
Python Mano Wrappers Documentation

Ashwin, Bhargavi, Deeksha

Apr 27, 2019

CONTENTS

1	Introduction	1
2	Example usage	3
3	Design Overview	5
3.1	Component Description	5
4	References	33
5	TODO	35
	Python Module Index	37

INTRODUCTION

Python MANO Wrappers (PMW) is a uniform python wrapper for various implementations of NFV Management and Network Orchestration (MANO) REST APIs. This project is developed as part of the project group SCrAMbLE at the University of Paderborn for the 5G-PICTURE project ([GitHub](#)).

PMW is easy to use and well documented. Code usage examples are available along with the function documentation.

PMW can be installed using pip by using pip.

```
pip install python-mano-wrappers
```


EXAMPLE USAGE

```
import wrappers

username = "admin"
password = "admin"
mano = "osm"
# mano = "sonata"
host = "vm-hadik3r-05.cs.uni-paderborn.de"

if mano == "osm":
    _client = wrappers.OSMClient.Auth(host)
elif mano == "sonata":
    _client = wrappers.SONATAClient.Auth(host)

response = _client.auth(
    username=username, password=password)

print(response)
```


DESIGN OVERVIEW

The design goal of PMW is to provide a unified, convenient and standards oriented access to MANO API.

To achieve this, an Abstract Base Class is defined which follows the conventions from the ETSI GS NFV-SOL 005 (SOL005) RESTful protocols specification.

3.1 Component Description

3.1.1 CommonInterface

The Abstract Base Class is called “CommonInterface” and it is divided into different sections as per the SOL005 into the following:

- **auth.py**: Authorization API
- **nsd.py**: NSD Management API
- **nsfm.py**: NS Fault Management API
- **nslcm.py**: Lifecycle Management API
- **nspm.py**: NS Performance Management API
- **vnfpkgm.py**: VNF Package Management API

To add python wrapper support for a specific MANO, the developer should inherit the Abstract Base Class and implement all the methods defined in the class which are supported by the target MANO.

CommonInterface package

wrappers.CommonInterface.auth

```
class wrappers.CommonInterface.auth.CommonInterfaceAuth
    Bases: abc.ABC

    Authorization API

    auth ()
        Authorization API

        POST method which returns an authorization token to be used by other calls.
```

wrappers.CommonInterface.nsd

Common Interface - nsd

Reference interface to implement REST API Wrappers for MANO Frameworks Defined according to the ETSI GS NFV-SOL 005 V2.4.1 (2018-02).

Defines abstract methods which are to be implemented by the wrappers.

class wrappers.CommonInterface.nsd.CommonInterfaceNsd

Bases: abc.ABC

NSD Management Interfaces

Base: {apiRoot}/nsd/v1

delete_ns_descriptors_nsdinfoId ()

NSD Management Interface - Individual NS Descriptor

/ns_descriptors/{nsdInfoId}: DELETE - Delete the content of NSD

delete_pnf_descriptors_pnfdinfoId (pnfdInfoId)

NSD Management interface - Individual PNF Descriptor

/pnf_descriptors/pnfdinfoId: DELETE - Delete an individual PNF descriptor resource.

delete_subscriptions_subscriptionId (subscriptionId)

NSD Management interface - Individual Subscription

/subscriptions/subscriptionId: DELETE - Terminate a subscription.

get_ns_descriptors ()

NSD Management Interface - NS Descriptors

/ns_descriptors: GET - Query information about multiple NS descriptor resources.

get_ns_descriptors_nsd_content ()

NSD Management Interface - NSD Content

/ns_descriptors/{nsdInfoId}/nsd_c: GET - Fetch the content of a NSD.

get_ns_descriptors_nsdinfoId ()

NSD Management Interface - Individual NS Descriptor

/ns_descriptors/{nsdInfoId}: Read information about an individual NS descriptor resource.

get_pnf_descriptors ()

NSD Management interface - PNF Descriptors

/pnf_descriptors: GET - Query information about multiple PNF descriptor resources.

get_pnf_descriptors_pnfd_content (pnfdInfoId)

NSD Management interface - PNFD Content

/pnf_descriptors/pnfdinfoId/pnfd_content: GET - Fetch the content of a PNFD.

get_pnf_descriptors_pnfdinfoId (pnfdInfoId)

NSD Management interface - Individual PNF Descriptor

/pnf_descriptors/pnfdinfoId: GET - Read an individual PNFD resource.

get_subscriptions (subscriptionId)

NSD Management interface - Subscriptions

/subscriptions: GET - Query multiple subscriptions.

get_subscriptions_subscriptionid (*subscriptionid*)
 NSD Management interface - Individual Subscription
/subscriptions/subscriptionId: GET - Read an individual subscription resource

patch_ns_descriptors_nsdinfoId ()
 NSD Management Interface - Individual NS Descriptor
/ns_descriptors/{nsdInfoId}: PATCH - Modify the operational state and/or the user defined data of an individual NS descriptor resource.

patch_pnf_descriptors_pnfinfoId (*pnfdInfoId*)
 NSD Management interface - Individual PNF Descriptor
/pnf_descriptors/pnfinfoId: PATCH - Modify the user defined data of an individual PNF descriptor resource.

post_ns_descriptors ()
 NSD Management Interface - NS Descriptors
/ns_descriptors: POST - Create a new NS descriptor resource.

post_pnf_descriptors ()
 NSD Management interface - PNF Descriptors
/pnf_descriptors: POST - Create a new PNF descriptor resource.

post_subscriptions (*pnfdInfoId*)
 NSD Management interface - Subscriptions
/subscriptions: POST - Subscribe to NSD and PNFD change notifications.

put_ns_descriptors_nsd_content ()
 NSD Management Interface - NSD Content
/ns_descriptors/{nsdInfoId}/nsd_c: PUT - Upload the content of NSD

put_pnf_descriptors_pnfd_content (*pnfdInfoId*)
 NSD Management interface - PNFD Content
/pnf_descriptors/pnfinfoId/pnfd_content: PUT - Upload the content of a PNFD.

wrappers.CommonInterface.nsfm

Common Interface - nsfm

Reference interface to implement REST API Wrappers for MANO Frameworks Defined according to the ETSI GS NFV-SOL 005 V2.4.1 (2018-02).

Defines abstract methods which are to be implemented by the wrappers.

```
class wrappers.CommonInterface.nsfm.CommonInterfaceNsfm
    Bases: abc.ABC

    NS Fault Management interface

    Base: {apiRoot}/nsfm/v1

delete_nsfm_subscriptions_subscriptionid (subscriptionId)
    NS Fault Management interface - Individual subscription

/nsfm_subscriptions_subscriptionid: DELETE - Terminate a subscription.
```

get_nsfm_alarms ()
NS Fault Management interface - Alarms
/nsfm_alarms: GET - Query alarms related to NS instances.

get_nsfm_alarms_alarmid (*alarmId*)
NS Fault Management interface - Individual alarm
/nsfm_alarms_alarmid: GET - Read individual alarm.

get_nsfm_subscriptions ()
NS Fault Management interface - Subscriptions
/nsfm_subscriptions: GET - Query multiple subscriptions.

get_nsfm_subscriptions_subscriptionid (*subscriptionId*)
NS Fault Management interface - Individual subscription
/nsfm_subscriptions_subscriptionid: GET - Read an individual subscription.

patch_nsfm_alarms_alarmid (*alarmId*)
NS Fault Management interface - Individual alarm
/nsfm_alarms_alarmid: PATCH - Acknowledge individual alarm.

post_nsfm_subscriptions ()
NS Fault Management interface - Subscriptions
/nsfm_subscriptions: POST - Subscribe to alarms related to NSs.

wrappers.CommonInterface.nslcm

Common Interface - nslcm

Reference interface to implement REST API Wrappers for MANO Frameworks Defined according to the ETSI GS NFV-SOL 005 V2.4.1 (2018-02).

