commandlets Installer from the [download server](https://www.igel.com/software-downloads/)<sup>4</sup>.
2. Launch the installer.

**i** You need administration privileges in order to install IMI Commandlets.

3. Close all other applications and confirm that you have done so.
4. Review and accept the **License Agreement**.
5. Select the destination location for the installation or leave the default.
6. Optional: Select a Start Menu folder for the program shortcuts.
7. Read the summary and start the installation process.  
The installer will install the IMI Commandlets and some sample scripts. The following shortcuts will be placed on the desktop:
  - **Starter-Interface:** A simple interactive terminal interface to *IMI*, suitable for new users
  - **IGEL-Shell:** A *Windows* PowerShell session with *IMI* Commandlets already loaded

Additionally, the *IMI* Commandlets Readme file is displayed.

<sup>4</sup> <https://www.igel.com/software-downloads/>



## Loading the IMI Cmdlets Snap-in

Before you can use IMI Cmdlets, you need to add the snap-in to your PowerShell session:

```
Add-PSSnapin *igel*
```

View all loaded IGEL Cmdlets:

```
Get-PSSnapin *igel*
```



## Authentication

IMI uses HTTP Basic Authentication with a username-password combination. IMI Cmdlets wrap authentication in the **Get-IgelRMServerLogin** cmdlet.

**Get-IgelRMServerLogin** logs in to IMI.

### Parameters

- **-Servername**  
Hostname of the of the UMS server with IMI
- **-Username**  
UMS user name
- **-ForceIMIVersion**  
Use a specific IMI version, `v1` or `v2`.
- **-IgnoreUntrustedCertificates**  
If this is set to `$true` the Cmdlets will ignore errors in TLS/SSL certificate validation.

The password is entered interactively.

### Example Command Line

Log in and save the credentials for the session in `$l` :

```
$l = Get-IgelRMServerLogin `
  -Servername localhost `
  -Username SPIELWIESE `
  -IgnoreUntrustedCertificates $true `
  -ForceIMIVersion v2
```

Supply the credentials to subsequent commands using

```
-Credentials $l
```



## Command Reference

- [Thin Client](#)(see page 41)
- [Profile](#)(see page 49)
- [Master Profile](#)(see page 56)
- [Thin Client Directory](#)(see page 63)
- [Profile Directory](#)(see page 71)
- [Master Profile Directory](#)(see page 77)
- [Firmware](#)(see page 83)
- [Server Status](#)(see page 85)





## Thin Client

- [Get-IgelTCInformation](#)(see page 42)
- [Start-IgelTCs](#)(see page 44)
- [Stop-IgelTC](#)(see page 45)
- [Restart-IgelTCs](#)(see page 46)
- [Move-IgelTCs](#)(see page 47)
- [Update-IgelTCSettings](#)(see page 48)



## Get-IgelTCInformation

### Summary

Gets information on thin clients.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login.
- **-tcID**  
Numeric ID of the thin client. If omitted, information is retrieved on all thin clients, including those in the Recycling Bin.
- **-Details**  
Level of detail in the output, one of:
  - **full** Gets all data available from IMI, including the online status.
  - **inventory** Gets all data available from IMI, apart from the online status.
  - **online** Gets data in a short format, including the online status.

If this parameter is omitted, only a short information format with the most important thin client properties is returned.

### Example Command Line

```
Get-IgelTCInformation -Credentials $l
```

### Example Output

```
mac : 000BCA050027
firmwareID : 45
lastIP : 172.30.91.237
id : 48335
name : UD2-D220
parentID : 76863
movedToBin : True
objectType : tc

mac : 005056934FDB
firmwareID : 28
lastIP : 172.30.91.9
id : 48366
name : westestx86
```



```
parentID : -1  
movedToBin : False  
objectType : tc  
  
mac : 00E0C561EEED  
firmwareID : 43  
lastIP : 172.30.91.30  
id : 72014  
name : IGEL-8KJ2GQPL3N  
parentID : 76863  
movedToBin : False  
objectType : tc
```



## Start-IgelTCs

### Summary

Wakes up thin clients.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-tcID / -tcIDs**  
Numeric IDs of the thin clients to wake up

### Example Command Line

```
Start-IgelTCs -Credentials $l -tcIDs 27
```

### Example Output

```
execID exectime message state
```

```
-----
```

```
ID-PM-MH-WIN7-UMS-49242-14... 1460647127627 OK SUCCESS
```



## Stop-IgelTC

### Summary

Shuts down thin clients.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-tcID / -tcIDs**  
Numeric IDs of the thin clients to shut down

### Example Command Line

```
Stop-IgelTCs -Credentials $l -tcID 2010
```

### Example Output

```
execID exectime message state
```

```
-----
```

```
ID-mhuber-50400-14605614057... 1460706641670 OK SUCCESS
```



## Restart-IgelTCs

### Summary

Reboots thin clients.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-tcIDs**  
Numeric IDs of the thin clients to reboot

### Example Command Line

```
Restart-IgelTCs -Credentials $l -tcIDs 2010,1435
```

### Example Output

```
execID exectime message state
-----
ID-mhuber-50400-14605614057... 1460991770313 OK SUCCESS
ID-mhuber-50400-14605614057... 1460991770313 OK SUCCESS
```



## Move-IgelTCs

### Summary

Moves a thin client to a thin client directory.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-tcIDs**  
Numeric ID of the thin client
- **-DirectoryID**  
Numeric ID of the target thin client directory

### Example Command Line

```
Move-IgelTCs `
-Credentials $l `
-TCIDs 2010 `
-DirectoryID 3147
```

### Example Output

```
id results
--
2010 successful
```



## Update-IgelTCSettings

### Summary

Sends settings from UMS to the thin client.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-tcID**  
Numeric ID of the thin client

### Example Command Line

```
Update-IgelTCSettings -Credentials $l -tcID 2010
```

### Example Output

```
execID : ID-mhuber-50400-1460561405782-16-0  
id : 2010  
mac : 000BCA050027  
exectime : 1460708102541  
message : OK  
state : SUCCESS
```





## Profile

- [Get-IgelTCProfile](#)(see page 50)
- [Rename-IgelTCProfile](#)(see page 51)
- [Get-IgelProfileAssignments](#)(see page 52)
- [Set-IgelProfileAssignment](#)(see page 53)
- [Remove-IgelProfileAssignment](#)(see page 54)
- [Remove-IgelTCProfile](#)(see page 55)



## Get-IgelTCProfile

### Summary

Gets information about profiles.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-ProfileID**  
Numeric ID of the profile. If this is omitted, information is retrieved for all profiles

### Example Command Line

```
Get-IgelTCProfile -Credentials $l -ProfileID 3150
```

### Example Output

```
firmwareID : 3
isMasterProfile : False
overridesSessions : False
id : 3150
name : My Profile
parentID : -2
movedToBin : False
objectType : profile
```



## Rename-IgelTCProfile

### Summary

Renames a profile.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-ProfileID**  
Numeric ID of the profile
- **-newProfileName**  
New profile name as string

### Example Command Line

```
Rename-IgelTCProfile `
-Credentials $l `
-ProfileID 3150 `
-newProfileName "Updated Profile"
```

### Example Output

```
message
-----
Update successful
```



## Get-IgelProfileAssignments

### Summary

Gets profile assignments.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-ProfileID**  
Numeric ID of the profile
- **-ObjectType**  
Get assignments to a specific object type, one of:
  - **tc** thin client
  - **directory** thin client directory

### Example Command Line

```
Get-IgelProfileAssignments `
-Credentials $l `
-ProfileID 3150 `
-ObjectType tc
```

### Example Output

```
type id assignmentPosition
---- --
tc 2010 0
```



## Set-IgelProfileAssignment

### Summary

Creates a profile or master profile assignment.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-ProfileID**  
Numeric ID of the profile or master profile
- **-ProfileType**  
Profile type, one of:
  - **profile** profile
  - **masterprofile** master profile
- **-TargetID**  
Numeric ID of the target object which the profile or master profile will be assigned to
- **-TargetType**  
Assignments to a specific object type, one of:
  - **tc** thin client
  - **tcdirectory** thin client directory

### Example Command Line

```
Set-IgelProfileAssignment `
-Credentials $l `
-ProfileID 3150 `
-ProfileType profile `
-TargetID 2010 `
-TargetType tc
```

### Example Output

```
message
-----
1 assignments successfully assigned
```



## Remove-IgelProfileAssignment

### Summary

Deletes a profile or master profile assignment.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-ProfileID**  
Numeric ID of the profile or master profile
- **-ProfileType**  
Profile type, one of:
  - **profile** profile
  - **masterprofile** master profile
- **-TargetID**  
Numeric ID of the target object which the profile or master profile is assigned to
- **-TargetType**  
Assignments to a specific object type, one of:
  - **tc** thin client
  - **tcdirectory** thin client directory

### Example Command Line

```
Remove-IgelProfileAssignment `
-Credentials $l `
-ProfileID 3151 `
-ProfileType profile `
-TargetType tcdirectory `
-TargetID 3201
```

### Example Output


```
message
-----
deleted profile assignment
```



## Remove-IgelTCProfile

### Summary

Deletes a profile.

 This does not move the profile to the Recycling Bin, but simply deletes it.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-ProfileID**  
Numeric ID of the profile

### Example Command Line

```
Remove-IgelProfile -Credentials $l -ProfileID 3150
```

### Example Output

```
message
```

```
-----
```

```
Deleted profile with id 3150
```



## Master Profile

- [Get-IgelTCMasterprofile](#)(see page 57)
- [Rename-IgelTCMasterprofile](#)(see page 58)
- [Get-IgelMasterprofileAssignments](#)(see page 59)
- [Set-IgelProfileAssignment](#)(see page 60)
- [Remove-IgelProfileAssignment](#)(see page 61)
- [Remove-IgelMasterprofile](#)(see page 62)





## Get-IgelTCMasterprofile

### Summary

Gets information about master profiles.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-MasterprofileID**  
Numeric ID of the master profile. If this is omitted, information is retrieved for all master profiles

### Example Command Line

```
Get-IgelTCMasterprofile -Credentials $l -MasterprofileID 3152
```

### Example Output

```
firmwareID : 3
isMasterProfile : True
overridesSessions : False
id : 3152
name : My Master Profile
parentID : -14
movedToBin : False
objectType : masterprofile
```



## Rename-IgelTCMasterprofile

### Summary

Renames a master profile.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-MasterprofileID**  
Numeric ID of the master profile
- **-newMasterProfileName**  
New master profile name as string

### Example Command Line

```
Rename-IgelTCMasterprofile `
-Credentials $l `
-MasterprofileID 3152 `
-newMasterProfileName "Renamed Master Profile"
```

### Example Output

```
message
-----
Update successful
```



## Get-IgelMasterprofileAssignments

### Summary

Gets master profile assignments.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-MasterprofileID**  
Numeric ID of the master profile
- **-ObjectType**  
Get assignments to a specific object type, one of:
  - **tc** thin client
  - **tcdirectory** thin client directory

### Example Command Line

```
Get-IgelMasterprofileAssignments `
-Credentials $l `
-MasterprofileID 3152 `
-ObjectType tc
```

### Example Output

```
type id assignmentPosition
---- --
tc 2010 0
```



## Set-IgelProfileAssignment

### Summary

Creates a profile or master profile assignment.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-ProfileID**  
Numeric ID of the profile or master profile
- **-ProfileType**  
Profile type, one of:
  - **profile** profile
  - **masterprofile** master profile
- **-TargetID**  
Numeric ID of the target object which the profile or master profile will be assigned to
- **-TargetType**  
Assignments to a specific object type, one of:
  - **tc** thin client
  - **tcdirectory** thin client directory

### Example Command Line

```
Set-IgelProfileAssignment `
-Credentials $l `
-ProfileID 3150 `
-ProfileType profile `
-TargetID 2010 `
-TargetType tc
```

### Example Output

```
message
-----
1 assignments successfully assigned
```



## Remove-IgelProfileAssignment

### Summary

Deletes a profile or master profile assignment.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-ProfileID**  
Numeric ID of the profile or master profile
- **-ProfileType**  
Profile type, one of:
  - **profile** profile
  - **masterprofile** master profile
- **-TargetID**  
Numeric ID of the target object which the profile or master profile is assigned to
- **-TargetType**  
Assignments to a specific object type, one of:
  - **tc** thin client
  - **tcdirectory** thin client directory

### Example Command Line

```
Remove-IgelProfileAssignment `
-Credentials $l `
-ProfileID 3151 `
-ProfileType profile `
-TargetType tcdirectory `
-TargetID 3201
```

### Example Output


```
message
-----
deleted profile assignment
```



## Remove-IgelMasterprofile

### Summary

Deletes a master profile.

 This does not move the master profile to the Recycling Bin, but simply deletes it.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-MasterprofileID**  
Numeric ID of the master profile

### Example Command Line

```
Remove-IgelMasterprofile -Credentials $l -MasterprofileID 3152
```

### Example Output

```
message
```

```
-----
```

```
Deleted profile with id 3152
```



## Thin Client Directory

- [Get-IgelTCDirectories](#)(see page 64)
- [Rename-IgelTCDirectory](#)(see page 66)
- [Set-IgelTCDirectory](#)(see page 67)
- [Move-IgelTCDirectory](#)(see page 68)
- [Set-IgelProfileAssignment](#)(see page 69)
- [Remove-IgelTCDirectory](#)(see page 70)



## Get-IgelTCDirectories

### Summary

Gets information on thin client directories.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-DirectoryID**  
Numeric ID of the thin client directory. If omitted, information is retrieved on all thin client directories, including those in the Recycling Bin
- **-details**  
Amount of detail in the output:
  - **\$true** The IDs of `DirectoryChildren` will be shown, which can be thin clients or thin client directories.

### Example Command Line

```
Get-IgelTCDirectories -Credentials $l -details $true
```

### Example Output

```
id : 3201
name : B Directory
parentID : -1
movedToBin : False
objectType : tcdirectory
DirectoryChildren : {3202, 1435}
```

```
id : 3202
name : Subdirectory
parentID : 3201
movedToBin : False
objectType : tcdirectory
DirectoryChildren : {}
```

```
id : 3147
name : A Directory
```





```
parentID : -1  
movedToBin : False  
objectType : tcdirectory  
DirectoryChildren : {2010}
```



## Rename-IgelTCDirectory

### Summary

Renames a thin client directory.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-directoryID**  
Numeric ID of the thin client directory
- **-newDirectoryName**  
New thin client directory name as string

### Example Command Line

```
Rename-IgelTCDirectory `
-Credentials $l `
-directoryID 3206 `
-newDirectoryName "Renamed TC Directory"
```

### Example Output

```
message
-----
Updated directory successfully.
```



## Set-IgelTCDirectory

### Summary

Creates a new thin client directory.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-directoryName**  
Name for the thin client directory
- **-directoryPosition**  
Parent directory for the new thin client directory. If omitted, it defaults to `-1`, which is the UMS directory **Thin Clients**.

### Example Command Line

```
Set-IgelTCDirectory `
-Credentials $l `
-directoryName "Brand New Directory" `
-directoryPosition 3201
```

### Example Output

```
message id name parentID
----- -- ----
Directory successfully inse... 3206 Brand New Directory 3201
```



## Move-IgelTCDirectory

### Summary

Moves one or more thin client directories.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-directoryIDs**  
Numerical IDs of the thin client directories to be moved
- **-targetDirectory**  
The target directory to move the thin client directories into

### Example Command Line

```
Move-IgelTCDirectory `
-Credentials $l `
-directoryIDs 3206 `
-targetDirectory
```

### Example Output

```
id results
--
3206 successful
```



## Set-IgelProfileAssignment

### Summary

Creates a profile or master profile assignment.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-ProfileID**  
Numeric ID of the profile or master profile
- **-ProfileType**  
Profile type, one of:
  - **profile** profile
  - **masterprofile** master profile
- **-TargetID**  
Numeric ID of the target object which the profile or master profile will be assigned to
- **-TargetType**  
Assignments to a specific object type, one of:
  - **tc** thin client
  - **tcdirectory** thin client directory

### Example Command Line

```
Set-IgelProfileAssignment `
-Credentials $l `
-ProfileID 3150 `
-ProfileType profile `
-TargetID 2010 `
-TargetType tc
```

### Example Output


```
message
-----
1 assignments successfully assigned
```



## Remove-IgelTCDirectory

### Summary

Deletes a thin client directory.

