



3D Juump Infinite Directory API

[DI_EN] version 3.1

Introduction	4
Accessing the API.....	6
1 - Location	6
2 - Authentication	6
Directory session	7
1 - Introduction.....	7
Directory management	8
1 - Principles	8
1.1 - Simple vs multiple get.....	8
1.2 - Filtering and sorting	9
2 - Information.....	9
3 - Statistics.....	10
4 - Users	11
4.1 - List users	11
4.2 - Create a user.....	12
4.3 - Change a user's teams	12
4.4 - Delete users	12
5 - Teams.....	12
5.1 - List teams.....	12
5.2 - Create/edit a team	13
5.3 - Delete teams.....	13
6 - Proxies	14
6.1 - List proxies.....	14
6.2 - Create/edit a proxy.....	15
6.3 - Send a command to a proxy	16
6.4 - Connections	16
6.4.1 - List the sources on a proxy	16
6.4.2 - Add/update a source to a proxy	16

6.4.3 - Remove a source from a proxy.....	17
6.5 - Delete proxies.....	17
7 - Projects	17
7.1 - List projects.....	17
7.2 - Create a project	18
7.3 - Send a command to a project	19
7.3.1 - Trigger build	19
7.4 - Delete a project	19
7.5 - List a project's builds.....	19
7.6 - Pack a project's build.....	21
7.7 - Delete an existing pack from a project's build.....	21
7.8 - Delete a project's build	21
8 - Settings	21
8.1 - Access the settings	22
8.2 - Change the settings.....	24

Introduction

This documentation is targeted to developers that want to automate 3D Juump Infinite operations:

- users administration,
- cluster administration,
- project administration.

Accessing the API

1 - Location

The Directory API is located under '/directoryapi' on your ∞Directory host.

```
GET /directoryapi
=>
{
  "comment": "directory http api",
  "type": "directoryapi",
  "version": {
    "backend": "3.1.0.10311 Beta",
    "number": "2.0"
  }
}
```

2 - Authentication

The Directory API is accessible through a *directory session* opened by an authenticated user with administrator privilege.

Alternatively, the API is also accessible through the *superuser* login procedure. Pass the credentials using the *Basic* authorization method.

Directory session

1 - Introduction

Directory management

Directory management API is available under the */manage* sub-path. Several endpoints are available.

1 - Principles

1.1 - Simple vs multiple get

Several endpoints can return one or more records from a collection. You can either use the collection endpoint (*users*, *teams*, *proxies*, *projects*, *builds*...) or the single item endpoint (*users/:sub*, *teams/:teamname*, *proxies/:proxyid*, *projects/:projectid*, *builds/:buildid*...). In either case, the result is a JSON document of the following form:

```
GET ...  
=> 200  
{  
  "count": <number of records>,  
  "data": [  
    <array of records>  
  ]  
}
```

Note: single item endpoints always return 0 or 1 record.

1.2 - Filtering and sorting

Collection endpoints support filtering and sorting.

```

POST ...
{
  "offset": <start of slice>,
  "limit": <size of slice>,
  "sort": {
    "field": <field used for sorting>,
    "sort_asc": <order of sorting>
  },
  "operator": <filters combination operator>,
  "filters": [
    {
      "field": <field to filter on>,
      "value": <value to match>,
      "partial": <partial or exact match>
    },
    ...
  ]
}
=> 200
{
  "count": <number of matching records>,
  "data": [
    <array of matching records>
  ]
}

```

Where:

- *"offset"* and *"limit"* allows to retrieve only a slice of the filtered collection,
- *"sort"* allows to sort the results along the chosen *"field"* in ascending (when *"sort_asc"* is *true*) or descending (when *"sort_asc"* is *false*) order,
- *"operator"* (*AND* or *OR*) and *"filters"* allows to search & filter the results, with a filter being composed of:
 - *"field"*: the name of the field of data on which to filter,
 - *"value"*: the value to match,
 - *"partial"*: a boolean telling whether the match is exact or absolute.

