

PPMS Utility Management API Documentation - revision 2013.2

PUMAPI Introduction:

The purpose of the PPMS Utility Management API (PUMAPI) is to provide a way to perform remote, secure and automated manipulations on PPMS data from third party scripts and programs.

- PUMAPI queries should use the URL: *<URL of your PPMS instance>/pumapi/*
- PUMAPI clients should be using encrypted HTTPS requests to the PUMAPI interface
- All parameters should be sent as HTTP POST data in the form: *parameter=value*
- Multiple parameters/value pairs should be separated in the HTTP POST data by the & character:
parameter1=value1¶meter2=value2¶meter3=value3
- All values should be **URLencoded** (and only the values); see [link](#) and example section

PUMAPI Global parameters and authentication:

psapwd - master password based authentication (the master password is set when initializing PPMS)

apikey - pre-shared key based authentication (multiple keys can be set)

id - identification number for a specific PPMS system (accessible in the PPMS system page)

code - access code for a specific PPMS system (accessible in the PPMS system page)

action - will determine what is the request's purpose; mandatory for all requests

format - CSV type output with *format=csv*, json type format with *format=json* (not available with all requests)

noheaders - if the option *noheaders=true* is set then CSV type outputs will not print header lines

Depending on the action, different types of authentication can be used for PUMAPI requests:

- the master password based authentication (the *psapwd* parameter)
- pre-shared API key based authentication (the *apikey* parameter)
- the system based authentication (for requests related to a specific system, using the *id* and *code* parameters).

PPMS Utility Management API Documentation - revision 2013.2

PUMAPI actions available:

action=newuser and ***action=upduser***

definition: creates (*action=newuser*) or updates (*action=upduser*) a PPMS user

authentication: master password based or API key based* authentication

parameter: *login* - the unique string identifier for the user. Mandatory for both creations and updates.

parameter: *lname* - the last name of the user. Mandatory for creations, optional for updates.

parameter: *fname* - the first name of the user. Mandatory for creations, optional for updates.

parameter: *email* - the email address of the user. Mandatory for creations, optional for updates.

parameter: *phone* - the phone number for the user. Optional for creations, optional for updates.

parameter: *bcode* - the account number to use for the user. Optional for creations, optional for updates.

parameter: *affiliation* - the affiliation for the user. Optional for creations, optional for updates.

parameter: *active* - the status of the user (value can be *true* or *false*). Cannot be used for user creation (the creation assumes the user is active), optional for updates.

parameter: *unitlogin* - the unique string identifier for the primary group of the user. Mandatory for creations, optional for updates.

parameter: *pwd* - the password to use for the user. Mandatory for creations, optional for updates. Passwords can be in clear text (will be hashed when saved in PPMS), a MD5 hash, or a SSHA hash (compatible with many LDAP servers). MD5 and SSHA hashes should be encoded in base-64 and preceded by the strings {SSHA} or {MD5} (in uppercase).

parameter: *mustchpwd* - the user will be required to change his/her password after next login (value can be *true* or *false*). Optional for creations, optional for updates.

parameter: *musthbcode* - the user will be required to change his/her account number after next login (value can be *true* or *false*). Optional for creations, optional for updates.

action=newgroup and ***action=updgroup***

definition: creates (*action=newgroup*) or updates (*action=updgroup*) a PPMS group

authentication: master password based or API key based* authentication

parameter: *unitlogin* : the unique string identifier for the group. Mandatory for creations and for updates.

parameter: *unitname* : the name of the group. Optional for creations, optional for updates.

parameter: *headname* : the name of the director/PI of the group (it is advised to use "last-name, first-name" for alphabetical ordering). Optional for creations, optional for updates.

parameter: *heademail* : the email of the director/PI of the group. Optional for creations, optional for updates.

parameter: *unitBcode* : the account number to use for the users of the group that do not have a personal account number. Optional for creations, optional for updates.

parameter: *department* : the department the group is part of (if relevant). Optional for creations, optional for updates.