 You can only delete empty thin client directories.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-directoryID**  
Numerical ID of the thin client directory to be deleted

### Example Command Line

```
Remove-IgelTCDirectory -Credentials $l -directoryID 3202
```

### Example Output

```
message
```

```
-----
```

```
Deletion successful.
```



## Profile Directory

- [Get-IgelProfileDirectory](#)(see page 72)
- [Rename-IgelTCProfileDirectory](#)(see page 73)
- [Move-IgelTCProfileDirectory](#)(see page 74)
- [Set-IgelProfileDirectory](#)(see page 75)
- [Remove-IgelProfileDirectory](#)(see page 76)



## Get-IgelProfileDirectory

### Summary

Gets information on profile directories.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-DirectoryID**  
Numeric ID of the profile directory. If omitted, information is retrieved on all profile directories, including those in the Recycling Bin.

### Example Command Line

```
Get-IgelProfileDirectory -Credentials $l -ProfileDirectoryID 3221
```

### Example Output

```
id : 3221
name : Profiles 2
parentID : -2
movedToBin : False
objectType : profiledirectory
```





## Rename-IgelTCProfileDirectory

### Summary

Renames a profile directory.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-directoryID**  
Numeric ID of the profile directory
- **-newDirectoryName**  
New profile directory name as string

### Example Command Line

```
Rename-IGELTCProfileDirectory `
-Credentials $l `
-directoryID 3220 `
-newDirectoryName "My Renamed Directory"
```

### Example Output

```
message
-----
Updated directory successfully.
```



## Move-IgelTCProfileDirectory

### Summary

Moves one or more profile directories.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-directoryIDs**  
Numerical IDs of the profile directories to be moved
- **-targetDirectory**  
The target directory to move the profiles directories into

### Example Command Line

```
Move-IgelProfileDirectory `
-Credentials $l `
-directoryIDs 4024 `
-targetDirectory 3220
```

### Example Output

```
id results
--
4024 successful
```



## Set-IgelProfileDirectory

### Summary

Creates a new profile directory.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-directoryName**  
Name for the profile directory
- **-directoryPosition**  
Parent directory for the new profile directory. If omitted, it defaults to `-2`, which is the UMS directory **Profiles**.

### Example Command Line

```
Set-IgelProfileDirectory `
-Credentials $l `
-directoryName "A New Directory" `
-directoryPosition -2
```

### Example Output


```
message id name parentID
----- -- ----
Directory successfully inse... 4030 A New Directory -2
```



## Remove-IgelProfileDirectory

### Summary

Deletes a profile directory.

 You can only delete empty thin client directories.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-directoryID**  
Numerical ID of the profile directory to be deleted

### Example Command Line

```
Remove-IgelProfileDirectory -Credentials $l -ProfileDirectoryID 4024
```

### Example Output

```
message
```

```
-----
```

```
Deletion successful.
```



## Master Profile Directory

- [Get-IgelMasterprofileDirectory](#)(see page 78)
- [Rename-IgelTCMasterprofileDirectory](#)(see page 79)
- [Move-IgelMasterprofileDirectory](#)(see page 80)
- [Set-IgelMasterprofileDirectory](#)(see page 81)
- [Remove-IgelMasterprofileDirectory](#)(see page 82)



## Get-IgelMasterprofileDirectory

### Summary

Gets information on master profile directories.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-DirectoryID**  
Numeric ID of the master profile directory. If omitted, information is retrieved on all master profile directories, including those in the Recycling Bin

### Example Command Line

```
Get-IgelMasterprofileDirectory -Credentials $l -MasterprofileDirectoryID 4034
```

### Example Output

```
id : 4034
name : New Master Profiles
parentID : -14
movedToBin : False
objectType : masterprofiledirectory
```



## Rename-IgelTCMasterprofileDirectory

### Summary

Renames a master profile directory.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-directoryID**  
Numeric ID of the master profile directory
- **-newDirectoryName**  
New master profile directory name as string

### Example Command Line

```
Rename-IGELTCMasterProfileDirectory `
-Credentials $l `
-directoryID 4038 `
-newDirectoryName "Renamed Master Profiles"
```

### Example Output

```
message
-----
Updated directory successfully.
```



## Move-IgelMasterprofileDirectory

### Summary

Moves one or more master profile directories.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-directoryIDs**  
Numerical IDs of the master profile directories to be moved
- **-targetDirectory**  
The target directory to move the master profiles directories into

### Example Command Line

```
Move-IgelMasterprofileDirectory `
-Credentials $l `
-directoryIDs 4033 `
-targetDirectory 4034
```

### Example Output

```
id results
--
4033 successful
```





## Set-IgelMasterprofileDirectory

### Summary

Creates a new master profile directory.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-directoryName**  
Name for the master profile directory
- **-directoryPosition**  
Parent directory for the new master profile directory. If omitted, it defaults to `-14`, which is the UMS directory **Master Profiles**.

### Example Command Line

```
Set-IgelMasterprofileDirectory -Credentials $l -directoryName "Brand New Directory"
```

### Example Output

```
message id name parentID
----- -- ----
Directory successfully inse... 4038 Brand New Directory -14
```



## Remove-IgelMasterprofileDirectory

### Summary

Deletes a master profile directory.

 You can only delete empty directories.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-directoryID**  
Numerical ID of the master profile directory to be deleted

### Example Command Line

```
Remove-IgelMasterprofileDirectory -Credentials $l -MasterprofileDirectoryID  
403 3
```

### Example Output

```
message
```

```
-----
```

```
Deletion successful.
```

Powershell



## Firmware

- [Get-IgelFirmware](#)(see page 84)



## Get-IgelFirmware

### Summary

Gets information on the firmwares registered with UMS.

### Parameters

- **-Credentials**  
Credential data, usually passed via a variable which it has been saved to at login
- **-firmwareID**  
Numeric ID of the firmware. If omitted, information is retrieved on all firmwares.

### Example Command Line

```
Get-IgelFirmware -Credentials $l -firmwareID 3
```

### Example Output

```
id product version firmwareType
--
3 IGEL Universal Desktop LX 5.09.100.01 LX
```



## Server Status

- [Get-IgelRMServerStatus](#)(see page 86)



## Get-IgelRMServerStatus

### Summary

Gets information on the UMS Server and IMI. No authentication is necessary for this.

### Parameters

- **-Servername**  
Hostname of the UMS server with IMI
- **-IgnoreUntrustedCertificates**  
If this is set to `$true` the Cmdlets will ignore errors in TLS/SSL certificate validation.

### Example Command Line

```
Get-IgelRMServerStatus -Servername 172.30.91.158 -IgnoreUntrustedCertificates $true
```

### Example Output

```
buildNumber : 26712
rmGuiServerVersion : 5.02.100.rc8
activeMQVersion : 5.6.0
derbyVersion : 10.12.1.1
serverUUID : a400f9da-333d-49d8-bb8d-2238543a4643
server : mhuber:8443
```



## IMI API V2 Reference

This document describes the resources and methods made available by the *IGEL Management Interface API Version 2* in a reference format.

- 
- [What's New in IMI V2?](#)(see page 88)
  - [Prerequisites](#)(see page 89)
  - [Client Applications and Libraries](#)(see page 90)
  - [Authentication](#)(see page 92)
  - [Basic Data Types](#)(see page 95)
  - [Resources](#)(see page 105)
  - [Error Codes](#)(see page 240)



## What's New in IMI V2?

- [Profiles and Master Profiles](#)(see page 104) are available as resources.
- The following [Assignments](#)(see page 97) can be made:
  - [Profile -> Thin Client](#)(see page 130)
  - [Master Profile -> Thin Client](#)(see page 172)
  - [Profile -> Thin Client Directory](#)(see page 158)
  - [Master Profile -> Thin Client Directory](#)(see page 177)
- Facets expose different views of objects and lists:
  - [details](#)(see page 120): Show detailed information
  - [online](#)(see page 123): Perform an online check on thin clients
  - [command](#): Execute command ([reboot](#), [shutdown](#), [wakeup](#)(see page 138), [update settings](#)(see page 140)) on a thin client
  - [shadow](#)(see page 115): Get credentials for Secure VNC shadowing
  - [children](#)(see page 189): List directory children recursively
- [Secure VNC Support](#)(see page 115): IMIv2 can supply the client certificate and one-time-password needed for TLS-encrypted VNC shadowing to third-party software.
- New syntax for [moving thin clients and directories](#)(see page 195), [profiles and profile directories](#)(see page 216).





## Prerequisites

### IGEL Universal Management Suite (UMS)

UMS 5.02.100

### Networking

In order to use *IGEL Management Interface* you need to be able to reach the API host via the network and connect to its API port, TCP 8443 by default.

The base URL is

- `https://[server]:8443/umsapi/`

The resources for IMI version 2 are available at:

- `https://[server]:8443/umsapi/v2/`

❗ *IMI* uses HTTPS to ensure the integrity and confidentiality of the network traffic. It is good practice to use a valid server certificate with a verifiable signature. Most clients and libraries can be configured to work with self-signed or invalid certificates, but that should not be done in a production environment.

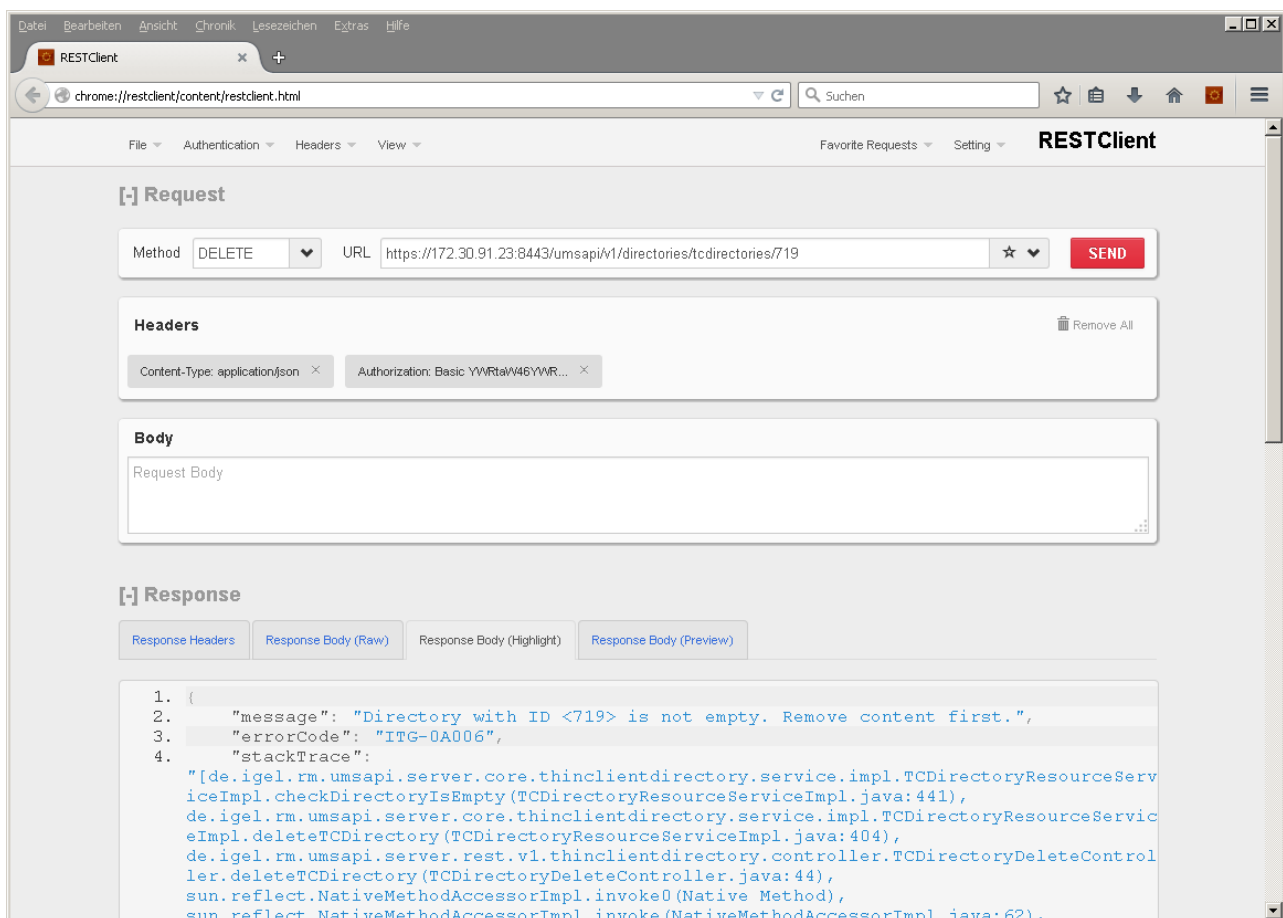


## Client Applications and Libraries

## Clients

The easiest way to try out the *IGEL Management Interface* is either

- with [RESTClient](#)<sup>5</sup>, an add-on for *Mozilla Firefox*:



or

- with **cURL**<sup>6</sup>, a commandline network client:

5 <http://restclient.net/>

6 <http://curl.haxx.se/>

A screenshot of a terminal window with a light blue title bar and standard window controls. The prompt is 'Administrator@PM-MH-WIN7-UMS ~'. The command '\$ curl -ik --request GET https://localhost:8443/umsapi/v1/serverstatus' has been executed. The output shows an HTTP 200 OK response with headers for Server, Set-Cookie, Content-Type, Transfer-Encoding, and Date. The body is a JSON object containing server version, build number, activeMQ version, derby version, server UUID, server address, and a self-link.

```
Administrator@PM-MH-WIN7-UMS ~  
$ curl -ik --request GET https://localhost:8443/umsapi/v1/serverstatus  
HTTP/1.1 200 OK  
Server: Apache-Coyote/1.1  
Set-Cookie: JSESSIONID=A645FEC36AA9C0373F0D869CCDFC8C0B; Path=/umsapi/; Secure; Http  
Only  
Content-Type: application/json  
Transfer-Encoding: chunked  
Date: Wed, 04 Mar 2015 12:44:19 GMT  
  
{  
  "rmGuiServerVersion": "4.09.100.rc1",  
  "buildNumber": "21741",  
  "activeMQVersion": "5.6.0",  
  "derbyVersion": "10.8.3.0",  
  "serverUUID": "dd188671-e485-4679-8d2d-485be8a6275c",  
  "server": "172.30.91.23:8443",  
  "links": [{"rel": "self", "href": "https://localhost:8443/umsapi/v1/serverstatus"}]  
}  
Administrator@PM-MH-WIN7-UMS ~  
$
```

Both are licensed as Open Source software and are available free of charge.

This guide uses *cURL* for examples, as the commandline makes all parameters visible in plain text. This need not stop you from using *RESTClient* if you are more comfortable with it. You can easily translate the commandline parameters into the fields of the *RESTClient* GUI.

## Programming

Most programming languages provide an HTTP and an SSL/TLS module, either in their standard library or as an extension, and a *JSON* library (for the API data format) as well.



## Authentication

You must be authenticated in order to use *IGEL Management Interface*, otherwise the server will return the HTTP status 401 "Unauthorized". Only querying the server status is allowed without authentication.

The login mechanism uses HTTP Basic Authentication (RFC 2617).

- 
- [POST /v2/login](#)(see page 93)
  - [POST /v2/logout](#)(see page 94)



## POST /v2/login

### Summary

Authenticates the client in regard to the IMI server.

### Resource URL

```
/v2/login
```

### Example Request

```
curl \
--request POST \
--user 'admin:W00t' \
https://[server]:8443/umsapi/v2/login
```

### Response Body

name	type	description
message	String	String containing the JSESSIONID

### Example Response

```
200 OK
Set-Cookie: JSESSIONID=3FB2F3F6A089FE9029DFD6DAFEF146DC; Path=/umsapi/; Secure; HttpOnly
---
{"message": "JSESSIONID=3FB2F3F6A089FE9029DFD6DAFEF146DC"}
```

**!** You can maintain the session by sending the JSESSIONID in the Cookie header with every subsequent request. Some clients will do this automatically for you, e.g. the *RESTClient Firefox* add-on.  
Cookie: JSESSIONID=3FB2F3F6A089FE9029DFD6DAFEF146DC



## POST /v2/logout

### Summary

Logs the client out from the *IMI* session.