2 - Information

Get information regarding the 3D Juump Infinite ∞Directory service.

```

GET /directoryapi/manage/daemoninfo
=> 200
{
  "softwareversion": "3.1.0.10311 Beta",
  "daemonversion": "3.0",
  "isrunning": false,
  "lmsserver": "6200@127.0.0.1",

```

```

    "lmxisok": true,
    "lmxfeature": "DIRECTORY-RF-FR-REALFUSIO-H1",
    "lmxclientfeature": "CLIENT-RF-FR-REALFUSIO-H1",
    "maxlic": 5,
    "borrowedlic": 0
  }

```

Where:

- *"softwareversion"*: the version of 3D Juump Infinite,
- *"daeomonversion"*: the version of the ∞Directory daemon,
- *"isrunning"*: whether the daemon is running or not,
- *"lmxserver"*: the adresse where the LMX server resides (ex: *6200@13.93.66.185*),
- *"lmxisok"*: whether the LMX server is accessible or not,
- *"lmxfeature"*: the licence feature for the ∞Directory,
- *"lmxclientfeature"*: the licence feature for the clients,
- *"maxlic"*: the maximum number of licences that can be acquired at the same time,
- *"borrowedlic"*: the number of borrowed licences.

3 - Statistics

Get overall statistics about the cluster.

```

GET /directoryapi/manage/stats
=> 200
{
  "nbusers": 2,
  "nbusers_active": 2,
  "nbteams": 3,
  "nbproxies": 1,
  "nberrors_proxies": 1,
  "nbprojects": 2,
  "nberrors_projects": 0,
  "nbbuilds": 1,
  "nberrors_builds": 0
}

```

Where:

- *"nbusers"*: the number of users,
- *"nbusers_active"*: the number of active users,
- *"nbteams"*: the number of teams,
- *"nbproxies"*: the number of ∞Servers/∞Proxies,
- *"nberrors_proxies"*: the number of ∞Servers/∞Proxies in error,
- *"nbprojects"*: the total number of projects,
- *"nberrors_projects"*: the number of projects in error,
- *"nbbuilds"*: the total number of builds,
- *"nberrors_builds"*: the number of builds in error.

4 - Users

The API allows to access the registered users collection.

4.1 - List users

GET /directoryapi/manage/users

=> 200

```
{
  "count": 2,
  "data": [
    {
      "enabled": true,
      "frontendaccess": true,
      "oidcsub": "auth0|5ccc073f90af3710f269d988",
      "tags": [
        "view:Link&Go",
        "export2d:Link&Go"
      ],
      "teams": [],
      "userinfo": {
        "email": "sylvain@infinitesupport.com",
        "email_verified": false,
        "name": "sylvain@infinitesupport.com",
        "nickname": "sylvain",
        "picture": "https://s.gravatar.com/avatar/33b367411e31b1f91539724c05d7a152?s=480&r=pg&d=https%3A%2F%2Fcdn.auth0.com%2Favatar-s%2Fsy.png",
        "sub": "auth0|5ccc073f90af3710f269d988"
      }
    },
    {
      "enabled": true,
      "frontendaccess": true,
      "oidcsub": "auth0|5cdc1611db4d301107026c77",
      "tags": [],
      "teams": [
        "akka"
      ],
      "userinfo": {
        "email": "david@infinitesupport.com",
        "email_verified": false,
        "name": "david@infinitesupport.com",
        "nickname": "david",
        "picture": "https://s.gravatar.com/avatar/18def08a9d6503e4941aaf2a0444b2b2ff?s=480&r=pg&d=https%3A%2F%2Fcdn.auth0.com%2Favatar-s%2Fda.png",
        "sub": "auth0|5cdc1611db4d301107026c77"
      }
    }
  ]
}
```

Where a user record has the following structure:

- `"enabled"`: a boolean telling whether the user is activated or not,
- `"fontendaccess"`: a boolean telling whether the user has administrator privilege,
- `"oidcsub"`: the identity of the user as delivered by the *Authenticator*,
- `"tags"`: the list of the user's personal tags,
- `"teams"`: the list of the teams the user belongs to,
- `"userinfo"`: the user profile as delivered by the *Authenticator*.