parameter: *institution* : the institution the group is part of. Optional for creations, optional for updates.

parameter: *address* : the address to use for invoicing. Optional for creations, optional for updates.

parameter: *ext* : defines if the group is from an external institution (value can be *true* or *false*). Optional for creations, optional for updates.

parameter: *affiliation* : the affiliation for the group. Optional for creations, optional for updates.

parameter: *active* : the status of the group (value can be *true* or *false*). Cannot be used for group creation (the creation assumes the group is active), optional for updates.

action=getusers

definition: get the list of the users in the PPMS database

authentication: master password based or API key based* authentication

parameter: *active* - optional, *active=true* returns only active users, *active=false* returns inactive users

action=getuser

definition: returns parameters for a given user. Parameters: login, lname, fname, email, phone, bcode, affiliation, unitlogin, mustchpwd, musthbcode, active

authentication: master password based or API key based* authentication

parameter: *login* - the unique string identifier for the user (mandatory)

PPMS Utility Management API Documentation - revision 2013.2

action=getgroups

definition: get the list of the groups in the PPMS database

authentication: master password based or API key based* authentication

parameter: *active* - optional, *active=true* returns only active groups, *active=false* returns inactive groups

action=getgroup

definition: returns all parameters for a given group. Parameters: unitlogin, unitname, headname, heademail, unitbcode, department, institution, address, affiliation, ext, active

authentication: master password based or API key based* authentication

parameter: *unitlogin* - the unique string identifier for the group (mandatory)

action=getgroupusers

definition: returns a list of users that belongs to a given group

authentication: master password based or API key based* authentication

parameter: *unitlogin* - the unique string identifier for the group (mandatory)

parameter: *active* - optional, *active=true* returns only active users, *active=false* returns inactive users

action=auth

definition: authenticate a PPMS user with his password. returns *OK* if successful.

authentication: master password, or API key*, or system based authentication

parameter: *login* - the unique string identifier for the user (mandatory)

parameter: *pwd* - the user's password (mandatory)

parameter: *rightcheck* - optional. When *rightcheck=true* and when using system based authentication, the system will check if the user is authorized to use the system. It will return *nopwd* if password is incorrect, *nosys* if user is not authorized and *notime* if the user cannot login at the current type due to access restrictions. It will also return a second line with *ADM* after *OK* if the authenticated user is an administrator or a superuser on this system..

action=getbooking

definition: returns what user is currently booked in the PPMS database for the current system, and the number of minutes left before the end of the booking.

authentication: master password, or API key*, or system based authentication

parameter: *id* - the identification number for the system (mandatory)

action=nextbooking

definition: returns what user is booked for the next booking in the PPMS database for the current system, and the number of minutes left before this booking starts

authentication: master password, or API key*, or system based authentication

parameter: *id* - the identification number for the system (mandatory)

action=creatinc

definition: creates a incident in the PPMS database

authentication: master password, or API key*, or system based authentication

parameter: *id* - the identification number for the system (mandatory)

parameter: *severity* - severity level for the incident (*severity=1* for low, *severity=2* for medium/partly dysfunctional, *severity=3* for high/down). Optional; default is *severity=3*.

parameter: *descr* - description of the incident. Optional.

parameter: *validated* - automatically validates the incident. Requires master password authentication.

action=getinc

definition: returns the description and severity of current incidents on the current system

authentication: master password, or API key*, or system based authentication

parameter: *id* - the identification number for the system (mandatory)

PPMS Utility Management API Documentation - revision 2013.2

action=getuserrights

definition: returns the rights to use instruments for a given user. The return format is a letter defining the right level (D: deactivated, A: autonomous, N: novice, S: superuser) and the id number that identifies the system for the right.

authentication: master password based or API key based* authentication

parameter: *login* - the unique string identifier for the user (mandatory)

action=getsysrights

definition: returns the users that have rights associated with the current system. The return format is a letter defining the right level (D: deactivated, A: autonomous, N: novice, S: superuser) followed by a colon and the login string that identifies the user.

