idoitAPI

unknown

Jul 10, 2022

CONTENTS:

idoitAPI is a Python Class to access the i-doit JSON-RPC API .

Note: This project is work in progress.

INDICES AND TABLES

- genindex

IDOITAPI

class idoit.**IdoitAPI**(*base_url*, *verify*, *language*, *username=None*, *password=None*, *apikey=None*)
Python3 class to access i-doit JSON-RPC API

API methods - Namespace idoit: Methods currently available in the i-doit JSON-RPC API

- idoit.search - `Implemented`
- idoit.version - `Implemented`
- idoit.constants - `Implemented`
- idoit.login - `Implemented`
- idoit.logout - `Implemented`

API methods - Namespace cmdb: Methods currently available in the i-doit JSON-RPC API

- cmdb.object.create - `Implemented`
- cmdb.object.read - `Implemented`
- cmdb.object.update - `Implemented`
- cmdb.object.delete - `Implemented`
- cmdb.object.recycle - `Implemented`
- cmdb.object.archive - `Not implemented`
- cmdb.object.purge - `Not implemented`
- cmdb.object.markAsTemplate - `Not implemented`
- cmdb.object.markAsMassChangeTemplate - `Not implemented`
- cmdb.objects.read - `Implemented`
- cmdb.category.save - `Implemented`
- cmdb.category.create - `Implemented but fiXXXme`
- cmdb.category.read - `Implemented`
- cmdb.category.update - `Not implemented`
- cmdb.category.delete - `Not implemented`
- cmdb.category.quickpurge - `Implemented – UNTESTED FIXXXME`
- cmdb.category.purge - `Implemented – UNTESTED FIXXXME`
- cmdb.category.recycle - `Not implemented`
- cmdb.category.archive - `Not implemented`

- cmdb.dialog.read - `Implemented`

- cmdb.dialog.create - `Not implemented`

- cmdb.dialog.update - `Not implemented`

- cmdb.dialog.delete - `Not implemented`

- cmdb.reports.read - `Not implemented`

Parameters

- **base_url** – `str`: Domain Name

- **verify** – `bool`: Verify SSL Connection?

- **language** – `str`: The Language your are using e.g. en or de

- **username** – `str`: your Username - can be left empty and set by environment variable IDOIT_USERNAME

- **password** – `str`: your Password - can be left empty and set by environment variable IDOIT_PASSWORD

- **apikey** – `str`: API Key for this i-doit instance - can be left empty and set by environment variable IDOIT_APIKEY

Raises

- **requests.HTTPError** – Raised by requests lib.

- **ValueError** – Raised when JSON-RPC PORT request returns not HTTP 200 statuscode.

- **SystemError** – Raised when credentials are missing (username, password or apikey).

log_json_request
bool: Set to *True* to log JSON requests send to i-doit.

return_query_do_not_send
bool: Guard to get JSON query from method instead of sending it. Can be used to get *obj_dict* for build_batch().

_api_login()
Login into i-doit and set session ID for later use

- Uses: `idoit.login`

api_logout()
Terminate active session to i-doit

- Uses: `idoit.logout`

build_batch(*obj_dict*)
Attach object to a list to be submited as batch request within a single call. Dataset ID equals to length of list, beginning with 1 as first ID as of JSON-RPC 2.0 IDs should not be null.

obj_dict can be created by calling the appropriate method with return_query_do_not_send guard set.

Parameters

- **grp_list** – `list`: List of JSON objects

- **obj_dict** – `dict`: JSON query / object

Returns List of JSON-RPC objects

create_category(*new_obj_id*, *category*, *data_dict*)
    Experimental

- Uses: `cmdb.category.create`

- **how to create unique obj_id?**

    Returns JSON object of response or raise exception.

create_object_by_type(*obj_type*, *title='Default Title - CHANGEME'*, *status=1*, *category=None*, *description=None*, *purpose=None*, *categories=None*)
    Create a new Object of a certain type

- Uses: `cmdb.object.create`

    Parameters

- **obj_type** – `str`: Object type constant to fetch.
- **title** – `str`: Descriptive title of object.
- **status_id** – `int`: CMDB status ID (planned, inoperative, etc.).
- **category** – `str`: **Category constant**.
- **description** – `str`: Description of object.
- **purpose** – `str`: Purpose (constant?) of object (production, test, dev, etc.).
- **categories** – `dict`: Categories to populate.