4.2 - Create a user

```
PUT /directoryapi/manage/users
{
  <oidcsub>: {
    "enabled": <user activated or not>,
    "frontendaccess": <administrator or not>,
    "tags": [
      <array of tags>
    ]
  }
}
=> 200
```

4.3 - Change a user's teams

It's possible to replace the teams of a user identified by his oidcsub.

```
PUT /directoryapi/manage/users/:oidcsub/teams
[<array of teams>]
=> 200
```

4.4 - Delete users

It's possible to delete one user.

```
DELETE /directoryapi/manage/users/:oidcsub
=> 200
```

It's also possible to batch-delete several users.

```
DELETE /directoryapi/manage/users
[<array of oidcsubs>]
=> 200
```

5 - Teams

The API allows to access the teams collection.

5.1 - List teams

```
GET /directoryapi/manage/teams
=> 200
{
  "count": 3,
  "data": [
```

```

    {
      "tags": ["view:Link&Go", "export2d:Link&Go", "export3d:Link&Go", "download:Link&Go", "connect:server", "connect:support"],
      "teamname": "admin",
      "teamuuid": "<unique id of the team>"
    },
    {
      "tags": ["view:Link&Go", "connect:server", "connect:support"],
      "teamname": "akka",
      "teamuuid": "<unique id of the team>"
    },
    {
      "tags": ["connect:server", "connect:support", "view:D551"],
      "teamname": "guest",
      "teamuuid": "<unique id of the team>"
    }
  ]
}

```

Where a team record has the following structure:

- `"tags"`: the list of the team's tags,
- `"teamname"`: the (unique) name of the team,
- `"teamuuid"`: the (unique) internal id of the team.

5.2 - Create/edit a team

PUT /directoryapi/manage/teams

```

{
  <teamname>: {
    "tags": [ <array of tags> ]
  }
}
=> 200
{
  "status": "upserted",
  "teamuuid": "<unique id of the newly created team>"
}

```

5.3 - Delete teams

It's possible to delete one team.

DELETE /directoryapi/manage/teams/:teamname
=> 200

It's also possible to batch-delete several teams.

DELETE /directoryapi/manage/users
[<array of teamnames>]
=> 200

6 - Proxies

The API allows to access the servers/proxies collection.

6.1 - List proxies

GET /directoryapi/manage/proxies

=> 200

```
{
  "count": 1,
  "data": [
    {
      "acceptnewconnections": true,
      "activeconnections": 0,
      "clientmessage": "",
      "haserror": false,
      "label": <name of the server/proxy>,
      "login": <postgres login>,
      "porthttp": <https port>,
      "portpg": <postgres port>,
      "proxyinfo": {
        "acceptgeneration": true,
        "acceptreplication": false,
        "acceptuserconnection": true,
        "couchserverpath": "/couchdb/",
        "couchserverversion": "1.6.1",
        "daemondbversion": "7.0",
        "daemonversion": "2.0",
        "elasticsearchserverpath": "/elastic/",
        "elasticsearchserverversion": "6.6.0",
        "error_info": "",
        "fileserverspath": "/",
        "httpserverversion": "6.0",
        "isallversionvalid": true,
        "iscouchonline": true,
        "isdaemonrunning": true,
        "isesonline": true,
        "ishttponline": true,
        "isloginvalid": true,
        "isonline": true,
        "pgserverversion": "10.7",
        "proxyrole": 6,
        "softwareversion": "3.1.0.10311 Beta"
      },
      "tags": ["support", "server"],
      "url": <host address>
    }
  ]
}
```

Where a proxy record has the following structure:

- **"label"**: the name of the server/proxy,
- **"tags"**: the list of the server/proxy's tags,

- `"url"`: the host address of the server/proxy,
- `"acceptnewconnections"`: whether the server/proxy accepts connections,
- `"activeconnections"`: the number of currently active connections,
- `"clientmessage"`: the content of the billboard message on this server/proxy,
- `"haserror"`: whether the server/proxy has pending errors,
- `"login"`: the PostgreSQL login for this server/proxy,
- `"porthttp"`: the HTTPS port,
- `"portpg"`: the PostgreSQL port,
- `"proxyinfo"`: a record containing detailed information about the server/proxy.

