

# Columbus REST

User Manual

Product version 7.6.3

# columbus

Issue: 08.2019

© brainwaregroup - 1997-2019 - All rights reserved.

Every documentation provided by the brainwaregroup is subject to copyright and owned by the brainwaregroup. The brainwaregroup does not guarantee nor accepts the legal responsibility or any liability whatsoever for the usage of this information, for their economic feasibility or error-free function for a certain purpose.

In the compilation of this document, every effort has been undertaken to ensure the correctness of the content. However, the brainwaregroup does not offer any guarantee related to this documentation nor does it offer a legal warranty for the marketable quality and suitability for a certain purpose. Furthermore, the brainwaregroup cannot be held liable for errors or unintended damages or consequential damages in relation with the provision, performance or usage of this document or the examples contained therein. The brainwaregroup reserves its right to change this documentation anytime without prior notice.

All names, company names or companies used in this document are fictitious and do not refer, neither in name nor content, to actually existing names, organizations, legal persons or institutions nor shall they represent them. Any similarity to existing people, organizations, legal persons or institutions is merely coincidental.

<b>Document title</b>	Columbus REST - User Manual
<b>Product version</b>	7.6.3
<b>Production and printing</b>	Brainware Consulting & Development AG Sumpfstrasse 15 CH-6312 Steinhausen <a href="http://www.brainwaregroup.com">www.brainwaregroup.com</a>
<b>Release date</b>	12.19

Neither the whole document nor parts of it may be copied, photocopied, reproduced or processed without prior written approval of the brainwaregroup.

## Additional Copyright

For more Information, please read the Documentation "Tec\_UserManual\_Col\_7\_ComponentLicenses.pdf".

# Contents

---

<b>1</b>	<b>Allgemeines</b>	<b>Error! Bookmark not defined.</b>
1.1	Licensing .....	5
1.2	Typographical conventions .....	6
1.3	Help .....	6
1.3.1	Support .....	6
1.3.2	Documentation .....	7
1.3.3	Training information .....	7
<b>2</b>	<b>Overview</b>	<b>8</b>
2.1	What is REST? .....	8
2.2	Infrastructure components .....	8
2.3	Configuration .....	8
2.4	Authentication .....	9
2.5	Resources .....	9
2.6	Methods .....	10
2.7	Access REST .....	10
<b>3</b>	<b>REST API</b>	<b>12</b>
3.1	General .....	12
3.1.1	Format .....	12
3.1.2	Lists .....	12
3.1.3	Response codes .....	13
3.2	devices .....	13
3.2.1	GET /{oid} .....	13
3.2.2	POST / .....	14
3.2.3	PUT /oid .....	15
3.2.4	DELETE /oid .....	15
3.2.5	POST /oid/actions .....	16
3.2.6	GET /oid/isActionPending/actionName .....	17
3.2.7	POST /oid/operatingSystemTemplates .....	18
3.2.8	POST /oid/softwareAssignments .....	20
3.2.9	DELETE /oid/softwareAssignments/packages/packageId .....	21
3.2.10	DELETE /oid/softwareAssignments/groups/groupId .....	23
3.2.11	POST /oid/variables .....	24
3.2.12	DELETE /oid/variables/variable .....	25
3.3	users .....	25
3.3.1	GET /{oid} .....	26
3.3.2	POST / .....	27
3.3.3	PUT /oid .....	28
3.3.4	DELETE /oid .....	28
3.3.5	POST /oid/variables .....	33
3.3.6	DELETE /oid/variables .....	34
3.3.7	POST /oid/softwareAssignments .....	29
3.3.8	DELETE /oid/softwareAssignments/packages/packageId .....	30
3.3.9	DELETE /oid/softwareAssignments/groups/groupId .....	32
3.4	actions .....	33
3.4.1	PUT /{actionid}/reschedule .....	35
3.5	companies .....	35
3.5.1	GET /{oid} .....	35
3.5.2	GET /oid/softwareInfo .....	36
3.5.3	GET /oid/operatingSystemTemplates .....	36
3.6	treeNodes .....	37
3.6.1	GET /{oid} .....	37

	3.6.2 POST .....	37
3.7	deviceClasses.....	38
	3.7.1 GET / .....	38

# 1 Preface

---

Thank you very much for choosing brainwaregroup and Columbus.

Columbus and Spider provide you with powerful tools for software and lifecycle management.

This manual will support you in using the manifold possibilities of Columbus in the best possible way for your company.

## 1.1 Licensing

---

Relevant for licensing is the number of managed assets based on the valid price list at the time of purchasing the license.

Managed assets are computer systems, e.g. laptops, desktops, servers, virtual servers, thin clients, mobile devices, tablets, every kind of asset for which software can be managed or for which another active management is technically available.

Such an active management can be e.g. the installation of the operating system, the inventorization or the distribution of software (no definitive list). The service desired by the customer determines the suites or modules to be licensed.

A license is basically bound to one computer, regardless whether the license is used only once or continuously.

Apart from computers, Columbus is also able to collect data from additional network devices (routers, switches, printers, etc.) which are administered as so-called *Not Managed Devices*. These devices require no Columbus license as long as they have the status *Not Managed*.

