Sushy Documentation

Release 5.2.1.dev3

OpenStack Foundation

Oct 22, 2024

CONTENTS

CHAPTER

ONE

OVERVIEW

Sushy is a Python library to communicate with Redfish based systems.

The goal of the library is to be extremely simple, small, have as few dependencies as possible and be very conservative when dealing with BMCs by issuing just enough requests to it (BMCs are very flaky).

Therefore, the scope of the library has been limited to what is supported by the OpenStack Ironic project. As the project grows and more features from Redfish are needed we can expand Sushy to fulfill those requirements.

- Free software: Apache license
- **Includes Redfish registry files licensed under**
  Creative Commons Attribution 4.0 License: https://creativecommons.org/licenses/by/4.0/
- Documentation: https://docs.openstack.org/sushy/latest/
- Usage: https://docs.openstack.org/sushy/latest/reference/usage.html
- Source: https://opendev.org/openstack/sushy
- Bugs: https://bugs.launchpad.net/sushy

CHAPTER

TWO

FEATURES

- Abstraction around the SystemCollection and System resources (Basic server identification and asset information)
- RAID in Redfish based Systems
- Redfish Ethernet Interface
- System mappings
- System processor
- Storage management
- Systems power management (Both soft and hard; Including NMI injection)
- Changing systems boot device, frequency (Once or permanently) and mode (UEFI or BIOS)
- Chassis management
- OEM extension
- Virtual media management
- Session Management

CHAPTER

THREE

DOCUMENTATION

3.1 Installing Sushy

At the command line:

```
$ pip install sushy
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv sushy
$ pip install sushy
```

3.2 Contributing to Sushy

3.2.1 How to contribute

If you would like to contribute to the development of OpenStack, you must follow the steps in this page:

> http://docs.openstack.org/infra/manual/developers.html

If you already have a good understanding of how the system works and your OpenStack accounts are set up, you can skip to the development workflow section of this documentation to learn how changes to OpenStack should be submitted for review via the Gerrit tool:

> http://docs.openstack.org/infra/manual/developers.html#development-workflow

Pull requests submitted through GitHub will be ignored.

Bugs should be filed in StoryBoard, not GitHub:

> https://storyboard.openstack.org/#!/project/960

3.2.2 Running a Redfish emulator

Testing and/or developing Sushy without owning a real baremetal machine that supports the Redfish protocol is possible by running an emulator, the sushy-tools project ships with two emulators that can be used for this purpose. To install it run:

```
sudo pip install --user sushy-tools
```

Note

Installing the dependencies requires libvirt development files. For example, run the following command to install them on Fedora:

```
sudo dnf install -y libvirt-devel
```

Static emulator

After installing sushy-tools you will have a new CLI tool named `sushy-static`. This tool creates a HTTP server to serve any of the Redfish mockups. The files are static so operations like changing the boot device or the power state **will not** have any effect. But that should be enough for enabling people to test parts of the library.

To use `sushy-static` we need the Redfish mockup files that can be downloaded from https://www.dmtf.org/standards/redfish, for example:

```
wget https://www.dmtf.org/sites/default/files/standards/documents/DSP2043_1.0.0.zip
```

After the download, extract the files somewhere in the file-system:

```
unzip DSP2043_1.0.0.zip -d <output-path>
```

Now run `sushy-static` pointing to those files. For example to serve the `DSP2043-server` mockup files, run:

```
sushy-static --mockup-files <output-path>/DSP2043-server
```

Libvirt emulator

The second emulator shipped by sushy-tools is the CLI tool named `sushy-emulator`. This tool starts a ReST API that users can use to interact with virtual machines using the Redfish protocol. So operations such as changing the boot device or the power state will actually affect the virtual machines. This allows users to test the library in a more dynamic way. To run it do

```
sushy-emulator

# Or, running with custom parameters
sushy-emulator --port 8000 --libvirt-uri "qemu:///system"
```

That's it, now you can test Sushy against the `http://localhost:8000` endpoint.

Enabling SSL

Both mockup servers supports SSL if you want Sushy with it. To set it up, first you need to generate key and certificate files with OpenSSL use following command:

```
openssl req -x509 -newkey rsa:2048 -keyout key.pem -out cert.pem -days 365
```

Start the mockup server passing the `--ssl-certificate` and `--ssl-key` parameters to it, for example:

```
sushy-emulator --ssl-key key.pem --ssl-certificate cert.pem
```

Now to connect with SSL to the server use the `verify` parameter pointing to the certificate file when instantiating Sushy, for example:

```
import sushy

# Note the HTTP"S"
s = sushy.Sushy('https://localhost:8000', verify='cert.pem', username='foo', password='bar')
```

3.3 Sushy Library Reference

3.3.1 Usage

Using Sushy

To use sushy in a project:

Specifying an authentication type

There are three authentication objects. By default we use SessionOrBasicAuth.

Authentication Modes:

- auth.SessionOrBasicAuth: Use session based authentication. If we are unable to create a session we will fallback to basic authentication.
- auth.BasicAuth: Use basic authentication only.
- auth.SessionAuth: Use session based authentication only.

```
import logging

import sushy
from sushy import auth

# Enable logging at DEBUG level
LOG = logging.getLogger('sushy')
LOG.setLevel(logging.DEBUG)
LOG.addHandler(logging.StreamHandler())
```

(continues on next page)

(continued from previous page)

```
basic_auth = auth.BasicAuth(username='foo', password='bar')
session_auth = auth.SessionAuth(username='foo', password='bar')
session_or_basic_auth = auth.SessionOrBasicAuth(username='foo',
                                                password='bar')

s = sushy.Sushy('http://localhost:8000/redfish/v1',
                auth=basic_auth)

s = sushy.Sushy('http://localhost:8000/redfish/v1',
                auth=session_auth)

s = sushy.Sushy('http://localhost:8000/redfish/v1',
                auth=session_or_basic_auth)

# It is important to note that you can
# call sushy without supplying an
# authentication object. In that case we
# will use the SessionOrBasicAuth authentication
# object in an attempt to connect to all different
# types of redfish servers.
s = sushy.Sushy('http://localhost:8000/redfish/v1',
                username='foo',
                password='bar')
```

Creating and using a sushy system object

```
import logging

import sushy

# Enable logging at DEBUG level
LOG = logging.getLogger('sushy')
LOG.setLevel(logging.DEBUG)
LOG.addHandler(logging.StreamHandler())

s = sushy.Sushy('http://localhost:8000/redfish/v1',
                username='foo', password='bar')

# Get the Redfish version
print(s.redfish_version)

# Instantiate a system object
sys_inst = s.get_system('/redfish/v1/Systems/437XR1138R2')


# Using system collections


# Instantiate a SystemCollection object
sys_col = s.get_system_collection()

# Print the ID of the systems available in the collection
print(sys_col.members_identities)
```

(continues on next page)

(continued from previous page)

```
# Get a list of systems objects available in the collection
sys_col_insts = sys_col.get_members()

# Instantiate a system object, same as getting it directly
# from the s.get_system()
sys_inst = sys_col.get_member(sys_col.members_identities[0])

# Refresh the system collection object
#
# See below for more options on how to refresh resources.
sys_col.refresh()


# Using system actions


# Power the system ON
sys_inst.reset_system(sushy.ResetType.ON)

# Get a list of allowed reset values
print(sys_inst.get_allowed_reset_system_values())

# Refresh the system object (with all its sub-resources)
sys_inst.refresh()

# Alternatively, you can only refresh the resource if it is stale by␣
↪passing
# force=False:
sys_inst.refresh(force=False)

# A resource can be marked stale by calling invalidate. Note that its
# subresources won't be marked as stale, and thus they won't be refreshed␣
↪by
# a call to refresh(force=False)
sys_inst.invalidate()

# Get the current power state
print(sys_inst.power_state)

# Set the next boot device to boot once from PXE in UEFI mode
sys_inst.set_system_boot_source(sushy.BootSource.PXE,
                                enabled=sushy.BootSourceOverrideEnabled.
↪ONCE,
                                mode=sushy.BootSourceOverrideMode.UEFI)

# Get the current boot source information
print(sys_inst.boot)

# Get a list of allowed boot source target values
print(sys_inst.get_allowed_system_boot_source_values())

# Get the memory summary
print(sys_inst.memory_summary)

# Get the processor summary
print(sys_inst.processors.summary)
```

Creating and using a sushy manager object

```
import logging

import sushy

# Enable logging at DEBUG level
LOG = logging.getLogger('sushy')
LOG.setLevel(logging.DEBUG)
LOG.addHandler(logging.StreamHandler())

s = sushy.Sushy('http://localhost:8000/redfish/v1',
                username='foo', password='bar')

# Instantiate a manager object
mgr_inst = s.get_manager('BMC')

# Get the manager name & description
print(mgr_inst.name)
print(mgr_inst.description)


# Using manager collections


# Instantiate a ManagerCollection object
mgr_col = s.get_manager_collection()

# Print the ID of the managers available in the collection
print(mgr_col.members_identities)

# Get a list of manager objects available in the collection
mgr_insts = mgr_col.get_members()

# Instantiate a manager object, same as getting it directly
# from the s.get_manager()
mgr_inst = mgr_col.get_member(mgr_col.members_identities[0])

# Refresh the manager collection object
mgr_col.invalidate()
mgr_col.refresh()


# Using manager actions


# Get supported graphical console types
print(mgr_inst.get_supported_graphical_console_types())

# Get supported serial console types
print(mgr_inst.get_supported_serial_console_types())

# Get supported command shell types
print(mgr_inst.get_supported_command_shell_types())

# Get a list of allowed manager reset values
```

(continues on next page)

(continued from previous page)

```
print(mgr_inst.get_allowed_reset_manager_values())

# Reset the manager
mgr_inst.reset_manager(sushy.ResetType.FORCE_RESTART)

# Refresh the manager object (with all its sub-resources)
mgr_inst.refresh(force=True)


# Using Virtual Media

# Instantiate a VirtualMediaCollection object
virtmedia_col = mgr_inst.virtual_media

# Print the ID of the VirtualMedia available in the collection
print(virtmedia_col.members_identities)

# Get a list of VirtualMedia objects available in the collection
virtmedia_insts = virtmedia_col.get_members()

# Instantiate a VirtualMedia object
virtmedia_inst = virtmedia_col.get_member(
    virtmedia_col.members_identities[0])


# Print out some of the VirtualMedia properties
print(virtmedia_inst.name,
      virtmedia_inst.media_types)

# Insert virtual media (invalidates virtmedia_inst contents)
virtmedia_inst.insert_media('https://www.dmtf.org/freeImages/Sardine.img')

# Refresh the resource to load actual contents
virtmedia_inst.refresh()

# Print out some of the VirtualMedia properties
print(virtmedia_inst.image,
      virtmedia_inst.image_path,
      virtmedia_inst.inserted,
      virtmedia_inst.write_protected)

# ... Boot the system off the virtual media...

# Eject virtual media (invalidates virtmedia_inst contents)
virtmedia_inst.eject_media()
```

Creating and using a sushy client with Sessions

```
import logging

import sushy

# Enable logging at DEBUG level
LOG = logging.getLogger('sushy')
LOG.setLevel(logging.DEBUG)
LOG.addHandler(logging.StreamHandler())

s = sushy.Sushy('http://localhost:8000/redfish/v1',
                username='foo', password='bar')

# Get the ComputerSystem object (if there is only one), otherwise
# the identity must be provided as a path to the system.
system = s.get_system()

# A session is created automatically for you.
# Print the boot field in the ComputerSystem.
print(system.boot)

# Upon session timeout, Sushy recreates the session based upon
# provided credentials. If this fails, an exception is raised.

# Explicitly request a session_key and session_uri.
# This is not stored, but may be useful.
session_key, session_uri = s.create_session(username='foo',
                                            password='bar')

# Retrieve the session
session = s.get_session(session_uri)

# Delete the session
session.delete()
```

Using OEM extensions

Before running this example, please make sure you have a Redfish BMC that includes the OEM piece for a specific vendor, as well as the Sushy OEM extension package installed in the system for the same vendor.

You can check the presence of the OEM extension within each Redfish resource by specifying the vendor ID and search for them.

In the following example, we are looking up “Acme” vendor extension to Redfish Manager resource.

```
import sushy

root = sushy.Sushy('http://localhost:8000/redfish/v1')

# Instantiate a system object
system = root.get_system('/redfish/v1/Systems/437XR1138R2')

print('Working on system resource %s' % system.identity)

```

(continues on next page)

(continued from previous page)

```
for manager in system.managers:

    print('Using System manager %s' % manager.identity)

    # Get a list of OEM extension names for the system manager
    oem_vendors = manager.oem_vendors

    print('Listing OEM extension name(s) for the System '
          'manager %s' % manager.identity )

    print(*oem_vendors, sep="\n")

    try:
        manager_oem = manager.get_oem_extension('Acme')

    except sushy.exceptions.OEMExtensionNotFoundError:
        print('ERROR: Acme OEM extension not found in '
              'Manager %s' % manager.identity)
        continue

    print('%s is an OEM extension of Manager %s'
           % (manager_oem.get_extension(), manager.identity))

    # set boot device to a virtual media device image
    manager_oem.set_virtual_boot_device(sushy.VirtualMediaType.CD,
                                        manager=manager)
```

If you do not have any real baremetal machine that supports the Redfish protocol you can look at the *Contributing to Sushy* page to learn how to run a Redfish emulator.

For the OEM extension example, presently, both of the emulators (static/dynamic) do not expose any OEM; as a result, users may need to add manually some OEM resources to emulators' templates. It may be easier to start with a static emulator.