The `proxyinfo` record has the following structure:

- `"acceptgeneration"`: whether this node has generation capability,
- `"acceptreplication"`: whether this node has replication capability,
- `"acceptuserconnection"`: whether this node accepts user connection,
- `"couchserverpath"`: the URI path to the CouchDB end-point,
- `"elasticsearchserverpath"`: the URI path to the ElasticSearch end-point,
- `"fileserverpath"`: the URI path to the file server end-point,
- `"softwareversion"`: the 3D Juump Infinite version,
- `"daemondbversion"`: the 3D Juump Infinite daemon database version,
- `"daemonversion"`: the 3D Juump Infinite daemon service version,
- `"pgserverversion"`: the PostgreSQL version,
- `"elasticsearchserverversion"`: the ElasticSearch version,
- `"httpserverversion"`: the HTTP server version,
- `"couchserverversion"`: the CouchDB version,
- `"isallversionvalid"`: whether all versions are valid,
- `"isonline"`: whether this server/proxy is online,
- `"isdaemonrunning"`: whether the 3D Juump Infinite daemon is running,
- `"isesonline"`: whether ElasticSearch is online,
- `"ishttponline"`: whether the HTTP server is online,
- `"iscouchonline"`: whether CouchDB is online,
- `"isloginvalid"`: whether the PostgreSQL login is valid,
- `"error_info"`: information on the potential error,
- `"proxyrole"`: the role of this particular node:
 - `"1"`: ∞Proxy,
 - `"2"`: ∞StaticProxy,
 - `"6"`: ∞Server.

6.2 - Create/edit a proxy

```
PUT /directoryapi/manage/proxies
{
  <label>: {
    "tags": [...],
    "url": <host address>
```

```

    "porthttp": 443,
    "portpg": 5432,
    "login": <postgres login>,
    "password": <postgres password>,
    "acceptnewconnections": false
  }
}
=> 200

```

6.3 - Send a command to a proxy

```

POST /directoryapi/manage/proxies/:proxylabel?command=:command
=> 200

```

Where the command can be either:

- *revokeusers*: to revoke all users connected to this proxy,
- *cleanup*: to trigger a cleanup routine on this proxy.

6.4 - Connections

6.4.1 - List the sources on a proxy

```

GET /directoryapi/manage/proxies/:proxylabel/sources
=> 200
{
  "count": 1,
  "data": [
    {
      "sourcedescriptor": <descriptor of the connection>,
      "tags": [ "Link&Go" ],
      "sourcetype": "metadata",
      "labelsrc": <label of source node>,
      "labeldest": <label of destination node>
    }
  ]
}

```

Where a proxy connection has the following structure:

- *"sourcedescriptor"*: the identifier of the connection,
- *"tags"*: the list of the connection's tags,
- *"sourcetype"*: the type of data that transit on this connection (either *metadata* or *asset*),
- *"labelsrc"*: the name of the source server/proxy,
- *"labeldst"*: the name of the destination server/proxy.

6.4.2 - Add/update a source to a proxy

```

PUT /directoryapi/manage/proxies/:proxylabel/sources/:uuid
{
  "sourcetype": "metadata",
  "proxysrclabel": "InfiniteSupport",
  "tags": ["Link&Go"]
}

```



```
}
=> 200
```

Where `:proxyLabel` is the name of the proxy and `:uuid` is a valid bracket-enclosed UUID.

The payload must contain the following fields:

- `"sourcetype"`: the type of data that transit on this connection (either `metadata` or `asset`),
- `"proxysrcLabel"`: the name of the source server/proxy,
- `"tags"`: the list of the connection's tags.

6.4.3 - Remove a source from a proxy

```
DELETE /directoryapi/manage/proxies/:proxyLabel/sources/:uuid
=> 200
```

Where `:proxyLabel` is the name of the proxy and `:uuid` is the id of the connection.