Defines abstract methods which are to be implemented by the wrappers.

```
class wrappers.CommonInterface.nslcm.CommonInterfaceNslcm
    Bases: abc.ABC

    Lifecycle Management interface

    Base: {apiRoot}/nslcm/v1

delete_ns_instances_nsinstanceid (nsInstanceId)
    NS Lifecycle Management interface - Individual NS instance
    /ns_instances_nsinstanceid: DELETE - Delete NS instance resource.

delete_ns_lcm_subscriptions_subscriptionid (subscriptionId)
    NS Lifecycle Management interface - Individual subscription
    /ns_lcm_subscriptions_subscriptionid: DELETE - Terminate a subscription.

get_ns_instances ()
    NS Lifecycle Management interface - NS instances
    /ns_instances: GET - Query multiple NS instances.

get_ns_instances_nsinstanceid (nsInstanceId)
    NS Lifecycle Management interface - Individual NS instance
    /ns_instances_nsinstanceid: GET - Read an individual NS instance resource.
```

get_ns_lcm_op_ops ()
 NS Lifecycle Management interface - NS lifecycle operation occurrences
/ns_lcm_op_ops: GET - Query multiple NS LCM operation occurrences.

get_ns_lcm_op_ops_nslcmopoccid (nsLcmOpOccId)
 NS Lifecycle Management interface - Individual NS lifecycle operation occurrence
/ns_lcm_op_ops_nslcmopoccid: GET - Read an individual NS LCM operation occurrence resource.

get_ns_lcm_subscriptions ()
 NS Lifecycle Management interface - Subscriptions
/ns_lcm_subscriptions: GET - Query multiple subscriptions.

get_ns_lcm_subscriptions_subscriptionid (subscriptionId)
 NS Lifecycle Management interface - Individual subscription
/ns_lcm_subscriptions_subscriptionid: GET - Read an individual subscription resource.

post_ns_instances ()
 NS Lifecycle Management interface - NS instances
/ns_instances: POST - Create a NS instance resource.

post_ns_instances_nsinstanceid_heal (nsInstanceId)
 NS Lifecycle Management interface - Heal NS task
/ns_instances_nsinstanceid_heal: POST - Heal a NS instance.

post_ns_instances_nsinstanceid_instantiate (nsInstanceId)
 NS Lifecycle Management interface - Instantiate NS task
/ns_instances_nsinstanceid_instantiate: POST - Instantiate a NS instance.

post_ns_instances_nsinstanceid_scale (nsInstanceId)
 NS Lifecycle Management interface - Scale NS task
/ns_instances_nsinstanceid_scale: POST - Scale a NS instance.

post_ns_instances_nsinstanceid_terminate (nsInstanceId)
 NS Lifecycle Management interface - Terminate NS task
/ns_instances_nsinstanceid_terminate: POST - Terminate a NS instance.

post_ns_instances_nsinstanceid_update (nsInstanceId)
 NS Lifecycle Management interface - Update NS task
/ns_instances_nsinstanceid_update: POST - Updates a NS instance.

post_ns_lcm_op_occs_nslcmopoccid_cancel (nsLcmOpOccId)
 NS Lifecycle Management interface - Cancel operation task
/ns_lcm_op_occs_nslcmopoccid_cancel: POST - Cancel a NS lifecycle management operation occurrence.

post_ns_lcm_op_occs_nslcmopoccid_continue (nsLcmOpOccId)
 NS Lifecycle Management interface - Continue operation task
/ns_lcm_op_occs_nslcmopoccid_continue: POST - Continue a NS lifecycle management operation occurrence.

post_ns_lcm_op_occs_nslcmopoccid_fail (nsLcmOpOccId)
 NS Lifecycle Management interface - Fail operation task

/ns_lcm_op_occs_nslcmopoccid_fail: POST - Mark a NS lifecycle management operation occurrence as failed.

post_ns_lcm_op_occs_nslcmopoccid_retry (*nsLcmOpOccId*)

NS Lifecycle Management interface - Retry operation task

/ns_lcm_op_occs_nslcmopoccid_retry: POST - Retry a NS lifecycle management operation occurrence.

post_ns_lcm_op_occs_nslcmopoccid_rollback (*nsLcmOpOccId*)

NS Lifecycle Management interface - Rollback operation task

/ns_lcm_op_occs_nslcmopoccid_rollback: POST - Rollback a NS lifecycle management operation occurrence.

post_ns_lcm_subscriptions ()

NS Lifecycle Management interface - Subscriptions

/ns_lcm_subscriptions: POST - Subscribe to NS lifecycle change notifications.

wrappers.CommonInterface.nspm

Common Interface - nspm

Reference interface to implement REST API Wrappers for MANO Frameworks Defined according to the ETSI GS NFV-SOL 005 V2.4.1 (2018-02).

Defines abstract methods which are to be implemented by the wrappers.

class wrappers.CommonInterface.nspm.CommonInterfaceNspm

Bases: abc.ABC

NS Performance Management Interface

Base: {apiRoot}/nspm/v1

delete_pm_jobs_pmjobid (*pmJobId*)

NS Performance Management Interface - Individual PM job

/pm_jobs_pmjobid: DELETE - Delete a PM job

delete_pm_subscriptions_subscriptionid (*subscriptionId*)

NS Performance Management Interface - Individual

/pm_subscriptions_subscriptionid: DELETE - Terminate a subscription

delete_pm_thresholds_thresholdid (*thresholdId*)

NS Performance Management Interface - Individual threshold

/pm_thresholds_thresholdid: DELETE - Delete a threshold

get_pm_jobs ()

NS Performance Management Interface - PM jobs

/pm_jobs: GET - Query PM jobs

get_pm_jobs_pmjobid (*pmJobId*)

NS Performance Management Interface - Individual PM job

/pm_jobs_pmjobid: GET - Read a single PM job

get_pm_jobs_pmjobid_reports_reportid (*pmJobId, reportId*)

NS Performance Management Interface - Individual performance report

/pm_jobs_pmjobid_reports_reportid: GET - Read an individual performance report

get_pm_subscriptions()
NS Performance Management Interface - Subscriptions
/pm_subscriptions: GET - Query PM related subscriptions

get_pm_subscriptions_subscriptionid(subscriptionId)
NS Performance Management Interface - Individual
/pm_subscriptions_subscriptionid: GET - Query a single PM related subscription

get_pm_thresholds()
NS Performance Management Interface - Thresholds
/pm_thresholds: GET - Query thresholds

get_pm_thresholds_thresholdid(thresholdId)
NS Performance Management Interface - Individual threshold
/pm_thresholds_thresholdid: GET - Query a single threshold

post_pm_jobs()
NS Performance Management Interface - PM jobs
/pm_jobs: POST - Create a PM job

post_pm_subscriptions()
NS Performance Management Interface - Subscriptions
/pm_subscriptions: POST - Subscribe to PM notifications

post_pm_thresholds()
NS Performance Management Interface - Thresholds
/pm_thresholds: POST - Create a threshold

wrappers.CommonInterface.sonpackage

Common Interface - nslcm

Reference interface to implement REST API Wrappers for MANO Frameworks Defined according to the ETSI GS NFV-SOL 005 V2.4.1 (2018-02).