3.3.2 Sushy Python API Reference

- modindex

sushy

sushy package

Subpackages

sushy.resources package

Subpackages

sushy.resources.certificateservice package

Submodules

sushy.resources.certificateservice.certificate module

class sushy.resources.certificateservice.certificate.**Certificate**(*connector*, *path=''*, *redfish_version=None*, *registries=None*, *reader=None*, *json_doc=None*, *root=None*)

Bases: *ResourceBase*

certificate_string = <sushy.resources.base.Field object>

Certificate in the format defined by certificate_type

certificate_type = <sushy.resources.base.MappedField object>

The format of the certificate

certificate_usage_type = <sushy.resources.base.MappedField object>

The types or purposes for this certificate

delete()

Delete this certificate.

description = <sushy.resources.base.Field object>

Certificate description

fingerprint = <sushy.resources.base.Field object>

The fingerprint of the certificate

fingerprint_hash_algorithm = <sushy.resources.base.Field object>

The hash algorithm for the fingerprint of the certificate

identity = <sushy.resources.base.Field object>

The certificate identity string

issuer = <sushy.resources.certificateservice.certificate.Identifier object>

The issuer of the certificate

key_usage = <sushy.resources.base.MappedListField object>

The key usage extension, which defines the purpose of the public keys in this certificate

name = <sushy.resources.base.Field object>

The certificate name

serial_number = <sushy.resources.base.Field object>

The serial number of the certificate

signature_algorithm = <sushy.resources.base.Field object>

The algorithm used for creating the signature of the certificate

subject = <sushy.resources.certificateservice.certificate.Identifier object>

The subject of the certificate

uefi_signature_owner = <sushy.resources.base.Field object>

The UEFI signature owner for this certificate

valid_not_after = <sushy.resources.base.Field object>

The date when the certificate is no longer valid

valid_not_before = <sushy.resources.base.Field object>

The date when the certificate becomes valid

class `sushy.resources.certificateservice.certificate.`**CertificateCollection**(*connect*
path,
redfish_ver
registries=
root=No

Bases: *MutableResourceCollectionBase*

create_member(*certificate_string*, *certificate_type*)

Create a new member of this collection.

Parameters

- **certificate_string** – the contents of the new certificate.
- **certificate_type** – the type of the new certificate, one of `sushy.CertificateType`.

class `sushy.resources.certificateservice.certificate.`**Identifier**(**args*, ***kwargs*)

Bases: *CompositeField*

The identifier information about a certificate.

city = <sushy.resources.base.Field object>

common_name = <sushy.resources.base.Field object>

country = <sushy.resources.base.Field object>

email = <sushy.resources.base.Field object>

organization = <sushy.resources.base.Field object>

organizational_unit = <sushy.resources.base.Field object>

state = <sushy.resources.base.Field object>

sushy.resources.certificateservice.certificateservice module

class sushy.resources.certificateservice.certificateservice.**ActionsField**(**args*, ***kwargs*)

Bases: *CompositeField*

generate_csr = <sushy.resources.common.ActionField object>

replace_certificate = <sushy.resources.common.ActionField object>

class sushy.resources.certificateservice.certificateservice.**CertificateLocations**

Bases: *ResourceLinksBase*

property members_identities

A sequence with members identities

name = <sushy.resources.base.Field object>

The name of the collection

class sushy.resources.certificateservice.certificateservice.**CertificateService**(*c*
p
re
fi
re
is
re
js
r

Bases: *ResourceBase*

property certificate_locations

Property to reference certificate locations instance

identity = <sushy.resources.base.Field object>

The certificate service identity

name = <sushy.resources.base.Field object>

The certificate service name

replace_certificate(*certificate_uri*, *certificate_string*, *certificate_type*)

Replace an existing certificate in the service.

Parameters

- **certificate_uri** – URI of an existing certificate.
- **certificate_string** – the contents of the new certificate.
- **certificate_type** – the type of the new certificate, one of sushy. CertificateType.

sushy.resources.certificateservice.constants module

class `sushy.resources.certificateservice.constants.`**`CertificateType`**`(`*value*`)`

Bases: `Enum`

An enumeration.

`PEM = 'PEM'`

A Privacy Enhanced Mail (PEM)-encoded single certificate.

`PEM_CHAIN = 'PEMchain'`

A Privacy Enhanced Mail (PEM)-encoded certificate chain.

`PKCS7 = 'PKCS7'`

A Privacy Enhanced Mail (PEM)-encoded PKCS7 certificate.

class `sushy.resources.certificateservice.constants.`**`CertificateUsageType`**`(`*value*`)`

Bases: `Enum`

An enumeration.

`BIOS = 'BIOS'`

This certificate is a BIOS certificate like those associated with UEFI.

`DEVICE = 'Device'`

This certificate is a device type certificate like those associated with SPDM and other standards.

`PLATFORM = 'Platform'`

This certificate is a platform type certificate like those associated with SPDM and other standards.

`SSH = 'SSH'`

This certificate is used for SSH.

`USER = 'User'`

This certificate is a user certificate like those associated with a manager account.

`WEB = 'Web'`

This certificate is a web or HTTPS certificate like those used for event destinations.

class `sushy.resources.certificateservice.constants.`**`KeyUsage`**`(`*value*`)`

Bases: `Enum`

An enumeration.

`CLIENT_AUTHENTICATION = 'ClientAuthentication'`

TLS WWW client authentication.

`CODE_SIGNING = 'CodeSigning'`

Signs downloadable executable code.

`CRL_SIGNING = 'CRLSigning'`

Verifies signatures on certificate revocation lists (CRLs).

`DATA_ENCIPHERMENT = 'DataEncipherment'`

Directly enciphers raw user data without an intermediate symmetric cipher.

DECIPHER_ONLY = 'DecipherOnly'

Deciphers data while performing a key agreement.

DIGITAL_SIGNATURE = 'DigitalSignature'

Verifies digital signatures, other than signatures on certificates and CRLs.

EMAIL_PROTECTION = 'EmailProtection'

Email protection.

ENCIPHER_ONLY = 'EncipherOnly'

Enciphers data while performing a key agreement.

KEY_AGREEMENT = 'KeyAgreement'

Key agreement.

KEY_CERT_SIGN = 'KeyCertSign'

Verifies signatures on public key certificates.

KEY_ENCIPHERMENT = 'KeyEncipherment'

Enciphers private or secret keys.

NON_REPUDIATION = 'NonRepudiation'

Verifies digital signatures, other than signatures on certificates and CRLs, and provides a non-repudiation service that protects against the signing entity falsely denying some action.

OCSP_SIGNING = 'OCSPSigning'

Signs OCSP responses.

SERVER_AUTHENTICATION = 'ServerAuthentication'

TLS WWW server authentication.

TIMESTAMPING = 'Timestamping'

Binds the hash of an object to a time.

Module contents

sushy.resources.chassis package

Subpackages

sushy.resources.chassis.power package

Submodules

sushy.resources.chassis.power.constants module

class sushy.resources.chassis.power.constants.**LineInputVoltageType**(*value*)

Bases: Enum

An enumeration.

AC_120V = 'AC120V'
AC 120V nominal input.

AC_240V = 'AC240V'
AC 240V nominal input.

AC_277V = 'AC277V'
AC 277V nominal input.

AC_AND_DC_WIDE_RANGE = 'ACandDCWideRange'
Wide range AC or DC input.

AC_HIGH_LINE = 'ACHighLine'
277V AC input.

AC_LOW_LINE = 'ACLowLine'
100-127V AC input.

AC_MID_LINE = 'ACMidLine'
200-240V AC input.

AC_WIDE_RANGE = 'ACWideRange'
Wide range AC input.

DC_240V = 'DC240V'
DC 240V nominal input.

DC_380V = 'DC380V'
High Voltage DC input (380V).

DC_NEG48V = 'DCNeg48V'
-48V DC input.

UNKNOWN = 'Unknown'
The power supply line input voltage type cannot be determined.

class sushy.resources.chassis.power.constants.**PowerInputType**(*value*)

Bases: Enum

An enumeration.

AC = 'AC'
Alternating Current (AC) input range.

DC = 'DC'
Direct Current (DC) input range.

class sushy.resources.chassis.power.constants.**PowerSupplyType**(*value*)

Bases: Enum

An enumeration.

AC = 'AC'
Alternating Current (AC) power supply.

AC_OR_DC = 'ACorDC'
The power supply supports both DC or AC.

DC = 'DC'

Direct Current (DC) power supply.

UNKNOWN = 'Unknown'

The power supply type cannot be determined.

sushy.resources.chassis.power.power module

class sushy.resources.chassis.power.power.**InputRangeListField**(**args*, ***kwargs*)

Bases: *ListField*

This type describes an input range for a power supply

input_type = <sushy.resources.base.MappedField object>

The Input type (AC or DC)

maximum_frequency_hz = <sushy.resources.base.Field object>

The maximum line input frequency at which this power supply input range is effective

maximum_voltage = <sushy.resources.base.Field object>

The maximum line input voltage at which this power supply input range is effective

minimum_frequency_hz = <sushy.resources.base.Field object>

The minimum line input frequency at which this power supply input range is effective

minimum_voltage = <sushy.resources.base.Field object>

The minimum line input voltage at which this power supply input range is effective

output_wattage = <sushy.resources.base.Field object>

The maximum capacity of this Power Supply when operating in this input range

class sushy.resources.chassis.power.power.**Power**(*connector*, *path=''*, *redfish_version=None*, *registries=None*, *reader=None*, *json_doc=None*, *root=None*)

Bases: *ResourceBase*

This class represents a Power resource.

identity = <sushy.resources.base.Field object>

Identifier of the resource

name = <sushy.resources.base.Field object>

The name of the resource

power_supplies = <sushy.resources.chassis.power.power.PowerSupplyListField object>

Details of a power supplies associated with this system or device

class sushy.resources.chassis.power.power.**PowerSupplyListField**(**args*, ***kwargs*)

Bases: *ListField*

The power supplies associated with this Power resource

firmware_version = <sushy.resources.base.Field object>

The firmware version for this Power Supply

identity = <sushy.resources.base.Field object>

Identifier of the Power Supply

indicator_led = <sushy.resources.base.MappedField object>

The state of the indicator LED, used to identify the power supply

input_ranges = <sushy.resources.chassis.power.power.InputRangeListField object>

This is the input ranges that the power supply can use

last_power_output_watts = <sushy.resources.base.Field object>

The average power output of this Power Supply

line_input_voltage = <sushy.resources.base.Field object>

The line input voltage at which the Power Supply is operating

line_input_voltage_type = <sushy.resources.base.MappedField object>

The line voltage type supported as an input to this Power Supply

manufacturer = <sushy.resources.base.Field object>

This is the manufacturer of this power supply

model = <sushy.resources.base.Field object>

The model number for this Power Supply

name = <sushy.resources.base.Field object>

Name of the Power Supply

part_number = <sushy.resources.base.Field object>

The part number for this Power Supply

power_capacity_watts = <sushy.resources.base.Field object>

The maximum capacity of this Power Supply

power_supply_type = <sushy.resources.base.MappedField object>

The Power Supply type (AC or DC)

serial_number = <sushy.resources.base.Field object>

The serial number for this Power Supply

spare_part_number = <sushy.resources.base.Field object>

The spare part number for this Power Supply

status = <sushy.resources.common.StatusField object>

Status of the sensor

Module contents

sushy.resources.chassis.thermal package

Submodules

sushy.resources.chassis.thermal.constants module

class sushy.resources.chassis.thermal.constants.**FanReadingUnit**(*value*)

Bases: `Enum`

An enumeration.

PERCENT = 'Percent'

The fan reading and thresholds are measured as a percentage.

RPM = 'RPM'

The fan reading and thresholds are measured in revolutions per minute.

sushy.resources.chassis.thermal.thermal module

class sushy.resources.chassis.thermal.thermal.**FansListField**(**args*, ***kwargs*)

Bases: *Sensor*

The Fan device/s associated with Thermal.

indicator_led = <sushy.resources.base.MappedField object>

The state of the indicator LED, used to identify the fan

manufacturer = <sushy.resources.base.Field object>

This is the manufacturer of this Fan

max_reading_range = <sushy.resources.base.Field object>

Maximum value for Reading

min_reading_range = <sushy.resources.base.Field object>

Minimum value for Reading

model = <sushy.resources.base.Field object>

The model of this Fan

part_number = <sushy.resources.base.Field object>

Part number of this Fan

reading = <sushy.resources.base.Field object>

Current Fan Speed

reading_units = <sushy.resources.base.MappedField object>

Units in which the reading and thresholds are measured

serial_number = <sushy.resources.base.Field object>

Serial number of this Fan

class sushy.resources.chassis.thermal.thermal.**Sensor**(**args*, ***kwargs*)

Bases: `ListField`

The sensor device/s associated with Thermal.

identity = <sushy.resources.base.Field object>

Identifier of the Sensor

lower_threshold_critical = <sushy.resources.base.Field object>

Below normal range but not yet fatal

lower_threshold_fatal = <sushy.resources.base.Field object>

Below normal range and is fatal

lower_threshold_non_critical = <sushy.resources.base.Field object>

Below normal range

name = <sushy.resources.base.Field object>

The name of this sensor

physical_context = <sushy.resources.base.Field object>

Area or device associated with this sensor

status = <sushy.resources.common.StatusField object>

Status of the sensor

upper_threshold_critical = <sushy.resources.base.Field object>

Above normal range but not yet fatal

upper_threshold_fatal = <sushy.resources.base.Field object>

Above normal range and is fatal

upper_threshold_non_critical = <sushy.resources.base.Field object>

Above normal range

class sushy.resources.chassis.thermal.thermal.**TemperaturesListField**(**args*, ***kwargs*)

Bases: `Sensor`

The Temperature device/s associated with Thermal.

max_allowable_operating_value = <sushy.resources.base.Field object>

Maximum allowable operating temperature for this equipment

max_reading_range_temp = <sushy.resources.base.Field object>

Maximum value for ReadingCelsius

min_allowable_operating_value = <sushy.resources.base.Field object>

Minimum allowable operating temperature for this equipment

min_reading_range_temp = <sushy.resources.base.Field object>

Minimum value for ReadingCelsius

reading_celsius = <sushy.resources.base.Field object>

Temperature

sensor_number = <sushy.resources.base.Field object>

A numerical identifier to represent the temperature sensor

class `sushy.resources.chassis.thermal.thermal.`**Thermal**(*connector*, *path=''*, *redfish_version=None*, *registries=None*, *reader=None*, *json_doc=None*, *root=None*)

Bases: `ResourceBase`

This class represents a Thermal resource.

fans = <sushy.resources.chassis.thermal.thermal.FansListField object>

A tuple of Fan identities

identity = <sushy.resources.base.Field object>

Identifier of the resource

name = <sushy.resources.base.Field object>

The name of the resource

status = <sushy.resources.common.StatusField object>

Status of the resource

temperatures = <sushy.resources.chassis.thermal.thermal.TemperaturesListField object>

A tuple of Temperature identities

Module contents

Submodules

sushy.resources.chassis.chassis module

class `sushy.resources.chassis.chassis.`**ActionsField**(**args*, ***kwargs*)

Bases: `CompositeField`

reset = <sushy.resources.common.ResetActionField object>

class `sushy.resources.chassis.chassis.`**Chassis**(*connector*, *identity*, *redfish_version=None*, *registries=None*, *root=None*)

Bases: `ResourceBase`

Chassis resource

The Chassis represents the physical components of a system. This resource represents the sheet-metal confined spaces and logical zones such as racks, enclosures, chassis and all other containers.