6.5 - Delete proxies

It's possible to delete one proxy.

```
DELETE /directoryapi/manage/proxies/:proxyLabel
=> 200
```

It's also possible to batch-delete several proxies.

```
DELETE /directoryapi/manage/proxies
[<array of proxy labels>]
=> 200
```

7 - Projects

The API allows to access the projects collection.

7.1 - List projects

```
GET /directoryapi/manage/projects
=> 200
{
  "count": 2,
  "data": [
    {
      "label": "prj_091d232caeae2c0d2e4e63b031534c5b",
      "generatorlabel": "InfiniteSupport",
      "projectstatus": 2,
      "lasterrormsg": "",
      "nbbuilds": 1,
      "nbbuilds_errors": 0,
      "diskfootprint": 42377621,
      "properties": {
        "compressionlevel": "Lzma",
        "generatorversion": "3.1.0",
        "projectcomment": "AKKA Link&Go",
        "projectid": "prj_091d232caeae2c0d2e4e63b031534c5b",

```

```

        "sourcemodelmirrorversion": "1~38f413f4b2684f888fa8daf
dab827d55",
        "unit": "millimeter"
    },
    ...
]
}

```

Where a project record has the following structure:

- **"label"**: the identifier of the project,
- **"generatorlabel"**: the name of the ∞ Server hosting this project,
- **"projectstatus"**: the current status of the project, either:
 - 1: the connector has reported an error,
 - 2: the project is waiting for connector update,
 - 3: the project is locked by the connector during update,
 - 4: the project is locked by the generator during build packing,
 - 5: the project is locked by the generator during build construction,
 - 7: the project's PostgreSQL database is missing,
 - 8: the project is currently locked by generator service, to be migrated to new database version,
 - 9: an error occurred during migration of this project,
 - 10: the project is locked by the generator during build unpacking,
 - 11: the project is locked because migration is needed,
 - 12: the project is locked to synchronize metadata from connector index,
 - 13: the project is locked to cleanup unused generation data,
 - 14: bad version.
- **"lasterrormsg"**: the last error message related to this project,
- **"nbbuilds"**: the number of builds available on this project,
- **"nbbuilds_errors"**: the number of builds with an error,
- **"diskfootprint"**: disk size occupied by the project (in bytes),
- **"properties"**: a project property block of the form:
 - **"projectid"**: the internal identifier of the project,
 - **"projectcomment"**: the human readable name of the project,
 - **"unit"**: the length unit of the project (either *millimeter*, *centimeter*, *decimeter* or *meter*),
 - **"generatorversion"**: the version of the generator,
 - **"compressionlevel"**: undocumented,
 - **"sourcemodelmirrorversion"**: undocumented.

7.2 - Create a project

```

PUT /directoryapi/manage/projects/:projectid
{
    "proxylabel": <name of the generator>,

```

```

    "projectcomment": <name of the project>,
    "unit": <length unit for this project>,
    "compressionlvl": "Lzma"
  }
=> 200

```

Where *:projectid* is a new project unique identifier of the form "prj_" followed by 32 alphanumeric characters.

The payload has the following fields:

- *"proxyLabel"*: the name of the generator server to use,
- *"projectcomment"*: the human readable name of the project,
- *"unit"*: the length unit of the project (either *millimeter*, *centimeter*, *decimeter* or *meter*),
- *"compressionlvl"*: undocumented (must be set to *Lzma*).

7.3 - Send a command to a project

```

POST /directoryapi/manage/projects/:projectid?command=:command
=> 200

```

Where the command can be either: *build*, *cancelbuild*, *tagbuildresultasignored*, *migrate*, *sync*, *cancelsync*. Some commands may require a payload, as explained here below.

7.3.1 - Trigger build

```

POST /directoryapi/manage/projects/:projectid?command=build
{
  "buildparametersdocid": <id of the build parameters document>
}
=> 200

```

7.4 - Delete a project

```

DELETE /directoryapi/manage/projects/:projectid
=> 200

```

Where *:projectid* is the internal identifier of the project (of the form "prj_" + 32 alphanumeric characters).