### Resource URL

```
/v2/logout
```

### Example Request

```
curl \
--request POST \
--header 'Cookie: JSESSIONID=11048CDA77DE2B45BE1562C8EED67858' \
https://[server]:8443/umsapi/v2/logout
```

### Example Response

```
200 OK
```

---

```
{"message": "Logout successful"}
```



## Basic Data Types

- [ApiObject](#)(see page 96)
- [Assignment](#)(see page 97)
- [MainDataResource](#)(see page 99)
- [TCResourceV2](#)(see page 100)
- [DirectoryResourceV2](#)(see page 103)
- [ProfileResource](#)(see page 104)



## ApiObject

### Summary

This object is the most basic representation of thin clients, profiles and directories. It is used in [Assignments](#)(see page 97).

### Properties

Name	Type	Mandatory	Description
id	String	yes	The object's ID in the UMS database
type	String	yes	One of: <ul style="list-style-type: none"><li>• tc</li><li>• tcdirectory</li><li>• profile</li><li>• profiledirectory</li><li>• masterprofile</li><li>• masterprofiledirectory</li></ul>

### Example

```
{  
  "id": "68257",  
  "type": "profile"  
}
```





## Assignment

### Summary

An Assignment links [ApiObjects](#)(see page 96), e.g. profiles to thin clients, or profiles to thin client directories.

### Properties

Name	Type	Mandatory	Description
assignee	<a href="#">ApiObject</a> (see page 96)	yes	The <a href="#">ApiObject</a> (see page 96) that is assigned
receiver	<a href="#">ApiObject</a> (see page 96)	yes	The <a href="#">ApiObject</a> (see page 96) that is assigned to
assignmentPosition	Integer	no	Position for the order in which profiles are applied
links	Array	no	Links that make IMI discoverable for an HTTP spider

### Example

```
[
{
  "assignee": {
    "id": "68257",
    "type": "profile"
  },
  "receiver": {
    "id": "48335",
    "type": "tc"
  },
  "assignmentPosition": 0,
  "links": [
    {
      "rel": "assigned",
      "href": "https://172.30.91.227:8443/umsapi/v2/profiles/68257"
```



```
  },  
  {  
    "rel": "receiver",  
    "href": "https://172.30.91.227:8443/umsapi/v2/thinclients/48335"  
  },  
  {  
    "rel": "self",  
    "href": "https://172.30.91.227:8443/umsapi/v2/profiles/68257/assignments/  
thinclients/48335"  
  }  
]  
}
```



## MainDataResource

### Summary

This resource contains the properties common to most UMS objects such as thin clients, directories and profiles. It is their default representation in lists.

### Properties

Name	Type	Mandatory	Description
id	String	no	Object ID in UMS
name	String	no	Object name
parentID	String	no	ID of parent object, usually a directory
movedToBin	Boolean	no	True if object is in the Recycling Bin
objectType		no	One of: <ul style="list-style-type: none"><li>• tc</li><li>• tcdirectory</li><li>• profile</li><li>• profiledirectory</li><li>• masterprofile</li><li>• masterprofiledirectory</li></ul>
links	Array	no	Links that make IMI discoverable for an HTTP spider



## TCResourceV2

### Summary

This resource extends [MainDataResource](#)(see page 99) by properties specific to a thin client.

### Properties

Name	Type	Mandatory	Description
id	String	no	Object ID in the UMS database
unitID	String	yes	Unique unit ID
name	String	no	Network name
parentID	String	no	Name of parent directory
movedToBin	Boolean	no	True if in Recycling Bin
objectType		no	<a href="#">tc</a>
links	Array	no	Links that make IMI discoverable for an HTTP spider
mac	String (12)	yes	MAC address
firmwareID	String	yes	Firmware ID
lastIP	String	no	Last known IP

### With facets=details

Name	Type	Mandatory	Description
site	String	no	Site
department	String	no	Department
costCenter	String	no	Cost center
comment	String	no	Comment
assetID	String	no	Asset ID
inserviceDate	String	no	Placed in service on
serialNumber	String	no	Serial Number
productId	String	no	Product ID



umsStructuralTag	String	no	Structure Tag
cpuSpeed	Integer	no	CPU speed (MHz)
cpuType	String	no	CPU type
deviceType	String	no	Device type
deviceSerialNumber	String	no	Device serial number
osType	String	no	Operating system type
flashSize	Integer	no	Flash size (MB)
memorySize	Integer	no	Memory size (MB)
networkSpeed	Integer	no	Network speed
duplexMode	String	no	Duplex mode
graphicsChipset0	String	no	First Graphics chipset
graphicsMemorySize0	Integer	no	First Graphics memory (MB)
graphicsChipset1	String	no	Second Graphics chipset
graphicsMemorySize1	Integer	no	Second Graphics memory (MB)
monitorVendor1	String	no	First Monitor vendor
monitorModel1	String	no	First Monitor model
monitorSerialnumber1	String	no	First Monitor serial number
monitorSize1	Number	no	First Monitor size
monitorNativeResolution1	String	no	First Monitor native resolution
monitor1YearOfManufacture	String	no	First Monitor year of manufacture
monitor1WeekOfManufacture	String	no	First Monitor week of manufacture
monitorVendor2	String	no	Second Monitor vendor
monitorModel2	String	no	Second Monitor model
monitorSerialnumber2	String	no	Second Monitor serial number
monitorSize2	Number	no	Second Monitor size
monitorNativeResolution2	String	no	Second Monitor native resolution



monitor2YearOfManufacture	String	no	Second Monitor year of manufacture
monitor2WeekOfManufacture	String	no	Second Monitor week of manufacture
biosVendor	String	no	BIOS vendor
biosVersion	String	no	BIOS version
biosDate	String	no	BIOS date
totalUsagetime	String	no	Total usage time
totalUptime	String	no	Total uptime
lastBoottime	String	no	Last boot time

With facets=online

online	Boolean	no	Online status
--------	---------	----	---------------

With facets=shadow

shadowSecret	String	no	Certificate and password for Secure VNC
--------------	--------	----	---



## DirectoryResourceV2

### Summary

This resource extends [MainDataResource](#)(see [page 99](#)) by properties specific to a directory.

### Properties

Name	Type	Mandatory	Description
id	String	yes	Object ID in the UMS database
name	String	yes	Directory name
parentID	String	no	ID of parent directory
movedToBin	Boolean	no	True if in Recycling Bin
objectType		no	One of: <ul style="list-style-type: none"> <li>• tcdirectory</li> <li>• profiledirectory</li> <li>• masterprofiledirectory</li> </ul>
links	Array	no	Links that make IMI discoverable for an HTTP spider

### With facets=children

DirectoryChildren	Array		Array of objects (thin clients, directories or profiles) contained in this directory
-------------------	-------	--	--



## ProfileResource

### Summary

This resource extends [MainDataResource](#)(see page 99) by properties specific to a profile.

### Properties

Name	Type	Mandatory	Description
id	String	yes	Object ID in the UMS database
name	String	no	(Master)profile name
parentID	String	no	ID of (Master)profile directory
movedToBin	Boolean	no	True if in Recycling Bin
objectType		yes	One of: <ul style="list-style-type: none"><li>• profile</li><li>• masterprofile</li></ul>
firmwareID	String	yes	Firmware ID
isMasterProfile	Boolean	no	True if master profile
overridesSessions	Boolean	no	True if profile overrides sessions
links	Array	no	Links that make IMI discoverable for an HTTP spider





## Resources

- [Thin Client](#)(see page 106)
- [Profile](#)(see page 142)
- [Master Profile](#)(see page 161)
- [Thin Client Directory](#)(see page 180)
- [Profile Directory](#)(see page 202)
- [Master Profile Directory](#)(see page 219)
- [Firmware](#)(see page 235)
- [Server Status](#)(see page 239)



## Thin Client

- [GET /v2/thinclients](#)(see page 107)
- [GET /v2/thinclients?facets=details](#)(see page 110)
- [GET /v2/thinclients?facets=online](#)(see page 113)
- [GET /v2/thinclients?facets=shadow](#)(see page 115)
- [GET /v2/thinclients/\[id\]](#)(see page 118)
- [GET /v2/thinclients/\[id\]?facets=details](#)(see page 120)
- [GET /v2/thinclients/\[id\]?facets=online](#)(see page 123)
- [GET /v2/thinclients/\[id\]?facets=shadow](#)(see page 125)
- [GET /v2/thinclients/\[id\]/assignments/profiles](#)(see page 127)
- [PUT /v2/thinclients/\[id\]/assignments/profiles](#)(see page 130)
- [PUT /v2/thinclients](#)(see page 131)
- [PUT /v2/thinclients/\[id\]](#)(see page 133)
- [DELETE /v2/thinclients/\[id\]](#)(see page 135)
- [DELETE /v1/thinclients/deletetcoffline/\[id\]](#)(see page 137)
- [POST /v2/thinclients?command={reboot|shutdown|wakeup}](#)(see page 138)
- [POST /v2/thinclients?command=settings2tc](#)(see page 140)



## GET /v2/thinclients

### Summary

Gets information on all thin clients registered with the UMS instance.

**⚠** This method will also list thin clients that are located in the **Recycle Bin** ( `"movedToBin": "true"` ), but you will not be able to call methods on those clients.

### Resource URL

`/v2/thinclients /`

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/thinclients
```

### Response Type

Returns a list of thin client resources, in the [MainDataResource](#)(see page 99) view plus

- `unitID`
- `mac`
- `firmwareID`
- `lastIP`

### Example Response

```
[
{
  "unitID": "00E0C54DCB8E",
  "mac": "00E0C54DCB8E",
  "firmwareID": "21",
  "lastIP": "172.30.91.43",
  "id": "27",
  "name": "Curry",
  "parentID": "-1",
  "movedToBin": false,
  "objectType": "tc",
  "links": [
```



```
{
  "rel": "self",
  "href": "https://172.30.91.227:8443/umsapi/latest/thinclients/27"
},
{
  "rel": "Parent",
  "href": "root"
},
{
  "rel": "Firmware",
  "href": "https://172.30.91.227:8443/umsapi/latest/firmwares/21"
}
],
{
  "unitID": "DC9C5207694E",
  "mac": "DC9C5207694E",
  "firmwareID": "13",
  "lastIP": "172.30.91.24",
  "id": "6888",
  "name": "UD3 M340C_Board",
  "parentID": "15592",
  "movedToBin": false,
  "objectType": "tc",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/latest/thinclients/6888"
    },
    {
      "rel": "Parent",
      "href": "https://172.30.91.227:8443/umsapi/latest/directories/tcdirectories/15592"
    }
  ],
}
```



```
{  
  "rel": "Firmware",  
  "href": "https://172.30.91.227:8443/umsapi/latest/firmwares/13"  
}  
]  
}
```



## GET /v2/thinclients?facets=details

### Summary

Gets detailed information on all thin clients.

**⚠** This method will also list thin clients that are located in the **Recycle Bin** ( `"movedToBin": "true"` ), but you will not be able to call methods on those clients.

### Resource URL

```
/v2/thinclients?facets=details
```

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/thinclients?facets=details
```

### Response Type

Returns a list of thin client resources as [TCResourceV2](#)([see page 100](#)) with full thin client details

### Example Response

```
[
{
  "unitID": "00E0C54DCB8E",
  "mac": "00E0C54DCB8E",
  "firmwareID": "21",
  "networkName": "Curry",
  "site": "1. Stock",
  "department": "Product Management",
  "lastIP": "172.30.91.43",
  "costCenter": "",
  "comment": "",
  "assetID": "",
  "inserviceDate": "01.06.215",
  "serialNumber": "",
  "productId": "UD6-LX 51cps",
```



```
"umsStructuralTag": "",
"cpuSpeed": 2416,
"cpuType": "Intel(R) Celeron(R) CPU J1900 @ 1.99GHz",
"deviceType": "IGEL H830C",
"deviceSerialNumber": "14D3D3C03B14470B9EM",
"osType": "IGEL Linux V5 (Kernel Version 3.13.11-ckt20)",
"flashSize": 1883,
"memorySize": 1853,
"networkSpeed": 1000,
"graphicsChipset0": "INTEL HD Graphics (Baytrail)",
"graphicsChipset1": "",
"monitorVendor1": "Samsung Electric Company",
"monitorModel1": "S24C650",
"monitorSerialnumber1": "H4MG404381",
"monitorSize1": 24,
"monitorNativeResolution1": "1920 x 1200",
"monitor1YearOfManufacture": "2015",
"monitor1WeekOfManufacture": "17",
"monitorVendor2": "Samsung Electric Company",
"monitorModel2": "S24C650",
"monitorSerialnumber2": "H4MG404389",
"monitorSize2": 24,
"monitorNativeResolution2": "1920 x 1200",
"monitor2YearOfManufacture": "2015",
"monitor2WeekOfManufacture": "17",
"biosVendor": "INSYDE Corp.",
"biosVersion": "H830C V:3.5.13-11282014",
"biosDate": "11/28/2014",
"totalUsagetime": "5798368000",
"totalUptime": "18513000",
"lastBoottime": "2015-09-29 08:30",
"id": "27",
"name": "Curry",
"parentID": "-1",
```



```
"movedToBin": false,
"objectType": "tc",
"links": [
{
"rel": "self",
"href": "https://172.30.91.227:8443/umsapi/latest/thinclients/27"
},
{
"rel": "Parent",
"href": "root"
},
{
"rel": "Firmware",
"href": "https://172.30.91.227:8443/umsapi/latest/firmwares/21"
}
],
{
"unitID": "DC9C5207694E",
"mac": "DC9C5207694E",
"firmwareID": "13",
"networkName": "IGEL-DC9C5207694E",
"lastIP": "172.30.91.24",
"productId": "UC1-LX cps",
"cpuSpeed": 1200,
[...]
```





## GET /v2/thinclients?facets=online

### Summary

Gets the online status for all thin clients.

**⚠** This method will also list thin clients that are located in the **Recycle Bin** ( `"movedToBin": "true"` ), but you will not be able to call methods on those clients.

**i** When you send this request, UMS cannot answer it from data in the database alone, but makes a network connection to each thin client, which may take some time.

### Resource URL

```
/v2/thinclients?facets=online
```

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/thinclients?facets=online
```

### Response Type

Returns a list of thin client resources in the [MainDataResource](#)(see page 99) view, but with the online status.

### Example Response

```
[
{
  "unitID": "00E0C54DCB8E",
  "mac": "00E0C54DCB8E",
  "firmwareID": "21",
  "lastIP": "172.30.91.43",
  "online": true,
  "id": "27",
  "name": "Curry
  "parentID": "-1",
  "movedToBin": false,
  "objectType": "tc",
```



```
"links": [  
  {  
    "rel": "self",  
    "href": "https://172.30.91.227:8443/umsapi/v2/thinclients/27"  
  },  
  {  
    "rel": "Parent",  
    "href": "root"  
  },  
  {  
    "rel": "Firmware",  
    "href": "https://172.30.91.227:8443/umsapi/v2/firmwares/21"  
  }  
],  
{  
  "unitID": "DC9C5207694E",  
  "mac": "DC9C5207694E",  
  "firmwareID": "13",  
  "lastIP": "172.30.91.24",  
  "online": false,  
  "id": "6888",  
  "name": "UD3 M340C_Board",  
  "parentID": "15592",
```



## GET /v2/thinclients?facets=shadow

### Summary

Gets the certificate and password for Secure VNC for all thin clients.

**i** This method will also list thin clients that are located in the **Recycle Bin** ( `"movedToBin": "true"` ), but you will not be able to call methods on those clients.

Learn more about shadowing from the How-To [Using Secure VNC via IGEL Management Interface \(IMI\)](#)(see page 245).

### Resource URL

```
/v2/thinclients?facets=shadow
```

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/thinclients?facets=shadow
```

### Response Type

Returns a list of thin client resources in the [MainDataResource](#)(see page 99) view, but with the shadowSecret object, consisting of the certificate and password fields.

