
dnac_api

Release 0.0.1

Kyle Kowalczyk

Aug 17, 2019

CONTENTS:

1	Project Goals	1
2	Installation Instructions	3
3	Version 1.1	5
3.1	Network Discovery	5
3.2	Network Device	10
3.3	Network Host	12
3.4	System	13
4	Indices and tables	15
	Index	17

PROJECT GOALS

The goal of this project is to have a community created and supported SDK to the Cisco DNA Center Rest API that includes support for all major versions of the API starting with version 1.1.

INSTALLATION INSTRUCTIONS

As of now the project has not been put on PyPi but will be in the future.

To install the project as of now download the project to you computer and install by running the setup.py file

```
`python setup.py install`
```


Documentation for Version 1.1 of the DNA Center API

3.1 Network Discovery

class dnac_api.v1_1.**NetworkDiscovery**(*dnac_server, username, password, verify=False*)

API to the Network Discovery section of the Cisco DNA Center REST API Version 1.1. Info on raw API calls can be found at <https://developer.cisco.com/site/dna-center-rest-api/?version=1.1>

cli (**kwargs)

This method is used to get global CLI credentials. This method gets to the api route /global-credential

Parameters **kwargs** – See Keyword Arguments below for available keyword arguments.

Keyword Arguments

- **sortBy** (str) – define sorting on the data returned
- **order** (str) – Define order on data returned

Returns Global CLI Credentials

create_cli_credentials (username, password, enable_password, comments, description)

Sends POST request to /global-credential/cli to create new cli credential entry.

Parameters

- **username** – Username used in the credentials
- **password** – Password used in the credentials
- **enable_password** – Enable password used in the credentials
- **comments** – Brief Comment
- **description** – Brief Description

Returns

create_http_read_credentials (username, password, port, secure, comments, description)

Creates new HTTP Read credentials by submitting a post request to /global-credential/http-read

Parameters

- **username** (str) –
- **password** (str) –
- **port** (int) –
- **secure** (bool) –
- **comments** (str) –
- **description** (str) –

Returns**Return type** ResponseObject**create_http_write_credentials** (*username, password, port, secure, comments, description*)

Creates new HTTP Write credentials by submitting a post request to /global-credential/http-write

Parameters

- **username** (*str*) –
- **password** (*str*) –
- **port** (*int*) –
- **secure** (*bool*) –
- **comments** (*str*) –
- **description** (*str*) –

Returns**Return type** ResponseObject**create_netconf_credentials** (*netconf_port, comments, description*)

Creates new netconf credentials by submitting a post request to /global-credential/netconf

Parameters

- **netconf_port** –
- **comments** –
- **description** –

Returns**create_snmpv2_read** (*community_string, comments, description*)

Creates new SNMPv2 Read credentials by submitting a post request to /global-credential/snmpv2-read-community

Parameters

- **community_string** (*str*) – Community string
- **comments** (*str*) – Comments
- **description** (*str*) – Description

Returns**create_snmpv2_write** (*community_string, comments, description*)

Creates new SNMPv2 Write credentials by submitting a post request to /global-credential/snmpv2-write-community

Parameters

- **community_string** (*str*) – Community string
- **comments** (*str*) – Comments
- **description** (*str*) – Description

Returns**create_snmpv3_credentials** (*privacy_password, privacy_type, snmp_mode, auth_type, auth_password, username, comments, description*)

Creates new SNMPv3 credentials by submitting a post request to /global-credential/snmpv3

Parameters

- **privacy_password** –
- **privacy_type** –
- **snmp_mode** –
- **auth_type** –
- **auth_password** –
- **username** –
- **comments** –

- **description** –

Returns

credential_sub_type(*credential_id*)

This method is used to get global credential for the given credential sub type

Parameters **credential_id** – Credential type as CLI / SN-MPV2_READ_COMMUNITY / SNMPV2_WRITE_COMMUNITY / SNMPV3 / HTTP_WRITE / HTTP_READ / NETCONF

Returns

discovery_by_id(*discovery_id*)

Gets discovery by specified ID, sends a GET request to /discovery/{id}

Parameters **discovery_id** –

Returns

discovery_jobs_by_id(*discovery_id*, ****kwargs**)

Returns discovery jobs by specified ID.

Parameters

- **discovery_id** –
- **kwargs** – See Keyword Arguments below

Keyword Arguments

- *offset* (str)
- *limit* (str)
- *ipAddress* (str)

Returns

discovery_jobs_for_ip(*ip*, ****kwargs**)

Return discovery jobs associated with an IP address

Parameters

- **ip** – IP address to get discovery jobs for.
- **kwargs** – See Keyword Arguments below

Keyword Arguments

- *offset* (str)
- *limit* (str)
- *name* (str)

Returns Discovery Jobs

http_read(****kwargs**)

This method is used to get global HTTP Read credentials. This method gets to the api route /global-credential

Parameters **kwargs** – See Keyword Arguments below for available keyword arguments.