Columbus has an active license check for all Columbus modules. License limit violations are signaled both on the Console as well as in the log file.

### Summary

The brainwaregroup licensing rules comprise the following:

- Each asset operated by Columbus requires a license.
- The transfer of a license to another computer is limited to replacement purchases.

In case of a license violation

- you will be notified at the start or during your work with the Console,
- entries are generated in the log files.

<b>Important</b>	If the number of managed assets increases, the customer shall bind himself/herself to obtain a new license for the additional ones within 30 days. The Software can restrict or stop the operation after an under-licensing period of 30 days.
------------------	--

A license comprises the temporally unlimited, non-exclusive usage right for the current version of the Software including all hot fixes within one year from the invoice date.

## 1.2 Typographical conventions

This manual uses various formats to highlight certain terms and actions. Specific notes and tips are shown with a different background color, according to their importance.

Format	Description
<b>Bold font</b>	Elements in the software or in the operating system, such as menu items, buttons or elements of a selection list
<i>Italic font</i>	Emphases (important details) and links to other chapters or documents
Triangle symbol ">"	Instruction step
Angle bracket ">"	Command menu sequences, e.g. <b>File &gt; Open</b>
<i>System font</i>	Directories, code and script samples
CAPITAL LETTERS	Key names, e.g. SHIFT, CTRL, or ALT
KEY+KEY	Key combinations, i.e. the user has to hold one key and press another simultaneously, e.g. CTRL+P or ALT+F4.

**Note** Used for notes or tips which facilitate the work or for additional information which enhances understanding for the product.

**Important** Information which should be observed by the user, because otherwise problems or additional work may be caused in operation.

**Attention** Information which should be observed by the user in order to prevent malfunctions of the system (crashes, data loss, system failure).

## 1.3 Help

Please visit our [Website](#) for additional information and support. Here you can find all documents, a [Knowledgebase](#) as well as a support form if you have questions for our specialists.

### 1.3.1 Support

The support page features a link to the [KnowledgeBase](#), the support form and information about our customer service.

- Open [www.brainwaregroup.com](http://www.brainwaregroup.com) in your browser and click on **Support and KnowledgeBase** on the right side.
- Start your search in the **KnowledgeBase** where you can find a list of the most common support cases

If nothing matches your search criteria, you may use the support form for advanced information. Please provide a detailed description of your problem so that our specialists can help you as quick as possible.

- Click on Support Form.
- Please fill in at least the fields marked with an asterisk (\*) and click **Send Question**.
  - ▮ You will receive a confirmation that your question has been received in the system.

The support page contains the telephone numbers necessary for telephone contacts.

## 1.3.2 Documentation

---

A separate download area is available on our web site for the currently available documentation. This area contains all manuals in PDF and CHM format in German and English language. At present, some of the technical manuals are only available in English.

- Open <http://www.brainwaregroup.com> in your browser and click on **Downloads + Licenses** on the right side.
- Click on **Login**, enter your **User name** and **Password** and then click on **Register**.
- Please select **Documentation** as download type, then select the desired product and click on **Search**
  - ▮ An overview of all available documents will be displayed.
- Click on the desired format and language to download the PDF/CHM file in the desired form.

You will find the Acrobat Reader on our support page or directly under [www.adobe.com](http://www.adobe.com).

---

<b>Hinweis</b>	During the installation the documents are copied by default to the directory named [C:\Program Files]\Columbus. In order to display the Console help in German you will have to rename the file CMC_de.chm in the \Console directory to CMC.chm.
----------------	--

---

## 1.3.3 Training information

---

brainwaregroup is taking all efforts to ensure that our products and solutions will be used and operated in an appropriate, qualified and proper way. Therefore, we are offering various training courses. You can find all information about training types and dates under the **Training** <https://www.brainwaregroup.com/de/support/trainings/> menu point

## 2 Overview

### 2.1 What is REST?

«REST» means «Representational State Transfer» and describes the binding transition from one status to the next. It is used for the machine-to-machine communication. It allows to request, add, edit or delete information using a pre-defined interface. This is done using simple http requests, which can nowadays be made from any browser.

In Columbus, this interface is used to request, create or delete information about the computer or the user in the database. This also applies for software assignments, actions and variables.

### 2.2 Infrastructure components

The REST interface is mapped in Columbus by the «REST» agents and only requires a OTB server as information supplier. This can be assigned as required in the agent's configuration. For this purpose, we recommend to use the Master-Server in order to obtain the best performance.

Company	Agent /	Last contact	Last status
✓ brainwaregroup	Preboot Services	14.03.2017 14:50:15	
✓ brainwaregroup	REST	14.03.2017 14:31:15	14.03.2017 14:30:47: Server ready to use on port 8080
✓ brainwaregroup	Replication		
✓ brainwaregroup	Reporting and Cleanup		
✓ brainwaregroup	SW Deployment		

### 2.3 Configuration

The REST interface is configured on the Infrastructure tab. Select the required server that shall provide the information on the «Infrastructure» screen. Then, select «Configure» in the ribbon to open the following dialog.

**REST Server**  
Configuration settings for the REST Server

Communication:

Port on which the REST server will operate (plain http)

Port on which the REST server will operate (with TLS: https)

OTB Server and port which will be used by the REST server

The authentication settings for REST are globally defined (see System -> Settings -> REST or click [here](#)).