`asset_tag = <sushy.resources.base.Field object>`

The user assigned asset tag of this chassis

`chassis_type = <sushy.resources.base.MappedField object>`

The type of physical form factor of the chassis

`depth_mm = <sushy.resources.base.Field object>`

Depth in millimeters The depth of the chassis. The value of this property shall represent the depth (length) of the chassis (in millimeters) as specified by the manufacturer.

`description = <sushy.resources.base.Field object>`

The chassis description

`get_allowed_reset_chassis_values``()`

Get the allowed values for resetting the chassis.

Returns

A set of allowed values.

Raises

MissingAttributeError, if Actions/#Chassis.Reset attribute not present.

`height_mm = <sushy.resources.base.Field object>`

Height in millimeters The height of the chassis. The value of this property shall represent the height of the chassis (in millimeters) as specified by the manufacturer.

`identity = <sushy.resources.base.Field object>`

Identifier for the chassis

`indicator_led = <sushy.resources.base.MappedField object>`

The state of the indicator LED, used to identify the chassis

`property managers`

A list of managers for this chassis.

Returns a list of *Manager* objects representing the managers that manage this chassis.

Raises

MissingAttributeError if '@odata.id' field is missing.

Returns

A list of *Manager* instances

`manufacturer = <sushy.resources.base.Field object>`

The manufacturer of this chassis

`model = <sushy.resources.base.Field object>`

The model number of the chassis

`name = <sushy.resources.base.Field object>`

The chassis name

property network_adapters

Property to reference *NetworkAdapterCollection* instance

It is set once when the first time it is queried. On refresh, this property is marked as stale (greedy-refresh not done). Here the actual refresh of the sub-resource happens, if stale.

part_number = <sushy.resources.base.Field object>

The part number of the chassis

physical_security = <sushy.resources.chassis.chassis.PhysicalSecurity object>

PhysicalSecurity This value of this property shall contain the sensor state of the physical security.

property power

Property to reference *Power* instance

It is set once when the first time it is queried. On refresh, this property is marked as stale (greedy-refresh not done). Here the actual refresh of the sub-resource happens, if stale.

power_state = <sushy.resources.base.MappedField object>

The current power state of the chassis

reset_chassis(*value*)

Reset the chassis.

Parameters

`value` – The target value.

Raises

InvalidParameterValueError, if the target value is not allowed.

serial_number = <sushy.resources.base.Field object>

The serial number of the chassis

set_indicator_led(*state*)

Set IndicatorLED to the given state.

Parameters

`state` – Desired LED state, an IndicatorLED value.

Raises

InvalidParameterValueError, if any information passed is invalid.

sku = <sushy.resources.base.Field object>

Stock-keeping unit number (SKU) The value of this property shall be the stock-keeping unit number for this chassis.

status = <sushy.resources.common.StatusField object>

Status and Health This property describes the status and health of the chassis and its children.

property systems

A list of systems residing in this chassis.

Returns a list of *System* objects representing systems being mounted in this chassis/cabinet.

Raises

MissingAttributeError if '@odata.id' field is missing.

Returns
A list of *System* instances

property thermal

Property to reference *Thermal* instance

It is set once when the first time it is queried. On refresh, this property is marked as stale (greedy-refresh not done). Here the actual refresh of the sub-resource happens, if stale.

uuid = <sushy.resources.base.Field object>

The Universal Unique Identifier (UUID) for this Chassis.

weight_kg = <sushy.resources.base.Field object>

Weight in kilograms The value of this property shall represent the published mass (commonly referred to as weight) of the chassis (in kilograms).

width_mm = <sushy.resources.base.Field object>

Width in millimeters The value of this property shall represent the width of the chassis (in millimeters) as specified by the manufacturer.

class sushy.resources.chassis.chassis.**ChassisCollection**(*connector*, *path*, *redfish_version=None*, *registries=None*, *root=None*)

Bases: `ResourceCollectionBase`

class sushy.resources.chassis.chassis.**PhysicalSecurity**(**args*, ***kwargs*)

Bases: `CompositeField`

intrusion_sensor = <sushy.resources.base.MappedField object>

IntrusionSensor This indicates the known state of the physical security sensor, such as if it is hardware intrusion detected.

intrusion_sensor_number = <sushy.resources.base.Field object>

A numerical identifier to represent the physical security sensor

intrusion_sensor_re_arm = <sushy.resources.base.MappedField object>

This indicates how the Normal state to be restored

sushy.resources.chassis.constants module

class sushy.resources.chassis.constants.**ChassisType**(*value*)

Bases: `Enum`

Chassis Types constants

BLADE = 'Blade'

An enclosed or semi-enclosed, typically vertically-oriented, system chassis that must be plugged into a multi-system chassis to function normally.

CARD = 'Card'

A loose device or circuit board intended to be installed in a system or other enclosure.

CARTRIDGE = 'Cartridge'

A small self-contained system intended to be plugged into a multi- system chassis.

COMPONENT = 'Component'

A small chassis, card, or device that contains devices for a particular subsystem or function.

DRAWER = 'Drawer'

An enclosed or semi-enclosed, typically horizontally-oriented, system chassis that can be slid into a multi-system chassis.

ENCLOSURE = 'Enclosure'

A generic term for a chassis that does not fit any other description.

EXPANSION = 'Expansion'

A chassis that expands the capabilities or capacity of another chassis.

IP_BASED_DRIVE = 'IPBasedDrive'

A chassis in a drive form factor with IP-based network connections.

MODULE = 'Module'

A small, typically removable, chassis or card that contains devices for a particular subsystem or function.

OTHER = 'Other'

A chassis that does not fit any of these definitions.

POD = 'Pod'

A collection of equipment racks in a large, likely transportable, container.

RACK = 'Rack'

An equipment rack, typically a 19-inch wide freestanding unit.

RACK_GROUP = 'RackGroup'

A group of racks that form a single entity or share infrastructure.

RACK_MOUNT = 'RackMount'

A single-system chassis designed specifically for mounting in an equipment rack.

ROW = 'Row'

A collection of equipment racks.

SHELF = 'Shelf'

An enclosed or semi-enclosed, typically horizontally-oriented, system chassis that must be plugged into a multi-system chassis to function normally.

SIDECAR = 'Sidecar'

A chassis that mates mechanically with another chassis to expand its capabilities or capacity.

SLED = 'Sled'

An enclosed or semi-enclosed, system chassis that must be plugged into a multi-system chassis to function normally similar to a blade type chassis.

STAND_ALONE = 'StandAlone'

A single, free-standing system, commonly called a tower or desktop chassis.

STORAGE_ENCLOSURE = 'StorageEnclosure'

A chassis that encloses storage.

ZONE = 'Zone'

A logical division or portion of a physical chassis that contains multiple devices or systems that cannot be physically separated.

class `sushy.resources.chassis.constants.`**`IntrusionSensor`**(*value*)

Bases: `Enum`

Chassis IntrusionSensor constants

HARDWARE_INTRUSION = 'HardwareIntrusion'

A door, lock, or other mechanism protecting the internal system hardware from being accessed is detected to be in an insecure state.

NORMAL = 'Normal'

No abnormal physical security condition is detected at this time.

TAMPERING_DETECTED = 'TamperingDetected'

Physical tampering of the monitored entity is detected.

class `sushy.resources.chassis.constants.`**`IntrusionSensorReArm`**(*value*)

Bases: `Enum`

Chassis IntrusionSensorReArm constants

AUTOMATIC = 'Automatic'

Because no abnormal physical security condition is detected, this sensor is automatically restored to the normal state.

MANUAL = 'Manual'

A manual re-arm of this sensor restores it to the normal state.

Module contents

sushy.resources.compositionservice package

Submodules

sushy.resources.compositionservice.compositionservice module

class `sushy.resources.compositionservice.compositionservice.`**`CompositionService`**(*co*
id
ti
re
fi
re
is
ro

Bases: `ResourceBase`

allow_overprovisioning = <sushy.resources.base.Field object>

This indicates whether this service is allowed to overprovision

allow_zone_affinity = <sushy.resources.base.Field object>

This indicates whether a client is allowed to request that given composition request

description = <sushy.resources.base.Field object>

The composition service description

identity = <sushy.resources.base.Field object>

The composition service identity string

name = <sushy.resources.base.Field object>

The composition service name

property resource_blocks

Property to reference *ResourceBlockCollection* instance

property resource_zones

Property to reference *ResourceZoneCollection* instance

service_enabled = <sushy.resources.base.Field object>

The status of composition service is enabled

status = <sushy.resources.common.StatusField object>

The status of composition service

sushy.resources.compositionservice.constants module

class sushy.resources.compositionservice.constants.**CompositionState**(*value*)

Bases: Enum

An enumeration.

COMPOSED = 'Composed'

Final successful state of a Resource Block which has participated in composition.

COMPOSED_AND_AVAILABLE = 'ComposedAndAvailable'

Indicates the Resource Block is currently participating in one or more compositions, and is available to be used in more compositions.

COMPOSING = 'Composing'

Intermediate state indicating composition is in progress.

FAILED = 'Failed'

The final composition resulted in failure and manual intervention may be required to fix it.

UNAVAILABLE = 'Unavailable'

Indicates the Resource Block has been made unavailable by the service, such as due to maintenance being performed on the Resource Block.

UNUSED = 'Unused'

Indicates the Resource Block is free and can participate in composition.

class `sushy.resources.compositionservice.constants.`**`ResourceBlockType`**`(`*value*`)`

Bases: `Enum`

An enumeration.

`COMPUTE = 'Compute'`

This Resource Block contains both Processor and Memory resources in a manner that creates a compute complex.

`COMPUTER_SYSTEM = 'ComputerSystem'`

This Resource Block contains ComputerSystem resources.

`EXPANSION = 'Expansion'`

This Resource Block is capable of changing over time based on its configuration. Different types of devices within this Resource Block can be added and removed over time.

`MEMORY = 'Memory'`

This Resource Block contains Memory resources.

`NETWORK = 'Network'`

This Resource Block contains Network resources, such as Ethernet Interfaces.

`PROCESSOR = 'Processor'`

This Resource Block contains Processor resources.

`STORAGE = 'Storage'`

This Resource Block contains Storage resources, such as Storage and Simple Storage.

sushy.resources.compositionservice.resourceblock module

class `sushy.resources.compositionservice.resourceblock.`**`CompositionStatusField`**`(`**ar*
****

Bases: *`CompositeField`*

`composition_state = <sushy.resources.base.MappedField object>`

Inform the client, state of the resource block

`max_compositions = <sushy.resources.base.Field object>`

The maximum number of compositions

`number_of_compositions = <sushy.resources.base.Field object>`

The number of compositions

`reserved_state = <sushy.resources.base.Field object>`

Inform the resource block has been identified by a client

`sharing_capable = <sushy.resources.base.Field object>`

Indicates if this Resource Block is capable of participating in multiple compositions simultaneously

`sharing_enabled = <sushy.resources.base.Field object>`

Indicates if this Resource Block is allowed to participate in multiple compositions simultaneously

class sushy.resources.compositionservice.resourceblock.**ResourceBlock**(*connector*, *identity*, *redfish_version=No*
registries=None, *root=None*)

Bases: *ResourceBase*

composition_status = <sushy.resources.compositionservice.resourceblock.CompositionStatusField object>

The composition state of resource block

description = <sushy.resources.base.Field object>

The resource block description

identity = <sushy.resources.base.Field object>

The resource block identity string

name = <sushy.resources.base.Field object>

The resource block name

resource_block_type = <sushy.resources.base.MappedField object>

The type of resource block

status = <sushy.resources.common.StatusField object>

The status of resource block

class sushy.resources.compositionservice.resourceblock.**ResourceBlockCollection**(*c*
id
ti
re
fi
re
is
ro

Bases: *ResourceCollectionBase*

description = <sushy.resources.base.Field object>

The resource block collection description

name = <sushy.resources.base.Field object>

The resource block collection name

sushy.resources.compositionservice.resourcezone module

class sushy.resources.compositionservice.resourcezone.**LinksField**(**args*, ***kwargs*)

Bases: *CompositeField*

endpoints = <sushy.resources.base.Field object>

The references to the endpoints that are contained in this zone

involved_switches = <sushy.resources.base.Field object>

The references to the switches in this zone

resource_blocks = <sushy.resources.base.Field object>

The references to the Resource Blocks that are used in this zone

class sushy.resources.compositionservice.resourcezone.**ResourceZone**(*connector*, *identity*, *redfish_version=None*, *registries=None*, *root=None*)

Bases: *ResourceBase*

description = <sushy.resources.base.Field object>

The resources zone description

identity = <sushy.resources.base.Field object>

The resource zone identity string

links = <sushy.resources.compositionservice.resourcezone.LinksField object>

The references to other resources that are related to this resource

name = <sushy.resources.base.Field object>

The resource zone name

status = <sushy.resources.common.StatusField object>

The resource zone status

class sushy.resources.compositionservice.resourcezone.**ResourceZoneCollection**(*conn* *identity*, *red-* *fish_* *reg-* *istrie* *root=*

Bases: *ResourceCollectionBase*

description = <sushy.resources.base.Field object>

The resource zone collection description

name = <sushy.resources.base.Field object>

The resource zone collection name

Module contents

sushy.resources.eventservice package

Submodules

sushy.resources.eventservice.constants module

class sushy.resources.eventservice.constants.**EventType**(*value*)

Bases: Enum

An enumeration.

ALERT = 'Alert'

A condition requires attention.

METRIC_REPORT = 'MetricReport'

The telemetry service is sending a metric report.

OTHER = 'Other'

Because EventType is deprecated as of Redfish Specification v1.6, the event is based on a registry or resource but not an EventType.

RESOURCE_ADDED = 'ResourceAdded'

A resource has been added.

RESOURCE_REMOVED = 'ResourceRemoved'

A resource has been removed.

RESOURCE_UPDATED = 'ResourceUpdated'

A resource has been updated.