Some Object Types
- ??: int: 'C__OBJTYPE__PATCH_PANEL'
- 35: int: 'C__OBJTYPE__OPERATING_SYSTEM'
- 7: int: 'C__OBJTYPE__ROUTER'
- 6: int: 'C__OBJTYPE__SWITCH'
- 5: int: 'C__OBJTYPE__SERVER'

CMDB status IDs
- 1: `int`: planned
- 2: `int`: ordered
- 3: `int`: delivered
- 4: `int`: assembled
- 5: `int`: tested
- 6: `int`: in operation - constant 'C__CMDB_STATUS__IN_OPERATION'
- 7: `int`: defect
- 8: `int`: under repair
- 9: `int`: delivered from repair
- 10: `int`: inoperative
- 11: `int`: stored
- 12: `int`: scrapped
- 15: `int`: swapped

    Returns JSON object of response or raise exception.

delete_object(*obj_id*, *move_to='C__RECORD_STATUS__ARCHIVED'*)
    Delete an object from CMDB, per default delete = archived

- Uses: `cmdb.object.delete`

> **Parameters**
>
> - **obj_id** – `str`: Object ID of i-doit object
>
> - **move_to** – `str`: **Status constant** to move to
>
> **Returns** JSON object of response or raise exception.

Status constants
- "C__RECORD_STATUS__ARCHIVED": `str`: Archived object
- "C__RECORD_STATUS__DELETED": `str`: Mark object as deleted

find_host_ip_serial(*host=None*, *ip_addr=None*, *serial=None*)
> Query i-doit for a set of hostname, IP address and serial number in a single call.
>
> **Parameters**
>
> - **host** – `str`: hostname to look for, might be empty
>
> - **ip** – `str`: ip address to look for, might be empty
>
> - **serial** – `str`: serial number to look for, might be empty
>
> **Returns** Dictionary with given host, IP, serial as keys and a list of object IDs with matching results as value. Only keys with search results will be present.

find_rack(*r_title*)
> Query i-doit for an object type 'enclosure' with a given title :param r_title: `str`: Enclosure title
>
> **Returns** JSON object of response or raise exception.

get_all_os()
> Get a list of all available operating systems
>
> - filter type 35 = OS
>
> - filter status 2 = Normal (not archived or deleted)
>
> **Returns** JSON object of response or raise exception.

get_all_routers()
> Get a list of all devices in category Hardware->Routers
>
> - filter type 7 = Router
>
> - filter status 2 = Normal (not archived or deleted)
>
> **Returns** JSON object of response or raise exception.

get_all_servers()
> Get a list of all devices in category Hardware->Servers
>
> - filter type 5 = Server - constant 'C__OBJTYPE__SERVER'
>
> - filter status 2 = Normal (not archived or deleted)
>
> **Returns** JSON object of response or raise exception.

get_all_switches()
> Get a list of all devices in category Hardware->Switches
>
> - filter type 6 = Switches
>
> - filter status 2 = Normal (not archived or deleted)

Returns JSON object of response or raise exception.

get_category_from_object(*obj_id*, *category*)
    Read a certain category of an object

- Uses: `cmdb.category.read`

    Parameters

    - **obj_id** – `str`: Object ID of i-doit object.

    - **category** – `str`: **Category constant**.

    Returns JSON object of response or raise exception.

get_category_hostaddress()
    get structure of category host address

    Returns JSON object of response or raise exception.

get_constants()
    Fetch defined constants from i-doit

- Uses: `idoit.constants`

    Returns JSON object of response or raise exception.

get_custom_category_info(*cat_c_id*)
    get structure of a certain custom category

    Parameters **cat_c_id** – `str`: Custom category constant

    Returns JSON object of response or raise exception.

get_dropdown_values(*category*, *property_name*)
    get values from drop down menus like manufacturer, model, CPU type etc. Use
    get_global_category_info(<category>) to find the property_name

get_filtered_objects(*filter_dict*, *categories=None*)
    Fetch a list of objects filtered by values in filter_dict and/or categories list

- Uses: `cmdb.objects.read`

    Parameters

    - **filter_dict** – `dict`: Dictionary with keys to filter request

    - **categories** – `str`: List of **Category constant(s)**.

    Returns JSON object of response or raise exception.

get_global_category_info(*cat_g_id*)
    get structure of a certain global category

    Parameters **cat_g_id** – `str`: Global category constant

    Returns JSON object of response or raise exception.

get_idoit_version()
    Fetch information about i-doit and the current user

- Uses: `idoit.version`

Returns JSON object of response or raise exception.

get_ipv4_address(*obj_id*, *primary=True*, *fqdn=False*)
Fetch IPv4 Address from object. Per default the 'primary_hostaddress' will be returned.

Parameters

- **obj_id** – str: Object ID of i-doit object

- **primary** – bool: When primery=False first match for 'hostaddress' will be returned.