OK Cancel

Option	Description
Port	Select the port on which the service shall be provided.



OTB server	Select the responsible OTB server that shall supply the data for the requests.
------------	--

## 2.4 Authentication

A HTTP Basic Authentication is used for authentication. If you wish to protect your interface from unauthorized access, you can define a user and set a password. Please observe that this login data must be entered for every access. For further information, please refer to: <https://de.wikipedia.org/wiki/HTTP-Authentifizierung>

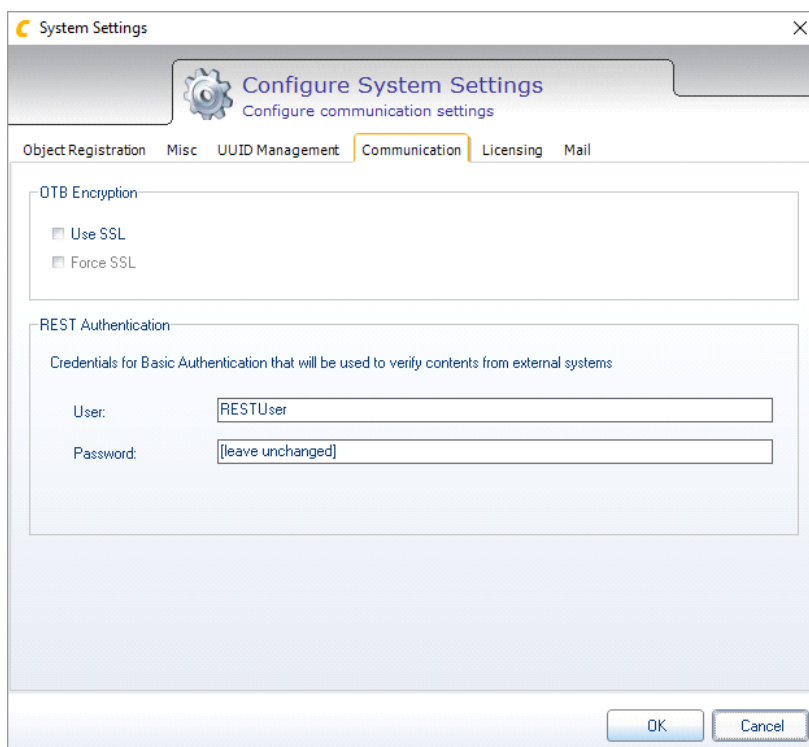
As an option, the interface can also be accessed through HTTPS, which makes interception more difficult.

The authentication of the REST interface is globally defined and therefore it is accessed through the «System configuration». For access, proceed as described below.

On the configuration of the REST agents, click on the link in the lower part of the dialog or select the «System» tab in the «Ribbon» and then the «Settings» button.

In the «Communication» tab, you can then define a user and a password for the REST interface under «REST Authentication».

**Important** This is a global setting and is valid for all REST agents.



## 2.5 Resources

A REST interface is divided into so-called «Resources» and gives the relevant information according to the request.

Currently, the following resources are defined:

Resource	Description
----------	-------------

devices	Provides the complete information on the computer and its inventory information, software assignments, actions, etc.
users	Provides the complete information on the user and their inventory information, software assignments, actions, etc.
companies	Provides the software and OS assigned to the company.
treeNodes	Provides information such as software or OS assignments on the relevant organizational unit.

## Helper resources

The following resources supply information that is not directly related to the objects or the organizational units (e.g. tree nodes).

Resource	Description
deviceClasses	Provides the available device classification.

## 2.6 Methods

There are basically 5 different methods to access the REST resources. Probably the most known is GET. It is mostly used to call up a website (e.g. [www.brainwaregroup.com](http://www.brainwaregroup.com)). With HTTP/REST there are additional methods to directly affect the same resource or URI and to control it accordingly.

Method	Description
GET	Gets information; e.g. from a device <a href="http://columbus:8084/latest/device/H3EEGkzBM9">http://columbus:8084/latest/device/H3EEGkzBM9</a>
POST	Calls up the same URI, but provides information in the invisible body. Used to create new data records (e.g. computer).
PUT	Calls up the same URI, but provides information in the invisible body. Used to edit or enhance existing data records (e.g. cost center of an object).
PATCH	The same as PUT, but contains several modifications on different objects.
DELETE	Calls up the same URI, but indicates that a data record must be deleted.

## 2.7 Access REST

Basically any application that can use HTTPS can access a REST interface. This is carried out by calling up an URI and the relevant resource. Additionally, filter parameters can be defined for restriction in order to obtain the required result.