### Example Response

```
[
{
  "unitID": "00E0C54DCB8E",
  "mac": "00E0C54DCB8E",
  "firmwareID": "21",
  "lastIP": "172.30.91.43",
  "shadowSecret": {
    "certificate": "-----BEGIN CERTIFICATE-----
MIIB0TCCATqgAwIBAgIERmWx9zANBgkqhkiG9w0BAQQFADA+MQswCQYDVQQGEwJE
RTEPMA0GA1UEBxMGQnJlbWVuMQ0wCwYDVQQKEwRJR0VMMB4XDTE0MDEwMTAwMDAw
```



```

MVoXDTM3MDEwMTAwMDAwMVowLTELMAkGA1UEBhMCREUxDzANBgNVBACTBkJyZW1l
bjENMA5GA1UEChMESUDFTDCBnzANBgkqhkiG9w0BAQEFAA0BjQAwYkCgYEA4b5q
QhcXhUHWndapMc9jKlQSCf0hSD4gBfzJncu2KryEHW0fCDXe44bhIYdICQ17/cIr
nE1ld4qylxnRouWUSJZkvC/+1fDqZRehPlp0nC3mjRUFotxSIL0u0+0IQ+dluEw/
lIR6kaYQbnk+CnnduG513MskeyQ5LTFs2To8W+kCAwD/OTANBgkqhkiG9w0BAQQF
AA0BgQA1E+79ZLLY7cF/5IfJjiHlBbXW2Dx67Qqs2SYwBvi31cr5fNm+qGh+7G4w
ayA0LXq2B4D6NyUsUN0JBx0C8aq8YxYvFQiD7vANVvk9cuXCeSB6d6cXlnS8IRI5
g7nAXacAPzrrORZPmFTW4R2wkJ6athrXCm+5fLyLuByRFhrh4Q==
-----END CERTIFICATE-----\n",
  "password": "371f0282-9f57-4135-b04e-68a84040ce07"
},
  "id": "27",
  "name": "Curry",
  "parentID": "-1",
  "movedToBin": false,
  "objectType": "tc",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/thinclients/27"
    },
    {
      "rel": "Parent",
      "href": "root"
    },
    {
      "rel": "Firmware",
      "href": "https://172.30.91.227:8443/umsapi/v2/firmwares/21"
    }
  ]
},
{
  "unitID": "DC9C5207694E",
  "mac": "DC9C5207694E",

```



```
"firmwareID": "13",
"lastIP": "172.30.91.24",
"shadowSecret": {
  "certificate": "-----BEGIN CERTIFICATE-----
MIIB0TCCATqgAwIBAgIEPCaQDzANBgkqhkiG9w0BAQQFADAtMQswCQYDVQQGEwJE
RTEPMA0GA1UEBxMGQnJlbWVuMQ0wCwYDVQQKEwRJR0VMMB4XDTEzMTIzMDAw
MVoXDTM2MTIzMTIzMDAwMVowLTELMAkGA1UEBhMCREUxDzANBgNVBACTBkJyZW1l
bjENMA5GA1UEChMESUdFTDCBnzANBgkqhkiG9w0BAQEFAA0BjQAwYkCgYEAwVAy
0cl0/+93LoUrmwzKK5X5HbzkZyGFEBR8KyBFpVPejxPNbTyDDK8PseEi47h4n6ci
PFK4p5CJVna0+WmQkHnc4t0gwXqalYCWsTnfZqBul1LspNkSGzC+1Svvl1gaLv4J
f1ydTwBc6Mtdoz2HYIxi/0o1FC+N/7oXZLSg9k0CAwD/OTANBgkqhkiG9w0BAQQF
AA0BgQBjYmTiWFi/MS1UYLXkt0+9ZnsLM9XUbJgM0kgm8DVxKhwl9Q9fLICa0BFO
uRlgBNGuFPvH3L01GAI5AkL2atRhF1Rut4BLr6fRHxJDqkZEONJ1tJYFNBRZfdr6
u1mLGE/x26C2D4jITRgKHy28wEgbUo4kWGjja6wNzDWGXDXG5Q==
-----END CERTIFICATE-----\n",
  "password": "883f7c16-ae68-4a9a-a6aa-f6851326b2d0"
},
"id": "6888",
"name": "UD3 M340C_Board",
"parentID": "15592",
[...]
```



GET /v2/thinclients/[id]

Summary

Gets information on the specified thin client.

Resource URL

/v2/thinclients/[id]

Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/thinclients/27
```

Request Path Variables

Name	Type	Mandatory	Description
id	String	yes	thin client ID

Response Type

Returns a [MainDataResource](#)(see [page 99](#)) for the thin client plus

- unitID
- mac
- firmwareID
- lastIP

Example Response

```
{
  "unitID": "00E0C54DCB8E",
  "mac": "00E0C54DCB8E",
  "firmwareID": "21",
  "lastIP": "172.30.91.43",
  "id": "27",
  "name": "Curry",
  "parentID": "-1",
  "movedToBin": false,
  "objectType": "tc",
```



```
"links": [  
  {  
    "rel": "self",  
    "href": "https://172.30.91.227:8443/umsapi/v2/thinclients/27"  
  },  
  {  
    "rel": "Parent",  
    "href": "root"  
  },  
  {  
    "rel": "Firmware",  
    "href": "https://172.30.91.227:8443/umsapi/v2/firmwares/21"  
  }  
]
```



## GET /v2/thinclients/[id]?facets=details

### Summary

Gets detailed information on the specified thin client.

### Resource URL

```
/v2/thinclients/[id]?facets=details
```

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/thinclients/27?facets=details
```

### Request Path Variables

Name	Type	Mandatory	Description
id	String	yes	thin client ID

### Response Type

Returns the detailed [TCResourceV2](#)(see page 100) for the thin client.

### Example Response

```
{
  "unitID": "00E0C54DCB8E",
  "mac": "00E0C54DCB8E",
  "firmwareID": "21",
  "networkName": "Curry",
  "site": "1. Stock",
  "department": "Product Management",
  "lastIP": "172.30.91.43",
  "costCenter": "",
  "comment": "",
  "assetID": "",
  "inserviceDate": "01.06.215",
  "serialNumber": "",
  "productId": "UD6-LX 51cps",
```





```
"umsStructuralTag": "",
"cpuSpeed": 2416,
"cpuType": "Intel(R) Celeron(R) CPU J1900 @ 1.99GHz",
"deviceType": "IGEL H830C",
"deviceSerialNumber": "14D3D3C03B14470B9EM",
"osType": "IGEL Linux V5 (Kernel Version 3.13.11-ckt20)",
"flashSize": 1883,
"memorySize": 1853,
"networkSpeed": 1000,
"graphicsChipset0": "INTEL HD Graphics (Baytrail)",
"graphicsChipset1": "",
"monitorVendor1": "Samsung Electric Company",
"monitorModel1": "S24C650",
"monitorSerialnumber1": "H4MG404381",
"monitorSize1": 24,
"monitorNativeResolution1": "1920 x 1200",
"monitor1YearOfManufacture": "2015",
"monitor1WeekOfManufacture": "17",
"monitorVendor2": "Samsung Electric Company",
"monitorModel2": "S24C650",
"monitorSerialnumber2": "H4MG404389",
"monitorSize2": 24,
"monitorNativeResolution2": "1920 x 1200",
"monitor2YearOfManufacture": "2015",
"monitor2WeekOfManufacture": "17",
"biosVendor": "INSYDE Corp.",
"biosVersion": "H830C V:3.5.13-11282014",
"biosDate": "11/28/2014",
"totalUsagetime": "5798368000",
"totalUptime": "18513000",
"lastBoottime": "2015-09-29 08:30",
"id": "27",
"name": "Curry",
"parentID": "-1",
```



```
"movedToBin": false,  
"objectType": "tc",  
"links": [  
  {  
    "rel": "self",  
    "href": "https://172.30.91.227:8443/umsapi/v2/thinclients/27"  
  },  
  {  
    "rel": "Parent",  
    "href": "root"  
  },  
  {  
    "rel": "Firmware",  
    "href": "https://172.30.91.227:8443/umsapi/v2/firmwares/21"  
  }  
]  
}
```



## GET /v2/thinclients/[id]?facets=online

### Summary

Gets the online status for the specified thin client.

**i** When you send this request, UMS cannot answer it from data in the database alone, but makes a network connection to the thin client, which may take some time.

### Resource URL

```
/v2/thinclients/[id]?facets=online
```

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/thinclients/27?facets=online
```

### Request Path Variables

Name	Type	Mandatory	Description
id	String	yes	thin client ID

### Response Type

Returns the [MainDataResource](#)(see page 99) for the thin client, but with the online status.

### Example Response

```
{
  "unitID": "00E0C54DCB8E",
  "mac": "00E0C54DCB8E",
  "firmwareID": "21",
  "lastIP": "172.30.91.43",
  "online": false,
  "id": "27",
  "name": "Curry",
  "parentID": "-1",
  "movedToBin": false,
  "objectType": "tc",
```



```
"links": [  
  {  
    "rel": "self",  
    "href": "https://172.30.91.227:8443/umsapi/v2/thinclients/27"  
  },  
  {  
    "rel": "Parent",  
    "href": "root"  
  },  
  {  
    "rel": "Firmware",  
    "href": "https://172.30.91.227:8443/umsapi/v2/firmwares/21"  
  }  
]
```



## GET /v2/thinclients/[id]?facets=shadow

### Summary

Gets the certificate and password for Secure VNC for the specified thin client.

Learn more about shadowing from the Best Practice document [Using Secure VNC via IGEL Management Interface \(IMI\)](#)(see [page 245](#)).

### Resource URL

```
/v2/thinclients/[id]?facets=shadow
```

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/thinclients/27?facets=shadow
```

Name	Type	Mandatory	Description
id	String	yes	thin client ID

### Response Type

Returns the [MainDataResource](#)(see [page 99](#)) for the specified thin client, but with the shadowSecret object, consisting of the certificate and password fields.

### Example Response

```
{
  "unitID": "00E0C54DCB8E",
  "mac": "00E0C54DCB8E",
  "firmwareID": "21",
  "lastIP": "172.30.91.43",
  "shadowSecret": {
    "certificate": "-----BEGIN CERTIFICATE-----
MIIB0TCCATqgAwIBAgIERmWx9zANBgkqhkiG9w0BAQQFADAtMQswCQYDVQQGEwJE
RTEPMA0GA1UEBxMGQnJlbWVuMQ0wCwYDVQQKEwRJR0VMMB4XDTE0MDEwMTAwMDAw
MVoxDjM0MDEwMTAwMDAwMVowLTELMAkGA1UEBhMCREUxDzANBgNVBACTBkYjZW1l
bjENMA5GA1UEChMESUdFTDCBnzANBgkqhkiG9w0BAQEFAA0BjQAwYkCgYEA4b5q
QhcxhUhwNdapMc9jKlQScf0hSD4gBfzJncu2KryEHW0fCDXe44bhIYdICQ17/cIr
nE1ld4qylxnRouWUSJZkvC/+1fDqZRehPlpOnC3mjRUFotxSIL0u0+0IQ+dluEw/
```



```
lIR6kaYQbnk+CnnduG513MskeyQ5LTFs2To8W+kCAwD/OTANBgkqhkiG9w0BAQQF
AA0BgQAlE+79ZLLY7cF/5IfJjiHlBbXW2Dx67Qqs2SYwBvi31cr5fNm+qGh+7G4w
ayA0LXq2B4D6NyUsuN0JBx0C8aq8YxYvFQiD7vANVvk9cuXCeSB6d6cXlnS8IRI5
g7nAXacAPzrrORZPmFTW4R2wkJ6athrXCm+5fLyLuByRFhrh4Q==
-----END CERTIFICATE-----\n",
  "password": "464abcbd-1399-47d0-b011-eba2e67a72b0"
},
  "id": "27",
  "name": "Curry",
  "parentID": "-1",
  "movedToBin": false,
  "objectType": "tc",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/thinclients/27"
    },
    {
      "rel": "Parent",
      "href": "root"
    },
    {
      "rel": "Firmware",
      "href": "https://172.30.91.227:8443/umsapi/v2/firmwares/21"
    }
  ]
}
```



## GET /v2/thinclients/[id]/assignments/profiles

### Summary

Gets the profile and master profile assignments for the specified thin client, in order of their application.

### Resource URL

```
/v2/thinclients/[id]/assignments/profiles
```

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/thinclients/48335/assignments/profiles
```

### Request Path Variables

Name	Type	Mandatory	Description
id	String	yes	thin client ID

### Response Type

Returns a list of Profile and Master Profile [Assignments](#)(see [page 97](#)).

### Example Response

```
[
{
  "assignee": {
    "id": "68257",
    "type": "profile"
  },
  "receiver": {
    "id": "48335",
    "type": "tc"
  },
  "assignmentPosition": 0,
  "links": [
    {
      "rel": "assigned",
```



```
"href": "https://172.30.91.227:8443/umsapi/v2/profiles/68257"
},
{
  "rel": "receiver",
  "href": "https://172.30.91.227:8443/umsapi/v2/thinclients/48335"
},
{
  "rel": "self",
  "href": "https://172.30.91.227:8443/umsapi/v2/profiles/68257/assignments/thinclients/48335"
}
]
},
{
  "assignee": {
    "id": "72063",
    "type": "masterprofile"
  },
  "receiver": {
    "id": "48335",
    "type": "tc"
  },
  "assignmentPosition": 1,
  "links": [
    {
      "rel": "assigned",
      "href": "https://172.30.91.227:8443/umsapi/v2/masterprofiles/72063"
    },
    {
      "rel": "receiver",
      "href": "https://172.30.91.227:8443/umsapi/v2/thinclients/48335"
    },
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/masterprofiles/72063/assignments/thinclients/48335"
    }
  ]
}
```





```
}  
]  
}  
]
```



## PUT /v2/thinclients/[id]/assignments/profiles

### Summary

Creates Profile and Master Profile Assignments for the specified Thin Client.

### Resource URL

```
/v2/thinclients/[id]/assignments/profiles
```

### Example Request

```
curl \
--request PUT \
--data '[{ "assignee": {
  "id": "68257",
  "type": "profile"
},
"receiver": {
  "id": "48335",
  "type": "tc"
}}]' \
https://[server]:8443/umsapi/v2/thinclients/48335/assignments/profiles
```

### Request Path Variables

Name	Type	Mandatory	Description
id	String	yes	thin client ID

### Request Body

A list of Profile and Master Profile [Assignments](#)(see page 97).

### Example Response

```
{
  "message": "1 assignments successfully assigned to thinclient <48335>"
}
```



## PUT /v2/thinclients

### Summary

Creates a new thin client resource.

### Resource URL

```
/v2/thinclients
```

### Example Request

```
curl \
--request PUT \
--header "Content-type: application/json" \
--data '{"mac":"00E0C53C3881", \
"firmwareID":"2"}' \
https://[server]:8443/umsapi/v2/thinclients/
```

### Request Body

Name	Type	Mandatory	Description
mac	String (12)	yes	thin client MAC address
firmwareID	String	yes	firmware ID
name	String	no	thin client name
parentID	String	no	ID of parent directory
site	String	no	thin client site
department	String	no	department
costCenter	String	no	cost center
lastIP	String	no	last known IP of thin client
comment	String	no	comment field
assetID	String	no	asset ID
inserviceDate	String	no	
serialNumber	String	no	serial number of thin client



Response Body

Name	Type	Description
message	String	success or error message
id	String	thin client ID
parentID	String	ID of parent directory
name	String	thin client name

Example Response

```
{
  "message": "Thin client successfully inserted.",
  "id": "7735",
  "name": "My Name",
  "parentID": "-1"
}
```



## PUT /v2/thinclients/[id]

### Summary

Updates properties of the specified thin client

### Resource URL

/v2/thinclients/[id]

### Example Request

```
curl \
--request PUT \
--header "Content-type: application/json" \
--data '{"name":"reception thin client", \
"site":"main campus"}' \
https://[server]:8443/umsapi/v1/thinclients/123
```

### Request Path Variables

Name	Type	Mandatory	Description
id	String	yes	thin client ID

### Request Body

Name	Type	Mandatory	Description
name	String	no	thin client name
site	String	no	site of operation
department	String	no	department
costCenter	String	no	cost center
lastIP	String	no	last known IP of thin client
comment	String	no	comment field
assetID	String	no	asset ID
inserviceDate	String	no	
serialNumber	String	no	serial number of thin client



### Response Type

Returns a success message.

### Example Response


```
{  
  "message": "Update successful"  
}
```



## DELETE /v2/thinclients/[id]

### Summary

Deletes a thin client.

 This does not move the thin client into the **Recycle Bin** but simply deletes it.

### Resource URL

```
/v2/thinclients/[id]
```

### Example Request

```
curl \
--request DELETE \
https://[server]:8443/umsapi/v2/thinclients/2704
```

### Request Path Variables

Name	Type	Mandatory	Description
id	String	yes	thin client ID

### Response Type

Returns a message.

### Example Response