STATUS_CHANGE = 'StatusChange'

The status of a resource has changed.

sushy.resources.eventservice.eventdestination module

class sushy.resources.eventservice.eventdestination.**EventDestination**(*connector*, *identity*, *redfish_version=No*, *registries=None*, *root=None*)

Bases: *ResourceBase*

context = <sushy.resources.base.Field object>

A client-supplied string that is stored with the event destination subscription

delete()

Delete an EventDestination

Raises
ConnectionError

Raises
HTTPError

description = <sushy.resources.base.Field object>

The description of the EventDestination resource

destination = <sushy.resources.base.Field object>

The URI of the destination Event Service

event_types = <sushy.resources.base.Field object>

The types of events that shall be sent to the destination

http_headers = <sushy.resources.base.Field object>

This is for setting HTTP headers, such as authorization information. This object will be null on a GET.

identity = <sushy.resources.base.Field object>

The EventDestination resource identity

name = <sushy.resources.base.Field object>

The EventDestination resource name

protocol = <sushy.resources.base.Field object>

Contain the protocol type that the event will use for sending the event to the destination. A value of Redfish shall be used to indicate that the event type shall adhere to that defined in the Redfish specification

class sushy.resources.eventservice.eventdestination.**EventDestinationCollection**(*co*
id
ti
re
fi
re
is
ro

Bases: `ResourceCollectionBase`

create(*payload*)

Create a Subscription

Parameters
payload – The payload representing the subscription.

Raises
ConnectionError

Raises
HTTPError

Returns
The new subscription

description = <sushy.resources.base.Field object>

The EventDestination collection description

name = <sushy.resources.base.Field object>

The EventDestination collection name

sushy.resources.eventservice.eventservice module

class sushy.resources.eventservice.eventservice.**ActionsField**(**args*, ***kwargs*)

Bases: *CompositeField*

submit_test_event = <sushy.resources.common.ActionField object>

class sushy.resources.eventservice.eventservice.**EventService**(*connector*, *identity*, *redfish_version=None*, *registries=None*, *root=None*)

Bases: *ResourceBase*

delivery_retry_attempts = <sushy.resources.base.Field object>

Number of attempts an event posting is retried before the subscription is terminated. This retry is at the service level, meaning the HTTP POST to the Event Destination was returned by the HTTP operation as unsuccessful (4xx or 5xx return code) or an HTTP timeout occurred this many times before the Event Destination subscription is terminated

delivery_retry_interval = <sushy.resources.base.Field object>

Number of seconds between retry attempts for sending any given Event

event_types_for_subscription = <sushy.resources.base.Field object>

Types of Events that can be subscribed to

get_event_types_for_subscription()

Get the Types of Events that can be subscribed to

Returns
A set with the types of Events that can be subscribed to.

identity = <sushy.resources.base.Field object>

The EventService resource identity

name = <sushy.resources.base.Field object>

The EventService resource name

service_enabled = <sushy.resources.base.Field object>

Indicates whether the EventService is enabled

status = <sushy.resources.common.StatusField object>

The status of the EventService

submit_test_event (*event_id*, *event_timestamp*, *event_type*, *message*, *message_args*, *message_id*, *origin*, *severity*)

Submit Test Event is used to to send a test event to the BMC

Parameters

- **event_id** – ID of event to be added.
- **event_timestamp** – time stamp of event to be added.
- **event_type** – type of event to be added.
- **message** – human readable message of event to be added.
- **message_args** – array of message arguments of the event to be added.
- **message_id** – message ID of event to be added.
- **origin** – string of the URL within the OriginOfCondition property of the event to be added
- **severity** – the Severity of event to be added.
- **target** – The link to invoke action.

Raises

MissingActionError if the EvenService does not have the action.

property subscriptions

Reference to a collection of Event Destination resources

Module contents

sushy.resources.fabric package

Submodules

sushy.resources.fabric.constants module

class sushy.resources.fabric.constants.**EntityRole** (*value*)

Bases: `Enum`

Entity role constants

BOTH = 'Both'

The entity can both send and receive commands, messages, and other requests to or from other entities on the fabric.

INITIATOR = 'Initiator'

The entity sends commands, messages, or other types of requests to other entities on the fabric, but cannot receive commands from other entities.

TARGET = 'Target'

The entity receives commands, messages, or other types of requests from other entities on the fabric, but cannot send commands to other entities.

class sushy.resources.fabric.constants.**EntityType**(*value*)

Bases: Enum

Entity type constants

ACCELERATION_FUNCTION = 'AccelerationFunction'

The entity is an acceleration function realized through a device, such as an FPGA.

BRIDGE = 'Bridge'

The entity is a PCI(e) bridge.

DISPLAY_CONTROLLER = 'DisplayController'

The entity is a display controller.

DRIVE = 'Drive'

The entity is a drive.

FABRIC_BRIDGE = 'FabricBridge'

The entity is a fabric bridge.

MANAGER = 'Manager'

The entity is a manager.

MEDIA_CONTROLLER = 'MediaController'

The entity is a media controller.

MEMORY_CHUNK = 'MemoryChunk'

The entity is a memory chunk.

NETWORK_CONTROLLER = 'NetworkController'

The entity is a network controller.

PROCESSOR = 'Processor'

The entity is a processor.

ROOT_COMPLEX = 'RootComplex'

The entity is a PCI(e) root complex.

STORAGE_EXPANDER = 'StorageExpander'

The entity is a storage expander.

STORAGE_INITIATOR = 'StorageInitiator'

The entity is a storage initiator.

STORAGE_SUBSYSTEM = 'StorageSubsystem'

The entity is a storage subsystem.

SWITCH = 'Switch'

The entity is a switch, not an expander. Use *Expander* for expanders.

VOLUME = 'Volume'

The entity is a volume.

sushy.resources.fabric.endpoint module

class sushy.resources.fabric.endpoint.**ConnectedEntitiesListField**(**args*, ***kwargs*)

Bases: *ListField*

All the entities connected to this endpoint.

entity_pci_id = <sushy.resources.fabric.endpoint.PciIdField object>

The PCI ID of the connected entity.

entity_role = <sushy.resources.base.MappedField object>

The role of the connected entity.

entity_type = <sushy.resources.base.MappedField object>

The type of the connected entity.

identifiers = <sushy.resources.common.IdentifiersListField object>

Identifiers for the remote entity.

pci_class_code = <sushy.resources.base.Field object>

The Class Code, Subclass code, and Programming Interface code of this PCIe function.

pci_function_number = <sushy.resources.base.Field object>

The PCI ID of the connected entity.

class sushy.resources.fabric.endpoint.**Endpoint**(*connector*, *path=''*, *redfish_version=None*, *registries=None*, *reader=None*, *json_doc=None*, *root=None*)

Bases: *ResourceBase*

This class represents a fabric endpoint.

It represents the properties of an entity that sends or receives protocol defined messages over a transport.

IP_transport_details = <sushy.resources.fabric.endpoint.IPTransportDetailsListField object>

This array contains details for each IP transport supported by this endpoint. The array structure can be used to model multiple IP addresses for this endpoint.

connected_entities = <sushy.resources.fabric.endpoint.ConnectedEntitiesListField object>

All entities connected to this endpoint.

description = <sushy.resources.base.Field object>

The endpoint description

endpoint_protocol = <sushy.resources.base.MappedField object>

The protocol supported by this endpoint.

host_reservation_memory_bytes = <sushy.resources.base.Field object>

The amount of memory in Bytes that the Host should allocate to connect to this endpoint.

identity = <sushy.resources.base.Field object>

Identifier for the endpoint

name = <sushy.resources.base.Field object>

The endpoint name

pci_id = <sushy.resources.fabric.endpoint.PciIdField object>

The PCI ID of the endpoint.

status = <sushy.resources.common.StatusField object>

The endpoint status

class sushy.resources.fabric.endpoint.**EndpointCollection**(*connector*, *path*, *redfish_version=None*, *registries=None*, *root=None*)

Bases: *ResourceCollectionBase*

Represents a collection of endpoints associated with the fabric.

class sushy.resources.fabric.endpoint.**IPTransportDetailsListField**(**args*, ***kwargs*)

Bases: *ListField*

IP transport details

This array contains details for each IP transport supported by this endpoint. The array structure can be used to model multiple IP addresses for this endpoint.

ipv4_address = <sushy.resources.fabric.endpoint.IPv4AddressField object>

The IPv4 address object.

ipv6_address = <sushy.resources.fabric.endpoint.IPv6AddressField object>

The IPv6 address object.

port = <sushy.resources.base.Field object>

The UDP or TCP port number used by the Endpoint.

transport_protocol = <sushy.resources.base.MappedField object>

The protocol used by the connection entity.

class sushy.resources.fabric.endpoint.**IPv4AddressField**(**args*, ***kwargs*)

Bases: *CompositeField*

address = <sushy.resources.base.Field object>

This is the IPv4 Address.

address_origin = <sushy.resources.base.MappedField object>

This indicates how the address was determined.

gateway = <sushy.resources.base.Field object>

This is the IPv4 gateway for this address.

subnet_mask = <sushy.resources.base.Field object>

This is the IPv4 Subnet mask.

class sushy.resources.fabric.endpoint.**IPv6AddressField**(**args*, ***kwargs*)

Bases: *CompositeField*

address = <sushy.resources.base.Field object>

This is the IPv6 Address.

address_origin = <sushy.resources.base.MappedField object>

This indicates how the address was determined.

address_state = <sushy.resources.base.MappedField object>

The current state of this address as defined in RFC 4862.

prefix_length = <sushy.resources.base.Field object>

This is the IPv6 Address Prefix Length.

class sushy.resources.fabric.endpoint.**PciIdField**(**args*, ***kwargs*)

Bases: *CompositeField*

device_id = <sushy.resources.base.Field object>

The Device ID of this PCIe function.

subsystem_id = <sushy.resources.base.Field object>

The Subsystem ID of this PCIefunction.

subsystem_vendor_id = <sushy.resources.base.Field object>

The Subsystem Vendor ID of thisPCIe function.

vendor_id = <sushy.resources.base.Field object>

The Vendor ID of this PCIe function.

sushy.resources.fabric.fabric module

class sushy.resources.fabric.fabric.**Fabric**(*connector*, *identity*, *redfish_version=None*, *registries=None*, *root=None*)

Bases: *ResourceBase*

Fabric resource

The Fabric represents a simple fabric consisting of one or more switches, zero or more endpoints, and zero or more zones.

description = <sushy.resources.base.Field object>

The fabric description

property endpoints

fabric_type = <sushy.resources.base.MappedField object>

The protocol being sent over this fabric

identity = <sushy.resources.base.Field object>

Identifier for the fabric

max_zones = <sushy.resources.base.Field object>

The maximum number of zones the switch can currently configure

name = <sushy.resources.base.Field object>

The fabric name

status = <sushy.resources.common.StatusField object>

The fabric status

class sushy.resources.fabric.fabric.**FabricCollection**(*connector*, *path*, *redfish_version=None*, *registries=None*, *root=None*)

Bases: ResourceCollectionBase

Module contents

sushy.resources.manager package

Submodules

sushy.resources.manager.constants module

class sushy.resources.manager.constants.**CommandConnectType**(*value*)

Bases: Enum

Command Shell constants

IPMI = 'IPMI'

The controller supports a command shell connection through the IPMI Serial Over LAN (SOL) protocol.

OEM = 'Oem'

The controller supports a command shell connection through an OEM- specific protocol.

SSH = 'SSH'

The controller supports a command shell connection through the SSH protocol.

TELNET = 'Telnet'

The controller supports a command shell connection through the Telnet protocol.

class sushy.resources.manager.constants.**ConnectedVia**(*value*)

Bases: Enum

Connected Via constants

APPLET = 'Applet'

Connected to a client application.

NOT_CONNECTED = 'NotConnected'

No current connection.

OEM = 'Oem'

Connected through an OEM-defined method.

URI = 'URI'

Connected to a URI location.

class `sushy.resources.manager.constants.`**GraphicalConnectType**(*value*)

Bases: `Enum`

Graphical Console constants

KVMIP = 'KVMIP'

The controller supports a graphical console connection through a KVM- IP (redirection of Keyboard, Video, Mouse over IP) protocol.

OEM = 'Oem'

The controller supports a graphical console connection through an OEM-specific protocol.

class `sushy.resources.manager.constants.`**ManagerType**(*value*)

Bases: `Enum`

Manager Type constants

AUXILIARY_CONTROLLER = 'AuxiliaryController'

A controller that provides management functions for a particular subsystem or group of devices.

BMC = 'BMC'

A controller that provides management functions for a single computer system.

ENCLOSURE_MANAGER = 'EnclosureManager'

A controller that provides management functions for a chassis or group of devices or systems.

MANAGEMENT_CONTROLLER = 'ManagementController'

A controller that primarily monitors or manages the operation of a device or system.

RACK_MANAGER = 'RackManager'

A controller that provides management functions for a whole or part of a rack.

SERVICE = 'Service'

A software-based service that provides management functions.

`sushy.resources.manager.constants.`**RESET_MANAGER_FORCE_RESTART = ResetType.FORCE_RESTART**

Perform an immediate (non-graceful) shutdown, followed by a restart

`sushy.resources.manager.constants.`**RESET_MANAGER_GRACEFUL_RESTART = ResetType.GRACEFUL_RESTART**

Perform a graceful shutdown followed by a restart of the system

class `sushy.resources.manager.constants.`**SerialConnectType**(*value*)

Bases: `Enum`

Serial Console constants

IPMI = 'IPMI'

The controller supports a serial console connection through the IPMI Serial Over LAN (SOL) protocol.

OEM = 'Oem'

The controller supports a serial console connection through an OEM- specific protocol.

SSH = 'SSH'

The controller supports a serial console connection through the SSH protocol.

TELNET = 'Telnet'

The controller supports a serial console connection through the Telnet protocol.

class `sushy.resources.manager.constants.`**TransferMethod**(*value*)

Bases: `Enum`

Transfer methods

STREAM = 'Stream'

Stream image file data from the source URI.

UPLOAD = 'Upload'

Upload the entire image file from the source URI to the service.

class `sushy.resources.manager.constants.`**VirtualMediaType**(*value*)

Bases: `Enum`

Supported Virtual Media Type constants

CD = 'CD'

A CD-ROM format (ISO) image.

DVD = 'DVD'

A DVD-ROM format image.

FLOPPY = 'Floppy'

A floppy disk image.