authentication: master password, or API key*, or system based authentication

parameter: *id* - the identification number for the system (mandatory)

action=setright

definition: adds into PPMS a right for a user to use a system

authentication: master password based or API key based* authentication

parameter: *id* - the identification number for the system (mandatory)

parameter: *login* - the unique string identifier for the user (mandatory)

parameter: *type* - the type of user right to set. Possible values are : *type=A* (Autonomous), *type=N* (Novice), *type=S* (Superuser), *type=D* (deactivated). Optional, default value if nothing is specified is A.

action=getprojects

definition: get the list of the projects. CSV format: columns: core facility ref., project name, phase, active, bcode, project ref., affiliation, project type

authentication: master password based or API key based* authentication

action=newproject* and *action=updproject

definition: create / update a project.

authentication: master password based or API key based* authentication

parameter: *projectid* - reference of the project (not available for creations, mandatory for updates)

parameter: *platformid* - reference of the core facility (mandatory for creations if no project type, not available for updates)

parameter: *name* - name of the project (mandatory for creations, optional for updates)

parameter: *bcode* - account number (optional for creations, optional for updates)

parameter: *phase* - phase of the project (optional for creations, optional for updates)

parameter: *affiliation* - affiliation of the project (optional for creations, optional for updates)

parameter: *ptype* - project type (mandatory for creations if no platformid, optional for updates)

parameter: *active* - active/deactive the project (not available for creations, optional for updates)

action=addprojectentry

definition: add an milestone/entry to a given project

authentication: master password based or API key based* authentication

parameter: *projectid* - reference of the project (mandatory)

parameter: *title* - title for the milestone/entry (mandatory)

parameter: *content* - content for the milestone/entry (mandatory)

action=getprojectusers

definition: get the list of the project members user (by login).

parameter: *projectid* - the project identification/reference number (mandatory)

authentication: master password based or API key based* authentication

action=getprojectgroups

definition: get the list of the project member groups (by unitlogin).

parameter: *projectid* - the project identification/reference number (mandatory)

authentication: master password based or API key based* authentication

PPMS Utility Management API Documentation - revision 2013.2

action=getorders

definition: get the list of the orders. CSV format: columns: core facility ref., order ref., user, group, project ref, quoted date, ordered date, completed date, status, invoiced

authentication: master password based or API key based* authentication

action=getmanifestlist

definition: get the list of manifests. CSV format: columns: manifest reference, manifest name, order ref. for manifest, core facility reference. headers include references to order and core facility.

authentication: master password based or API key based* authentication

action=getmanifest

definition: get a manifest. CSV format, columns: sampleID, plate, x, y, sample reference, any additional columns from template

authentication: master password based or API key based* authentication

action=addsampleresult

definition: attach a result or any value to a sample

authentication: master password based or API key based* authentication

parameter: *sampleID* - the PPMS reference for the sample (mandatory),

parameter: *title* - the name of the value (mandatory)

parameter: *value* - the value to add (mandatory).

action=getinvoicelist

definition: get the list of invoices currently in the PPMS database

authentication: master password based or API key based* authentication

action=getinvoice

definition: get a specific invoice. CSV format: column 1: account number, column 2: amount.

authentication: master password based or API key based* authentication

parameter: *invoiceid* - the reference for the invoice (mandatory)

action=getinvoicedetails

definition: get a specific invoice's details. CSV format.

authentication: master password based or API key based* authentication

parameter: *invoiceid* - the reference for the invoice (mandatory).

parameter: *bcode* - only get the invoice details for the account number *bcode* (optional),

parameter: *assistant* - adds assistant value for assisted session when *assistant=show*

output details for sessions: reference, user, group, project, affiliation, session type, system type, system, date, start time, duration (booked), duration (used), used by others, booked by others, cancelled, cancelled date, special env. 1, special env. 2, user comments, notes, account number, account number (subsidy), rebate, fee, subsidy, final amount.

output details for trainings: reference, user, group, affiliation, session type, system type, system, date, start time, duration,,,,,,, account number, account number (subsidy),,, subsidy, final amount.