```
200 OK
---
{
  "CommandExecList": [
    {
      "execID": "ID-PM-MH-WIN7-UMS-63885-1424682219085-5-0",
      "mac": "008064AD82FB",
      "message": "OK",
      "exectime": 1424698605821,
      "state": "SUCCESS"
    }
  ]
}
```



}






## DELETE /v1/thinclients/deletetcoffline/[id]

### Summary

Deletes a thin client even if it is offline.

 This does not move the thin client into the **Recycle Bin** but simply deletes it.

### Resource URL

```
/v2/thinclients/[id]/deletetcoffline
```

### Example Request

```
curl \
--request DELETE \
https://[server]:8443/umsapi/v2/thinclients/2704/deletetcoffline
```

### Request Path Variables

Name	Type	Mandatory	Description
id	String	yes	thin client ID

### Response Type

Returns a message.

### Example Response

```
200 OK
---
{
  "message": "Offline deletion successful"
}
```



## POST /v2/thinclients?command={reboot|shutdown|wakeup}

### Summary

Sends a command to all thin clients listed in the request body.

### Resource URL

```
/v2/thinclients/?command={reboot|shutdown|wakeup}
```

### Example Request

```
curl \
--request POST \
--data ' [{"id": "27", "type": "tc"}, {"id": "72014", "type": "tc"} ] '
https://[server]:8443/umsapi/v2/thinclients?command=wakeup
```

### Request Body

A list of [APIObjects](#) (see page 96).

### Example Response

```
{
  "CommandExecList": [
    {
      "execID": "ID-PM-MH-WIN7-UMS-54530-1456839861871-5-0",
      "id": "72014",
      "mac": "00E0C561EEED",
      "exectime": "1456845240566",
      "message": "OK",
      "state": "SUCCESS"
    },
    {
      "execID": "ID-PM-MH-WIN7-UMS-54530-1456839861871-5-0",
      "id": "27",
      "mac": "00E0C54DCB8E",
      "exectime": "1456845240560",
      "message": "OK",
      "state": "SUCCESS"
    }
  ]
}
```



```
}  
]  
}
```



## POST /v2/thinclients?command=settings2tc

### Summary

Sends settings modified in the UMS database to all thin clients listed in the request body immediately.

### Resource URL

```
/v2/thinclients/?command=settings2tc
```

### Example Request

```
curl \
--request POST \
--data ' [{"id": "27", "type": "tc"}, {"id": "72014", "type": "tc"} ] '
https://[server]:8443/umsapi/v2/thinclients?command=settings2tc
```

### Request Body

A list of [APIObjects](#) (see page 96).

### Example Response

```
{
  "CommandExecList": [
    {
      "execID": "ID-PM-MH-WIN7-UMS-54530-1456839861871-5-0",
      "id": "72014",
      "mac": "00E0C561EEED",
      "exectime": "1456845240566",
      "message": "OK",
      "state": "SUCCESS"
    },
    {
      "execID": "ID-PM-MH-WIN7-UMS-54530-1456839861871-5-0",
      "id": "27",
      "mac": "00E0C54DCB8E",
      "exectime": "1456845240560",
      "message": "OK",
      "state": "SUCCESS"
    }
  ]
}
```



```
}  
]  
}
```



## Profile

- [GET /v2/profiles](#)(see page 143)
- [GET /v2/profiles/\[profileid\]](#)(see page 146)
- [PUT /v2/profiles/\[profileid\]](#)(see page 148)
- [DELETE /v2/profiles/\[profileid\]](#)(see page 149)
- [GET /v2/profiles/\[profileid\]/assignments/thinclients](#)(see page 150)
- [PUT /v2/profiles/\[profileid\]/assignments/thinclients/](#)(see page 153)
- [DELETE /v2/profiles/\[profileid\]/assignments/thinclients/\[id\]](#)(see page 155)
- [GET /v2/profiles/\[profileid\]/assignments/tcdirectories](#)(see page 156)
- [PUT /v2/profiles/\[profileid\]/assignments/tcdirectories](#)(see page 158)
- [DELETE /v2/profiles/\[profileid\]/assignments/tcdirectories/\[id\]](#)(see page 160)



## GET /v2/profiles

### Summary

Gets information on all profiles on the UMS instance.

 This method will also list profiles that are located in the **Recycle Bin** ( "movedToBin": "true" ).

### Resource URL

/v2/profiles /

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/profiles
```

### Response Type

Returns a list of profiles.

### Example Response

```
[
{
  "firmwareID": "2",
  "isMasterProfile": false,
  "overridesSessions": false,
  "id": "6585",
  "name": "Benchmark intern",
  "parentID": "20453",
  "movedToBin": false,
  "objectType": "profile",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/profiles/6585"
    }
  ],
}
```



```
{
  "rel": "Parent",
  "href": "https://172.30.91.227:8443/umsapi/v2/directories/profiledirectories/20453"
},
{
  "rel": "Firmware",
  "href": "https://172.30.91.227:8443/umsapi/v2/firmwares/2"
}
],
{
  "firmwareID": "2",
  "isMasterProfile": false,
  "overridesSessions": false,
  "id": "6592",
  "name": "Benchmark extern",
  "parentID": "20453",
  "movedToBin": true,
  "objectType": "profile",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/profiles/6592"
    },
    {
      "rel": "Parent",
      "href": "https://172.30.91.227:8443/umsapi/v2/directories/profiledirectories/20453"
    },
    {
      "rel": "Firmware",
      "href": "https://172.30.91.227:8443/umsapi/v2/firmwares/2"
    }
  ]
}
```





```
},
```

```
[...]
```

```
]
```



## GET /v2/profiles/[profileid]

### Summary

Gets information on the specified profile.

### Resource URL

```
/v2/profiles/[profileid]
```

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/profiles/6585
```

### Request Path Variables

Name	Type	Mandatory	Description
profileid	String	yes	profile ID

### Response Type

Returns a profile.

### Example Response

```
{
  "firmwareID": "2",
  "isMasterProfile": false,
  "overridesSessions": false,
  "id": "6585",
  "name": "Benchmark intern",
  "parentID": "20453",
  "movedToBin": false,
  "objectType": "profile",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/profiles/6585"
```



```
  },  
  {  
    "rel": "Parent",  
    "href": "https://172.30.91.227:8443/umsapi/v2/directories/profiledirectories/  
20453"  
  },  
  {  
    "rel": "Firmware",  
    "href": "https://172.30.91.227:8443/umsapi/v2/firmwares/2"  
  }  
]  
}
```



## PUT /v2/profiles/[profileid]

### Summary

Updates a profile name.

### Resource URL

/v2/profiles/[profileid]

### Example Request

```
curl \
--request PUT \
--header "Content-type: application/json" \
--data '{"name":"New Profile Name"}' \
https://[server]:8443/umsapi/v2/profiles/6585
```

### Request Path Variables

Name	Type	Mandatory	Description
profileid	String	yes	profile ID

### Request Body

Name	Type	Mandatory	Description
name	String	no	profile name

### Response Type

Returns a success message.

### Example Response


```
{
  "message": "Update successful"
}
```



## DELETE /v2/profiles/[profileid]

### Summary

Deletes profile.

 This does not move the master profile into the **Recycle Bin** but simply deletes it.

### Resource URL

```
/v2/profiles/[profileid]
```

### Example Request

```
curl \  
--request DELETE \  
https://[server]:8443/umsapi/v2/profiles/72372
```

### Request Path Variables

Name	Type	Mandatory	Description
id	String	yes	profile ID

### Response Type

Returns a message.

### Example Response

```
{  
  "message": "Deleted profile with id 72372"  
}
```



## GET /v2/profiles/[profileid]/assignments/thinclients

### Summary

Gets the thin clients the profile is assigned to.

### Resource URL

```
/v2/profile/[profileid]/assignments/thinclients
```

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/profiles/20467/assignments/thinclients
```

### Request Path Variables

Name	Type	Mandatory	Description
profileid	String	yes	profile ID

### Response Type

Returns a list of [assignments](#)([see page 97](#)).

### Example Response

```
[
{
  "assignee": {
    "id": "20467",
    "type": "profile"
  },
  "receiver": {
    "id": "23028",
    "type": "tc"
  },
  "assignmentPosition": 0,
  "links": [
    {
      "rel": "assigned",
```



```
"href": "https://172.30.91.227:8443/umsapi/v2/profiles/20467"
},
{
  "rel": "receiver",
  "href": "https://172.30.91.227:8443/umsapi/v2/thinclients/23028"
},
{
  "rel": "self",
  "href": "https://172.30.91.227:8443/umsapi/v2/profiles/20467/assignments/
thinclients/23028"
}
]
},
{
  "assignee": {
    "id": "20467",
    "type": "profile"
  },
  "receiver": {
    "id": "48335",
    "type": "tc"
  },
  "assignmentPosition": 1,
  "links": [
    {
      "rel": "assigned",
      "href": "https://172.30.91.227:8443/umsapi/v2/profiles/20467"
    },
    {
      "rel": "receiver",
      "href": "https://172.30.91.227:8443/umsapi/v2/thinclients/48335"
    },
    {
      "rel": "self",
```



```
"href": "https://172.30.91.227:8443/umsapi/v2/profiles/20467/assignments/  
thinclients/48335"
```

```
}
```

```
]
```

```
}
```

```
]
```





PUT /v2/profiles/[profileid]/assignments/thinclients/

Summary

Assigns a profile to one or more thin clients.

Resource URL

/v2/profiles/[profileid]/assignments/thinclients/

Example Request

```
curl \
--request PUT \
--header "Content-type: application/json" \
--data ' [{ "assignee": {
  "id": "20452",
  "type": "profile"
},
  "receiver": {
    "id": "23028",
    "type": "tc"}
}]' \
https://[server]:8443/umsapi/v2/profiles/20452/assignments/thinclients/
```

Request Path Variables

Name	Type	Mandatory	Description
profileid	String	yes	master profile ID

Request Body

A list of [Assignments](#)(see [page 97](#))

Response Type

Returns a success message.

Example Response

```
{
  "message": "1 assignments successfully assigned to thinclient <23028>"
}
```



}



## DELETE /v2/profiles/[profileid]/assignments/thinclients/[id]

### Summary

Deletes assignment of the specified profile to the specified thin client.

### Resource URL

```
/v2/profiles/[profileid]/assignments/thinclients/[id]
```

### Example Request

```
curl \
--request DELETE\
https://[server]:8443/umsapi/v2/profiles/68257/assignments/thinclients/48335
```

### Request Path Variables

Name	Type	Mandatory	Description
profileid	String	yes	profile ID
id	String	yes	thin client ID

### Response Type

Returns a message.

### Example Response

```
200 OK
---
{
  "message": "deleted profile assignment"
}
```



## GET /v2/profiles/[profileid]/assignments/tcdirectories

### Summary

Gets the thin clients the profile is assigned to.

### Resource URL

```
/v2/profile/[profileid]/assignments/thinclients
```

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/profiles/35549/assignments/tcdirectories/76462
```

### Request Path Variables

Name	Type	Mandatory	Description
profileid	String	yes	profile ID

### Response Type

Returns a list of [assignments](#)(see [page 97](#)).

### Example Response

```
[
{
  "assignee": {
    "id": "35549",
    "type": "profile"
  },
  "receiver": {
    "id": "76462",
    "type": "tcdirectory"
  },
  "assignmentPosition": 0,
  "links": [
    {
      "rel": "assigned",
```



```
"href": "https://172.30.91.227:8443/umsapi/v2/profiles/35549"
},
{
  "rel": "receiver",
  "href": "https://172.30.91.227:8443/umsapi/v2/directories/tcdirectories/76462"
},
{
  "rel": "self",
  "href": "https://172.30.91.227:8443/umsapi/v2/profiles/35549/assignments/
tcdirectories/76462"
}
]
}
```



PUT /v2/profiles/[profileid]/assignments/tcdirectories

Summary

Assigns a profile to one or more thin client directories.

Resource URL

/v2/profiles/[profileid]/assignments/tcdirectories/

Example Request

```
curl \
--request PUT \
--header "Content-type: application/json" \
--data ' [{ "assignee": {
  "id": "35549",
  "type": "profile"
},
"receiver": {
  "id": "76462",
  "type": "tcdirectory"}
}]' \
https://[server]:8443/umsapi/v2/profiles/35549/assignments/tcdirectories/
```

Request Path Variables

Name	Type	Mandatory	Description
profileid	String	yes	master profile ID

Request Body

A list of [Assignments](#)(see page 97)

Response Type

Returns a success message.

Example Response

```
{
```



```
"message": "1 assignments successfully assigned to thinclient directory  
<35549>"  
}
```



## DELETE /v2/profiles/[profileid]/assignments/tcdirectories/[id]

### Summary

Deletes assignment of the specified profile to the specified thin client directory.

### Resource URL

```
/v2/profiles/[profileid]/assignments/tcdirectory/[id]
```

### Example Request

```
curl \
--request DELETE\
https://[server]:8443/umsapi/v2/profiles/35549/assignments/tcdirectories/76462
```

### Request Path Variables

Name	Type	Mandatory	Description
profileid	String	yes	profile ID
id	String	yes	thin client directory ID

### Response Type

Returns a message.

### Example Response

```
200 OK
---
{
  "message": "deleted profile assignment"
}
```





## Master Profile

- [GET /v2/masterprofiles](#)(see page 162)
- [GET /v2/masterprofiles/\[profileid\]](#)(see page 165)
- [PUT /v2/masterprofile/\[id\]](#)(see page 167)
- [DELETE /v2/masterprofiles/\[profileid\]](#)(see page 168)
- [GET /v2/masterprofiles/\[profileid\]/assignments/thinclients](#)(see page 169)
- [PUT /v2/masterprofiles/\[profileid\]/assignments/thinclients/](#)(see page 172)
- [DELETE /v2/masterprofiles/\[profileid\]/assignments/thinclients/\[id\]](#)(see page 174)
- [GET /v2/masterprofiles/\[profileid\]/assignments/tcdirectories](#)(see page 175)
- [PUT /v2/masterprofiles/\[profileid\]/assignments/tcdirectories/](#)(see page 177)
- [DELETE /v2/masterprofiles/\[profileid\]/assignments/tcdirectories/\[id\]](#)(see page 179)



## GET /v2/masterprofiles

### Summary

Gets information on all master profiles on the UMS instance.

**i** This method will also list master profiles that are located in the **Recycle Bin** ( `"movedToBin": "true"` )

### Resource URL

`/v2/masterprofiles /`

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/masterprofiles
```

### Response Type

Returns a list of profiles.

### Example Response

```
[
{
  "firmwareID": "32",
  "isMasterProfile": true,
  "overridesSessions": false,
  "id": "72054",
  "name": "Company Master Profile",
  "parentID": "-14",
  "movedToBin": false,
  "objectType": "masterprofile",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/masterprofiles/72054"
```



```
{
  },
  {
    "rel": "Parent",
    "href": "root"
  },
  {
    "rel": "Firmware",
    "href": "https://172.30.91.227:8443/umsapi/v2/firmwares/32"
  }
]
},
{
  "firmwareID": "23",
  "isMasterProfile": true,
  "overridesSessions": false,
  "id": "72055",
  "name": "SpECIAL Master Profile",
  "parentID": "-14",
  "movedToBin": false,
  "objectType": "masterprofile",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/masterprofiles/72055"
    },
    {
      "rel": "Parent",
      "href": "root"
    },
    {
      "rel": "Firmware",
      "href": "https://172.30.91.227:8443/umsapi/v2/firmwares/23"
    }
  ]
}
```



```
}  
]
```



## GET /v2/masterprofiles/[profileid]

### Summary

Gets information on the specified master profile.