Structure of the URI

```
http://server:port/version/resource
```

For Columbus 7.5 / 7.6, the version must be given the value «1». Alternatively, “latest” will redirect always to the latest version. E.G:

```
http://server:port/latest/ressource
```

### Examples

In the example below, on the server «columbus», port 8084, the resource «devices» is called up.

```
http://columbus:8084/1/devices
```

The above mentioned request returns all devices that are located in the database. The search can also be restricted, e.g. on the server:

```
http://columbus:8084/1/devices?filter=name:srv*
```

or domain

```
http://columbus:8084/1/devices?filter=fqdn:acme.inc
```

Example with CURL and active authentication

```
curl.exe -u ColumbusRest:Brainware123 -H "Accept:application/json" -X GET "http://columbus:8084/1/devices"
```

Below, the URIs are only shown in the abbreviated form, without indicating the server and version.

```
http://columbus:8084/latest/devices
```

will become

```
/devices
```

## 3 REST API

---

### 3.1 General

---

#### 3.1.1 Format

---

The output can be formatted using the parameter `pretty`.

```
/devices?pretty=true
```

The output ...

```
{"total":0,"maxResults":0,"startAt":0,"items":[]}
```

... will then be shown in a legible format:

```
{
  "total": 0,
  "maxResults": 0,
  "startAt": 0,
  "items": []
}
```

#### 3.1.2 Lists

---

Use the following parameters to restrict the output. This currently applies to the following resources:

- Devices
- Users

Name	Description	Type	Default value
startAt	Start of the first element (0 is the first element)	Integer	0
maxResults	Maximum number of returned objects	Integer	1000

Example for the elements 100 to 150

```
/devices?startAt=100&maxResults=50
```

### 3.1.3 Common response codes

The following response codes are valid for all requests.

Code	Description
200	Request successful / Object return
404	Object not found
50x	Server error
501	Resource not implemented

## 3.2 devices

### 3.2.1 GET /{oid}

Outputs information about the devices found in the database. This can be restricted to one device using **oid**.

#### URI parameters

Name	Description
{oid}	(Optional) OID of a device. E.g. <code>ExAmpL3d3v</code>
filter	Filters used to restrict the results. Separate multiple filters by using a comma ",". Use * as placeholder. Separate multiple information by using a comma – see the examples. <code>name</code> – Hostname <code>workgroup</code> – Workgroup <code>fqdn</code> – Fully qualified domain name <code>uuid</code> – UUID of the device <code>mac</code> – MAC address <code>companyId</code> – OID of the company <code>homeId</code> – OID of the organizational unit <code>costCenter</code> – Cost center
include	<code>softwareStatus</code> – Softwarezuweisungen <code>hardwareInventory</code> – Hardware inventory <code>softwareInventory</code> – Software inventory (MS uninstall)

#### Examples

Returns all devices

```
/devices
```

Returns as specific device on given OID

```
/devices/ExAmpL3d3v
```

Returns as specific device on given OID, including its software assignments and status as well the hardware specs

```
/devices/ExAmpL3d3v?include=softwareStatus,hardwareInventory
```

Only devices from the domain «bwg.corp» having the prefix «srv-»

```
/devices?filter=fqdn:bwg.corp,name:srv-*
```

### Response codes

Code	Description
200	Object return
404	Object not found

## 3.2.2 POST /

Creates a device. During creation, it checks whether the required object already exists (features: name, fqdn, MAC, UUID).

### Request parameters

Parameter	optional	Description
Name	No	Hostname
workgroup	No *	Workgroup
fqdn	No *	Fully qualified domain name
uuid	Yes	UUID from BIOS
mac	Yes	Ethernet MAC address
homeId	Yes **	OID of the organizational unit or company (only required when several companies are available)
deviceClassId	Yes	Device class – the helper function «deviceClass» returns all available classes
activationState	Yes	MANAGED   LOCKED   UNMANAGED   LOCKED
costcenter	Yes	String – Max 30 characters

\* Enter either `fqdn` or `workgroup`

\*\* If several companies are available, define the `homeId`.

### Example

POST /devices

```
{
  "name": "wrk-p-rest000",
  "workgroup": "restdom",
  "fqdn": "rest.dom",
  "uuid": "12345678-1234-1234-1234-1234567890AB",
  "mac": "1234567890AB",
  "homeId": "5qQEAMP13",
  "deviceClassId": "~TypP1VDI",
  "activationState": "LOCKED",
  "costCenter": "REST-DEMO"
}
```

### Response codes

Code	Description
200	The device already exists in the database
201	The device has been created successfully
400	Error when creating the object

### 3.2.3 PUT /oid

Updates the properties of a device.

#### URI parameters

Name	Description
oid	OID of a device

#### Body parameters

Parameter	optional	Description
Name	Yes	Hostname
workgroup	Yes	Workgroup
fqdn	Yes	Fully qualified domain name
uuid	Yes	UUID from BIOS
mac	Yes	Ethernet MAC address
homeld	Yes	OID of the organizational unit
deviceClassId	Yes	Device class – the helper function «deviceClass» returns all available classes
activationState	Yes	ACTIVE   LOCKED   UNMANAGED
costcenter	Yes	String – Max 30 characters

#### Example

```
PUT /devices/ExAmpL3d3v
```

```
{
  "activationState": "UNMANAGED",
  "costCenter": "Inventory"
}
```

#### Response codes

Code	Description
200	The request was sent successfully
400	Invalid body
404	Object not found

### 3.2.4 DELETE /oid

Deletes the specified device

#### Parameter

Name	Description
{oid}	OID of a device

#### Example

```
DELETE /devices/ExAmpL3123
```

### Response codes

Code	Description
202	The device is deleted
400	No device ID has been specified

## 3.2.5 GET /oid/actions (7.6.3)

Lists all pending actions of a device.