Keyword Arguments

- *sortBy* (str) – define sorting on the data returned
- *order* (str) – Define order on data returned

Returns Global HTTP Read Credentials

http_write(****kwargs**)

This method is used to get global HTTP Write credentials. This method gets to the api route /global-credential

Parameters **kwargs** – See Keyword Arguments below for available keyword arguments.

Keyword Arguments

- *sortBy* (str) – define sorting on the data returned
- *order* (str) – Define order on data returned

Returns Global HTTP Write Credentials

netconf (**kwargs)

This method is used to get global Netconf credentials. This method gets to the api route /
global-credential

Parameters **kwargs** – See Keyword Arguments below for available keyword arguments.

Keyword Arguments

- *sortBy* (str) – define sorting on the data returned
- *order* (str) – Define order on data returned

Returns Global Netconf Credentials

network_devices_from_discovery_by_filters (discovery_id, **kwargs)

Parameters

- **discovery_id** –
- **kwargs** – See Keyword Arguments below

Keyword Arguments

- *offset* (str)
- *limit* (str)
- *ipAddress* (str)
- *taskId* (str)
- *sortBy* (str)
- *sortOrder* (str)
- *ipAddress* (str)
- *pingStatus* (str)
- *snmpStatus* (str)
- *cliStatus* (str)
- *netconfStatus* (str)
- *httpStatus* (str)

Returns

num_network_devices_in_discovery (discovery_id)

Get number of network devices in a discovery. Sends a post request to /discovery/
{discovery-id}/network-device/count

Parameters **discovery_id** – Discovery ID

Returns number of network devices in discovery

property number_of_discoveries

Gets number of discoveries by sending a get request to /discovery/count

Returns

physical_topology ()

Sends get request to /topology/physical-topology to get the physical topology data.

Returns Physical topology data

snmpv2_read (**kwargs)

This method is used to get global SNMPv2 Read credentials. This method gets to the api route
/global-credential

Parameters **kwargs** – See Keyword Arguments below for available keyword arguments.

Keyword Arguments

- *sortBy* (str) – define sorting on the data returned
- *order* (str) – Define order on data returned

Returns Global SNMPv2 Read Credentials

snmpv2_write (**kwargs)

This method is used to get global SNMPv2 Write credentials. This method gets to the api route
/global-credential

Parameters *kwargs* – See Keyword Arguments below for available keyword arguments.

Keyword Arguments

- *sortBy* (*str*) – define sorting on the data returned
- *order* (*str*) – Define order on data returned

Returns Global SNMPv2 Write Credentials

snmpv3 (***kwargs*)

This method is used to get global SNMPv3 credentials. This method gets to the api route /global-credential

Parameters *kwargs* – See Keyword Arguments below for available keyword arguments.

Keyword Arguments

- *sortBy* (*str*) – define sorting on the data returned
- *order* (*str*) – Define order on data returned

Returns Global SNMPv3 Credentials

start_discovery_process (***kwargs*)

Initiates discovery with the given parameters

Parameters *kwargs* – See Keyword Arguments below

Keyword Arguments

- *snmpMode* (*str*)
- *netconfPort* (*str*)
- *preferredMgmtIPMethod* (*str*)
- *name* (*str*)
- *globalCredentialIdList* (*list* (*str*))
- **httpReadCredential: (dict)**
 - *port* (*integer*)
 - *secure* (*boolean*)
 - *username* (*string*)
 - *password* (*string*)
 - *comments* (*string*)
 - *credentialType* (*string*)
 - *description* (*string*)
 - *id* (*string*)
 - *instanceUuid* (*string*)
- **httpWriteCredential: (dict)**
 - *port* (*integer*)
 - *secure* (*boolean*)
 - *username* (*string*)
 - *password* (*string*)
 - *comments* (*string*)
 - *credentialType* (*string*)
 - *description* (*string*)
 - *id* (*string*)
 - *instanceUuid* (*string*)
- *parentDiscoveryId* (*str*)
- *snmpROCommunityDesc* (*str*)
- *snmpRWCommunityDesc* (*str*)
- *snmpUserName* (*str*)
- *timeout* (*int*)
- *snmpVersion* (*str*)
- *ipAddressList* (*str*)
- *cdpLevel* (*int*)
- *enablePasswordList*: (*list* (*string*))