7.5 - List a project's builds

```

GET /directoryapi/manage/projects/:projectid/builds
=> 200
{
  "count": 1,
  "data": [
    {
      "buildid": <internal id of the build>,
      "tags": ["Link&Go"],
      "haswebdata": true,
      "status": 5,
      "statustimestamp": "2019-06-14 09:40:08.372539",
      "packstatus": 0,
    }
  ]
}

```

```

        "errorlog": "",
        "proxylabel": "InfiniteSupport",
        "proxyrole": 6,
        "proxyhaserror": false,
        "builddbcommentlight": {
            "buildcomment": "Build comment",
            "buildid": <internal id of the build>,
            "buildversion": "7.0",
            "compressionlevel": "Lzma",
            "generationdate": "2019-05-03T14:34:24+00:00",
            "projectcomment": "AKKA Link&Go",
            "projectid": "prj_091d232caeae2c0d2e4e63b031534c5b",
            "tags": ["Link&Go"]
        }
    }
}
]
}

```

Where a build record has the following fields:

- *"buildid"*: the identifier of the build,
- *"tags"*: an array containing the build's tags,
- *"haswebdata"*: whether this build contains data for the web application,
- *"status"*: the current status of the build, either:
 - 1: replication in progress,
 - 2: restoration in progress,
 - 3: metadata synchronization in progress,
 - 4: publication in progress,
 - 22: this build has been unpublished and migration is needed,
 - 18: migration in progress,
 - 20: pending publication post migration,
 - 27-28: index verification in progress,
 - 5: build is ready,
 - 6: replication error,
 - 7: restoration error,
 - 8: synchronization error,
 - 29: resynchronization error,
 - 9: publication error,
 - 10: unknown error,
 - 11-17: internal error,
 - 19: migration error,
 - 21: missing files,
 - 24-26: internal error,
 - 30: bad version,
- *"statustimestamp"*: the human readable time of last status update,
- *"packstatus"*: the current status of the build's pack, either:

- 1: the build's pack is available,
- 2: the build's pack has been requested,
- 3: packing in progress,
- 4: packing error,
- `"errorLog"`: the logged error (if any),
- `"proxyLabel"`: the name of the hosting server/proxy,
- `"proxyrole"`: the role of the hosting server/proxy (see the corresponding chapter),
- `"proxyhaserror"`: whether the server/proxy itself has errors,
- `"bulddbcommentlight"`: the descriptor of the build, with the following fields:
 - `"buildcomment"`: the comment associated to the build,
 - `"buildid"`: the internal id of the build,
 - `"buildversion"`: the build version,
 - `"compressionlevel"`: undocumented,
 - `"generationdate"`: the date of the generation of this build,
 - `"projectcomment"`: the name of the project,
 - `"projectid"`: the id of the project,
 - `"tags"`: an array containing the build's tags.

7.6 - Pack a project's build

```
POST /directoryapi/manage/projects/:projectid/builds/:buildid?command=
pack
{
  "passphrase": <passphrase to protect the pack>
}
=> 200
```

Where `"passphrase"` contains the phrase the user will have to type to use the packed build file.

7.7 - Delete an existing pack from a project's build

```
POST /directoryapi/manage/projects/:projectid/builds/:buildid?command=
unpack
=> 200
```

7.8 - Delete a project's build

```
DELETE /directoryapi/manage/projects/:projectid/builds/:buildid?proxylabel=:proxyid
=> 200
```

Where `:proxyid` optionnaly identifies the proxy from which to delete the build. If no proxy is given, all instances of the build are removed from the whole cluster.

8 - Settings

The API allows to access the ∞Directory settings.