### URI Parameter

Name	Description
oid	OID of the device

### Example

```
GET /devices/ExAmpL312D3V/actions
```

### Response Codes

Code	Description
200	OK
400	Invalid request
404	Object not found

## 3.2.6 POST /oid/actions

Sets an action for a device. The actions must be specified in capital letters.

### URI parameters

Name	Description
oid	OID of the device

### Request parameters

Name	Description
code	<b>SWPROC</b> – Software push <b>POWERON</b> – Wake On Lan <b>REBOOT</b> - Restart <b>SHUTDOWN</b> – Shutdown <b>CMBACKUP</b> – Backup of the user data <b>AUTOSMARTDEPLOY</b> – Smart OSDeploy <b>EPSCAN</b> – UpdateManagement Patch scan <b>EPSCANINSTALL</b> – UpdateManagement Patch scan and installation <b>OCOMPBACK</b> – Online backup <b>CLIUNINST</b> – De-installation of the management client



## Example

POST /devices/ExAmpL312D3V/actions

```
{
  "code": "SWPROC"
}
```

## Response codes

Code	Description
201	Action created
400	Unknown action
404	Specified object (OID) not found

## 3.2.7 DELETE /oid/actions/aoid

---

Deletes a device action.

### URI parameters

Name	Description
oid	OID of the device
aoid	OID of the action. A list of actions can be retrieved with /oid/actions

## Example

DELETE /devices/ExAmpL312D3V/actions/ExAmpL312Ac7

## Response codes

Code	Description
201	Action deleted
400	Unknown action
404	Specified object (OID) not found

## 3.2.8 GET /oid/isActionPending/actionName

---

Checks whether the action `actionName` is pending on a device.

### URI parameters

Name	Description
oid	OID of a device, e.g. ExAmpL3D3v

actionName	<b>SWPROC</b> – Software push <b>POWERON</b> – Wake On Lan <b>REBOOT</b> - Restart <b>SHUTDOWN</b> – Shutdown <b>CMBACKUP</b> – Backup of the user data <b>AUTOSMARTDEPLOY</b> – Smart OSDeploy <b>EPSCAN</b> – UpdateManagement Patch scan <b>EPSCANINSTALL</b> – UpdateManagement Patch scan and installation <b>OCOMPBACK</b> – Online backup <b>CLIUNINST</b> – De-installation of the management client
------------	---

### Example

```
GET /devices/ExAmpL3D3v/isActionPending/REBOOT
```

### Response codes

Code	Description
201	Request OK
400	Unknown action
404	Specified object (OID) not found
422	Device not activated
501	Oid is not a device

## 3.2.9 GET /oid/inventory[/hw|/sw] (7.6.3)

Lists all hardware and/or software inventory of a device

### URI Parameter

Name	Description
oid	OID of the device

### Example

```
GET /devices/ExAmpL3123/inventory
GET /devices/ExAmpL3123/inventory/hw
GET /devices/ExAmpL3123/inventory/sw
```

### Response Codes

Code	Description
200	OK
400	Invalid request
404	Object not found

### 3.2.10 POST /oid/operatingSystemTemplates

---

Assigns an OS template.

#### Request parameters

Name	Description
oid	Machine OID

The possible OS list can be requested using the helper function «operatingSystems».

#### Example

```
POST /devices/ExAmpL3d3v/operatingSystemTemplates
```

```
{  
  "Id": 7  
}
```

## Response codes

Code	Description
201	Release template assigned
400	The template does not exist
403	OS deploy not licensed
404	The specified object does not exist.
422	Device not activated

## 3.2.11 POST /oid/softwareAssignments (legacy)

**Important:** This function has been improved. Please use the calls «PUT /oid/software/packages/{packageId}» and «PUT/oid/software/groups/{packageId}».

Weist ein Paket oder Paketgruppe zu.

### URI Parameter

Name	Description
oid	OID des Objektes

### Request Parameter

Name	Description
id	Name des Paketes oder Identifier des Paketes. Format: <Identifier>_<Sprache>_<Version> z.B: 900000_REMOTELYANYWHERE_12_CONFIG_CLIENT_ENU_01 903000_COLUMBUS_PACKAGINGTOOLS_7_UNI_01
softwareType	PACKAGE   GROUP

### Example

POST /devices/ExAmpL3123/softwareAssignments

```
{
  "id": "900000_REMOTELYANYWHERE_12_CONFIG_CLIENT_ENU_01",
  "softwareType": "PACKAGE"
}
```

### Response Codes

Code	Beschreibung
201	OK
400	Invalid Request
403	Softwaredeploy not licensed
404	Object not found
422	Device not activated

### 3.2.12 GET /oid/software[/packages]/groups (7.6.3)

Lists all assigned groups and/or software packages

#### URI Parameter

Name	Description
oid	OID of the object

#### Beispiel

```
GET /devices/ExAmpL3123/software
GET /devices/ExAmpL3123/software/packages
GET /devices/ExAmpL3123/software/groups
```