Defines abstract methods which are to be implemented by the wrappers.

```
class wrappers.CommonInterface.sonpackage.CommonInterfaceSonPackage
    Bases: abc.ABC

    delete_son_packages_PackageId(PackageId)
        Delete a particular sonata package

    get_son_packages()
        Receive the list of stored packages

    get_son_packages_PackageId(PackageId)
        Receive a particular stored packages

    post_son_packages()
        Send a particular package
```

wrappers.CommonInterface.vnfpgm

Common Interface - vnfpgm

Reference interface to implement REST API Wrappers for MANO Frameworks Defined according to the ETSI GS NFV-SOL 005 V2.4.1 (2018-02).

Defines abstract methods which are to be implemented by the wrappers.

```
class wrappers.CommonInterface.vnfpgm.CommonInterfaceVnfPkgm
    Bases: abc.ABC

    VNF Package Management Interfaces

    Base: {apiRoot}/vnfpgm/v1

    delete_vnf_packages_vnfpkgid (vnfPkgId)
        VNF Package Management Interface - Individual VNF package
        /vnf_packages/{vnfPkgId}: DELETE - Delete an individual VNF package

    get_vnf_packages ()
        VNF Package Management Interface - VNF packages
        /vnf_packages: GET - Query VNF packages information

    get_vnf_packages_subscriptions ()
        VNF Package Management Interface - Subscriptions
        /subscriptions: GET - Query multiple subscriptions

    get_vnf_packages_vnfpkgid (vnfPkgId)
        VNF Package Management Interface - Individual VNF package
        /vnf_packages/{vnfPkgId}: GET - Read information about an individual VNF package

    get_vnf_packages_vnfpkgid_artifacts_artifactpath (vnfPkgId, artifactPath)
        VNF Package Management Interface - Individual VNF package artifact
        /vnf_packages/{vnfPkgId}/artifacts/{artifactPath}: GET - Fetch individual VNF package artifact

    get_vnf_packages_vnfpkgid_package_content (vnfPkgId)
        VNF Package Management Interface - VNF package content
        /vnf_packages/{vnfPkgId}/package_content: GET - Fetch an on-boarded VNF package

    get_vnf_packages_vnfpkgid_vnfd (vnfPkgId)
        VNF Package Management Interface - VNFD of an individual VNF package
        /vnf_packages/{vnfPkgId}/vnfd: GET - Read VNFD of an on-boarded VNF package

    patch_vnf_packages_vnfpkgid (vnfPkgId)
        VNF Package Management Interface - Individual VNF package
        /vnf_packages/{vnfPkgId}: PATCH - Update information about an individual VNF package

    post_vnf_packages ()
        VNF Package Management Interface - VNF packages
        /vnf_packages: POST - Create a new individual VNFpackage resource

    post_vnf_packages_subscriptions ()
        VNF Package Management Interface - Subscriptions
        /subscriptions: POST - Subscribe to notifications related to on-boarding and/or changes of VNF packages
```


post_vnf_packages_vnfpkgid_package_content (*vnfPkgId*)

VNF Package Management Interface - Upload VNF package from URI task

/vnf_packages/{vnfPkgId}/package_content/upload_from_uri: POST - Upload a VNF package by providing the address information of the VNF package

put_vnf_packages_vnfpkgid_package_content (*vnfPkgId*)

VNF Package Management Interface - VNF package content

/vnf_packages/{vnfPkgId}/package_content: PUT - Upload a VNF package by providing the content of the VNF package

3.1.2 SONATAClient | OSMClient

As a starting point, support for SONATA and OpenSource MANO (OSM) is implemented.

Methods which are implemented in the target MANO and the ones supported by PWM are tracked using a google sheet [here](#)

Documentation and usage examples are available [here](#).

OSMClient package

wrappers.OSMClient.admin

wrappers.OSMClient.auth

class wrappers.OSMClient.auth.**Auth** (*host, port=9999*)

Bases: *wrappers.CommonInterface.auth.CommonInterfaceAuth*

auth (*username, password, host=None, port=None*)

Authorization API

POST method which returns an authorization token to be used by other calls.

Parameters

- **username** – username for login
- **password** – password for login
- **host** – host url
- **port** – port where the MANO API can be accessed

Example:

```
osm_c = OSMClient.Auth(HOST_URL)
response = json.loads(osm_c.auth(
    username=USERNAME,
    password=PASSWORD))
```

wrappers.OSMClient.helpers

wrappers.OSMClient.nsd**class** wrappers.OSMClient.nsd.Nsd(*host, port=9999*)Bases: *wrappers.CommonInterface.nsd.CommonInterfaceNsd*

NSD Management Interfaces

delete_ns_descriptors_nsdinfoId(*token, nsdinfoId, host=None, port=None*)

NSD Management Interface - Individual NS Descriptor

/ns_descriptors/{nsdInfoId}: DELETE - Delete the content of NSD**Parameters**

- **token** – auth token retrieved by the auth call
- **nsdinfoId** – id of the individual NSD
- **host** – host url
- **port** – port where the MANO API can be accessed

delete_pnf_descriptors_pnfinfoId(*pnfinfoId*)

NSD Management interface - Individual PNF Descriptor

/pnf_descriptors/pnfinfoId: DELETE - Delete an individual PNF descriptor resource.**delete_subscriptions_subscriptionId**(*subscriptionId*)

NSD Management interface - Individual Subscription

/subscriptions/subscriptionId: DELETE - Terminate a subscription.**get_ns_descriptors**(*token, _filter=None, host=None, port=None*)

NSD Management Interface - NS Descriptors

/ns_descriptors: GET - Query information about multiple NS descriptor resources.**Parameters**

- **token** – auth token retrieved by the auth call
- **_filter** – content query filter
- **host** – host url
- **port** – port where the MANO API can be accessed

Example:

```
osm_nsd = OSMClient.Nsd(HOST_URL)
osm_auth = OSMClient.Auth(HOST_URL)

_token = json.loads(osm_auth.auth(
    username=USERNAME,
    password=PASSWORD))
_token = json.loads(_token["data"])

response = json.loads(osm_nsd.get_ns_descriptors(
    token=_token["id"]))
response = json.loads(response["data"])
```

get_ns_descriptors_nsd_content (*token, nsdinfoId, host=None, port=None*)

NSD Management Interface - NSD Content

/ns_descriptors/{nsdInfoId}/nsd_c: GET - Fetch the content of a NSD.

Parameters

- **token** – auth token retrieved by the auth call
- **nsdinfoId** – id of the individual NSD
- **host** – host url
- **port** – port where the MANO API can be accessed

get_ns_descriptors_nsdinfoId (*token, nsdinfoId, host=None, port=None*)

NSD Management Interface - Individual NS Descriptor

/ns_descriptors/{nsdInfoId}: Read information about an individual NS descriptor resource.