8.1 - Access the settings

GET /directoryapi/manage/settings

=> 200

```
{
  "count": 15,
  "data": [{
    "propcomment": "Max token offline time (in sec)",
    "propmax": 8035200,
    "propmin": 1,
    "propname": "max_token_offline_time_sec",
    "proptype": "T_INT32",
    "propvalue": "604800"
  }, {
    "propcomment": "Proxy access token maxAge (in sec)",
    "propmax": 600,
    "propmin": 30,
    "propname": "proxy_access_token_maxage_sec",
    "proptype": "T_INT32",
    "propvalue": "300"
  }, {
    "propcomment": "sso management, interval between two user
consent, if 0 sso is disabled (in sec)",
    "propmax": 604800,
    "propmin": 0,
    "propname": "sso_user_consent_interval_sec",
    "proptype": "T_INT32",
    "propvalue": "86400"
  }, {
    "propcomment": "Verbose",
    "propmax": 0,
    "propmin": 0,
    "propname": "verbose",
    "proptype": "T_BOOL",
    "propvalue": "true"
  }, {
    "propcomment": "Webhook alert url minimum call interval (i
n sec)",
    "propmax": 2147483647,
    "propmin": 60,
    "propname": "webhook_report_interval_sec",
    "proptype": "T_INT32",
    "propvalue": "300"
  }, {
    "propcomment": "Webhook alert url",
    "propmax": 0,
    "propmin": 0,
    "propname": "webhook_url",
    "proptype": "T_STRING",
    "propvalue": ""
  }, {
    "propcomment": "Auth allowed callback URLs",
    "propmax": 0,
    "propmin": 0,

```

```

        "propname": "auth_allowed_callback_urls",
        "proptype": "T_STRINGLIST",
        "propvalue": "https://realfusio-support.westeurope.cloudap
p.azure.com:443/directoryapi/directorysession/onauthenticationdone,htt
ps://13.93.66.185/directory,https://realfusio-support.westeurope.cloud
app.azure.com/directory"
    }, {
        "propcomment": "Client hardware report",
        "propmax": 0,
        "propmin": 0,
        "propname": "client_hardwarereport",
        "proptype": "T_BOOL",
        "propvalue": "true"
    }, {
        "propcomment": "Client usage report",
        "propmax": 0,
        "propmin": 0,
        "propname": "client_usagereport",
        "proptype": "T_BOOL",
        "propvalue": "true"
    }, {
        "propcomment": "Data session max heartbeat interval (in se
c)",
        "propmax": 7200,
        "propmin": 60,
        "propname": "data_session_max_heartbeat_interval_sec",
        "proptype": "T_INT32",
        "propvalue": "1800"
    }, {
        "propcomment": "Directory session max client idle duration
(in sec)",
        "propmax": 2678400,
        "propmin": 3600,
        "propname": "directory_session_max_client_idle_duration_se
c",
        "proptype": "T_INT32",
        "propvalue": "7200"
    }, {
        "propcomment": "Directory session max duration (in sec)",
        "propmax": 2678400,
        "propmin": 3600,
        "propname": "directory_session_max_duration_sec",
        "proptype": "T_INT32",
        "propvalue": "86400"
    }, {
        "propcomment": "Directory session max heartbeat interval (
in sec)",
        "propmax": 14400,
        "propmin": 60,
        "propname": "directory_session_max_heartbeat_interval_sec"
    },
    {
        "proptype": "T_INT32",
        "propvalue": "3600"
    }, {

```

```

        "propcomment": "Directory session wait for auth (in sec)",
        "propmax": 900,
        "propmin": 30,
        "propname": "directory_session_wait_for_auth_sec",
        "proptype": "T_INT32",
        "propvalue": "300"
    }, {
        "propcomment": "Max DMU rent time (in sec)",
        "propmax": 8035200,
        "propmin": 1,
        "propname": "max_dmu_rent_time_sec",
        "proptype": "T_INT32",
        "propvalue": "604800"
    }
]
}

```

Each individual setting is of the form:

- *"propname"*: the identifier of the setting,
- *"propcomment"*: the description of the setting,
- *"proptype"*: the type of the setting (either "T_INT32", "T_BOOL", "T_STRING" or "T_STRINGLIST"),
- *"propmin"*: the minimum value of the setting (if applicable),
- *"propmax"*: the maximum value of the setting (if applicable),
- *"propvalue"*: the current value of the setting.

8.2 - Change the settings

```

POST /directoryapi/manage/settings
{
    <property name>: {
        "proptype": <property type>,
        "propvalue": <property value>
    },
    ...
}
=> 200

```