#### Response Codes

Code	Description
200	OK
400	Invalid request
404	Object not found

### 3.2.13 PUT /oid/software/packages/packageid (7.6.3)

Assigns a package

#### URI Parameter

Name	Description
oid	OID of the object
packageid	Package identifier. Observe the format: <Identifier>_<Language>_<Version> e.g.: 900000_REMOTELYANYWHERE_12_CONFIG_CLIENT_ENU_01 903000_COLUMBUS_PACKAGINGTOOLS_7_UNI_01

#### Example

```
PUT /devices/ExAmpL3123/software/packages/903000_COLUMBUS_PACKAGINGTOOLS_7_UNI_01
```

#### Response Codes

Code	Description
201	The request is executed
400	Invalid request body
403	Software deploy not licensed
404	Object not found
422	Device not activated

### 3.2.14 PUT /oid/software/groups/groupid (7.6.3)

Assigns a package

#### URI Parameter

Name	Description
oid	OID of the object
groupid	Group id e.g. "Client-Standard-2019"  If the group contains a space, it must be url encoded "Client Standard 2019" -> "Client%20Standard%202019"

#### Example

```
PUT /devices/ExAmpL3123/software/groups/Client-Default-2019
```

#### Response Codes

Code	Description
201	The request is executed
400	Invalid request body
403	Software deploy not licensed
404	Object not found
422	Device not activated

### 3.2.15 GET /oid/softwareDepot (7.6.3)

Returns all available packages

#### URI Parameter

Name	Description
oid	OID of the devices

#### Example

```
GET /devices/ExAmpL3123/softwareDepot
```

#### Response Codes

Code	Description
200	OK
400	Invalid request
404	Object not found

### 3.2.16 DELETE /oid/softwareAssignments/packages/packageId

Removes an assigned software

### URI parameters

Name	Description
oid	OID of the device
packageId	Package identifier. Observe the format: <Identifier>_<Language>_<Version> e.g.: 900000_REMOTELYANYWHERE_12_CONFIG_CLIENT_ENU_01 903000_COLUMBUS_PACKAGINGTOOLS_7_UNI_01

### Example

```
DELETE
/devices/ExAmpL3d3v/softwareAssignments/packages/900000_REMOTELYANYWHERE_12_CONFIG_CLIENT_ENU_01
```

### Response codes

Code	Description
201	The request is executed
400	Invalid request body
403	Software deploy not licensed
404	Object not found
422	Device not activated

## 3.2.17 DELETE /oid/softwareAssignments/groups/groupId

Removes an assigned software

### URI parameters

Name	Description
oid	OID of the device
groupId	Group id e.g. "Client-Standard-2019" If the group contains a space, it must be url encoded "Client Standard 2019" -> "Client%20Standard%202019"

### Example

```
DELETE /devices/ExAmpL3123/softwareAssignments/groups/Client-Standard-2019
```

Code	Description
201	The request is executed
400	Invalid request body
403	Software deploy not licensed
404	Object not found
422	Device not activated

### 3.2.18 GET /oid/variables (7.6.3)

Lists all assigned variables.

#### URI Parameter

Name	Description
oid	OID of the device

#### Example

```
GET /devices/ExAmpL312D3V/variables
```

#### Response Codes

Code	Description
201	OK
400	Invalid request
404	Object not found

### 3.2.19 POST/PUT /oid/variables

Creates or modifies a directly assigned variable.

#### URI parameters

Name	Description
oid	OID of the device

#### Request parameters

Name	Description
Name	Name of the variable to be created
value	Value of the variable to be created

#### Example

```
POST /devices/ExAmpL3123/variables
```

```
PUT POST /devices/ExAmpL3123/variables
```

```
{
  "name": "c_MachineLanguage",
  "value": "de-DE"
}
```

#### Response codes

Code	Description
201	Variable created
400	No information in the request body
404	Specified object not found



### 3.2.20 DELETE /oid/variables/variable

---

Deletes an assigned variable

#### URI parameters

Name	Description
oid	OID of the device
variable	Name of the variable

#### Example

```
DELETE /devices/ExAmpL3123/variables/c_restvar
```

#### Response codes

Code	Description
202	The variable is deleted
400	No information in the request body
404	Specified device not found

### 3.2.21 GET /oid/os (7.6.3)

---

Lists the assigned OS templates.

#### URI Parameter

Name	Description
oid	OID of the device

#### Example

```
GET /devices/ExAmpL3123/os
```

#### Response Codes

Code	Description
200	OK
400	Invalid request
404	Object not found

### 3.2.22 GET /oid/osDepot (7.6.3)

---

Lists all available OS templates

#### URI Parameter

Name	Description
oid	OID of the device

#### Beispiel

GET /devices/ExAmpL3123/osDepot

### Response Codes

Code	Description
200	OK
400	Invalid request
404	Object not found

## 3.3 users

### 3.3.1 GET /{oid}

Outputs information about the users found in the database. This can be restricted to one user using `oid`.