- *ipFilterList*: (list (string))
- *passwordList*: (list (string))
- *protocolOrder* (str)
- *reDiscovery* (bool)
- *retry* (int)
- *snmpAuthPassphrase* (str)
- *snmpAuthProtocol* (str)
- *snmpPrivPassphrase* (str)
- *snmpPrivProtocol* (str)
- *snmpROCommunity* (str)
- *snmpRWCommunity* (str)
- *userNameList* (list (string))
- *discoveryType* (str)

Returns

3.2 Network Device

```
class dnac_api.v1_1.NetworkDevice (dna_server, username, password, verify=False)
```

```
devices_at_location (location_id)
```

Get devices at location by location ID. Sends get request to /network-device/location/{location_id}

Parameters *location_id* – ID of location to search by

Returns List of devices

```
property devices_with_location
```

Get location data about devices. Sends get request to /network-device/location

Returns

```
location_by_device_id (device_id)
```

Returns the location of a device when specifying the device ID. Sends get request to /network-device/{device_id}/location

Parameters *device_id* – ID of network device

Returns Location information

```
module_info_by_id (module_id)
```

Gets info related to module by the ID of the module. Sends get request to /network-device/module/{module_id}

Parameters *module_id* – ID of module to return data about

Returns Module info

```
modules_in_device (device_id, **kwargs)
```

Returns the modules in device. Sends a get request to /network-device/module

Parameters

- *device_id* – ID of network device
- *kwargs* – See Keyword Arguments below

Keyword Arguments

- *limit* (str)
- *offset* (str)
- *nameList* (str)
- *vendorEquipmentTypeList* (str)
- *partNumberList* (str)
- *operationalStateCodeList* (str)
- *deviceId* (str)

Returns**network_device_brief_by_id** (id)Similar to `network_device_by_id(id)` but returns brief data.**Parameters** `id` – ID to search by**Returns** Network device data - brief**network_device_by_id** (id)Gets network device by ID by sending a get request to `/network-device/{id}`**Parameters** `id` – ID to search by**Returns** Network device**network_device_by_ip** (ip)Sends get request to `/network-device/ip-address/{ip}`**Parameters** `ip` – IP address to search by**Returns** Network device associated with IP**network_device_by_serial_number** (device_serial_number)Gets network device by serial number by sending a get request to `/network-device/serial-number/{sn}`**Parameters** `device_serial_number` –**Returns****property network_device_count**gets the number of network devices by sending a get request to `/network-device/count`**Returns** number of network devices**network_devices** (id=None)

Returns the network devices

Parameters `id` – Filters devices returned that match ID**Returns****number_of_modules_in_device** (device_id, **kwargs)Returns the number of modules in a device. Sends get request to `/network-device/module/count`**Parameters**

- **device_id** – Device ID of network device
- **kwargs** – See Keyword Arguments below

Keyword Arguments

- *deviceId* (str)
- *nameList* (str)
- *vendorEquipmentTypeList* (str)
- *partNumberList* (str)
- *operationalStateCodeList* (str)

Returns

3.3 Network Host

```
class dnac_api.v1_1.NetworkHost (dna_server, username, password, verify=False)
```

```
host_by_id (id)
```

Returns host by ID. Sends a get request to /host/{id}

Parameters *id* – ID of host

Returns

```
hosts_by_filter (**kwargs)
```

Parameters *kwargs* – See Keyword Arguments below for available keyword arguments.

Keyword Arguments

- *limit* (str)
- *offset* (str)
- *sortBy* (str)
- *order* (str)
- *hostName* (str)
- *hostMac* (str)
- *hostType* (str)
- *connectedInterfaceName* (str)
- *hostIp* (str)
- *connectedNetworkDeviceIpAddress* (str)
- *connectedNetworkDeviceName* (str)
- *hostDeviceType* (str)
- *subType* (str) Available values: 'UNKNOWN' or 'IP_PHONE' or 'TELEPRESENCE' or 'VIDEO_SURVEILLANCE_IP_CAMERA' or 'VIDEO_ENDPOINT'. Only exact match filtering supported on this field

```
num_of_hosts (**kwargs)
```

Returns number of hosts. Sends get request to /host/count

param kwargs See Keyword Arguments below for available keyword arguments.