Parameters

- **token** – auth token retrieved by the auth call
- **nsdinfoId** – id of the individual NSD
- **host** – host url
- **port** – port where the MANO API can be accessed

Example:

```
osm_nsd = OSMClient.Nsd(HOST_URL)
osm_auth = OSMClient.Auth(HOST_URL)

_token = json.loads(osm_auth.auth(username=USERNAME, password=PASSWORD))
_token = json.loads(_token["data"])

_nsd = "24d8ea1c-3d96-47d6-8fc4-473c9a6f1ad2"

response = json.loads(osm_nsd.get_ns_descriptors_nsdinfoId(
    token=_token["id"], nsdinfoId=_nsd))
```

get_pnf_descriptors ()

NSD Management interface - PNF Descriptors

/pnf_descriptors: GET - Query information about multiple PNF descriptor resources.

get_pnf_descriptors_pnfd_content (*pnfdInfoId*)

NSD Management interface - PNFD Content

/pnf_descriptors/pnfdinfoId/pnfd_content: GET - Fetch the content of a PNFD.

get_pnf_descriptors_pnfdinfoId (*pnfdInfoId*)

NSD Management interface - Individual PNF Descriptor

/pnf_descriptors/pnfdinfoId: GET - Read an individual PNFD resource.

get_subscriptions (*subscriptionId*)

NSD Management interface - Subscriptions

/subscriptions: GET - Query multiple subscriptions.

get_subscriptions_subscriptionid (*subscriptionid*)

NSD Management interface - Individual Subscription

/subscriptions/subscriptionId: GET - Read an individual subscription resource

patch_ns_descriptors_nsinfoid ()

NSD Management Interface - Individual NS Descriptor

/ns_descriptors/{nsdInfoId}: PATCH - Modify the operational state and/or the user defined data of an individual NS descriptor resource.

patch_pnf_descriptors_pnfinfoid (*pnfdInfoId*)

NSD Management interface - Individual PNF Descriptor

/pnf_descriptors/pnfinfoid: PATCH - Modify the user defined data of an individual PNF descriptor resource.

post_ns_descriptors (*token, package_path, host=None, port=None*)

NSD Management Interface - NS Descriptors

/ns_descriptors: POST - Create a new NS descriptor resource.

Parameters

- **token** – auth token retrieved by the auth call
- **package_path** – file path of the package
- **host** – host url
- **port** – port where the MANO API can be accessed

Example:

```
osm_vnfpkgm = OSMClient.VnfPkgm(HOST_URL)
osm_nsd = OSMClient.Nsd(HOST_URL)
osm_auth = OSMClient.Auth(HOST_URL)
_token = json.loads(osm_auth.auth(username=USERNAME, password=PASSWORD))
_token = json.loads(_token["data"])

osm_vnfpkgm.post_vnf_packages(token=_token["id"],
                             package_path="tests/samples/test_osm_cirros_vnfd.tar.gz")

response = json.loads(osm_nsd.post_ns_descriptors(token=_token["id"],
                                                  package_path="tests/samples/test_osm_cirros_nsd.tar.gz",
                                                  ↪))
response = json.loads(response["data"])
```

post_pnf_descriptors ()

NSD Management interface - PNF Descriptors

/pnf_descriptors: POST - Create a new PNF descriptor resource.

post_subscriptions (*pnfdInfoId*)

NSD Management interface - Subscriptions

/subscriptions: POST - Subscribe to NSD and PNFD change notifications.

put_ns_descriptors_nsd_content (*token, data_path, nsdinfoId, host=None, port=None*)

NSD Management Interface - NSD Content

/ns_descriptors/{nsdInfoId}/nsd_c: PUT - Upload the content of NSD

Parameters

- **token** – auth token retrieved by the auth call
- **data_path** – file location of the NSD
- **nsdinfoId** – id of the individual NSD
- **host** – host url
- **port** – port where the MANO API can be accessed

put_pnf_descriptors_pnfd_content (*pnfdInfoId*)

NSD Management interface - PNFD Content

/pnf_descriptors/pnfdinfoId/pnfd_content: PUT - Upload the content of a PNFD.

wrappers.OSMClient.nsfm

class wrappers.OSMClient.nsfm.Nsfm

Bases: *wrappers.CommonInterface.nsfm.CommonInterfaceNsfm*

delete_nsfm_subscriptions_subscriptionid (*subscriptionId*)

NS Fault Management interface - Individual subscription

/nsfm_subscriptions_subscriptionid: DELETE - Terminate a subscription.

get_nsfm_alarms ()

NS Fault Management interface - Alarms

/nsfm_alarms: GET - Query alarms related to NS instances.

get_nsfm_alarms_alarmid (*alarmId*)

NS Fault Management interface - Individual alarm

/nsfm_alarms_alarmid: GET - Read individual alarm.

get_nsfm_subscriptions ()

NS Fault Management interface - Subscriptions

/nsfm_subscriptions: GET - Query multiple subscriptions.

get_nsfm_subscriptions_subscriptionid (*subscriptionId*)

NS Fault Management interface - Individual subscription

/nsfm_subscriptions_subscriptionid: GET - Read an individual subscription.

patch_nsfm_alarms_alarmid (*alarmId*)

NS Fault Management interface - Individual alarm

/nsfm_alarms_alarmid: PATCH - Acknowledge individual alarm.

post_nsfm_subscriptions ()

NS Fault Management interface - Subscriptions

/nsfm_subscriptions: POST - Subscribe to alarms related to NSs.

wrappers.OSMClient.nslcm

class wrappers.OSMClient.nslcm.Nslcm (*host, port=9999*)

Bases: *wrappers.CommonInterface.nslcm.CommonInterfaceNslcm*

Lifecycle Management interface

delete_ns_instances_nsinstanceid (*nsInstanceId*)

NS Lifecycle Management interface - Individual NS instance

/ns_instances_nsinstanceid: DELETE - Delete NS instance resource.

delete_ns_lcm_subscriptions_subscriptionid (*subscriptionId*)

NS Lifecycle Management interface - Individual subscription

/ns_lcm_subscriptions_subscriptionid: DELETE - Terminate a subscription.

get_ns_instances (*token, host=None, port=None*)

NS Lifecycle Management interface - NS instances

/ns_instances: GET - Query multiple NS instances.

Parameters

- **token** – auth token retrieved by the auth call
- **host** – host url
- **port** – port where the MANO API can be accessed

Example:

```
osm_nslcm = OSMClient.Nslcm(HOST_URL)
osm_auth = OSMClient.Auth(HOST_URL)
_token = json.loads(osm_auth.auth(username=USERNAME, password=PASSWORD))
_token = json.loads(_token["data"])

response = json.loads(osm_nslcm.get_ns_instances(token=_token["id"]))
response = json.loads(response["data"])
```

get_ns_instances_nsinstanceid (*token, nsInstanceId, host=None, port=None*)

NS Lifecycle Management interface - Individual NS instance

/ns_instances_nsinstanceid: GET - Read an individual NS instance resource.