#### URI parameters

Name	Description
{oid}	(Optional) OID of a user. E.g. <code>ExAmpL3uSR</code>
Filter	Filters used to restrict the results. Separate multiple filters by using a comma ",". Use * as placeholder. <code>name</code> – Hostname <code>workgroup</code> – Workgroup <code>fqdn</code> – Fully qualified domain name <code>companyId</code> – OID of the company <code>homeId</code> – OID of the organizational unit <code>costCenter</code> – Cost center <code>eMail</code> – E-mail address <code>phone</code> – Phone number (if synchronized with AD or manually guided) <code>fullName</code> – Name displayed (if synchronized with AD or manually guided)

#### Examples

Returns all users

```
/users
```

Only users from the domain «bwg.training» having the prefix «training»

```
/users?filter=fqdn:bwg.training,name:training*
```

### 3.3.2 POST /

Creates a user. During creation, it checks whether the required object already exists (features: name, fqdn, workgroup).

#### Request parameters

Parameter	optional	Description
Name	No	Hostname
workgroup	No *	Workgroup
fqdn	No *	Fully qualified domain name
homeld	Yes **	OID of the organizational unit or company
activationState	Yes	ACTIVE   LOCKED   UNMANAGED
costcenter	Yes	String – Max 30 characters
eMail	Yes	E-mail address
phone	Yes	Phone number

\* Enter either "fqdn" or "workgroup"

\*\* If several companies are available, define the homeld.

#### Example

POST /users

```
{
  "name": "rest-User",
  "workgroup": "restdom",
  "fqdn": "rest.dom",
  "homeId": "5qQExAMP13",
  "activationState": "ACTIVE",
  "costCenter": "REST-DEMO"
}
```

#### Response codes

Code	Description
200	The user already exists in the database
201	The user has been created successfully
400	Error when creating the object

### 3.3.3 PUT /oid

Updates the properties of a user.

#### URI parameters

Name	Description
oid	OID of a user

#### Body parameters

Parameter	optional	Description
Name	Yes	Hostname
workgroup	Yes *	Workgroup
fqdn	Yes *	Fully qualified domain name
homeld	Yes	OID of the organizational unit
activationState	Yes	ACTIVE   LOCKED   UNMANAGED
costcenter	Yes	String – Max 30 characters
eMail	Yes	E-mail address
phone	Yes	Phone number

#### Example

PUT /users/ExAmpL3uSR

```
{
  "activationState": "UNMANAGED",
  "costCenter": "Inactive - Exit"
}
```

#### Response codes

Code	Description
200	Update successful
400	Invalid body
404	Object not found

### 3.3.4 DELETE /oid

Deletes the specified user

#### URI parameters

Name	Description
oid	OID of the user

#### Example

DELETE /users/ExAmpL3uSR

### Response codes

Code	Description
202	The user is deleted
404	User not found

## 3.3.5 POST /oid/softwareAssignments **(legacy)**

**IMPORTANT:** This function has been superseded. Please use «PUT /oid/software/packages/{packageId}» or «PUT/oid/software/groups/{packageId}».

Assigns a package or package group. Only «User packages» (conditional usage) can be assigned to the users.

URI parameters

Name	Description
oid	OID of the user

Request parameters

Name	Description
id	Group name or package identifier. Observe the format: <Identifier>_<Language>_<Version> e.g.: 900000_REMOTELYANYWHERE_12_CONFIG_CLIENT_ENU_01 903000_COLUMBUS_PACKAGINGTOOLS_7_UNI_01
softwareType	PACKAGE   GROUP

### Example

POST /users/ExAmpL3uSR/softwareAssignments

```
{
  "id": "900000_REMOTELYANYWHERE_12_CONFIG_CLIENT_ENU_01",
  "softwareType": "PACKAGE"
}
```

Response codes

Code	Description
201	Assignment is created
400	Invalid request body
404	Object not found

### 3.3.6 GET /oid/software[/packages]/groups (7.6.3)

Lists all assigned packages and/or groups.

#### URI Parameter

Name	Description
oid	OID of the user

#### Example

```
GET /users/ExAmpL3uSR/software
GET /users/ExAmpL3uSR/software/packages
GET /users/ExAmpL3uSR/software/groups
```

#### Response Codes

Code	Description
201	OK
400	Invalid request
404	Object not found

### 3.3.7 PUT /oid/software/packages/packageid (7.6.3)

Assigns a package. Its only possible to assign package with conditional usage (user icon)

#### URI Parameter

Name	Description
oid	OID of the user
packageid	Package identifier. Format <Identifier>_<Language>_<Version> z.B: 900000_REMOTELYANYWHERE_12_CONFIG_CLIENT_ENU_01 903000_COLUMBUS_PACKAGINGTOOLS_7_UNI_01

#### Example

```
PUT
/users/ExAmpL3uSR/software/packages/900000_REMOTELYANYWHERE_12_CONFIG_CLIENT_ENU_01
```

#### Response Codes

Code	Description
201	Request OK
400	Invalid request
404	Object not found

### 3.3.8 PUT /oid/software/groups/groupid (7.6.3)

---

Assigning a package group.

#### URI Parameter

Name	Beschreibung
oid	OID des Users
groupid	Name of the group. If the group contains a blank, the whitespace must be escaped. E.g. "Client Default 2019" becomes "Client%20Default%202019"

#### Example

```
PUT /users/ExAmpL3uSR/software/Client-Default-2019
```

#### Response Codes

Code	Beschreibung
201	OK
400	Invalid request
404	Object not found

### 3.3.9 GET /oid/softwareDepot (7.6.3)

Lists all available packages and groups.