USB_STICK = 'USBStick'

An emulation of a USB storage device.

sushy.resources.manager.manager module

class `sushy.resources.manager.manager.`**ActionsField**(**args*, ***kwargs*)

Bases: *CompositeField*

reset = <sushy.resources.common.ResetActionField object>

class `sushy.resources.manager.manager.`**Manager**(*connector*, *identity*, *redfish_version=None*, *registries=None*, *root=None*)

Bases: *ResourceBase*

auto_dst_enabled = <sushy.resources.base.Field object>

Indicates whether the manager is configured for automatic DST adjustment

property chassis

A list of chassis managed by this manager.

Returns a list of *Chassis* objects representing the chassis or cabinets managed by this manager.

Raises

MissingAttributeError if '@odata.id' field is missing.

Returns

A list of *Chassis* instances

command_shell = <sushy.resources.manager.manager.RemoteAccessField object>

A dictionary containing the remote access support service via command shell (e.g. Telnet, SSH) and max concurrent sessions

description = <sushy.resources.base.Field object>

The manager description

firmware_version = <sushy.resources.base.Field object>

The manager firmware version

get_allowed_reset_manager_values()

Get the allowed values for resetting the manager.

Returns

A set of allowed values.

Raises

MissingAttributeError, if Actions/#Manager.Reset attribute not present.

get_supported_command_shell_types()

Get the supported values for Command Shell connection types.

Returns

A set of supported values.

get_supported_graphical_console_types()

Get the supported values for Graphical Console connection types.

Returns

A set of supported values.

get_supported_serial_console_types()

Get the supported values for Serial Console connection types.

Returns

A set of supported values.

graphical_console = <sushy.resources.manager.manager.RemoteAccessField object>

A dictionary containing the remote access support service via graphical console (e.g. KVMIP) and max concurrent sessions

identity = <sushy.resources.base.Field object>

The manager identity string

manager_type = <sushy.resources.base.MappedField object>

The manager type

model = <sushy.resources.base.Field object>

The manager model

name = <sushy.resources.base.Field object>

The manager name

reset_manager(*value*)

Reset the manager.

Parameters

value – The target value.

Raises

InvalidParameterValueError, if the target value is not allowed.

serial_console = <sushy.resources.manager.manager.RemoteAccessField object>

A dictionary containing the remote access support service via serial console (e.g. Telnet, SSH, IPMI) and max concurrent sessions

property systems

A list of systems managed by this manager.

Returns a list of *System* objects representing systems being managed by this manager.

Raises

MissingAttributeError if '@odata.id' field is missing.

Returns

A list of *System* instances

uuid = <sushy.resources.base.Field object>

The manager UUID

property virtual_media

class sushy.resources.manager.manager.**ManagerCollection**(*connector*, *path*, *redfish_version=None*, *registries=None*, *root=None*)

Bases: *ResourceCollectionBase*

class sushy.resources.manager.manager.**RemoteAccessField**(**args*, ***kwargs*)

Bases: *CompositeField*

connect_types_supported = <sushy.resources.base.Field object>

max_concurrent_sessions = <sushy.resources.base.Field object>

service_enabled = <sushy.resources.base.Field object>

sushy.resources.manager.virtual_media module

class sushy.resources.manager.virtual_media.**ActionsField**(**args*, ***kwargs*)

Bases: `CompositeField`

eject_media = <sushy.resources.common.ActionField object>

insert_media = <sushy.resources.common.ActionField object>

class sushy.resources.manager.virtual_media.**VirtualMedia**(*connector*, *path=''*, *redfish_version=None*, *registries=None*, *reader=None*, *json_doc=None*, *root=None*)

Bases: `ResourceBase`

property certificates

Get the collection of certificates for this device.

connected_via = <sushy.resources.base.MappedField object>

Current virtual media connection methods

Applet: Connected to a client application NotConnected: No current connection Oem: Connected via an OEM-defined method URI: Connected to a URI location

eject_media()

Detach remote media from virtual media

After ejecting media inserted will be False and image_name will be empty.

identity = <sushy.resources.base.Field object>

Virtual Media resource identity string

image = <sushy.resources.base.Field object>

A URI providing the location of the selected image

image_name = <sushy.resources.base.Field object>

The image name

insert_media(*image*, *inserted=True*, *write_protected=True*, *username=None*, *password=None*, *transfer_method=None*)

Attach remote media to virtual media

Parameters

- **image** – a URI providing the location of the selected image
- **inserted** – specify if the image is to be treated as inserted upon completion of the action.
- **write_protected** – indicates the media is write protected
- **username** – User name for the image URI.
- **password** – Password for the image URI.

- **transfer_method** – Transfer method (stream or upload) to use for the image.

inserted = <sushy.resources.base.Field object>

Indicates if virtual media is inserted in the virtual device

is_transfer_method_required(*error=None*)

Check the response code and body and in case of failure

Try to determine if it happened due to missing TransferMethod

is_transfer_protocol_required(*error=None*)

Check the response code and body and in case of failure

Try to determine if it happened due to missing TransferProtocolType.

media_types = <sushy.resources.base.MappedListField object>

List of supported media types as virtual media

name = <sushy.resources.base.Field object>

The name of resource

set_verify_certificate(*verify_certificate*)

Enable or disable certificate validation.

status = <sushy.resources.common.StatusField object>

The virtual media status

transfer_method = <sushy.resources.base.MappedField object>

The transfer method to use with the Image

user_name = <sushy.resources.base.Field object>

The user name to access the Image parameter-specified URI

verify_certificate = <sushy.resources.base.Field object>

Whether to verify the certificate of the server for the Image

write_protected = <sushy.resources.base.Field object>

Indicates the media is write protected

class sushy.resources.manager.virtual_media.**VirtualMediaCollection**(*connector, path, redfish_version=None, registries=None, root=None*)

Bases: *ResourceCollectionBase*

A collection of virtual media attached to a Manager

Module contents

sushy.resources.oem package

Submodules

sushy.resources.oem.base module

class sushy.resources.oem.base.**OEMResourceBase**(*connector*, *path=''*, *redfish_version=None*, *registries=None*, *reader=None*, *root=None*)

Bases: *ResourceBase*

set_parent_resource(*parent_resource*, *vendor_id*)

sushy.resources.oem.common module

sushy.resources.oem.common.**get_resource_extension_by_vendor**(*resource_name*, *vendor*, *resource*)

Helper method to get Resource specific OEM extension object for vendor

Parameters

- **resource_name** – The underscore joined name of the resource e.g. 'system' / 'ethernet_interface' / 'update_service'
- **vendor** – This is the OEM vendor string which is the vendor-specific extensibility identifier. Examples are: 'Contoso', 'Hpe'. As a matter of fact the lowercase of this string will be the plugin entry point name.
- **resource** – The Sushy resource instance

Returns
The object returned by `plugin(*args, **kwds)` of extension.

Raises
OEMExtensionNotFoundError – if no valid resource OEM extension found.

sushy.resources.oem.fake module

class sushy.resources.oem.fake.**ContosoActionsField**(**args*, ***kwargs*)

Bases: *CompositeField*

reset = <sushy.resources.common.ResetActionField object>

class sushy.resources.oem.fake.**FakeOEMSystemExtension**(*connector*, *path=''*, *redfish_version=None*, *registries=None*, *reader=None*, *root=None*)

Bases: *OEMResourceBase*

data_type = <sushy.resources.base.Field object>

get_reset_system_path()

name = <sushy.resources.base.Field object>

production_location = <sushy.resources.oem.fake.ProductionLocationField object>

class sushy.resources.oem.fake.**ProductionLocationField**(**args*, ***kwargs*)

Bases: *CompositeField*

country = <sushy.resources.base.Field object>

facility_name = <sushy.resources.base.Field object>

sushy.resources.oem.fake.**get_extension**(**args*, ***kwargs*)

Module contents

sushy.resources.oem.**get_resource_extension_by_vendor**(*resource_name*, *vendor*, *resource*)

Helper method to get Resource specific OEM extension object for vendor

Parameters

- **resource_name** – The underscore joined name of the resource e.g. ‘system’ / ‘ethernet_interface’ / ‘update_service’
- **vendor** – This is the OEM vendor string which is the vendor-specific extensibility identifier. Examples are: ‘Contoso’, ‘Hpe’. As a matter of fact the lowercase of this string will be the plugin entry point name.
- **resource** – The Sushy resource instance

Returns

The object returned by `plugin(*args, **kwds)` of extension.

Raises

OEMExtensionNotFoundError – if no valid resource OEM extension found.

sushy.resources.registry package

Submodules

sushy.resources.registry.attribute_registry module

class sushy.resources.registry.attribute_registry.**AttributeListField**(**args*, ***kwargs*)

Bases: *ListField*

allowable_values = <sushy.resources.base.Field object>
An array of the possible values for enumerated attribute values

attribute_type = <sushy.resources.base.Field object>
The attribute type

default_value = <sushy.resources.base.Field object>
The default value for the attribute

display_name = <sushy.resources.base.Field object>
User-readable display string for attribute in the defined language

immutable = <sushy.resources.base.Field object>
An indication of whether this attribute is immutable

lower_bound = <sushy.resources.base.Field object>
The lower limit for an integer attribute

max_length = <sushy.resources.base.Field object>
The maximum character length of the string attribute

min_length = <sushy.resources.base.Field object>
The minimum character length of the string attribute

name = <sushy.resources.base.Field object>
The unique name for the attribute

read_only = <sushy.resources.base.Field object>
An indication of whether this attribute is read-only

reset_required = <sushy.resources.base.Field object>
Whether a System reset is required to change this attribute

unique = <sushy.resources.base.Field object>
Indicates whether this attribute is unique for this system

upper_bound = <sushy.resources.base.Field object>
The upper limit for an integer attribute

class sushy.resources.registry.attribute_registry.**AttributeRegistry**(*connector*, *path=''*, *redfish_version=None*, *registries=None*, *reader=None*, *json_doc=None*, *root=None*)

Bases: *ResourceBase*

description = <sushy.resources.base.Field object>
Human-readable description of the registry

identity = <sushy.resources.base.Field object>
The Attribute registry identity string

language = <sushy.resources.base.Field object>

RFC 5646 compliant language code for the registry

name = <sushy.resources.base.Field object>

The name of the attribute registry

owning_entity = <sushy.resources.base.Field object>

Organization or company that publishes this registry

registry_entries = <sushy.resources.registry.attribute_registry.AttributeRegistryEntryField object>

Field containing Attributes, Dependencies, Menus etc.

registry_version = <sushy.resources.base.Field object>

The version of this registry

supported_systems = <sushy.resources.base.Field object>

The system that this registry supports

class sushy.resources.registry.attribute_registry.**AttributeRegistryEntryField**(*ar
**k

Bases: *CompositeField*

attributes = <sushy.resources.registry.attribute_registry.AttributeListField object>

List of attributes in this registry

sushy.resources.registry.constants module

class sushy.resources.registry.constants.**MessageParamType**(*value*)

Bases: Enum

Message Registry message parameter type related constants.

NUMBER = 'number'

STRING = 'string'

sushy.resources.registry.message_registry module

class sushy.resources.registry.message_registry.**MessageDictionaryField**(*args, **kwargs)

Bases: *DictionaryField*

description = <sushy.resources.base.Field object>

Indicates how and when the message is returned by the Redfish service

message = <sushy.resources.base.Field object>

Template text of the message

Template can include placeholders for message arguments in form %<integer> where <integer> denotes a position passed from MessageArgs.

number_of_args = <sushy.resources.base.Field object>

Number of arguments to be expected to be passed in as MessageArgs for this message

param_types = <sushy.resources.base.Field object>

Mapped MessageArg types, in order, for the message

resolution = <sushy.resources.base.Field object>

Suggestions on how to resolve the situation that caused the error

severity = <sushy.resources.base.MappedField object>

Mapped severity of the message

class sushy.resources.registry.message_registry.**MessageRegistry**(*connector*, *path=''*, *redfish_version=None*, *registries=None*, *reader=None*, *json_doc=None*, *root=None*)

Bases: `ResourceBase`

description = <sushy.resources.base.Field object>

Human-readable description of the message registry

identity = <sushy.resources.base.Field object>

The Message registry identity string

language = <sushy.resources.base.Field object>

RFC 5646 compliant language code for the registry

messages = <sushy.resources.registry.message_registry.MessageDictionaryField object>

List of messages in this registry

name = <sushy.resources.base.Field object>

The name of the message registry

owning_entity = <sushy.resources.base.Field object>

Organization or company that publishes this registry

registry_prefix = <sushy.resources.base.Field object>

Prefix used in messageIDs which uniquely identifies all of the messages in this registry as belonging to this registry

registry_version = <sushy.resources.base.Field object>

Message registry version which is used in the middle portion of a messageID

sushy.resources.registry.message_registry.**parse_message**(*message_registries*, *message_field*)

Parse the messages in registries and substitute any params

Check only registries that support messages.

Parameters

- **message_registries** – dict of Message Registries
- **message_field** – settings.MessageListField to parse

Returns

parsed settings.MessageListField with missing attributes filled

sushy.resources.registry.message_registry_file module

class sushy.resources.registry.message_registry_file.**LocationListField**(**args*, ***kwargs*)

Bases: *ListField*

Location for each registry file of languages supported

There are 3 options where the file can be hosted:

- locally as a single file,
- locally as a part of archive (zip or other),
- publicly on the Internet.

archive_file = <sushy.resources.base.Field object>

File name for registry if using archive_uri

archive_uri = <sushy.resources.base.Field object>

Location URI for archive file

language = <sushy.resources.base.Field object>

File's RFC5646 language code or the string 'default'

publication_uri = <sushy.resources.base.Field object>

Location URI of publicly available schema

uri = <sushy.resources.base.Field object>

Location URI for co-located registry file with the Redfish service

class sushy.resources.registry.message_registry_file.**MessageRegistryFile**(*connector*,
path='',
redfish_versio
registries=Nor
reader=Nc
json_doc=
root=None

Bases: *ResourceBase*

description = <sushy.resources.base.Field object>

Description of Message Registry file resource

get_attribute_registry(*language*, *public_connector*)

Get an Attribute Registry from the location

Parameters

- **`language`** – RFC 5646 language code for registry files
- **`public_connector`** – connector to use when downloading registry from the Internet

Returns
an AttributeRegistry or None if not found

`get_message_registry`(*language*, *public_connector*)
Get a Message Registry from the location

Parameters

- **`language`** – RFC 5646 language code for registry files
- **`public_connector`** – connector to use when downloading registry from the Internet

Returns
a MessageRegistry or None if not found

`identity = <sushy.resources.base.Field object>`
Identity of Message Registry file resource

`languages = <sushy.resources.base.Field object>`
List of RFC 5646 language codes supported by this resource

`location = <sushy.resources.registry.message_registry_file.LocationListField object>`
List of locations of Registry files for each supported language

`name = <sushy.resources.base.Field object>`
Name of Message Registry file resource

`registry = <sushy.resources.base.Field object>`
Prefix for MessageId used for messages from this resource

This attribute is in form Registry_name.Major_version.Minor_version

class `sushy.resources.registry.message_registry_file.`**`MessageRegistryFileCollecti`**

Bases: *`ResourceCollectionBase`*

Collection of Message Registry Files

class `sushy.resources.registry.message_registry_file.`**`RegistryType`**(*connector*, *path=''*, *redfish_version=None*, *registries=None*, *reader=None*, *json_doc=None*, *root=None*)

Bases: `ResourceBase`

Module contents

sushy.resources.sessionservice package

Submodules

sushy.resources.sessionservice.session module

class `sushy.resources.sessionservice.session.`**`Session`**(*connector*, *identity*, *redfish_version=None*, *registries=None*, *root=None*)

Bases: `ResourceBase`

`delete`()

Method for deleting a Session.