### Resource URL

```
/v2/masterprofiles/[profileid]
```

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/profiles/72054
```

### Request Path Variables

Name	Type	Mandatory	Description
profileid	String	yes	master profile ID

### Response Type

Returns a master profile.

### Example Response

```
{
  "firmwareID": "32",
  "isMasterProfile": true,
  "overridesSessions": false,
  "id": "72054",
  "name": "Company Master Profile",
  "parentID": "-14",
  "movedToBin": false,
  "objectType": "masterprofile",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/masterprofiles/72054"
```



```
  },  
  {  
    "rel": "Parent",  
    "href": "root"  
  },  
  {  
    "rel": "Firmware",  
    "href": "https://172.30.91.227:8443/umsapi/v2/firmwares/32"  
  }  
]  
}
```



## PUT /v2/masterprofile/[id]

### Summary

Updates a master profile name.

### Resource URL

```
/v2/masterprofiles/[profileid]
```

### Example Request

```
curl \
--request PUT \
--header "Content-type: application/json" \
--data '{"name":"New Master Profile Name"}' \
https://[server]:8443/umsapi/v2/masterprofiles/6585
```

### Request Path Variables

Name	Type	Mandatory	Description
profileid	String	yes	master profile ID

### Request Body

Name	Type	Mandatory	Description
name	String	no	master profile name

### Response Type

Returns a success message.

### Example Response


```
{
  "message": "Update successful"
}
```



## DELETE /v2/masterprofiles/[profileid]

### Summary

Deletes a master profile.

 This does not move the master profile into the **Recycle Bin** but simply deletes it.

### Resource URL

```
/v2/masterprofiles/[profileid]
```

### Example Request

```
curl \
--request DELETE \
https://[server]:8443/umsapi/v2/masterprofiles/72303
```

### Request Path Variables

Name	Type	Mandatory	Description
id	String	yes	master profile ID

### Response Type

Returns a message.

### Example Response

```
{
  "message": "Deleted profile with id 72303"
}
```





## GET /v2/masterprofiles/[profileid]/assignments/thinclients

### Summary

Gets the thin clients the master profile is assigned to.

### Resource URL

```
/v2/masterprofiles/[profileid]/assignments/thinclients
```

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/masterprofiles/72308/assignments/thinclients
```

### Request Path Variables

Name	Type	Mandatory	Description
profileid	String	yes	profile ID

### Response Type

Returns a list of [assignments](#)([see page 97](#)).

### Example Response

```
[
{
  "assignee": {
    "id": "72308",
    "type": "masterprofile"
  },
  "receiver": {
    "id": "48335",
    "type": "tc"
  },
  "assignmentPosition": 0,
  "links": [
    {
      "rel": "assigned",
```



```
"href": "https://172.30.91.227:8443/umsapi/v2/masterprofiles/72308"
},
{
  "rel": "receiver",
  "href": "https://172.30.91.227:8443/umsapi/v2/thinclients/48335"
},
{
  "rel": "self",
  "href": "https://172.30.91.227:8443/umsapi/v2/masterprofiles/72308/
assignments/thinclients/48335"
}
]
},
{
  "assignee": {
    "id": "72308",
    "type": "masterprofile"
  },
  "receiver": {
    "id": "72014",
    "type": "tc"
  },
  "assignmentPosition": 1,
  "links": [
    {
      "rel": "assigned",
      "href": "https://172.30.91.227:8443/umsapi/v2/masterprofiles/72308"
    },
    {
      "rel": "receiver",
      "href": "https://172.30.91.227:8443/umsapi/v2/thinclients/72014"
    },
    {
      "rel": "self",
```



```
"href": "https://172.30.91.227:8443/umsapi/v2/masterprofiles/72308/assignments/thinclients/72014"
```

```
}
```

```
]
```

```
}
```

```
]
```



PUT /v2/masterprofiles/[profileid]/assignments/thinclients/

Summary

Assigns a master profile to one or more thin clients.

Resource URL

/v2/masterprofiles/[profileid]/assignments/thinclients/

Example Request

```
curl \
--request PUT \
--header "Content-type: application/json" \
--data ' [{ "assignee": {
  "id": "72066",
  "type": "masterprofile"
},
"receiver": {
  "id": "48335",
  "type": "tc"}
}]' \
https://[server]:8443/umsapi/v2/masterprofiles/72066/assignments/thinclients/
```

Request Path Variables

Name	Type	Mandatory	Description
profileid	String	yes	master profile ID

Request Body

A list of [Assignments](#)(see [page 97](#))

Response Type

Returns a success message.

Example Response

```
{
  "message": "1 assignments successfully assigned to thinclient <48335>"
}
```



}



## DELETE /v2/masterprofiles/[profileid]/assignments/thinclients/[id]

### Summary

Deletes assignment of the specified master profile to the specified thin client.

### Resource URL

```
/v2/masterprofiles/[profileid]/assignments/thinclients/[id]
```

### Example Request

```
curl \
--request DELETE\
https://[server]:8443/umsapi/v2/masterprofiles/68257/assignments/thinclients/48335
```

### Request Path Variables

Name	Type	Mandatory	Description
profileid	String	yes	master profile ID
id	String	yes	thin client ID

### Response Type

Returns a message.

### Example Response

```
200 OK
---
{
  "message": "deleted profile assignment"
}
```



## GET /v2/masterprofiles/[profileid]/assignments/tcdirectories

### Summary

Gets the thin client directories the master profile is assigned to.

### Resource URL

```
/v2/masterprofiles/[profileid]/assignments/tcdirectories
```

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/masterprofiles/72098/assignments/tcdirectories
```

### Request Path Variables

Name	Type	Mandatory	Description
profileid	String	yes	Master profile ID

### Response Type

Returns a list of [assignments](#)(see [page 97](#)).

### Example Response

```
[
{
  "assignee": {
    "id": "72098",
    "type": "masterprofile"
  },
  "receiver": {
    "id": "76462",
    "type": "tcdirectory"
  },
  "assignmentPosition": 0,
  "links": [
    {
      "rel": "assigned",
```



```
"href": "https://172.30.91.227:8443/umsapi/v2/masterprofiles/72098"
},
{
  "rel": "receiver",
  "href": "https://172.30.91.227:8443/umsapi/v2/directories/tcdirectories/76462"
},
{
  "rel": "self",
  "href": "https://172.30.91.227:8443/umsapi/v2/masterprofiles/72098/
assignments/tcdirectories/76462"
}
]
}
```





PUT /v2/masterprofiles/[profileid]/assignments/tcdirectories/

Summary

Assigns a master profile to one or more thin client directory.

Resource URL

/v2/masterprofiles/[profileid]/assignments/tcdirectories/

Example Request

```
curl \
--request PUT \
--header "Content-type: application/json" \
--data ' [{ "assignee": {
  "id": "72098",
  "type": "masterprofile"
},
"receiver": {
  "id": "76462",
  "type": "tcdirectory"}
}]' \
https://[server]:8443/umsapi/v2/masterprofiles/72066/assignments/thinclients/
```

Request Path Variables

Name	Type	Mandatory	Description
profileid	String	yes	master profile ID

Request Body

A list of [Assignments](#)(see [page 97](#))

Response Type

Returns a success message.

Example Response

```
{
```



```
"message": "1 assignments successfully assigned to thinclient directory  
<72098>"  
}
```



## DELETE /v2/masterprofiles/[profileid]/assignments/tcdirectories/[id]

### Summary

Deletes assignment of the specified master profile to the specified thin client directory.

### Resource URL

```
/v2/masterprofiles/[profileid]/assignments/tcdirectory/[id]
```

### Example Request

```
curl \
--request DELETE\
https://[server]:8443/umsapi/v2/masterprofiles/72098/assignments/tcdirectories/76462
```

### Request Path Variables

Name	Type	Mandatory	Description
profileid	String	yes	master profile ID
id	String	yes	thin client directory ID

### Response Type

Returns a message.

### Example Response

```
200 OK
---
{
  "message": "deleted profile assignment"
}
```



## Thin Client Directory

- [GET /v2/directories/tcdirectories](#)(see page 181)
- [GET /v2/directories/tcdirectories?facets=children](#)(see page 183)
- [GET /v2/directories/tcdirectories/\[id\]](#)(see page 187)
- [GET /v2/directories/tcdirectories/\[id\]?facets=children](#)(see page 189)
- [PUT /v2/directories/tcdirectories/](#)(see page 193)
- [PUT /v2/directories/tcdirectories/\[id\]](#)(see page 194)
- [PUT /v2/directories/tcdirectories/\[id\]?operation=move](#)(see page 195)
- [DELETE /v2/directories/tcdirectories/\[id\]](#)(see page 197)
- [GET /v2/directories/tcdirectories/\[id\]/assignments/profiles](#)(see page 198)
- [PUT /v2/directories/tcdirectories/\[id\]/assignments/profiles](#)(see page 201)



## GET /v2/directories/tcdirectories

### Summary

Gets information on all Thin Client Directories in a flat format.

 This method will also list directories that are located in the **Recycle Bin** ( "movedToBin": "true" ).

### Resource URL

/v2/directories/tcdirectories

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/directories/tcdirectories
```

### Response Type

Returns a list of [Directories](#)(see page 103).

### Example Response

```
[
{
  "id": "50705",
  "name": "New",
  "parentID": "-1",
  "movedToBin": false,
  "objectType": "tcdirectory",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/directories/tcdirectories/50705"
    },
    {
      "rel": "Parent",
      "href": "root"
    }
  ]
}
```



```
]
},
{
  "id": "15592",
  "name": "Pool",
  "parentID": "-1",
  "movedToBin": false,
  "objectType": "tcdirectory",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/directories/tcdirectories/15592"
    },
    {
      "rel": "Parent",
      "href": "root"
    }
  ]
}
]
```



## GET /v2/directories/tcdirectories?facets=children

### Summary

Gets information on all Thin Client Directories, recursively listing their children.

 This method will also list directories that are located in the **Recycle Bin** ( `"movedToBin": "true"` ).

### Resource URL

```
/v2/directories/tcdirectories?facets=children
```

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/directories/tcdirectories/facets=children
```

### Response Type

Returns a list of [Directories](#)([see page 103](#)), with their DirectoryChildren fields containing other objects.

### Example Response

```
[
{
  "id": "50705",
  "name": "New",
  "parentID": "-1",
  "movedToBin": true,
  "objectType": "tcdirectory",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/directories/tcdirectories/50705"
    },
    {
      "rel": "Parent",
      "href": "root"
    }
  ]
}
```



```
]
},
{
  "DirectoryChildren": [
    {
      "objectType": "tc",
      "id": "7117",
      "link": {
        "rel": "tc",
        "href": "https://172.30.91.227:8443/umsapi/v2/thinclients/7117"
      }
    },
    {
      "objectType": "tc",
      "id": "6888",
      "link": {
        "rel": "tc",
        "href": "https://172.30.91.227:8443/umsapi/v2/thinclients/6888"
      }
    },
    {
      "objectType": "tc",
      "id": "23028",
      "link": {
        "rel": "tc",
        "href": "https://172.30.91.227:8443/umsapi/v2/thinclients/23028"
      }
    }
  ],
  "id": "72327",
  "name": "Sub",
  "parentID": "15592",
  "movedToBin": false,
  "objectType": "tcdirectory",
```





```
"links": [  
  {  
    "rel": "self",  
    "href": "https://172.30.91.227:8443/umsapi/v2/directories/tcdirectories/72327"  
  },  
  {  
    "rel": "Parent",  
    "href": "https://172.30.91.227:8443/umsapi/v2/directories/tcdirectories/15592"  
  }  
],  
{  
  "DirectoryChildren": [  
    {  
      "objectType": "tcdirectory",  
      "id": "72327",  
      "link": {  
        "rel": "tcdirectory",  
        "href": "https://172.30.91.227:8443/umsapi/v2/directories/tcdirectories/72327"  
      }  
    },  
    {  
      "objectType": "tc",  
      "id": "7121",  
      "link": {  
        "rel": "tc",  
        "href": "https://172.30.91.227:8443/umsapi/v2/thinclients/7121"  
      }  
    },  
    {  
      "objectType": "tc",  
      "id": "6907",  
      "link": {  
        "rel": "tc",
```



```
"href": "https://172.30.91.227:8443/umsapi/v2/thinclients/6907"
```

```
}
```

```
},
```

```
{
```

```
[...]
```



## GET /v2/directories/tcdirectories/[id]

### Summary

Gets information on a specific Thin Client Directory.

 This method will also list directories that are located in the **Recycle Bin** ( "movedToBin": "true" ).

### Resource URL

/v2/directories/tcdirectories

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/directories/tcdirectories/15592
```

### Response Type

Returns a [Directory](#)(see page 103).

### Example Response

```
{
  "id": "15592",
  "name": "Pool",
  "parentID": "-1",
  "movedToBin": false,
  "objectType": "tcdirectory",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/directories/tcdirectories/15592"
    },
    {
      "rel": "Parent",
      "href": "root"
    }
  ]
}
```



}



## GET /v2/directories/tcdirectories/[id]?facets=children

### Summary

Gets information a Thin Client Directory, recursively listing its children.

 This method will also list directories that are located in the **Recycle Bin** ( "movedToBin": "true" ).

### Resource URL

```
/v2/directories/tcdirectories/[id]?facets=children
```

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/directories/tcdirectories/15592?facets=children
```

### Response Type

Returns a [Directory](#)(see page 103), with its DirectoryChildren field containing other objects.

### Example Response

```
{
  "DirectoryChildren": [
    {
      "objectType": "tcdirectory",
      "id": "72327",
      "link": {
        "rel": "tcdirectory",
        "href": "https://172.30.91.227:8443/umsapi/v2/directories/tcdirectories/72327"
      }
    },
    {
      "objectType": "tc",
      "id": "7121",
      "link": {
        "rel": "tc",
        "href": "https://172.30.91.227:8443/umsapi/v2/thinclients/7121"
      }
    }
  ]
}
```



```
}  
},  
{  
  "objectType": "tc",  
  "id": "6907",  
  "link": {  
    "rel": "tc",  
    "href": "https://172.30.91.227:8443/umsapi/v2/thinclients/6907"  
  }  
},  
{  
  "objectType": "tc",  
  "id": "11219",  
  "link": {  
    "rel": "tc",  
    "href": "https://172.30.91.227:8443/umsapi/v2/thinclients/11219"  
  }  
},  
{  
  "objectType": "tc",  
  "id": "15223",  
  "link": {  
    "rel": "tc",  
    "href": "https://172.30.91.227:8443/umsapi/v2/thinclients/15223"  
  }  
},  
{  
  "objectType": "tc",  
  "id": "15553",  
  "link": {  
    "rel": "tc",  
    "href": "https://172.30.91.227:8443/umsapi/v2/thinclients/15553"  
  }  
},
```



```
{
  "objectType": "tc",
  "id": "7113",
  "link": {
    "rel": "tc",
    "href": "https://172.30.91.227:8443/umsapi/v2/thinclients/7113"
  }
},
{
  "objectType": "tc",
  "id": "6899",
  "link": {
    "rel": "tc",
    "href": "https://172.30.91.227:8443/umsapi/v2/thinclients/6899"
  }
},
{
  "objectType": "tc",
  "id": "6912",
  "link": {
    "rel": "tc",
    "href": "https://172.30.91.227:8443/umsapi/v2/thinclients/6912"
  }
},
{
  "objectType": "tc",
  "id": "26836",
  "link": {
    "rel": "tc",
    "href": "https://172.30.91.227:8443/umsapi/v2/thinclients/26836"
  }
}
],
{id": "15592",
```



```
"name": "Pool",
"parentID": "-1",
"movedToBin": false,
"objectType": "tcdirectory",
"links": [
{
"rel": "self",
"href": "https://172.30.91.227:8443/umsapi/v2/directories/tcdirectories/15592"
},
{
"rel": "Parent",
"href": "root"
}
]
```





## PUT /v2/directories/tcdirectories/

### Summary

Creates a Thin Client Directory.

### Resource URL

```
/v2/directories/tcdirectories/
```

### Example Request

```
curl \
--request PUT \
--header "Content-type: application/json" \
--data '{"name":"Test"}' \
https://[server]:8443/umsapi/v2/directories/tcdirectories/
```

### Request Body

Name	Type	Mandatory	Description
name	String	no	directory name

### Response Type

Returns a success message.

### Example Response

```
{
  "message": "Directory successfully inserted.",
  "id": "72340",
  "name": "Test"
}
```



## PUT /v2/directories/tcdirectories/[id]

### Summary

Updates a Thin Client Directory name.

### Resource URL

```
/v2/directories/tcdirectories/[id]
```

### Example Request

```
curl \
--request PUT \
--header "Content-type: application/json" \
--data '{"name":"New Directory Name"}' \
https://[server]:8443/umsapi/v2/directories/tcdirectories/72340
```

### Request Path Variables

Name	Type	Mandatory	Description
id	String	yes	directory ID

### Request Body

Name	Type	Mandatory	Description
name	String	no	directory name

### Response Type

Returns a success message.

### Example Response

```
{
  "message": "Updated directory successfully."
}
```



## PUT /v2/directories/tcdirectories/[id]?operation=move

### Summary

Moves Thin Clients and Thin Client Directories into the specified Thin Client Directory.

### Resource URL

```
/v2/directories/tcdirectories/[id]?operation=move
```

### Example Request

```
curl \
--request PUT \
--header "Content-type: application/json" \
--data '[{"id": "76462", "type": "tcdirectory"}, {"id": "7121", "type": "tc"}]' \
https://[server]:8443/umsapi/v2/directories/tcdirectories/72340?operation=move
```

### Request Path Variables

Name	Type	Mandatory	Description
id	String	yes	Thin Client Directory ID

### Request Body

A list of [ApiObjects](#)(see page 96)

### Response Type

Returns a success message.

### Example Response

```
[
{
  "id": "7121",
  "results": "successful"
},
{
  "id": "76462",
  "results": "successful"
}
```




]



## DELETE /v2/directories/tcdirectories/[id]

### Summary

Deletes a Thin Client Directory.

 This deletes only empty directories. The attempt to delete a non-empty directory results in an error.

### Resource URL

```
/v2/directories/tcdirectories/[id]
```

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/directories/tcdirectories/72327
```

### Response Type

Returns a success message.