- **fqdn** – str: If fqdn is true a tupple (ip,fqdn) will be returned. **fqdn might be 'None'!**

Returns JSON object of response or raise exception.

get_object(*obj_id*)
Read general informations of an object specified by it's ID

- Uses: cmdb.object.read

Parameters obj_id – str: Object ID of i-doit object

Returns JSON object of response or raise exception.

get_object_type_categories_by_obj_type(*obj_type*)
Fetch all categories that belong to a certain object

Parameters obj_type – str: Object type constant to fetch

Returns JSON object of response or raise exception.

get_objects_by_type(*obj_type*, *status=2*, *title=None*, *limit=None*)
Fetch a list of objects filtered by obj_type, title, limit

- Uses: cmdb.objects.read

Parameters

- **obj_type** – int / str: Number or constant of object type

- **status** – int default 2 = Normal (not archived or deleted)

- **title** – str: Title of object

- **limit** – int / str: default no limit, can be fixed or with offset

limit:

- 1000: int: get 1000 objects

- '1000, 1000': str: with offset, here *get the next thousand object*

Returns JSON object of response or raise exception.

get_specific_category_info(*cat_s_id*)
get structure of a certain specific category

Parameters cat_s_id – str: Specific category constant

Returns JSON object of response or raise exception.

get_version()
Get version of this class

purge_object(*obj_id*, *category*, *entry*)
UNTESTED FIXXXME - Purge category entry of an object from CMDB

- Uses: `cmdb.category.purge`

    Parameters

    - **obj_id** – `int`: Object ID of i-doit object

    - **category** – `str`: Category constant

    - **entry** – `int`: Entry identificator

    Returns JSON object of response or raise exception.

quickpurge_object(*obj_id*, *category*, *cat_e_id*)
UNTESTED FIXXXME - Purge category entry of an object from CMDB - if quickpurge is active

- Uses: `cmdb.category.quickpurge`

    Parameters

    - **obj_id** – `int`: Object ID of i-doit object

    - **category** – `str`: Category constant

    - **cat_e_id** – `int`: Entry identificator

    Returns JSON object of response or raise exception.

recycle_object(*obj_id*)
Move object from archived or deleted back to normal

- Uses: `cmdb.object.recycle`

    Parameters obj_id – `str`: Object ID of i-doit object

    Returns JSON object of response or raise exception.

search_text(*text*)
Simple search - like in i-doit top right

- Uses: `idoit.search`

    Parameters text – `str`: Text to search for.

    Returns JSON object of response or raise exception.

send_batch()
Submit the currently queued requests and clear the list of queued requests.

    Returns Tuple with result of batch request and batch list that has been send.

send_rpc(*method*, *params_dict*, *header=None*)
Generic method to send json-rpc call to server. Only set method and method specific params. Parameters like **apikey** and **language** etc will be added.

    Parameters

    - **method** – `dict`: JSON-RPC method to use

    - **params_dict** – `dict`: Method specific parameters.

    - **header** – `dict`: Header if other than default needs to be used.

Returns JSON object of response or raise exception.

send_rpc_d(*data*, *batch=False*)

Generic method to send json-rpc call to server

Parameters

- **data** – dict: supply your own data dictionary

- **batch** – bool: RPC call is batch request.

Returns JSON object of response or raise exception or when batch is True dictionary with object IDs as keys.

update_cmdb_status(*obj_id*, *status_id*)

Update CMDB Status of an Object - status IDs are listed in `create_object_by_type()`

Parameters

- **obj_id** – str: Object ID of i-doit object.

- **status_id** – int: CMDB status ID (planned, inoperative, etc.).

Returns JSON object of response or raise exception.

update_object_category(*obj_id*, *category*, *data_dict*)

Update some entries of an object in a certain category.

- Uses: `cmdb.category.save`

Parameters

- **obj_id** – str: Object ID of i-doit object.

- **category** – str: **Category constant**.

- **data_dict** – dict: Dict with entries to update.

Returns JSON object of response or raise exception.

update_object_category_entry(*obj_id*, *category*, *entry_id*, *data_dict*)

Update an entry in a category of an object.

- Uses: `cmdb.category.save`

Parameters

- **obj_id** – str: Object ID of i-doit object.

- **category** – str: **Category constant**.

- **entry_id** – str: ID of entry to be updated.

- **data_dict** – dict: Dict with entries to update.

Returns JSON object of response or raise exception.

update_object_title(*obj_id*, *title*)

Update the title of a certain object

- Uses: `cmdb.object.update`

Parameters

- **obj_id** – str: Object ID of i-doit object.

- **title** – str: Descriptive title of object.

Returns JSON object of response or raise exception.

MODULES

3.1 idoitSettings

A `dict` which should be named **settings** as configuration file for a specific i-doit instance. Pass it as **parameter** when creating a new object from this class.

username `` ``str`` ``
> your Username

password `` ``str`` ``
> your Password

apikey `` ``str`` ``
> API Key for this i-doit instance

base_url `` ``str`` ``
> Domain Name

verify `` ``bool`` ``
> Verify SSL Connection?

language `` ``str`` ``
> The Language your are using e.g. en or de

PYTHON MODULE INDEX

i