Raises

ServerSideError

`description = <sushy.resources.base.Field object>`

The session service description

`identity = <sushy.resources.base.Field object>`

The session service identify string

`name = <sushy.resources.base.Field object>`

The session service name

`username = <sushy.resources.base.Field object>`

The UserName for the account for this session.

class `sushy.resources.sessionservice.session.`**`SessionCollection`**(*connector*, *identity*, *redfish_version=None*, *registries=None*, *root=None*)

Bases: `ResourceCollectionBase`

`description = <sushy.resources.base.Field object>`

The session collection description

`name = <sushy.resources.base.Field object>`

The session collection name

sushy.resources.sessionservice.sessionservice module

class sushy.resources.sessionservice.sessionservice.**SessionService**(*connector*, *identity*, *redfish_version=None*, *registries=None*, *root=None*)

Bases: *ResourceBase*

close_session(*session_uri*)

This function is for closing a session based on its id.

Raises

ServerSideError

create_session(*username*, *password*, *target_uri=None*)

This function will try to create a session.

Create a session and return the associated key and URI.

Parameters

- **username** – the username of the user requesting a new session
- **password** – the password associated to the user requesting a new session
- **target_uri** – the "Sessions" uri, usually in the form: '/redfish/v1/SessionService/Sessions'

Returns

A session key and uri in the form of a tuple

Raises

MissingXAuthToken

Raises

ConnectionError

Raises

AccessError

Raises

HTTPError

description = <sushy.resources.base.Field object>

The session service description

identity = <sushy.resources.base.Field object>

The session service identify string

name = <sushy.resources.base.Field object>

The session service name

service_enabled = <sushy.resources.base.Field object>

Tells us if session service is enabled

session_timeout = <sushy.resources.base.Field object>

The session service timeout

property sessions

Property to provide reference to the *SessionCollection* instance

It is calculated once when the first time it is queried. On refresh, this property gets reset.

Module contents

sushy.resources.system package

Subpackages

sushy.resources.system.network package

Submodules

sushy.resources.system.network.adapter module

class `sushy.resources.system.network.adapter.`**NetworkAdapter**(*connector*, *path=''*, *redfish_version=None*, *registries=None*, *reader=None*, *json_doc=None*, *root=None*)

Bases: `ResourceBase`

description = <sushy.resources.base.Field object>

Human-readable description of the resource

identity = <sushy.resources.base.Field object>

The network adapter identity string

manufacturer = <sushy.resources.base.Field object>

The manufacturer of this network adapter

model = <sushy.resources.base.Field object>

The model of this network adapter

name = <sushy.resources.base.Field object>

The name of the network adapter

property network_device_functions

Property to reference *NetworkDeviceFunctionCollection* instance

It is set once when the first time it is queried. On refresh, this property is marked as stale (greedy-refresh not done). Here the actual refresh of the sub-resource happens, if stale.

property network_ports

Property to reference *NetworkPortCollection* instance

It is set once when the first time it is queried. On refresh, this property is marked as stale (greedy-refresh not done). Here the actual refresh of the sub-resource happens, if stale.

part_number = <sushy.resources.base.Field object>

The part number of the network adapter

serial_number = <sushy.resources.base.Field object>

The serial number of the network adapter

status = <sushy.resources.common.StatusField object>

The status

class sushy.resources.system.network.adapter.**NetworkAdapterCollection**(*connector*, *path*, *redfish_version=N* *registries=None*, *root=None*)

Bases: `ResourceCollectionBase`

sushy.resources.system.network.constants module

class sushy.resources.system.network.constants.**FlowControl**(*value*)

Bases: `Enum`

An enumeration.

NONE = 'None'

No IEEE 802.3x flow control is enabled on this port.

RX = 'RX'

The link partner can initiate IEEE 802.3x flow control.

TX = 'TX'

This station can initiate IEEE 802.3x flow control.

TX_RX = 'TX_RX'

This station or the link partner can initiate IEEE 802.3x flow control.

class sushy.resources.system.network.constants.**IPAddressType**(*value*)

Bases: `Enum`

An enumeration.

IPV4 = 'IPv4'

IPv4 addressing is used for all IP-fields in this object.

IPV6 = 'IPv6'

IPv6 addressing is used for all IP-fields in this object.

class `sushy.resources.system.network.constants.`**`LinkStatus`**(*value*)

Bases: `Enum`

An enumeration.

`DOWN = 'Down'`

The port is enabled but link is down.

`STARTING = 'Starting'`

This link on this interface is starting. A physical link has been established, but the port is not able to transfer data.

`TRAINING = 'Training'`

This physical link on this interface is training.

`UP = 'Up'`

The port is enabled and link is good (up).

class `sushy.resources.system.network.constants.`**`NetworkAuthenticationMethod`**(*value*)

Bases: `Enum`

An enumeration.

`CHAP = 'CHAP'`

iSCSI Challenge Handshake Authentication Protocol (CHAP) authentication is used.

`MUTUAL_CHAP = 'MutualCHAP'`

iSCSI Mutual Challenge Handshake Authentication Protocol (CHAP) authentication is used.

`NONE = 'None'`

No iSCSI authentication is used.

class `sushy.resources.system.network.constants.`**`NetworkBootMode`**(*value*)

Bases: `Enum`

An enumeration.

`DISABLED = 'Disabled'`

Do not indicate to UEFI/BIOS that this device is bootable.

`FIBRE_CHANNEL = 'FibreChannel'`

Boot this device by using the embedded Fibre Channel support and configuration. Only applicable if the NetDevFuncType is *FibreChannel*.

`FIBRE_CHANNEL_OVER_ETHERNET = 'FibreChannelOverEthernet'`

Boot this device by using the embedded Fibre Channel over Ethernet (FCoE) boot support and configuration. Only applicable if the NetDevFuncType is *FibreChannelOverEthernet*.

`PXE = 'PXE'`

Boot this device by using the embedded PXE support. Only applicable if the NetDevFuncType is *Ethernet* or *InfiniBand*.

`SCSI = 'iSCSI'`

Boot this device by using the embedded iSCSI boot support and configuration. Only applicable if the NetDevFuncType is *iSCSI* or *Ethernet*.

class sushy.resources.system.network.constants.**NetworkDeviceTechnology**(*value*)

Bases: Enum

An enumeration.

DISABLED = 'Disabled'

Neither enumerated nor visible to the operating system.

ETHERNET = 'Ethernet'

Appears to the operating system as an Ethernet device.

FIBRE_CHANNEL = 'FibreChannel'

Appears to the operating system as a Fibre Channel device.

FIBRE_CHANNEL_OVER_ETHERNET = 'FibreChannelOverEthernet'

Appears to the operating system as an FCoE device.

INFINI_BAND = 'InfiniBand'

Appears to the operating system as an InfiniBand device.

iSCSI = 'iSCSI'

Appears to the operating system as an iSCSI device.

sushy.resources.system.network.device_function module

class sushy.resources.system.network.device_function.**BootTargetsField**(**args*, ***kwargs*)

Bases: *ListField*

lun_id = <sushy.resources.base.Field object>

The logical unit number (LUN) ID from which to boot on the device

priority = <sushy.resources.base.Field object>

The relative priority for this entry in the boot targets array.

wwpn = <sushy.resources.base.Field object>

The World Wide Port Name (WWPN) from which to boot.

class sushy.resources.system.network.device_function.**EthernetField**(**args*, ***kwargs*)

Bases: *CompositeField*

mac_address = <sushy.resources.base.Field object>

The currently configured MAC address of the resource

mtu_size = <sushy.resources.base.Field object>

The Maximum Transmission Unit (MTU) configured for this resource

permanent_mac_address = <sushy.resources.base.Field object>

The permanent MAC address assigned to this resource

vlan = <sushy.resources.system.network.device_function.VLANField object>

The VLAN for this interface

class sushy.resources.system.network.device_function.**FibreChannelField**(**args*, ***kwargs*)

Bases: *CompositeField*

boot_targets = <sushy.resources.system.network.device_function.BootTargetsField object>

An array of Fibre Channel boot targets configured for this resource.

class sushy.resources.system.network.device_function.**ISCSIBootField**(**args*, ***kwargs*)

Bases: *CompositeField*

authentication_method = <sushy.resources.base.MappedField object>

The configured capability of this network device function.

initiator_default_gateway = <sushy.resources.base.Field object>

The IPv6 or IPv4 iSCSI boot default gateway.

initiator_ip_address = <sushy.resources.base.Field object>

The IPv6 or IPv4 address of the iSCSI initiator.

initiator_netmask = <sushy.resources.base.Field object>

The IPv6 or IPv4 netmask of the iSCSI boot initiator.

ip_address_type = <sushy.resources.base.MappedField object>

The type of IP address being populated IP address fields.

primary_dns = <sushy.resources.base.Field object>

The IPv6 or IPv4 address of the primary DNS server.

primary_lun = <sushy.resources.base.Field object>

The logical unit number (LUN) for the primary iSCSI boot target.

primary_target_ip_address = <sushy.resources.base.Field object>

The IPv4 or IPv6 address for the primary iSCSI boot target.

primary_target_tcp_port = <sushy.resources.base.Field object>

The TCP port for the primary iSCSI boot target.

primary_vlan_enabled = <sushy.resources.base.Field object>

An indication of whether the primary VLAN is enabled.

primary_vlan_id = <sushy.resources.base.Field object>

The 802.1q VLAN ID to use for iSCSI boot from the primary target.

secondary_dns = <sushy.resources.base.Field object>

The IPv6 or IPv4 address of the secondary DNS server.

secondary_lun = <sushy.resources.base.Field object>

The logical unit number (LUN) for the secondary iSCSI boot target.

secondary_target_ip_address = <sushy.resources.base.Field object>

The IPv4 or IPv6 address for the secondary iSCSI boot target.

secondary_target_tcp_port = <sushy.resources.base.Field object>

The TCP port for the secondary iSCSI boot target.

secondary_vlan_enabled = <sushy.resources.base.Field object>

An indication of whether the secondary VLAN is enabled.

secondary_vlan_id = <sushy.resources.base.Field object>

The 802.1q VLAN ID to use for iSCSI boot from the secondary target.

class sushy.resources.system.network.device_function.**NetworkDeviceFunction**(*connect path='', redfish_ver registries= reader= json_do root=N*

Bases: *ResourceBase*

property assignable_physical_ports

An array of physical ports to which this resource may be assigned.

Network ports to which this network device function may be assigned.

Raises

MissingAttributeError if '@odata.id' field is missing.

Returns

A list of *NetworkPort* instances

capabilities = <sushy.resources.base.MappedListField object>

An array of capabilities for this network device function.

description = <sushy.resources.base.Field object>

The network device function description

ethernet = <sushy.resources.system.network.device_function.EthernetField object>

The Ethernet capabilities, status, and configuration values.

fibre_channel = <sushy.resources.system.network.device_function.FibreChannelField object>

The Fibre Channel capabilities, status, and configuration values.

identity = <sushy.resources.base.Field object>

Identifier for the network device function

iscsi_boot = <sushy.resources.system.network.device_function.ISCSIBootField object>

The iSCSI boot capabilities, status, and configuration for a network device function.

max_virtual_functions = <sushy.resources.base.Field object>

The number of virtual functions that are available for this network device function.

name = <sushy.resources.base.Field object>

The network device function name

status = <sushy.resources.common.StatusField object>

The status of the resource

type = <sushy.resources.base.MappedField object>

The configured capability of this network device function.

class sushy.resources.system.network.device_function.**NetworkDeviceFunctionCollec**

Bases: *ResourceCollectionBase*

class sushy.resources.system.network.device_function.**VLANField**(**args, **kwargs*)

Bases: *CompositeField*

vlan_enabled = <sushy.resources.base.Field object>

vlan_id = <sushy.resources.base.Field object>

sushy.resources.system.network.port module

class sushy.resources.system.network.port.**NetworkPort**(*connector, path='', redfish_version=None, registries=None, reader=None, json_doc=None, root=None*)

Bases: *ResourceBase*

associated_network_addresses = <sushy.resources.base.Field object>

The array of configured network addresses that are associated.

current_link_speed_mbps = <sushy.resources.base.Field object>

The network port current link speed.

description = <sushy.resources.base.Field object>

The network port description

flow_control_configuration = <sushy.resources.base.MappedField object>

The locally configured 802.3x flow control setting.

flow_control_status = <sushy.resources.base.MappedField object>

The 802.3x flow control behavior negotiated with the link partner

identity = <sushy.resources.base.Field object>

The network port identity

link_status = <sushy.resources.base.MappedField object>

The link status of the network port.

name = <sushy.resources.base.Field object>

The network port name

physical_port_number = <sushy.resources.base.Field object>

The physical port number label for this port.

status = <sushy.resources.common.StatusField object>

The network port status

class sushy.resources.system.network.port.**NetworkPortCollection**(*connector*, *path*, *redfish_version=None*, *registries=None*, *root=None*)

Bases: *ResourceCollectionBase*

Module contents

sushy.resources.system.storage package

Submodules

sushy.resources.system.storage.constants module

class sushy.resources.system.storage.constants.**RAIDType**(*value*)

Bases: Enum

An enumeration.