Parameters

- **token** – auth token retrieved by the auth call
- **nsInstanceId** – id of the NS instance
- **host** – host url
- **port** – port where the MANO API can be accessed

get_ns_lcm_op_ops (*token, id, host=None, port=None*)

NS Lifecycle Management interface - NS lifecycle operation occurrences

/ns_lcm_op_ops: GET - Query multiple NS LCM operation occurrences.

get_ns_lcm_op_ops_nslcmopoccid (*token, id, host=None, port=None*)

NS Lifecycle Management interface - Individual NS lifecycle operation occurrence

/ns_lcm_op_ops_nslcmopoccid: GET - Read an individual NS LCM operation occurrence resource.

get_ns_lcm_subscriptions ()

NS Lifecycle Management interface - Subscriptions

/ns_lcm_subscriptions: GET - Query multiple subscriptions.

get_ns_lcm_subscriptions_subscriptionid (*subscriptionId*)

NS Lifecycle Management interface - Individual subscription

/ns_lcm_subscriptions_subscriptionid: GET - Read an individual subscription resource.

post_ns_instances (*nsInstanceId*)

NS Lifecycle Management interface - NS instances

/ns_instances: POST - Create a NS instance resource.

post_ns_instances_nsinstanceid_heal (*nsInstanceId*)

NS Lifecycle Management interface - Heal NS task

/ns_instances_nsinstanceid_heal: POST - Heal a NS instance.

post_ns_instances_nsinstanceid_instantiate (*token, nsDescription, nsName, nsdId, vimAccountId, host=None, port=None*)

NS Lifecycle Management interface - Instantiate NS task

/ns_instances_nsinstanceid_instantiate: POST - Instantiate a NS instance.

Parameters

- **token** – auth token retrieved by the auth call
- **nsDescription** – NS instantiation description
- **nsName** – Name for instantiation reference
- **nsdId** – id of the NSD to be instantiate
- **vimAccountId** – ID of the VIM to be instantiated on
- **host** – host url
- **port** – port where the MANO API can be accessed

Example:

```
osm_nsd = OSMClient.Nsd(HOST_URL)
osm_nslcm = OSMClient.Nslcm(HOST_URL)
osm_auth = OSMClient.Auth(HOST_URL)

_token = json.loads(osm_auth.auth(username=USERNAME, password=PASSWORD))
_token = json.loads(_token["data"])

_nsd_list = json.loads(osm_nsd.get_ns_descriptors(token=_token["id"]))
_nsd_list = json.loads(_nsd_list["data"])
_nsd = None

for _n in _nsd_list:
    if "test_osm_cirros_2vnf_nsd" == _n['_id']:
        _nsd = _n['_id']

response = json.loads(osm_nslcm.post_ns_instances_nsinstanceid_
↳instantiate(token=_token["id"],
              nsDescription=NSDESCRIPTION,
              nsName=NSNAME,
              nsdId=_nsd,
              vimAccountId=VIMACCOUNTID))

response = json.loads(response["data"])
```

post_ns_instances_nsinstanceid_scale (*nsInstanceId*)

NS Lifecycle Management interface - Scale NS task

/ns_instances_nsinstanceid_scale: POST - Scale a NS instance.

post_ns_instances_nsinstanceid_terminate (*token*, *nsInstanceId*, *host=None*,
port=None, *force=None*)

NS Lifecycle Management interface - Terminate NS task

/ns_instances_nsinstanceid_terminate: POST - Terminate a NS instance.

Parameters

- **token** – auth token retrieved by the auth call
- **nsInstanceId** – id of the NS instance
- **host** – host url
- **port** – port where the MANO API can be accessed
- **force** – true/false whether to force terminate

post_ns_instances_nsinstanceid_update (*nsInstanceId*)

NS Lifecycle Management interface - Update NS task

/ns_instances_nsinstanceid_update: POST - Updates a NS instance.

post_ns_lcm_op_occs_nslcmopoccid_cancel (*nsLcmOpOccId*)

NS Lifecycle Management interface - Cancel operation task

/ns_lcm_op_occs_nslcmopoccid_cancel: POST - Cancel a NS lifecycle management operation occurrence.

post_ns_lcm_op_occs_nslcmopoccid_continue (*nsLcmOpOccId*)

NS Lifecycle Management interface - Continue operation task

/ns_lcm_op_occs_nslcmopoccid_continue: POST - Continue a NS lifecycle management operation occurrence.

post_ns_lcm_op_occs_nslcmopoccid_fail (*nsLcmOpOccId*)

NS Lifecycle Management interface - Fail operation task

/ns_lcm_op_occs_nslcmopoccid_fail: POST - Mark a NS lifecycle management operation occurrence as failed.

post_ns_lcm_op_occs_nslcmopoccid_retry (*nsLcmOpOccId*)

NS Lifecycle Management interface - Retry operation task

/ns_lcm_op_occs_nslcmopoccid_retry: POST - Retry a NS lifecycle management operation occurrence.

post_ns_lcm_op_occs_nslcmopoccid_rollback (*nsLcmOpOccId*)

NS Lifecycle Management interface - Rollback operation task

/ns_lcm_op_occs_nslcmopoccid_rollback: POST - Rollback a NS lifecycle management operation occurrence.

post_ns_lcm_subscriptions ()

NS Lifecycle Management interface - Subscriptions

/ns_lcm_subscriptions: POST - Subscribe to NS lifecycle change notifications.

wrappers.OSMClient.nspm**wrappers.OSMClient.vnfpkgm**

class wrappers.OSMClient.vnfpkgm.VnfPkgm(*host, port=9999*)

Bases: *wrappers.CommonInterface.vnfpkgm.CommonInterfaceVnfPkgm*

VNF Package Management Interfaces

delete_vnf_packages_vnfpkgid(*token, vnfPkgId, host=None, port=None*)

VNF Package Management Interface - Individual VNF package

/vnf_packages/{vnfPkgId}: DELETE - Delete an individual VNF package

Parameters

- **token** – auth token retrieved by the auth call
- **vnfPkgId** – id of the vnf package to fetch
- **host** – host url
- **port** – port where the MANO API can be accessed

get_vnf_packages(*token, _filter=None, host=None, port=None*)

VNF Package Management Interface - VNF packages

/vnf_packages: GET - Query VNF packages information

Parameters

- **token** – auth token retrieved by the auth call
- **_filter** – content query filter
- **host** – host url
- **port** – port where the MANO API can be accessed

get_vnf_packages_subscriptions()

VNF Package Management Interface - Subscriptions

/subscriptions: GET - Query multiple subscriptions

get_vnf_packages_vnfpkgid(*token, vnfPkgId, host=None, port=None*)

VNF Package Management Interface - Individual VNF package

/vnf_packages/{vnfPkgId}: GET - Read information about an individual VNF package

Parameters

- **token** – auth token retrieved by the auth call
- **vnfPkgId** – id of the vnf package to fetch
- **host** – host url
- **port** – port where the MANO API can be accessed

get_vnf_packages_vnfpkgid_artifacts_artifactpath(*token, vnfPkgId, host=None, port=None*)

VNF Package Management Interface - Individual VNF package artifact

/vnf_packages/{vnfPkgId}/artifacts/{artifactPath}: GET - Fetch individual VNF package artifact

Parameters

- **token** – auth token retrieved by the auth call
- **vnfPkgId** – id of the vnf package to delete
- **host** – host url
- **port** – port where the MANO API can be accessed

get_vnf_packages_vnfpkgid_package_content (*token, vnfPkgId, host=None, port=None*)
VNF Package Management Interface - VNF package content

/vnf_packages/{vnfPkgId}/package_content: GET - Fetch an on-boarded VNF package

Parameters

- **token** – auth token retrieved by the auth call
- **vnfPkgId** – id of the vnf package to delete
- **host** – host url
- **port** – port where the MANO API can be accessed

get_vnf_packages_vnfpkgid_vnfd (*token, vnfPkgId, host=None, port=None*)
VNF Package Management Interface - VNFD of an individual VNF package