Keyword Arguments

- *limit* (str)
- *hostName* (str)
- *hostMac* (str)
- *hostType* (str)
- *connectedInterfaceName* (str)
- *hostIp* (str)
- *connectedNetworkDeviceIpAddress* (str)
- *connectedNetworkDeviceName* (str)
- *hostDeviceType* (str)
- *subType* (str)

Returns

3.4 System

```
class dnac_api.v1_1.System(dna_server, username, password, verify=False)
```

property available_namespaces

Lists available namespaces. Sends get request to /file/namespaces

Returns**file_checksum_by_field** (field)

Gets file checksum by field. Sends a get request to /file/{field}/checksum

Parameters field –**Returns** File checksum**files_under_namespace** (namespace)

Gets files located under namespace. sends get request to /file/namespaces/{namespace}

Parameters namespace –**Returns** Files

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

A

`available_namespaces()` (*dnac_api.v1_1.System* property), 13

C

`cli()` (*dnac_api.v1_1.NetworkDiscovery* method), 5
`create_cli_credentials()` (*dnac_api.v1_1.NetworkDiscovery* method), 5
`create_http_read_credentials()` (*dnac_api.v1_1.NetworkDiscovery* method), 5
`create_http_write_credentials()` (*dnac_api.v1_1.NetworkDiscovery* method), 6
`create_netconf_credentials()` (*dnac_api.v1_1.NetworkDiscovery* method), 6
`create_snmpv2_read()` (*dnac_api.v1_1.NetworkDiscovery* method), 6
`create_snmpv2_write()` (*dnac_api.v1_1.NetworkDiscovery* method), 6
`create_snmpv3_credentials()` (*dnac_api.v1_1.NetworkDiscovery* method), 6
`credential_sub_type()` (*dnac_api.v1_1.NetworkDiscovery* method), 7

D

`devices_at_location()` (*dnac_api.v1_1.NetworkDevice* method), 10
`devices_with_location()` (*dnac_api.v1_1.NetworkDevice* property), 10
`discovery_by_id()` (*dnac_api.v1_1.NetworkDiscovery* method), 7
`discovery_jobs_by_id()` (*dnac_api.v1_1.NetworkDiscovery* method), 7
`discovery_jobs_for_ip()` (*dnac_api.v1_1.NetworkDiscovery* method), 7

F

`file_checksum_by_field()` (*dnac_api.v1_1.System* method), 13
`files_under_namespace()` (*dnac_api.v1_1.System* method), 13

H

`host_by_id()` (*dnac_api.v1_1.NetworkHost* method), 12
`hosts_by_filter()` (*dnac_api.v1_1.NetworkHost* method), 12
`http_read()` (*dnac_api.v1_1.NetworkDiscovery* method), 7
`http_write()` (*dnac_api.v1_1.NetworkDiscovery* method), 7

L

`location_by_device_id()` (*dnac_api.v1_1.NetworkDevice* method), 10

M

`module_info_by_id()` (*dnac_api.v1_1.NetworkDevice* method), 10
`modules_in_device()` (*dnac_api.v1_1.NetworkDevice* method), 10

N

`netconf()` (*dnac_api.v1_1.NetworkDiscovery* method), 7
`network_device_brief_by_id()` (*dnac_api.v1_1.NetworkDevice* method), 11
`network_device_by_id()` (*dnac_api.v1_1.NetworkDevice* method), 11
`network_device_by_ip()` (*dnac_api.v1_1.NetworkDevice* method), 11
`network_device_by_serial_number()` (*dnac_api.v1_1.NetworkDevice* method), 11
`network_device_count()` (*dnac_api.v1_1.NetworkDevice* property), 11

`network_devices()`
 (*dnac_api.v1_1.NetworkDevice method*),
 11
`network_devices_from_discovery_by_filters()`
 (*dnac_api.v1_1.NetworkDiscovery method*), 8
`NetworkDevice` (*class in dnac_api.v1_1*), 10
`NetworkDiscovery` (*class in dnac_api.v1_1*), 5
`NetworkHost` (*class in dnac_api.v1_1*), 12
`num_network_devices_in_discovery()`
 (*dnac_api.v1_1.NetworkDiscovery method*), 8
`num_of_hosts()` (*dnac_api.v1_1.NetworkHost*
 method), 12
`number_of_discoveries()`
 (*dnac_api.v1_1.NetworkDiscovery property*), 8
`number_of_modules_in_device()`
 (*dnac_api.v1_1.NetworkDevice method*),
 11

P

`physical_topology()`
 (*dnac_api.v1_1.NetworkDiscovery method*), 8

S

`snmpv2_read()` (*dnac_api.v1_1.NetworkDiscovery*
 method), 8
`snmpv2_write()` (*dnac_api.v1_1.NetworkDiscovery*
 method), 8
`snmpv3()` (*dnac_api.v1_1.NetworkDiscovery method*),
 9
`start_discovery_process()`
 (*dnac_api.v1_1.NetworkDiscovery method*), 9
`System` (*class in dnac_api.v1_1*), 13