NONE = 'None'

A placement policy with no redundancy at the device level.

RAID0 = 'RAID0'

A placement policy where consecutive logical blocks of data are uniformly distributed across a set of independent storage devices without offering any form of redundancy.

RAID00 = 'RAID00'

A placement policy that creates a RAID 0 stripe set over two or more RAID 0 sets.

RAID01 = 'RAID01'

A data placement policy that creates a mirrored device (RAID 1) over a set of striped devices (RAID 0).

RAID1 = 'RAID1'

A placement policy where each logical block of data is stored on more than one independent storage device.

RAID10 = 'RAID10'

A placement policy that creates a striped device (RAID 0) over a set of mirrored devices (RAID 1).

RAID10E = 'RAID10E'

A placement policy that uses a RAID 0 stripe set over two or more RAID 10 sets.

RAID10_TRIPLE = 'RAID10Triple'

A placement policy that uses a striped device (RAID 0) over a set of triple mirrored devices (RAID 1Triple).

RAID1E = 'RAID1E'

A placement policy that uses a form of mirroring implemented over a set of independent storage devices where logical blocks are duplicated on a pair of independent storage devices so that data is uniformly distributed across the storage devices.

RAID1_TRIPLE = 'RAID1Triple'

A placement policy where each logical block of data is mirrored three times across a set of three independent storage devices.

RAID3 = 'RAID3'

A placement policy using parity-based protection where logical bytes of data are uniformly distributed across a set of independent storage devices and where the parity is stored on a dedicated independent storage device.

RAID4 = 'RAID4'

A placement policy using parity-based protection where logical blocks of data are uniformly distributed across a set of independent storage devices and where the parity is stored on a dedicated independent storage device.

RAID5 = 'RAID5'

A placement policy using parity-based protection for storing stripes of 'n' logical blocks of data and one logical block of parity across a set of 'n+1' independent storage devices where the parity and data blocks are interleaved across the storage devices.

RAID50 = 'RAID50'

A placement policy that uses a RAID 0 stripe set over two or more RAID 5 sets of independent storage devices.

RAID6 = 'RAID6'

A placement policy using parity-based protection for storing stripes of 'n' logical blocks of data and two logical blocks of independent parity across a set of 'n+2' independent storage devices where the parity and data blocks are interleaved across the storage devices.

RAID60 = 'RAID60'

A placement policy that uses a RAID 0 stripe set over two or more RAID 6 sets of independent storage devices.

RAID6TP = 'RAID6TP'

A placement policy that uses parity-based protection for storing stripes of 'n' logical blocks of data and three logical blocks of independent parity across a set of 'n+3' independent storage devices where the parity and data blocks are interleaved across the storage devices.

class `sushy.resources.system.storage.constants.`**VolumeInitializeType**(*value*)

Bases: `Enum`

An enumeration.

FAST = 'Fast'

The volume is prepared for use quickly, typically by erasing just the beginning and end of the space so that partitioning can be performed.

SLOW = 'Slow'

The volume is prepared for use slowly, typically by completely erasing the volume.

class `sushy.resources.system.storage.constants.`**VolumeType**(*value*)

Bases: `Enum`

An enumeration.

MIRRORED = 'Mirrored'

The volume is a mirrored device.

NON_REDUNDANT = 'NonRedundant'

The volume is a non-redundant storage device.

RAW_DEVICE = 'RawDevice'

The volume is a raw physical device without any RAID or other virtualization applied.

SPANNED_MIRRORS = 'SpannedMirrors'

The volume is a spanned set of mirrored devices.

SPANNED_STRIPES_WITH_PARITY = 'SpannedStripesWithParity'

The volume is a spanned set of devices which uses parity to retain redundant information.

STRIPED_WITH_PARITY = 'StripedWithParity'

The volume is a device which uses parity to retain redundant information.

sushy.resources.system.storage.controller module

class `sushy.resources.system.storage.controller.`**ControllerCollection**(*connector*, *path*, *redfish_version=No*, *registries=None*, *root=None*)

Bases: *ResourceCollectionBase*

property summary

Summary of storage controllers

Returns

dictionary of controller id-s and their status in format

```
{'RAID.Integrated.1-1': {'Health': sushy.Health.OK,
                         'State': sushy.State.ENABLED}}
```

class sushy.resources.system.storage.controller.**StorageController**(*connector*, *path=''*, *redfish_version=None*, *registries=None*, *reader=None*, *json_doc=None*, *root=None*)

Bases: *ResourceBase*

Storage controller

controller_protocols = <sushy.resources.base.MappedListField object>

The protocols by which this storage controller can be communicated to

device_protocols = <sushy.resources.base.MappedListField object>

The protocols that can be used to communicate with attached devices

identifiers = <sushy.resources.common.IdentifiersListField object>

The Durable names for the storage controller.

identity = <sushy.resources.base.Field object>

The storage controller identity

name = <sushy.resources.base.Field object>

The name of the storage controller

property pending_settings

Pending Storage Controller settings resource

raid_types = <sushy.resources.base.MappedListField object>

The set of RAID types supported by the storage controller.

speed_gbps = <sushy.resources.base.Field object>

The maximum speed of the storage controller's device interface.

status = <sushy.resources.common.StatusField object>

Describes the status and health of the resource and its children.

property supported_apply_times

List of supported BIOS update apply times

Returns

List of supported update apply time names

update (*payload*, *apply_time=None*, *maint_window_start_time=None*, *maint_window_duration=None*)

Updates writable properties

Supports updating properties that require reboot.

Parameters

- **payload** – dictionary with properties to update
- **apply_time** – When to update the attributes. Optional. A `sushy.ApplyTime` value.
- **maint_window_start_time** – The start time of a maintenance window, datetime. Required when updating during maintenance window and default maintenance window not set by the system.
- **maint_window_duration** – Duration of maintenance time since maintenance window start time in seconds. Required when updating during maintenance window and default maintenance window not set by the system.

Returns

TaskMonitor if async task or None

sushy.resources.system.storage.drive module

class sushy.resources.system.storage.drive.**Drive** (*connector*, *path=''*, *redfish_version=None*, *registries=None*, *reader=None*, *json_doc=None*, *root=None*)

Bases: *ResourceBase*

This class represents a disk drive or other physical storage medium.

block_size_bytes = <sushy.resources.base.Field object>

The size of the smallest addressable unit of this drive in bytes

capacity_bytes = <sushy.resources.base.Field object>

The size in bytes of this Drive

identifiers = <sushy.resources.common.IdentifiersListField object>

The Durable names for the drive

identity = <sushy.resources.base.Field object>

The Drive identity string

indicator_led = <sushy.resources.base.MappedField object>

Whether the indicator LED is lit or off

manufacturer = <sushy.resources.base.Field object>

This is the manufacturer of this drive

media_type = <sushy.resources.base.Field object>

The type of media contained in this drive

model = <sushy.resources.base.Field object>

This is the model number for the drive

name = <sushy.resources.base.Field object>

The name of the resource

part_number = <sushy.resources.base.Field object>

The part number for this drive

protocol = <sushy.resources.base.MappedField object>

Protocol this drive is using to communicate to the storage controller

revision = <sushy.resources.base.Field object>

The firmware/hardware version of the drive.

serial_number = <sushy.resources.base.Field object>

The serial number for this drive

set_indicator_led(*state*)

Set IndicatorLED to the given state.

Parameters

state – Desired LED state, an IndicatorLED value.

Raises

InvalidParameterValueError, if any information passed is invalid.

status = <sushy.resources.common.StatusField object>

This type describes the status and health of the drive

property volumes

A list of volumes that this drive is part of.

Volumes that this drive either wholly or only partially contains.

Raises

MissingAttributeError if '@odata.id' field is missing.

Returns

A list of *Volume* instances

sushy.resources.system.storage.storage module

class sushy.resources.system.storage.storage.**Storage**(*connector*, *path=''*, *redfish_version=None*, *registries=None*, *reader=None*, *json_doc=None*, *root=None*)

Bases: `ResourceBase`

This class represents the storage subsystem resources.

A storage subsystem represents a set of storage controllers (physical or virtual) and the resources such as drives and volumes that can be accessed from that subsystem.

property controllers

The storage controllers allocated to this storage subsystem.

Replaces *storage_controllers* since Redfish v1.9 to allow storage controllers be their own resource.

property drives

Return a list of *Drive* objects present in the storage resource.

It is set once when the first time it is queried. On subsequent invocations, it returns a cached list of *Drives* objects until it is marked stale.

Returns
A list of *Drive* objects

Raises
ResourceNotFoundError

drives_identities = <sushy.resources.base.Field object>

A tuple with the drive identities

property drives_max_size_bytes

Max size available in bytes among all Drives of this collection.

property drives_sizes_bytes

Sizes of all Drives in bytes in Storage resource.

Returns the list of cached values until it (or its parent resource) is refreshed.

get_drive(*drive_identity*)

Given the drive identity return a `Drive` object

Parameters
drive_identity – The identity of the `Drive`

Returns
The `Drive` object

Raises
ResourceNotFoundError

identity = <sushy.resources.base.Field object>

The Storage identity string

name = <sushy.resources.base.Field object>

The name of the resource

status = <sushy.resources.common.StatusField object>

Describes the status and health of the resource and its children.

storage_controllers = <sushy.resources.system.storage.storage.StorageControllersListField object>

The storage devices associated with this resource.

Deprecated since Redfish v1.13 to allow storage controllers be their own resource. Use *controllers* where available.

property volumes

Property to reference *VolumeCollection* instance

It is set once when the first time it is queried. On refresh, this property is marked as stale (greedy-refresh not done at that point). Here only the actual refresh of the sub-resource happens, if resource is stale.

class sushy.resources.system.storage.storage.**StorageCollection**(*connector*, *path*, *redfish_version=None*, *registries=None*, *root=None*)

Bases: *ResourceCollectionBase*

This class represents the collection of Storage resources

property drives_sizes_bytes

Sizes of each Drive in bytes in Storage collection resource.

Returns the list of cached values until it (or its parent resource) is refreshed.

property max_drive_size_bytes

Max size available (in bytes) among all Drive resources.

Returns the cached value until it (or its parent resource) is refreshed.

property max_volume_size_bytes

Max size available (in bytes) among all Volume resources.

Returns the cached value until it (or its parent resource) is refreshed.

property volumes_sizes_bytes

Sizes of each Volume in bytes in Storage collection resource.

Returns the list of cached values until it (or its parent resource) is refreshed.

class sushy.resources.system.storage.storage.**StorageControllersListField**(**args*, ***kwargs*)

Bases: *ListField*

The set of storage controllers represented by this resource.

controller_protocols = <sushy.resources.base.MappedListField object>

The protocols by which this storage controller can be communicated to

device_protocols = <sushy.resources.base.MappedListField object>

The protocols which the controller can use tocommunicate with devices

identifiers = <sushy.resources.common.IdentifiersListField object>

The Durable names for the storage controller.

member_id = <sushy.resources.base.Field object>

Uniquely identifies the member within the collection.

name = <sushy.resources.base.Field object>

The name of the storage controller

raid_types = <sushy.resources.base.MappedListField object>

The set of RAID types supported by the storage controller.

speed_gbps = <sushy.resources.base.Field object>

The maximum speed of the storage controller's device interface.

status = <sushy.resources.common.StatusField object>

Describes the status and health of the resource and its children.

sushy.resources.system.storage.volume module

class `sushy.resources.system.storage.volume.`**ActionsField**(**args*, ***kwargs*)

Bases: `CompositeField`

initialize = <sushy.resources.common.InitializeActionField object>

class `sushy.resources.system.storage.volume.`**Volume**(*connector*, *path=''*, *redfish_version=None*, *registries=None*, *reader=None*, *json_doc=None*, *root=None*)

Bases: `ResourceBase`

This class adds the Storage Volume resource

block_size_bytes = <sushy.resources.base.Field object>

The size of the smallest addressable unit of this volume in bytes.

capacity_bytes = <sushy.resources.base.Field object>

The size in bytes of this Volume.

delete(*payload=None*, *apply_time=None*, *timeout=500*)

Delete the volume.

Parameters

- **payload** – May contain @Redfish.OperationApplyTime property
- **apply_time** – When to update the attributes. Optional. An `sushy.ApplyTime` value.
- **timeout** – Max time in seconds to wait for blocking async call.

Raises

ConnectionError

Raises

HTTPError

Returns

TaskMonitor if async task or None if successful deletion

encrypted = <sushy.resources.base.Field object>

Is this Volume encrypted.

get_allowed_initialize_volume_values()

Get the allowed values for initializing the volume.

Returns

A set with the allowed values.

identifiers = <sushy.resources.common.IdentifiersListField object>

The Durable names for the volume.

identity = <sushy.resources.base.Field object>

The Volume identity string

initialize(*value=VolumeInitializeType.FAST*, *apply_time=None*, *timeout=500*)

Initialize the volume.

Parameters

- **value** – The InitializeType value.
- **apply_time** – When to update the attributes. Optional. An `sushy.ApplyTime` value.
- **timeout** – Max time in seconds to wait for blocking async call.

Raises

InvalidParameterValueError, if the target value is not allowed.

Raises

ConnectionError

Raises

HTTPError

Returns

TaskMonitor if async task or None if successful init

name = <sushy.resources.base.Field object>

The name of the resource

operation_apply_time_support = <sushy.resources.common.OperationApplyTimeSupportField object>

Indicates if a client is allowed to request for a specific apply time of a create, delete, or action operation of a given resource

raid_type = <sushy.resources.base.MappedField object>

The RAID type of this volume.

volume_type = <sushy.resources.base.MappedField object>

The type of this volume.

class sushy.resources.system.storage.volume.**VolumeCollection**(*connector*, *path*, *redfish_version=None*, *registries=None*, *root=None*)

Bases: *ResourceCollectionBase*

This class represents the Storage Volume collection

create(*payload*, *apply_time=None*, *timeout=500*)

Create a volume.

Parameters

- **payload** – The payload representing the new volume to create.
- **apply_time** – When to update the attributes. Optional. An sushy.ApplyTime value.
- **timeout** – Max time in seconds to wait for blocking async call.

Raises

ConnectionError

Raises

HTTPError

Returns

Newly created Volume resource or TaskMonitor if async task

property max_size_bytes

Max size available (in bytes) among all Volume resources.