### Example Response

```
{
  "message": "Deletion successful."
}
```



## GET /v2/directories/tcdirectories/[id]/assignments/profiles

### Summary

Gets the profile and master profile assignments for the specified thin client directory, in order of their application.

### Resource URL

```
/v2/directories/tcdirectories/[id]/assignments/profiles
```

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/directories/tcdirectories/76462/assignments/
profiles
```

### Request Path Variables

Name	Type	Mandatory	Description
id	String	yes	Thin Client Directory ID

### Response Type

Returns a list of Profile and Master Profile [Assignments](#)(see page 97).

### Example Response

```
[
{
  "assignee": {
    "id": "35549",
    "type": "profile"
  },
  "receiver": {
    "id": "76462",
    "type": "tcdirectory"
  },
  "assignmentPosition": 0,
  "links": [
    {
```



```
"rel": "assigned",
"href": "https://172.30.91.227:8443/umsapi/v2/profiles/35549"
},
{
  "rel": "receiver",
  "href": "https://172.30.91.227:8443/umsapi/v2/directories/tcdirectories/76462"
},
{
  "rel": "self",
  "href": "https://172.30.91.227:8443/umsapi/v2/profiles/35549/assignments/
tcdirectories/76462"
}
]
},
{
  "assignee": {
    "id": "72098",
    "type": "masterprofile"
  },
  "receiver": {
    "id": "76462",
    "type": "tcdirectory"
  },
  "assignmentPosition": 1,
  "links": [
    {
      "rel": "assigned",
      "href": "https://172.30.91.227:8443/umsapi/v2/masterprofiles/72098"
    },
    {
      "rel": "receiver",
      "href": "https://172.30.91.227:8443/umsapi/v2/directories/tcdirectories/76462"
    },
    {
```



```
"rel": "self",  
"href": "https://172.30.91.227:8443/umsapi/v2/masterprofiles/72098/assignments/  
tcdirectories/76462"  
}  
]  
}  
]
```





## PUT /v2/directories/tcdirectories/[id]/assignments/profiles

### Summary

Creates profile and master profile assignments for the specified thin client directory.

### Resource URL

```
/v2/directories/tcdirectories/[id]/assignments/profiles
```

### Example Request

```
curl \
--request PUT \
--data '[{ "assignee": {
  "id": "35549",
  "type": "profile"
},
"receiver": {
  "id": "76462",
  "type": "tcdirectory"
}]' \
https://[server]:8443/umsapi/v2/directories/tcdirectories/76462/assignments/
profiles
```

### Request Path Variables

Name	Type	Mandatory	Description
id	String	yes	Thin Client Directory ID

### Request Body

A list of Profile and Master Profile [Assignments](#)(see page 97).

### Example Response

```
{
"message": "1 assignments successfully assigned to thinclient directory <76462>"
}
```



## Profile Directory

- [GET /v2/directories/profiledirectories](#)(see page 203)
- [GET /v2/directories/profiledirectories?facets=children](#)(see page 206)
- [GET /v2/directories/profiledirectories\[id\]](#)(see page 210)
- [GET /v2/directories/profiledirectories\[id\]?facets=children](#)(see page 212)
- [PUT /v2/directories/profiledirectories/](#)(see page 214)
- [PUT /v2/directories/profiledirectories/\[id\]](#)(see page 215)
- [PUT /v2/directories/tcdirectories/\[id\]?operation=move](#)(see page 216)
- [DELETE /v2/directories/profiledirectories/\[id\]](#)(see page 218)



## GET /v2/directories/profiledirectories

### Summary

Gets information on all Profile Directories in a flat format..

**i** This method will also list profile directories that are located in the **Recycle Bin** ( "movedToBin":"true" ).

### Resource URL

/v2/directories/profiledirectories

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/directories/profiledirectories
```

### Response Type

Returns a list of Profile Directories.

### Example Response

```
[
{
  "id": "72064",
  "name": "New Profile Directory",
  "parentID": "-2",
  "movedToBin": true,
  "objectType": "profiledirectory",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/directories/profiledirectories/72064"
    }
  ],
}
```



```
"rel": "Parent",
"href": "root"
},
{
  "id": "20451",
  "name": "Custom Partitions",
  "parentID": "-2",
  "movedToBin": false,
  "objectType": "profiledirectory",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/directories/profiledirectories/20451"
    },
    {
      "rel": "Parent",
      "href": "root"
    }
  ],
  "id": "20707",
  "name": "Standards",
  "parentID": "-2",
  "movedToBin": false,
  "objectType": "profiledirectory",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/directories/profiledirectories/20707"
    },
    {
```



```
{
  "rel": "Parent",
  "href": "root"
},
{
  "id": "20453",
  "name": "Benchmarks",
  "parentID": "-2",
  "movedToBin": false,
  "objectType": "profiledirectory",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/directories/profiledirectories/20453"
    }
  ],
  "rel": "Parent",
  "href": "root"
}
]
```



## GET /v2/directories/profiledirectories?facets=children

### Summary

Gets information on all Profile Directories, recursively listing their children.

**i** This method will also list profile directories that are located in the **Recycle Bin** ( "movedToBin": "true" ).

### Resource URL

/v2/directories/profiledirectories?facets=children

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/directories/profiledirectories/facets=children
```

### Response Type

Returns a list of Profile Directories, with their DirectoryChildren fields containing other objects.

### Example Response

```
[
{
  "DirectoryChildren": [
    {
      "objectType": "profile",
      "id": "20467",
      "link": {
        "rel": "profile",
        "href": "https://172.30.91.227:8443/umsapi/v2/profiles/20467"
      }
    },
    {
      "objectType": "profile",
      "id": "20452",
```



```
"link": {
  "rel": "profile",
  "href": "https://172.30.91.227:8443/umsapi/v2/profiles/20452"
},
{
  "objectType": "profile",
  "id": "20449",
  "link": {
    "rel": "profile",
    "href": "https://172.30.91.227:8443/umsapi/v2/profiles/20449"
  }
},
{
  "id": "20707",
  "name": "Standards",
  "parentID": "-2",
  "movedToBin": false,
  "objectType": "profiledirectory",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/directories/profiledirectories/20707"
    },
    {
      "rel": "Parent",
      "href": "root"
    }
  ],
  "DirectoryChildren": [
    {
```



```
"objectType": "profiledirectory",
"id": "72068",
"link": {
  "rel": "profiledirectory",
  "href": "https://172.30.91.227:8443/umsapi/v2/directories/profiledirectories/72068"
},
{
  "objectType": "profile",
  "id": "6592",
  "link": {
    "rel": "profile",
    "href": "https://172.30.91.227:8443/umsapi/v2/profiles/6592"
  },
  {
    "objectType": "profile",
    "id": "6585",
    "link": {
      "rel": "profile",
      "href": "https://172.30.91.227:8443/umsapi/v2/profiles/6585"
    },
    }
  ],
  "id": "20453",
  "name": "Benchmarks",
  "parentID": "-2",
  "movedToBin": false,
  "objectType": "profiledirectory",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/directories/profiledirectories/20453"
```






```
{
  },
  {
    "rel": "Parent",
    "href": "root"
  }
],
{
  "DirectoryChildren": [],
  "id": "72068",
  "name": "A Subdirectory",
  "parentID": "20453",
  "movedToBin": false,
  "objectType": "profiledirectory",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/directories/profiledirectories/72068"
    },
    {
      "rel": "Parent",
      "href": "https://172.30.91.227:8443/umsapi/v2/directories/profiledirectories/20453"
    }
  ]
}
```



## GET /v2/directories/profiledirectories[id]

### Summary

Gets information on a specific Profile Directory.

 This method will also show directories that are located in the **Recycle Bin** ( "movedToBin": "true" ).

### Resource URL

/v2/directories/profiledirectories

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/directories/profiledirectories/20707
```

### Response Type

Returns a Profile Directory.

### Example Response

```
{
  "id": "20707",
  "name": "Standards",
  "parentID": "-2",
  "movedToBin": false,
  "objectType": "profiledirectory",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/directories/profiledirectories/20707"
    },
    {
      "rel": "Parent",
      "href": "root"
    }
  ]
}
```




```
]
}
```



## GET /v2/directories/profiledirectories[id]?facets=children

### Summary

Gets information on a specific Profile Directory, recursively listing its children.

 This method will also show directories that are located in the **Recycle Bin** ( "movedToBin": "true" ).

### Resource URL

```
/v2/directories/profiledirectories[id]?facets=children
```

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/directories/profiledirectories/20707?
facets=children
```

### Response Type

Returns a Profile Directory, with its DirectoryChildren fields containing other objects

### Example Response

```
{
  "DirectoryChildren": [
    {
      "objectType": "profile",
      "id": "20467",
      "link": {
        "rel": "profile",
        "href": "https://172.30.91.227:8443/umsapi/v2/profiles/20467"
      }
    },
    {
      "objectType": "profile",
      "id": "20452",
      "link": {
```



```
"rel": "profile",
"href": "https://172.30.91.227:8443/umsapi/v2/profiles/20452"
},
{
  "objectType": "profile",
  "id": "20449",
  "link": {
    "rel": "profile",
    "href": "https://172.30.91.227:8443/umsapi/v2/profiles/20449"
  }
},
{
  "id": "20707",
  "name": "Standards",
  "parentID": "-2",
  "movedToBin": false,
  "objectType": "profiledirectory",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/directories/profiledirectories/20707"
    },
    {
      "rel": "Parent",
      "href": "root"
    }
  ]
}
```



## PUT /v2/directories/profiledirectories/

### Summary

Creates a Profile Directory.

### Resource URL

```
/v2/directories/profiledirectories/
```

### Example Request

```
curl \
--request PUT \
--header "Content-type: application/json" \
--data '{"name":"New Profiles"}' \
https://[server]:8443/umsapi/v2/directories/profiledirectories/
```

### Request Body

Name	Type	Mandatory	Description
name	String	no	directory name

### Response Type

Returns a success message.

### Example Response

```
{
  "message": "Directory successfully inserted.",
  "id": "72090",
  "name": "New Profiles"
}
```



## PUT /v2/directories/profiledirectories/[id]

### Summary

Updates a Profile Directory name.

### Resource URL

```
/v2/directories/profiledirectories/[id]
```

### Example Request

```
curl \
--request PUT \
--header "Content-type: application/json" \
--data '{"name":"New Directory Name"}' \
https://[server]:8443/umsapi/v2/directories/profiledirectories/72340
```

### Request Path Variables

Name	Type	Mandatory	Description
id	String	yes	directory ID

### Request Body

Name	Type	Mandatory	Description
name	String	no	directory name

### Response Type

Returns a success message.

### Example Response

```
{
  "message": "Updated directory successfully."
}
```



## PUT /v2/directories/tcdirectories/[id]?operation=move

### Summary

Moves Profiles and Profile Directories into the specified Profile Directory.

### Resource URL

```
/v2/directories/profiledirectories/[id]?operation=move
```

### Example Request

```
curl \
--request PUT \
--header "Content-type: application/json" \
--data ' [{"id": "20451", "type": "profiledirectory"},
{"id": "6592", "type": "profile"} ] ' \
https://[server]:8443/umsapi/v2/directories/profiledirectories/72068?
operation=move
```

### Request Path Variables

Name	Type	Mandatory	Description
id	String	yes	directory ID

### Request Body

A list of [ApiObjects](#)(see page 96)

### Response Type

Returns a success message.

### Example Response

```
[
{
  "id": "20451",
  "results": "successful"
},
{
  "id": "6592",
```





```
"results": "successful"
```

```
}
```


```
]
```



## DELETE /v2/directories/profiledirectories/[id]

### Summary

Deletes a Profile Directory.

 This deletes only empty directories. The attempt to delete a non-empty directory results in an error.

### Resource URL

/v2/directories/profiledirectories/[id]

### Example Request

```
curl \
--request DELETE\
https://[server]:8443/umsapi/v2/directories/profiledirectories/72327
```

### Response Type

Returns a success message.

### Example Response

```
{
  "message": "Deletion successful."
}
```



## Master Profile Directory

- [GET /v2/directories/masterprofiledirectories](#)(see page 220)
- [GET /v2/directories/masterprofiledirectories?facets=children](#)(see page 223)
- [GET /v2/directories/masterprofiledirectories\[id\]](#)(see page 226)
- [GET /v2/directories/masterprofiledirectories\[id\]?facets=children](#)(see page 228)
- [PUT /v2/directories/masterprofiledirectories/](#)(see page 230)
- [PUT /v2/directories/masterprofiledirectories/\[id\]](#)(see page 231)
- [PUT /v2/directories/tcdirectories/\[id\]?operation=move](#)(see page 232)
- [DELETE /v2/directories/masterprofiledirectories\[id\]](#)(see page 234)



## GET /v2/directories/masterprofiledirectories

### Summary

Gets information on all Master Profile Directories in a flat format.

**i** This method will also list master profile directories that are located in the **Recycle Bin** ( "movedToBin": "true" ).

### Resource URL

```
/v2/directories/masterprofiledirectories
```

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/directories/masterprofiledirectories
```

### Response Type

Returns a list of Master Profile Directories.

### Example Response

```
[
{
  "id": "72101",
  "name": "Basic",
  "parentID": "-14",
  "movedToBin": false,
  "objectType": "masterprofiledirectory",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/directories/masterprofiledirectories/72101"
    },
    {
      "rel": "Parent",
```



```
"href": "root"
}
],
{
  "id": "72100",
  "name": "Important",
  "parentID": "76552",
  "movedToBin": false,
  "objectType": "masterprofiledirectory",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/directories/
masterprofiledirectories/72100"
    },
    {
      "rel": "Parent",
      "href": "https://172.30.91.227:8443/umsapi/v2/directories/
masterprofiledirectories/76552"
    }
  ],
},
{
  "id": "76552",
  "name": "New Directory",
  "parentID": "-14",
  "movedToBin": false,
  "objectType": "masterprofiledirectory",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/directories/
masterprofiledirectories/76552"
    },
  ],
}
```



```
{
  "rel": "Parent",
  "href": "root"
},
{
  "id": "76556",
  "name": "Old Directory",
  "parentID": "-14",
  "movedToBin": true,
  "objectType": "masterprofiledirectory",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/directories/masterprofiledirectories/76556"
    }
  ],
  "rel": "Parent",
  "href": "root"
}
]
```



## GET /v2/directories/masterprofiledirectories?facets=children

### Summary

Gets information on all Master Profile Directories, recursively listing their children.

**i** This method will also list master profile directories that are located in the **Recycle Bin** ( "movedToBin": "true" ).