#### URI Parameter

Name	Description
oid	OID of the user

#### Example

```
GET /users/ExAmpL3123/softwareDepot
```

#### Response Codes

Code	Description
200	OK
400	Invalid request
404	Object not found

### 3.3.10 DELETE /oid/softwareAssignments/packages/packageId

Removes a directly assigned software

#### URI parameters

Name	Description
oid	OID of the user
packageId	Package identifier. Format: <Identifier>_<Language>_<Version> e.g.: 900000_REMOTELYANYWHERE_12_CONFIG_CLIENT_ENU_01 903000_COLUMBUS_PACKAGINGTOOLS_7_UNI_01

#### Example

```
DELETE /users/ExAmpL3uSR/softwareAssignments/packages/900000_REMOTELYANYWHERE_12_CONFIG_CLIENT_ENU_01
```

#### Response codes

Code	Description
202	The request is executed
400	Invalid request body
404	Object not found

### 3.3.11 DELETE /oid/softwareAssignments/groups/groupId

Removes a directly assigned software group



### URI parameters

Name	Description
oid	Name of the variable to be created
groupId	Name of the software group

### Example

```
DELETE /users/ExAmpL3uSR/softwareAssignments/groups/CH_Standard_Client_2016
```

### Response codes

Code	Description
202	The request is executed
400	Invalid request body
404	Object not found

## 3.3.12 GET /oid/variables (7.6.3)

---

Lists all assigned variables

### URI Parameter

Name	Description
oid	OID of the user

### Example

```
GET /devices/ExAmpL312D3V/variables
```

### Response Codes

Code	Description
201	OK
400	Invalid request
404	Object not found

## 3.3.13 POST /oid/variables

---

Creates or modifies a directly assigned variable.

### URI parameters

Name	Description
oid	OID of the user

### Request parameters

Name	Description
Name	Name of the variable to be created
value	Value of the variable to be created

## Example

POST /devices/ExAmpL3uSR/variables

```
{
  "name": "c_UserLanguage",
  "value": "de-DE"
}
```

## Response codes

Code	Description
200	Variable created
400	No or invalid information in the request body
404	Specified object not found

## 3.3.14 DELETE /oid/variables

---

Deletes an assigned variable

### URI parameters

Name	Description
oid	OID of the user

### Request parameters

Name	Description
Name	Name of the variable to be deleted

## Example

```
DELETE /ExAmpL3uSR/variables
```

## Response codes

Code	Description
202	Variable deleted
400	No/invalid information in the request body
404	Specified device not found

## 3.4 actions

---

### 3.4.1 PUT /{actionid}/reschedule

---

Sets the time of execution to the next interval. This call-up can only be used for repetitive actions.

#### URI parameters

Name	Description
{actionoid}	OID of an action, e.g. ExAmpL3Ac7

#### Examples

Returns all actions of a device

```
GET /devices/ExAmpL3D3v/actions
```

Sets the timer to the next interval

```
PUT /actions/ExAmpL3Ac7/reschedule
```

#### Response codes

Code	Description
200	Request successfully executed
202	Invalid request

## 3.5 companies

---

### 3.5.1 GET /{oid}

---

Outputs information about the companies found in the database. This can be restricted to one user using `oid`.

#### URI parameters

Name	Description
{oid}	(Optional) OID of a company. E.g. ExAmpL30rG

#### Examples

Returns all companies

GET /companies

Returns the information about a specific company

GET /companies/ExAmpL30rG

### Response codes

Code	Description
200	Request successfully executed
400	Empty company {oid}
404	Object not found

## 3.5.2 GET /oid/softwareInfo

---

Lists all software packages and software groups of a company.

### URI parameters

Name	Description
oid	OID of the organizational unit. E.g. ExAmpL30rG

### Example

GET /companies/ExAmpL30rG/softwareInfo

### Response codes

Code	Description
200	Request successfully executed
400	No valid oid
404	Object not found

## 3.5.3 GET /oid/operatingSystemTemplates

---

Lists all software packages and software groups of a company.

### URI parameters

Name	Description
oid	OID of the organizational unit. E.g. ExAmpL30rG

### Example

GET /companies/ExAmpL30rG/operatingSystemTemplates

### Response codes

Code	Description
200	Request successfully executed
400	No valid oid
404	Object not found

## 3.6 treeNodes

---

### 3.6.1 GET /{oid}

---

Outputs information about the organizational units.

#### URI parameters

Field	Beschreibung
{oid}	(Optional) OID of the organizational unit. E.g. <code>ExAmpL30rG</code>

#### Example

Returns the complete organization tree

```
GET /treeNodes
```

Returns a specific organizational unit

```
GET /treeNodes/ExAmpL30rG
```

#### Response Codes

Code	Beschreibung
200	Request OK
400	Keine OID angegeben
404	OU not found

### 3.6.2 POST

---

Creates an organizational unit.

#### Example

```
POST /treeNodes
```

```
{
  "fullpath": "Organisations.brainwaregroup.Neue OU"
}
```

#### Response codes

Code	Description
201	OU created
400	Company not found

## 3.7 deviceClasses

---

### 3.7.1 GET /

---

Returns all device classes

#### Example

GET /deviceClasses

#### Response codes

Code	Description
200	OK