output details for services/consumables: reference, user, group, project, affiliation, type,,, date completed,,,,,,,,,,,,, account number, account number (subsidy),,, subsidy, final amount.

output details for subsidies: reference,,,,, type,,,,,,,,,,,,, account number,,,,, final amount

action=getbcodes

definition: get the list of account numbers used by users, groups, projects and subsidies. CSV format: columns: account number, user, group, project, subsidy.

authentication: master password based or API key based* authentication

action=getvalidbcodes

definition: get the list of valid account numbers (if any have been added)

authentication: master password based or API key based* authentication

PPMS Utility Management API Documentation - revision 2013.2

action=addvalidbcode

definition: add an account number to the list of valid account numbers

authentication: master password based or API key based* authentication

parameter: *bcode* - the account number to add (mandatory)

parameter: *expirationdate* - an expiration date for the account number (option)

action=delvalidbcode

definition: remove an account number from the list of valid account numbers

authentication: master password based or API key based* authentication

parameter: *bcode* - the account number to remove (mandatory)



PPMS Utility Management API Documentation - revision 2013.2

Notes for generating API keys

To create API keys, you will need to log in PPMS in the administrative interface, as the super-administrator. A menu called "API" will give you access to API keys management.

You can delete unused API keys, and create new keys. When creating new keys, you will be asked the scope of access of the key: whenever the keys allows to edit or modify users and groups (user management option), and whenever the key has its rights restricted to a specific core facility.

Notes for user and group manipulation:

- When creating/updating a user, if the *unitlogin* supplied is not in the list of groups already in PPMS, a group will automatically be created with this unique identifier.
- If you try to update a user or a group that does not exist, then it will be created instead (if enough information is there).
- If you try to create a user or a group that already exists, then it will be updated instead.

Notes for developers:

- We **highly** recommend that you **do not work** on production PPMS databases at first. Please contact support@stratocore.com to get access to a complimentary test environment to develop and test your routines.

PPMS Utility Management API Documentation - revision 2013.2

Examples using the UNIX *curl* command line utility:

To URLEncode values you can use the following perl command for each parameters (only the value should be encoded) - most scripting or programming languages have a built-in feature to URLEncode values.:

```
echo "The Rockefeller University" | perl -MURI::Escape -lne 'print uri_escape($_)'  
result: The%20Rockefeller%20University
```

To add the "birc" group on the the server ppms.example.org (remove any line break ↵):

```
curl -d "action=newgroup&psapwd=ppmssecret&unitlogin=birc&  
unitname=Bio%20Imaging%20Resource%20Center&headname=North%2C%20Alison&  
heademail=northa@rockefeller.edu&institution=The%20Rockefeller%20University&ext=false"  
"https://ppms.example.org/pumapi/"
```

To add the user "John Smith" in the "birc" group (remove any line break ↵):

```
curl -d "action=newuser&psapwd=ppmssecret&login=jsmith&fname=John&lname=Smith&  
email=jsmith@example.org&phone=x7489&unitlogin=birc&pwd=secret" "https://ppms.example.org/pumapi/"
```

To update John Smith's phone and password (using a SSHA hash) (remove any line break ↵):

```
curl -d "action=upduser&psapwd=ppmssecret&login=jsmith&phone=x7487&  
pwd=%7BSSHA%2BhAgfFiMIQGo%2Bk3RxRazhCpVa6qE%3D" "https://ppms.example.org/pumapi/"
```

To get the current booking on a system (using system based auth):

```
curl -d "action=getbooking&id=4&code=123456" "https://ppms.example.org/pumapi/"
```

To get the current accounts number used in PPMS:

```
curl -d "action=getbcodes&psapwd=secret" "https://ppms.example.org/pumapi/"
```

Force the user jsmith to update his account number after next login:

```
curl -d "action=upduser&login=jsmith&mustchbcode=true&psapwd=secret" "https://ppms.example.org/pumapi/"
```