/vnf_packages/{vnfPkgId}/vnfd: GET - Read VNFD of an on-boarded VNF package

Parameters

- **token** – auth token retrieved by the auth call
- **vnfPkgId** – id of the vnf package to delete
- **host** – host url
- **port** – port where the MANO API can be accessed

patch_vnf_packages_vnfpkgid (*vnfPkgId*)
VNF Package Management Interface - Individual VNF package

/vnf_packages/{vnfPkgId}: PATCH - Update information about an individual VNF package

post_vnf_packages (*token, package_path, host=None, port=None*)
VNF Package Management Interface - VNF packages

/vnf_packages: POST - Create a new individual VNFpackage resource

Parameters

- **token** – auth token retrieved by the auth call
- **package_path** – file path of the package
- **host** – host url
- **port** – port where the MANO API can be accessed

Example:

```

osm_vnfpkgm = OSMClient.VnfPkgm(HOST_URL)
osm_auth = OSMClient.Auth(HOST_URL)

_token = json.loads(osm_auth.auth(username=USERNAME, password=PASSWORD))
_token = json.loads(_token["data"])

response = json.loads(osm_vnfpkgm.post_vnf_packages(token=_token["id"],
                                                    package_path="tests/samples/test_osm_cirros_vnfd.tar.
↪gz"))
response = json.loads(response["data"])

```

post_vnf_packages_subscriptions()

VNF Package Management Interface - Subscriptions

/subscriptions: POST - Subscribe to notifications related to on-boarding and/or changes of VNF packages

post_vnf_packages_vnfpkgid_package_content(vnfPkgId)

VNF Package Management Interface - Upload VNF package from URI task

/vnf_packages/{vnfPkgId}/package_content/upload_from_uri: POST - Upload a VNF package by providing the address information of the VNF package

put_vnf_packages_vnfpkgid_package_content(token, data_path, vnfPkgId, host=None, port=None)

VNF Package Management Interface - VNF package content

/vnf_packages/{vnfPkgId}/package_content: PUT - Upload a VNF package by providing the content of the VNF package

Parameters

- **token** – auth token retrieved by the auth call
- **data_path** – file location of the VNFD
- **vnfPkgId** – id of the vnf package to delete
- **host** – host url
- **port** – port where the MANO API can be accessed

SONATAClient package

wrappers.SONATAClient.admin

wrappers.SONATAClient.auth

class wrappers.SONATAClient.auth.**Auth**(host, port=80)

Bases: *wrappers.CommonInterface.auth.CommonInterfaceAuth*

auth(username, password, host=None, port=None)

Authorization API

POST method which returns an authorization token to be used by other calls.

Parameters

- **username** – username for login
- **password** – password for login

- **host** – host url
- **port** – port where the MANO API can be accessed

Example:

```
sonata_auth = SONATAClient.Auth(HOST_URL)
response = json.loads(sonata_auth.auth(
    username=USERNAME,
    password=PASSWORD))
```

wrappers.SONATAClient.nsd

class wrappers.SONATAClient.nsd.**Nsd**(*host, port=4002*)

Bases: *wrappers.CommonInterface.nsd.CommonInterfaceNsd*

NSD Management Interfaces

delete_ns_descriptors_nsdinfoId(*token, nsdinfoId, host=None, port=None*)

NSD Management Interface - Individual NS Descriptor

/ns_descriptors/{nsdInfoId}: DELETE - Delete the content of NSD

Parameters

- **token** – auth token retrieved by the auth call
- **nsdinfoId** – id of the individual NSD
- **host** – host url
- **port** – port where the MANO API can be accessed

delete_pnf_descriptors_pnfinfoId(*pnfinfoId*)

NSD Management interface - Individual PNF Descriptor

/pnf_descriptors/pnfinfoId: DELETE - Delete an individual PNF descriptor resource.

delete_subscriptions_subscriptionId(*subscriptionId*)

NSD Management interface - Individual Subscription

/subscriptions/subscriptionId: DELETE - Terminate a subscription.

get_ns_descriptors(*token, _filter=None, host=None, port=None*)

NSD Management Interface - NS Descriptors

/ns_descriptors: GET - Query information about multiple NS descriptor resources.

Parameters

- **token** – auth token retrieved by the auth call
- **_filter** – content query filter
- **host** – host url
- **port** – port where the MANO API can be accessed

Example:

```

sonata_nsd = SONATAClient.Nsd(HOST_URL)
sonata_auth = SONATAClient.Auth(HOST_URL)
_token = json.loads(sonata_auth.auth(
    username=USERNAME,
    password=PASSWORD))
_token = json.loads(_token["data"])

response = json.loads(sonata_nsd.get_ns_descriptors(
    token=_token["token"]["access_token"]))
response = json.loads(response["data"])

```

get_ns_descriptors_nsd_content()

NSD Management Interface - NSD Content

/ns_descriptors/{nsdInfoId}/nsd_c: GET - Fetch the content of a NSD.

get_ns_descriptors_nsdinfoId(token, nsdinfoId, host=None, port=None)

NSD Management Interface - Individual NS Descriptor

/ns_descriptors/{nsdInfoId}: Read information about an individual NS descriptor resource.

Parameters

- **token** – auth token retrieved by the auth call
- **nsdinfoId** – id of the individual NSD
- **host** – host url
- **port** – port where the MANO API can be accessed

get_pnf_descriptors()

NSD Management interface - PNF Descriptors

/pnf_descriptors: GET - Query information about multiple PNF descriptor resources.

get_pnf_descriptors_pnfd_content(pnfdInfoId)

NSD Management interface - PNFD Content

/pnf_descriptors/pnfdinfoId/pnfd_content: GET - Fetch the content of a PNFD.

get_pnf_descriptors_pnfdinfoId(pnfdInfoId)

NSD Management interface - Individual PNF Descriptor

/pnf_descriptors/pnfdinfoId: GET - Read an individual PNFD resource.

get_subscriptions(subscriptionId)

NSD Management interface - Subscriptions

/subscriptions: GET - Query multiple subscriptions.

get_subscriptions_subscriptionId(subscriptionId)

NSD Management interface - Individual Subscription

/subscriptions/subscriptionId: GET - Read an individual subscription resource

patch_ns_descriptors_nsdinfoId()

NSD Management Interface - Individual NS Descriptor

/ns_descriptors/{nsdInfoId}: PATCH - Modify the operational state and/or the user defined data of an individual NS descriptor resource.

patch_pnf_descriptors_pnfdinfoId (*pnfdInfoId*)
NSD Management interface - Individual PNF Descriptor

/pnf_descriptors/pnfdinfoId: PATCH - Modify the user defined data of an individual PNF descriptor resource.

post_ns_descriptors (*token, package_path, host=None, port=None*)
NSD Management Interface - NS Descriptors

/ns_descriptors: POST - Create a new NS descriptor resource.