### Resource URL

/v2/directories/masterprofiledirectories?facets=children

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/directories/masterprofiledirectories/
facets=children
```

### Response Type

Returns a list of Master Profile Directories, with their DirectoryChildren fields containing other objects.

### Example Response

```
[
{
  "DirectoryChildren": [
    {
      "objectType": "masterprofile",
      "id": "72098",
      "link": {
        "rel": "masterprofile",
        "href": "https://172.30.91.227:8443/umsapi/v2/masterprofiles/72098"
      }
    }
  ],
  "id": "72101",
```



```
"name": "Basic",
"parentID": "-14",
"movedToBin": false,
"objectType": "masterprofiledirectory",
"links": [
{
"rel": "self",
"href": "https://172.30.91.227:8443/umsapi/v2/directories/
masterprofiledirectories/72101"
},
{
"rel": "Parent",
"href": "root"
}
],
{
"DirectoryChildren": [
{
"objectType": "masterprofile",
"id": "72096",
"link": {
"rel": "masterprofile",
"href": "https://172.30.91.227:8443/umsapi/v2/masterprofiles/72096"
}
}
],
"id": "72100",
"name": "Important",
"parentID": "-14",
"movedToBin": false,
"objectType": "masterprofiledirectory",
"links": [
{
```





```
"rel": "self",  
  "href": "https://172.30.91.227:8443/umsapi/v2/directories/  
masterprofiledirectories/72100"  
},  
{  
  "rel": "Parent",  
  "href": "root"  
}  
]  
}
```



## GET /v2/directories/masterprofiledirectories[id]

### Summary

Gets information on a specific Master Profile Directory.

**i** This method will also show master profile directories that are located in the **Recycle Bin** ( "movedToBin": "true" ).

### Resource URL

/v2/directories/masterprofiledirectories[id]

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/directories/masterprofiledirectories/20707
```

### Response Type

Returns a Master Profile Directory.

### Example Response

```
{
  "id": "72101",
  "name": "Basic",
  "parentID": "-14",
  "movedToBin": false,
  "objectType": "masterprofiledirectory",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/directories/masterprofiledirectories/72101"
    },
    {
      "rel": "Parent",
```



```
"href": "root"  
}  
]  
}
```



## GET /v2/directories/masterprofiledirectories[id]?facets=children

### Summary

Gets information on a specific Master Profile Directory, recursively listing its children.

**i** This method will also show master profile directories that are located in the **Recycle Bin** ( "movedToBin":"true" ).

### Resource URL

```
/v2/directories/masterprofiledirectories[id]?facets=children
```

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/directories/masterprofiledirectories/72101?
facets=children
```

### Response Type

Returns a Master Profile Directory, with its DirectoryChildren fields containing other objects

### Example Response

```
{
  "DirectoryChildren": [
    {
      "objectType": "masterprofiledirectory",
      "id": "72112",
      "link": {
        "rel": "masterprofiledirectory",
        "href": "https://172.30.91.227:8443/umsapi/v2/directories/
masterprofiledirectories/72112"
      }
    },
    {
      "objectType": "masterprofile",
```



```
"id": "72098",
"link": {
  "rel": "masterprofile",
  "href": "https://172.30.91.227:8443/umsapi/v2/masterprofiles/72098"
}
},
{
  "id": "72101",
  "name": "Basic",
  "parentID": "-14",
  "movedToBin": false,
  "objectType": "masterprofiledirectory",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/directories/
masterprofiledirectories/72101"
    },
    {
      "rel": "Parent",
      "href": "root"
    }
  ]
}
```



## PUT /v2/directories/masterprofiledirectories/

### Summary

Creates a Master Profile Directory.

### Resource URL

```
/v2/directories/masterprofiledirectories/
```

### Example Request

```
curl \
--request PUT \
--header "Content-type: application/json" \
--data '{"name":"Yet Another Directory"}' \
https://[server]:8443/umsapi/v2/directories/masterprofiledirectories/
```

### Request Body

Name	Type	Mandatory	Description
name	String	yes	Master Profile directory name

### Response Type

Returns a success message.

### Example Response

```
{
  "message": "Directory successfully inserted.",
  "id": "76560",
  "name": "Yet Another Directory"
}
```



## PUT /v2/directories/masterprofiledirectories/[id]

### Summary

Renames a Master Profile Directory.

### Resource URL

```
/v2/directories/masterprofiledirectories/[id]
```

### Example Request

```
curl \
--request PUT \
--header "Content-type: application/json" \
--data '{"name":"New Master Profiles"}' \
https://[server]:8443/umsapi/v2/directories/masterprofiledirectories/76560
```

### Request Body

Name	Type	Mandatory	Description
name	String	yes	Master directory name

### Response Type

Returns a success message.

### Example Response

```
{
  "message": "Updated directory successfully."
}
```



## PUT /v2/directories/tcdirectories/[id]?operation=move

### Summary

Moves Master Profiles and Master Profile Directories into the specified Master Profile Directory.

### Resource URL

```
/v2/directories/masterprofiledirectories/[id]?operation=move
```

### Example Request

```
curl \
--request PUT \
--header "Content-type: application/json" \
--data '["{"id": "72100", "type": "masterprofiledirectory"}, {"id": "72098", "type": "masterprofile"}]' \
https://[server]:8443/umsapi/v2/directories/masterprofiledirectories/76552?operation=move
```

### Request Path Variables

Name	Type	Mandatory	Description
id	String	yes	directory ID

### Request Body

A list of [ApiObjects](#)(see page 96)

### Response Type

Returns a success message.

### Example Response

```
[
{
  "id": "72098",
  "results": "successful"
},
{
  "id": "72100",
  "results": "successful"
}
```






```
}  
]
```



## DELETE /v2/directories/masterprofiledirectories[id]

### Summary

Deletes a Master Profile Directory.

 This deletes only empty directories. The attempt to delete a non-empty directory results in an error.

### Resource URL

/v2/directories/masterprofiledirectories/[id]

### Example Request

```
curl \
--request DELETE\
https://[server]:8443/umsapi/v2/directories/masterprofiledirectories/76559
```

### Response Type

Returns a success message.

### Example Response

```
{
  "message": "Deletion successful."
}
```



## Firmware

- [GET /v2/firmwares](#)(see page 236)
- [GET /v2/firmwares/\[id\]](#)(see page 238)



## GET /v2/firmwares

### Summary

Gets information on all Firmwares known to the UMS.

### Resource URL

```
/v2/firmwares/
```

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/firmwares/
```

### Response Type

Returns a list of Firmwares.

### Example Response

```
{
  "FwResource": [
    {
      "id": "2",
      "product": "IGEL Universal Desktop LX",
      "version": "5.06.100.01",
      "firmwareType": "LX",
      "links": [
        {
          "rel": "self",
          "href": "https://172.30.91.227:8443/umsapi/v2/firmwares/2"
        }
      ]
    },
    {
      "id": "4",
      "product": "IGEL Universal Desktop LX",
```



```
"version": "4.14.120.01",  
"firmwareType": "LX",  
"links": [  
  {  
    "rel": "self",  
    "href": "https://172.30.91.227:8443/umsapi/v2/firmwares/4"  
  }  
]  
}
```



## GET /v2/firmwares/[id]

### Summary

Gets information on a specific Firmware.

### Resource URL

```
/v2/firmwares/[id]
```

### Example Request

```
curl \
--request GET \
https://[server]:8443/umsapi/v2/firmwares/45
```

### Response Type

Returns a Firmware.

### Example Response

```
{
  "id": "45",
  "product": "IGEL Universal Desktop LX",
  "version": "5.09.100.01",
  "firmwareType": "LX",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/firmwares/45"
    }
  ]
}
```



## Server Status

### Summary

Gets status information about the UMS server instance.

 Server Status is the only IMI resource that can be queried without logging in.

### Resource URL

```
/v2/serverstatus
```

### Example Request

```
curl \
--request GET\
https://[server]:8443/umsapi/v2/serverstatus
```

### Example Response

```
{
  "rmGuiServerVersion": "5.02.100.rc4",
  "buildNumber": "26596",
  "activeMQVersion": "5.6.0",
  "derbyVersion": "10.12.1.1",
  "serverUUID": "d5210098-091b-464d-b24d-9f989450931d",
  "server": "PM-MH-WIN7-UMS.IGEL.LOCAL:8443",
  "links": [
    {
      "rel": "self",
      "href": "https://172.30.91.227:8443/umsapi/v2/serverstatus"
    }
  ]
}
```



## Error Codes

- [HTTP Status Codes](#)(see page 241)
- [API Error Messages](#)(see page 242)





## HTTP Status Codes

### Client-Side Errors

HTTP Status	Title	Description
200	OK	The request was successful. However, it is a good idea to read the API message.
400	Bad Request	The request does not match the API, e.g. it uses the wrong HTTP method
401	Unauthorized	The client has not logged in or has sent the wrong credentials.
404	Not Found	The endpoint does not exist, it may be misspelled.
415	Unsupported Media Type	The body content, e.g. JSON, does not match the Content-Type header or is not well-formed.

### Server-Side Errors

HTTP Status	Title	Description
500	Internal Server Error	The server has encountered an error, check the server logfiles <code>catalina.log</code> and <code>stderr</code>



## API Error Messages

The API server will send a message in the response body, should an error occur:

```
{
  "message": "Request method 'GET' not supported",
  "errorCode": "ITG-0A011",
  "stackTrace": "[org.springframework.web.servlet.mvc.m ...
  [...]]
}
```

Error Code	Message
Client Errors	
ITG-0A001	Thin client with ID [id] not found.
ITG-0A002	No target directory with ID [id] found.
ITG-0A003	[command] is an invalid command.
ITG-0A004	Invalid request body
ITG-0A005	"Could not read JSON: [...]"
ITG-0A006	Directory with ID [id] is not empty. Remove content first.
ITG-0A007	No MAC defined.
ITG-0A008	[...] The statement was aborted because it would have caused a duplicate key value [...]
ITG-0A009	No firmware defined.
ITG-0A010	Number format error
ITG-0A011	Request method [method] not supported.
ITG-0A012	IDList is empty.
ITG-0A013	No valid property in body
ITG-0A014	MAC must be 12 characters long.
ITG-0A015	Firmware with ID [id] does not exist.
ITG-0A016	Parent with ID [id] does not exist.
ITG-0A017	Target directory is in Recycling Bin. In order to use it revert it first.



ITG-0A018	Thin client is in Recycling Bin. In order to use it, revert it first.
ITG-0A019	Invalid assignment receiver
ITG-0A020	No handler is mapped for this request.
ITG-0A021	Invalid assignment
ITG-0A022	Profile is in Recycling Bin.
Authentication Errors	
ITG-0B001	No authentication header found in request.
ITG-0B002	Invalid login credentials.
ITG-0B004	No session ID found.
ITG-0B004	No session found for requested session ID [id].
ITG-0B005	Access denied.
ITG-0B006	No valid IMI API version.
ITG-0B007	No user found in session with sessionID
Licensing Errors	
ITG-0C001	License required
ITG-0C002	Invalid IMI version
Server Errors	
ITG-FF001	Internal error
ITG-FF002	Database error
ITG-FF003	Cache error
ITG-FF004	Server not available
ITG-FF005	Object mapping error
ITG-FF006	Invalid object type



## IMI How-Tos

- [Using Secure VNC via IGEL Management Interface \(IMI\)](#)(see page 245)



## Using Secure VNC via IGEL Management Interface (IMI)

This document describes making SSL/TLS-secured VNC connections to *IGEL* Linux thin clients using the credentials provided by *IGEL Management Interface (IMI)* Version 2 .

### Prerequisites

- *Universal Management Suite (UMS)* 5.02.100
- *IMI* maintenance subscription covering version 2
- *IGEL* LX thin clients with *IGEL* Linux version 5.03.190 or newer

Apart from that, you will have to implement a [custom VNC proxy](#)([see page 247](#)).



## Thin Client Configuration

In the thin clients' configuration, either locally, via UMS or in a profile, do the following:

1. Go to **System > Remote Access > Shadow**
2. Enable **Allow Remote Shadowing**
3. Optional: Enable **Prompt User to allow Remote Session**

⚠ In a number of countries, unannounced shadowing is prohibited by law. Do not disable this option if you are in one of these countries!

4. Optional: Enable **Allow Input from Remote**
5. Enable **Secure Mode**

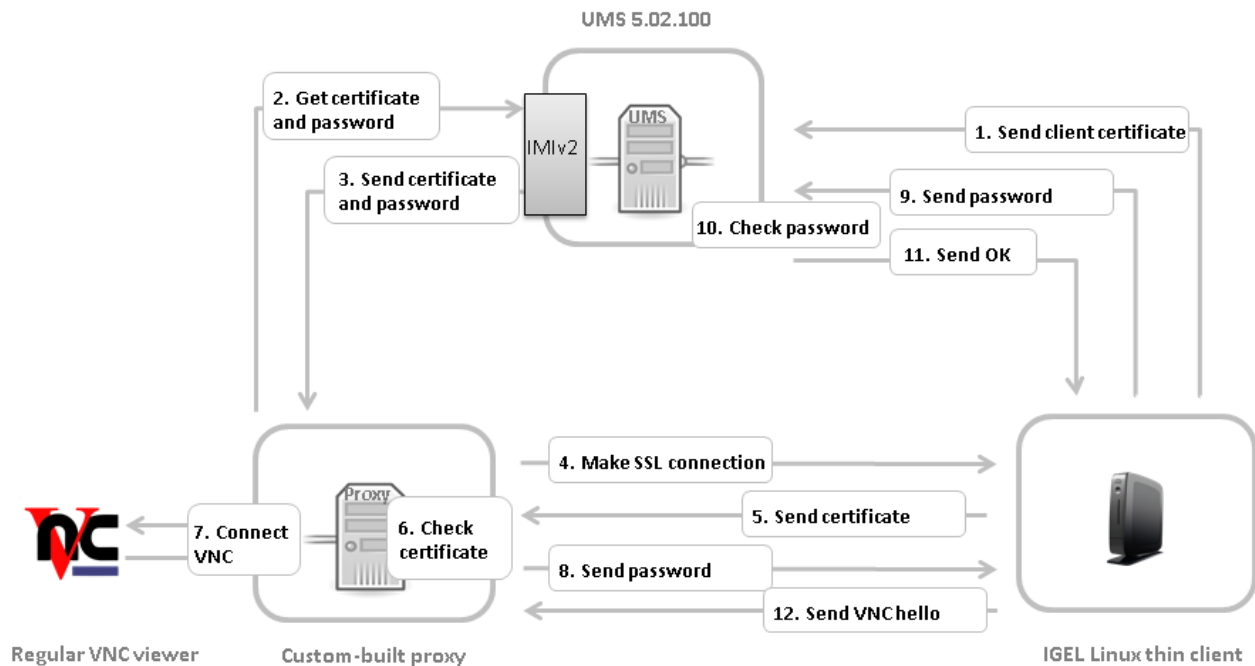
ℹ Alternatively, you can globally activate Secure VNC for all thin clients under **UMS Administration > Remote Access > Secure VNC**.

6. Optional: Enable **Allow User to disconnect Remote Shadowing**



## Secure VNC with IMIv2

This section describes the Secure VNC procedure, parts of which your custom VNC proxy has to implement.



1. The thin client sends its certificate to UMS on boot.

- i** For a thin client newly registered to the UMS, either
- reboot the thin client once or
  - use UMS Console to send **Settings UMS > TC** and send **Settings TC > UMS** once

2. Via the IMIv2 REST API, the custom proxy asks UMS for the credentials necessary for shadowing the thin client:  

```
GET /umsapi/v2/thinclients/[thin client ID]?facets=shadow
```
3. The custom proxy receives the credentials:
  - the thin client certificate in Base64 encoding, or an empty string if UMS has not received a certificate
  - a one-time-password in the form of a Java UUID for logging in within the next 5 min.
4. The custom proxy initiates a SSL connection to the thin client's TCP port 5900.
5. The thin client sends its certificate.
6. The custom proxy checks the certificate against the one received from UMS and decides whether to accept it and proceed with the connection.
7. The custom proxy opens a TCP server socket for an external VNC viewer to connect to. The custom proxy must connect the network streams involved as follows:
  - write to the VNC viewer what it reads from the SSL connection with the thin client
  - write to the SSL connection with the thin client what it reads from the VNC viewer



Make the external VNC viewer connect to the custom proxy's server socket.

8. On accepting the connection from the external VNC viewer, the custom proxy writes the String `PROCYCMD PW_[one-time password]` to the SSL connection with the thin client.
9. The thin client sends the password to UMS.
10. UMS checks the password.
11. If the password is correct, UMS tells the thin client to proceed. Otherwise the connection will be closed.
12. The thin client sends a string like `RFB 003.008\n` as a VNC hello which initiates the VNC session with the external VNC viewer.