Parameters

- **token** – auth token retrieved by the auth call
- **package_path** – file path of the package
- **host** – host url
- **port** – port where the MANO API can be accessed

Example:

```
sonata_vnfpkgm = SONATAClient.VnfPkgm(HOST_URL)
sonata_nsd = SONATAClient.Nsd(HOST_URL)
sonata_auth = SONATAClient.Auth(HOST_URL)

_token = json.loads(sonata_auth.auth(username=USERNAME,
    ↪password=PASSWORD))
_token = json.loads(_token["data"])

sonata_vnfpkgm.post_vnf_packages(token=_token["token"]["access_token"],
    package_path="tests/samples/vnfd_example.yml")

response = json.loads(sonata_nsd.post_ns_descriptors(
    token=_token["token"]["access_token"],
    package_path="tests/samples/nsd_example.yml"))
```

post_pnf_descriptors ()
NSD Management interface - PNF Descriptors

/pnf_descriptors: POST - Create a new PNF descriptor resource.

post_subscriptions (*pnfdInfoId*)
NSD Management interface - Subscriptions

/subscriptions: POST - Subscribe to NSD and PNFD change notifications.

put_ns_descriptors_nsd_content ()
NSD Management Interface - NSD Content

/ns_descriptors/{nsdInfoId}/nsd_c: PUT - Upload the content of NSD

put_pnf_descriptors_pnfd_content (*pnfdInfoId*)
NSD Management interface - PNFD Content

/pnf_descriptors/pnfdinfoId/pnfd_content: PUT - Upload the content of a PNFD.

wrappers.SONATAClient.nslcm

class wrappers.SONATAClient.nslcm.Nslcm(*host, port=32001*)
 Bases: *wrappers.CommonInterface.nslcm.CommonInterfaceNslcm*

Lifecycle Management interface

delete_ns_instances_nsinstanceid()

NS Lifecycle Management interface - Individual NS instance

/ns_instances_nsinstanceid: DELETE - Delete NS instance resource.

delete_ns_lcm_subscriptions_subscriptionid(*subscriptionId*)

NS Lifecycle Management interface - Individual subscription

/ns_lcm_subscriptions_subscriptionid: DELETE - Terminate a subscription.

get_ns_instances(*token, host=None, port=None*)

NS Lifecycle Management interface - NS instances

/ns_instances: GET - Query multiple NS instances.

Parameters

- **token** – auth token retrieved by the auth call
- **host** – host url
- **port** – port where the MANO API can be accessed

Example:

```
sonata_nslcm = SONATAClient.Nslcm(HOST_URL)
sonata_auth = SONATAClient.Auth(HOST_URL)
_token = json.loads(sonata_auth.auth(username=USERNAME,
↳password=PASSWORD))
_token = json.loads(_token["data"])
response = json.loads(sonata_nslcm.get_ns_instances(
    token=_token["token"]["access_token"]))
response = json.loads(response["data"])
```

get_ns_instances_nsinstanceid(*token, nsInstanceId, host=None, port=None*)

NS Lifecycle Management interface - Individual NS instance

/ns_instances_nsinstanceid: GET - Read an individual NS instance resource.

Parameters

- **token** – auth token retrieved by the auth call
- **nsInstanceId** – id of the NS instance
- **host** – host url
- **port** – port where the MANO API can be accessed

get_ns_lcm_op_ops()

NS Lifecycle Management interface - NS lifecycle operation occurrences

/ns_lcm_op_ops: GET - Query multiple NS LCM operation occurrences.

get_ns_lcm_op_ops_nslcmopoccid()
 NS Lifecycle Management interface - Individual NS lifecycle operation occurrence
/ns_lcm_op_ops_nslcmopoccid: GET - Read an individual NS LCM operation occurrence resource.

get_ns_lcm_subscriptions()
 NS Lifecycle Management interface - Subscriptions
/ns_lcm_subscriptions: GET - Query multiple subscriptions.

get_ns_lcm_subscriptions_subscriptionid(subscriptionId)
 NS Lifecycle Management interface - Individual subscription
/ns_lcm_subscriptions_subscriptionid: GET - Read an individual subscription resource.

post_ns_instances()
 NS Lifecycle Management interface - NS instances
/ns_instances: POST - Create a NS instance resource.

post_ns_instances_nsinstanceid_heal(nsInstanceId)
 NS Lifecycle Management interface - Heal NS task
/ns_instances_nsinstanceid_heal: POST - Heal a NS instance.

post_ns_instances_nsinstanceid_instantiate(token, nsInstanceId, egresses=[], ingresses=[], host=None, port=None)
 NS Lifecycle Management interface - Instantiate NS task
/ns_instances_nsinstanceid_instantiate: POST - Instantiate a NS instance.

Parameters

- **token** – auth token retrieved by the auth call
- **nsInstanceId** – NS instantiation description
- **ingresses** – ingresses list
- **egresses** – egresses list
- **host** – host url
- **port** – port where the MANO API can be accessed

Example:

```
sonata_nslcm = SONATAClient.Nslcm(HOST_URL)
sonata_auth = SONATAClient.Auth(HOST_URL)
sonata_nsd = SONATAClient.Nsd(HOST_URL)

_token = json.loads(sonata_auth.auth(username=USERNAME,
    ↪password=PASSWORD))
_token = json.loads(_token["data"])

_nsd_list = json.loads(sonata_nsd.get_ns_descriptors(token=_token["token"]
    ↪)["access_token"])
_nsd_list = json.loads(_nsd_list["data"])

_ns = None
for _n in _nsd_list:
    if "A dummy Example." == _n['nsd']['description']:
        _ns = _n['uuid']
```

(continues on next page)

(continued from previous page)

```

if _ns:
    response = json.loads(
        sonata_nslcm.post_ns_instances_nsinstanceid_instantiate(
            token=_token["token"]["access_token"], nsInstanceId=_
↪ns) )

```

post_ns_instances_nsinstanceid_scale (*nsInstanceId*)

NS Lifecycle Management interface - Scale NS task

/ns_instances_nsinstanceid_scale: POST - Scale a NS instance.

post_ns_instances_nsinstanceid_terminate (*token*, *nsInstanceId*, *host=None*,
port=None)

NS Lifecycle Management interface - Terminate NS task

/ns_instances_nsinstanceid_terminate: POST - Terminate a NS instance.

Parameters

- **token** – auth token retrieved by the auth call
- **nsInstanceId** – id of the NS instance
- **host** – host url
- **port** – port where the MANO API can be accessed
- **force** – true/false whether to force terminate

post_ns_instances_nsinstanceid_update (*nsInstanceId*)

NS Lifecycle Management interface - Update NS task

/ns_instances_nsinstanceid_update: POST - Updates a NS instance.

post_ns_lcm_op_occs_nslcmopoccid_cancel (*nsLcmOpOccId*)

NS Lifecycle Management interface - Cancel operation task

/ns_lcm_op_occs_nslcmopoccid_cancel: POST - Cancel a NS lifecycle management operation occurrence.

post_ns_lcm_op_occs_nslcmopoccid_continue (*nsLcmOpOccId*)

NS Lifecycle Management interface - Continue operation task

/ns_lcm_op_occs_nslcmopoccid_continue: POST - Continue a NS lifecycle management operation occurrence.

post_ns_lcm_op_occs_nslcmopoccid_fail (*nsLcmOpOccId*)

NS Lifecycle Management interface - Fail operation task

/ns_lcm_op_occs_nslcmopoccid_fail: POST - Mark a NS lifecycle management operation occurrence as failed.

post_ns_lcm_op_occs_nslcmopoccid_retry (*nsLcmOpOccId*)

NS Lifecycle Management interface - Retry operation task

/ns_lcm_op_occs_nslcmopoccid_retry: POST - Retry a NS lifecycle management operation occurrence.

post_ns_lcm_op_occs_nslcmopoccid_rollback (*nsLcmOpOccId*)

NS Lifecycle Management interface - Rollback operation task

/ns_lcm_op_occs_nslcmopoccid_rollback: POST - Rollback a NS lifecycle management operation occurrence.

post_ns_lcm_subscriptions ()
NS Lifecycle Management interface - Subscriptions

/ns_lcm_subscriptions: POST - Subscribe to NS lifecycle change notifications.

wrappers.SONATAClient.sonpackage

```
class wrappers.SONATAClient.sonpackage.Package (host, port=4002)
    Bases: wrappers.CommonInterface.sonpackage.CommonInterfaceSonPackage

    delete_son_packages_PackageId (token, id, host=None, port=None)
        Delete a particular sonata package

    get_son_packages (token, _filter=None, host=None, port=None)
        Receive the list of stored packages

    get_son_packages_PackageId (token, id, host=None, port=None)
        Receive a particular stored packages

    post_son_packages (token, package_path, host=None, port=None)
        Send a particular package
```

wrappers.SONATAClient.vnfpkgm

```
class wrappers.SONATAClient.vnfpkgm.VnfPkgm (host, port=4002)
    Bases: wrappers.CommonInterface.vnfpkgm.CommonInterfaceVnfPkgm

    VNF Package Management Interfaces

    delete_vnf_packages_vnfpkgid (token, vnfPkgId, host=None, port=None)
        VNF Package Management Interface - Individual VNF package

    /vnf_packages/{vnfPkgId}: DELETE - Delete an individual VNF package
```

Parameters

- **token** – auth token retrieved by the auth call
- **vnfPkgId** – id of the vnf package to fetch
- **host** – host url
- **port** – port where the MANO API can be accessed

```
get_vnf_packages (token, _filter=None, host=None, port=None)
    VNF Package Management Interface - VNF packages

/vnf_packages: GET - Query VNF packages information
```

Parameters

- **token** – auth token retrieved by the auth call
- **_filter** – content query filter
- **host** – host url
- **port** – port where the MANO API can be accessed

get_vnf_packages_subscriptions()

VNF Package Management Interface - Subscriptions

/subscriptions: GET - Query multiple subscriptions

get_vnf_packages_vnfpkgid(token, vnfPkgId, host=None, port=None)

VNF Package Management Interface - Individual VNF package

/vnf_packages/{vnfPkgId}: GET - Read information about an individual VNF package

Parameters

- **token** – auth token retrieved by the auth call
- **vnfPkgId** – id of the vnf package to fetch
- **host** – host url
- **port** – port where the MANO API can be accessed

get_vnf_packages_vnfpkgid_artifacts_artifactpath(vnfPkgId, artifactPath)

VNF Package Management Interface - Individual VNF package artifact

/vnf_packages/{vnfPkgId}/artifacts/{artifactPath}: GET - Fetch individual VNF package artifact

get_vnf_packages_vnfpkgid_package_content(vnfPkgId)

VNF Package Management Interface - VNF package content

/vnf_packages/{vnfPkgId}/package_content: GET - Fetch an on-boarded VNF package

get_vnf_packages_vnfpkgid_vnfd(vnfPkgId)

VNF Package Management Interface - VNFD of an individual VNF package

/vnf_packages/{vnfPkgId}/vnfd: GET - Read VNFD of an on-boarded VNF package

patch_vnf_packages_vnfpkgid(vnfPkgId)

VNF Package Management Interface - Individual VNF package

/vnf_packages/{vnfPkgId}: PATCH - Update information about an individual VNF package

post_vnf_packages(token, package_path, host=None, port=None)

VNF Package Management Interface - VNF packages

/vnf_packages: POST - Create a new individual VNFpackage resource

Parameters

- **token** – auth token retrieved by the auth call
- **package_path** – file path of the package
- **host** – host url
- **port** – port where the MANO API can be accessed

Example:

```
sonata_vnfpgm = SONATAClient.VnfPkgm(HOST_URL)
sonata_auth = SONATAClient.Auth(HOST_URL)

_token = json.loads(sonata_auth.auth(username=USERNAME,
↳password=PASSWORD))
_token = json.loads(_token["data"])
```

(continues on next page)

(continued from previous page)

```
response = json.loads(sonata_vnfpkgm.post_vnf_packages(  
    token=_token["token"] ["access_token"],  
    package_path="tests/samples/vnfd_example.yml"))
```

post_vnf_packages_subscriptions()

VNF Package Management Interface - Subscriptions

/subscriptions: POST - Subscribe to notifications related to on-boarding and/or changes of VNF packages

post_vnf_packages_vnfpkgid_package_content(vnfPkgId)

VNF Package Management Interface - Upload VNF package from URI task

/vnf_packages/{vnfPkgId}/package_content/upload_from_uri: POST - Upload a VNF package by providing the address information of the VNF package

put_vnf_packages_vnfpkgid_package_content(vnfPkgId)

VNF Package Management Interface - VNF package content

/vnf_packages/{vnfPkgId}/package_content: PUT - Upload a VNF package by providing the content of the VNF package

REFERENCES

- ETSI GS NFV-SOL 005 - https://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/005/02.04.01_60/gs_NFV-SOL005v020401p.pdf
- OSM
 - OSM NBI Project - https://osm.etsi.org/gitweb/?p=osm/NBI.git;a=tree;f=osm_nbi;hb=HEAD
 - Old OSM document - <https://osm.etsi.org/wikipub/images/2/24/Osm-r1-so-rest-api-guide.pdf>
 - OSM architecture - https://osm.etsi.org/wikipub/index.php/Developer_HowTo_for_RO_Module#Directory_Organization
- Sonata
 - Sonata Gatekeeper - <https://github.com/sonata-nfv/son-gkeeper/tree/master/son-gtkapi>
 - Sonata Routes - <https://github.com/sonata-nfv/son-gkeeper/tree/master/son-gtkapi/routes>
 - Sonata Usage doc - <https://github.com/sonata-nfv/son-gkeeper/wiki/Usage-documentation>

TODO

- Include more MANOs (OpenBaton, 5GTango..)
- Support more versions

PYTHON MODULE INDEX

W

- `wrappers.CommonInterface.auth`, 5
- `wrappers.CommonInterface.nsd`, 6
- `wrappers.CommonInterface.nsfm`, 7
- `wrappers.CommonInterface.nslcm`, 8
- `wrappers.CommonInterface.nspm`, 10
- `wrappers.CommonInterface.sonpackage`, 11
- `wrappers.CommonInterface.vnfpkgm`, 12
- `wrappers.OSMClient.admin`, 13
- `wrappers.OSMClient.auth`, 13
- `wrappers.OSMClient.helpers`, 13
- `wrappers.OSMClient.nsd`, 14
- `wrappers.OSMClient.nsfm`, 17
- `wrappers.OSMClient.nslcm`, 17
- `wrappers.OSMClient.nspm`, 21
- `wrappers.OSMClient.vnfpkgm`, 21
- `wrappers.SONATAClient.admin`, 23
- `wrappers.SONATAClient.auth`, 23
- `wrappers.SONATAClient.nsd`, 24
- `wrappers.SONATAClient.nslcm`, 27
- `wrappers.SONATAClient.sonpackage`, 30
- `wrappers.SONATAClient.vnfpkgm`, 30