Returns the cached value until it (or its parent resource) is refreshed.

property max_volume_size_bytes

Max size available (in bytes) among all Volume resources.

Returns the cached value until it (or its parent resource) is refreshed.

operation_apply_time_support = <sushy.resources.common.OperationApplyTimeSupportField object>

Indicates if a client is allowed to request for a specific apply time of a create, delete, or action operation of a given resource

property volumes_sizes_bytes

Sizes of all Volumes in bytes in VolumeCollection resource.

Returns the list of cached values until it (or its parent resource) is refreshed.

Module contents

Submodules

sushy.resources.system.bios module

class sushy.resources.system.bios.**ActionsField**(**args*, ***kwargs*)

Bases: `CompositeField`

change_password = <sushy.resources.common.ActionField object>

reset_bios = <sushy.resources.common.ActionField object>

class sushy.resources.system.bios.**Bios**(*connector*, *path*, *redfish_version=None*, *registries=None*, *root=None*)

Bases: `ResourceBase`

property apply_time_settings

attributes = <sushy.resources.base.Field object>

Vendor-specific key-value dict of effective BIOS attributes

Attributes cannot be updated directly. To update use `set_attribute()` or `set_attributes()`

change_password(*new_password*, *old_password*, *password_name*)

Change BIOS password

description = <sushy.resources.base.Field object>

Human-readable description of the BIOS resource

get_attribute_registry(*language='en'*)

Get the Attribute Registry associated with this BIOS instance

Parameters

language – RFC 5646 language code for Message Registries. Indicates language of registry to be used. Defaults to 'en'.

Returns

the BIOS Attribute Registry

identity = <sushy.resources.base.Field object>

The Bios resource identity string

maintenance_window = <sushy.resources.settings.MaintenanceWindowField object>

Indicates if a given resource has a maintenance window assignment for applying settings or operations

name = <sushy.resources.base.Field object>

The name of the resource

property pending_attributes

Pending BIOS attributes

BIOS attributes that have been committed to the system, but for them to take effect system restart is necessary

reset_bios()

Reset the BIOS attributes to default

set_attribute(*key*, *value*, *apply_time=None*, *maint_window_start_time=None*, *maint_window_duration=None*)

Update an attribute

Attribute update is not immediate but requires system restart. Committed attributes can be checked at `pending_attributes` property

Parameters

- **key** – Attribute name
- **value** – Attribute value
- **apply_time** – When to update the attribute. Optional. An `sushy.ApplyTime` value.
- **maint_window_start_time** – The start time of a maintenance window, datetime. Required when updating during maintenance window and default maintenance window not set by the system.
- **maint_window_duration** – Duration of maintenance time since maintenance window start time in seconds. Required when updating during maintenance window and default maintenance window not set by the system.

set_attributes(*value*, *apply_time=None*, *maint_window_start_time=None*, *maint_window_duration=None*)

Update many attributes at once

Attribute update is not immediate but requires system restart. Committed attributes can be checked at `pending_attributes` property

Parameters

- **value** – Key-value pairs for attribute name and value
- **apply_time** – When to update the attributes. Optional. An `sushy.ApplyTime` value.
- **maint_window_start_time** – The start time of a maintenance window, datetime. Required when updating during maintenance window and default maintenance window not set by the system.
- **maint_window_duration** – Duration of maintenance time since maintenance window start time in seconds. Required when updating during maintenance window and default maintenance window not set by the system.

property supported_apply_times

List of supported BIOS update apply times

Returns

List of supported update apply time names

property update_status

Status of the last attribute update

Returns

sushy.resources.settings.SettingsUpdate object containing status and any messages

sushy.resources.system.constants module

class sushy.resources.system.constants.**BootProgressStates**(*value*)

Bases: `Enum`

Boot System Progress Indicator constants

BUS = 'BusInitializationStarted'

Initialization of the buses has started.

HARDWARE_COMPLETE = 'SystemHardwareInitializationComplete'

Hardware Initialization is completed.

MEMORY = 'MemoryInitializationStarted'

Initialization of memory has started.

NONE = 'None'

The system is not booting.

OEM = 'OEM'

OEM Defined Boot Progress State.

OS_BOOT_STARTED = 'OSBootStarted'

Boot of the Operating System has started.

OS_RUNNING = 'OSRunning'

Operating System Running.

PCI_RESOURCE_CONFIG = 'PCIResourceConfigStarted'

Initalizatoin of PCI Resources has started.

PRIMARY_PROCESSOR = 'PrimaryProcessorInitializationStarted'

Initialization of the Primary Processor has started.

SECONDARY_PROCESSOR = 'SecondaryProcessorInitializationStarted'

Secondary Prcessors have started initialization.

SETUP = 'SetupEntered'

System is in the Setup utility.

class sushy.resources.system.constants.**BootSource**(*value*)

Bases: `Enum`

Boot source target constants

BIOS_SETUP = 'BiosSetup'

Boot to the BIOS setup utility.

CD = 'Cd'

Boot from the CD or DVD.

DIAGS = 'Diags'

Boot to the manufacturer's diagnostics program.

FLOPPY = 'Floppy'

Boot from the floppy disk drive.

HDD = 'Hdd'

Boot from a hard drive.

NONE = 'None'

Boot from the normal boot device.

PXE = 'Pxe'

Boot from the Pre-Boot EXecution (PXE) environment.

REMOTE_DRIVE = 'RemoteDrive'

Boot from a remote drive, such as an iSCSI target.

SD_CARD = 'SDCard'

Boot from an SD card.

UEFI_BOOT_NEXT = 'UefiBootNext'

Boot to the UEFI device that the BootNext property specifies.

UEFI_HTTP = 'UefiHttp'

Boot from a UEFI HTTP network location.

UEFI_SHELL = 'UefiShell'

Boot to the UEFI Shell.

UEFI_TARGET = 'UefiTarget'

Boot to the UEFI device specified in the UefiTargetBootSourceOverride property.

USB = 'Usb'

Boot from a system BIOS-specified USB device.

USB_CD = 'UsbCd'

Boot from a USB CD device as specified by the system BIOS.

This is NOT a standard value! On SuperMicro X11 and X12 machines, virtual media is presented as an USB CD drive as opposed to a CD drive. Both are present in the list of boot devices, however only selecting UsbCd as the boot source results in a successful boot from vMedia. If CD is selected, boot fails even if vMedia is inserted.

UTILITIES = 'Utilities'

Boot to the manufacturer's utilities program or programs.

class sushy.resources.system.constants.**BootSourceOverrideEnabled**(*value*)

Bases: Enum

Boot source enabled constants

CONTINUOUS = 'Continuous'

The system boots to the target specified in the BootSourceOverrideTarget property until this property is *Disabled*.

DISABLED = 'Disabled'

The system boots normally.

ONCE = 'Once'

On its next boot cycle, the system boots one time to the boot source override target. Then, the BootSourceOverrideEnabled value is reset to *Disabled*.

class sushy.resources.system.constants.**BootSourceOverrideMode**(*value*)

Bases: Enum

Boot source mode constants

LEGACY = 'Legacy'

The system boots in non-UEFI boot mode to the boot source override target.

UEFI = 'UEFI'

The system boots in UEFI boot mode to the boot source override target.

class sushy.resources.system.constants.**InstructionSet**(*value*)

Bases: Enum

Processor InstructionSet constants

ARM_A32 = 'ARM-A32'

ARM 32-bit.

ARM_A64 = 'ARM-A64'

ARM 64-bit.

IA_64 = 'IA-64'

Intel IA-64.

MIPS32 = 'MIPS32'

MIPS 32-bit.

MIPS64 = 'MIPS64'

MIPS 64-bit.

OEM = 'OEM'

OEM-defined.

POWER_ISA = 'PowerISA'

PowerISA-64 or PowerISA-32.

X86 = 'x86'

x86 32-bit.

X86_64 = 'x86-64'

x86 64-bit.

class `sushy.resources.system.constants.`**`ProcessorArchitecture`**(*value*)

Bases: `Enum`

Processor Architecture constants

`ARM = 'ARM'`

ARM.

`IA_64 = 'IA-64'`

Intel Itanium.

`MIPS = 'MIPS'`

MIPS.

`OEM = 'OEM'`

OEM-defined.

`POWER = 'Power'`

Power.

`X86 = 'x86'`

x86 or x86-64.

class `sushy.resources.system.constants.`**`ProcessorType`**(*value*)

Bases: `Enum`

Processor type constants

`ACCELERATOR = 'Accelerator'`

An accelerator.

`CORE = 'Core'`

A core in a processor.

`CPU = 'CPU'`

A CPU.

`DSP = 'DSP'`

A DSP.

`FPGA = 'FPGA'`

An FPGA.

`GPU = 'GPU'`

A GPU.

`OEM = 'OEM'`

An OEM-defined processing unit.

`THREAD = 'Thread'`

A thread in a processor.

`sushy.resources.system.constants.`**`SYSTEM_POWER_STATE_OFF = PowerState.OFF`**

The system is powered off, although some components may continue to have AUX power such as management controller sushy.resources.system.constants.**SYSTEM_POWER_STATE_ON = PowerState.ON**

The system is powered on

sushy.resources.system.constants.**SYSTEM_POWER_STATE_POWERING_OFF = PowerState.POWERING_OFF**

A temporary state between On and Off. The power off action can take time while the OS is in the shutdown process

sushy.resources.system.constants.**SYSTEM_POWER_STATE_POWERING_ON = PowerState.POWERING_ON**

A temporary state between Off and On. This temporary state can be very short

class sushy.resources.system.constants.**SecureBootCurrentBoot**(*value*)

Bases: Enum

An enumeration.

DISABLED = 'Disabled'

UEFI Secure Boot is currently disabled.

ENABLED = 'Enabled'

UEFI Secure Boot is currently enabled.

class sushy.resources.system.constants.**SecureBootDatabaseId**(*value*)

Bases: Enum

An enumeration.

ALLOWED_KEYS_DATABASE = 'db'

DEFAULT_ALLOWED_KEYS_DATABASE = 'dbDefault'

DEFAULT_DENIED_KEYS_DATABASE = 'dbxDefault'

DEFAULT_KEY_EXCHANGE_KEYS = 'KEKDefault'

DEFAULT_PLATFORM_KEY = 'PKDefault'

DEFAULT_RECOVERY_KEYS_DATABASE = 'dbrDefault'

DEFAULT_TIMESTAMP_DATABASE = 'dbtDefault'

DENIED_KEYS_DATABASE = 'dbx'

KEY_EXCHANGE_KEYS = 'KEK'

PLATFORM_KEY = 'PK'

RECOVERY_KEYS_DATABASE = 'dbr'

TIMESTAMP_DATABASE = 'dbt'

class sushy.resources.system.constants.**SecureBootMode**(*value*)

Bases: Enum

An enumeration.

AUDIT = 'AuditMode'

UEFI Secure Boot is currently in Audit Mode.

DEPLOYED = 'DeployedMode'

UEFI Secure Boot is currently in Deployed Mode.

SETUP = 'SetupMode'

UEFI Secure Boot is currently in Setup Mode.

USER = 'UserMode'

UEFI Secure Boot is currently in User Mode.

class `sushy.resources.system.constants.`**SecureBootResetKeysType**(*value*)

Bases: `Enum`

An enumeration.

DELETE_ALL_KEYS = 'DeleteAllKeys'

Delete the contents of all UEFI Secure Boot key databases, including the PK key database. This puts the system in Setup Mode.

DELETE_PK = 'DeletePK'

Delete the contents of the PK UEFI Secure Boot database. This puts the system in Setup Mode.

RESET_ALL_KEYS_TO_DEFAULT = 'ResetAllKeysToDefault'

Reset the contents of all UEFI Secure Boot key databases, including the PK key database, to the default values.

class `sushy.resources.system.constants.`**SystemType**(*value*)

Bases: `Enum`

System type constants

COMPOSED = 'Composed'

A computer system constructed by binding resource blocks together.

DPU = 'DPU'

A computer system that performs the functions of a data processing unit, such as a SmartNIC.

OS = 'OS'

An operating system instance.

PHYSICAL = 'Physical'

A computer system.

PHYSICALLY_PARTITIONED = 'PhysicallyPartitioned'

A hardware-based partition of a computer system.

VIRTUAL = 'Virtual'

A virtual machine instance running on this system.

VIRTUALLY_PARTITIONED = 'VirtuallyPartitioned'

A virtual or software-based partition of a computer system.

sushy.resources.system.ethernet_interface module

class `sushy.resources.system.ethernet_interface.`**`EthernetInterface`**(*connector*, *path=''*, *redfish_version=None*, *registries=None*, *reader=None*, *json_doc=None*, *root=None*)

Bases: `ResourceBase`

This class adds the EthernetInterface resource

`description = <sushy.resources.base.Field object>`

Description

`identity = <sushy.resources.base.Field object>`

The Ethernet Interface identity string

`mac_address = <sushy.resources.base.Field object>`

This is the currently configured MAC address of the interface.

`name = <sushy.resources.base.Field object>`

The name of the resource or array element

`permanent_mac_address = <sushy.resources.base.Field object>`

This is the permanent MAC address assigned to this interface (port)

`speed_mbps = <sushy.resources.base.Field object>`

This is the current speed in Mbps of this interface.

`status = <sushy.resources.common.StatusField object>`

Describes the status and health of this interface.

class `sushy.resources.system.ethernet_interface.`**`EthernetInterfaceCollection`**(*conne*
path,
redfish_v
registries
root=

Bases: `ResourceCollectionBase`

property summary

Summary of MAC addresses and interfaces state

This filters the MACs whose health is OK, which means the MACs in both ‘Enabled’ and ‘Disabled’ States are returned.

Returns

dictionary in the format {‘aa:bb:cc:dd:ee:ff’: sushy.State.ENABLED, ‘aa:bb:aa:aa:aa:aa’: sushy.State.DISABLED}

sushy.resources.system.processor module

class sushy.resources.system.processor.**Processor**(*connector*, *identity*, *redfish_version=None*, *registries=None*, *root=None*)

Bases: `ResourceBase`

identity = <sushy.resources.base.Field object>

The processor identity string

instruction_set = <sushy.resources.base.MappedField object>

The instruction set of the processor

manufacturer = <sushy.resources.base.Field object>

The processor manufacturer

max_speed_mhz = <sushy.resources.base.Field object>

The maximum clock speed of the processor in MHz.

model = <sushy.resources.base.Field object>

The product model number of this device

processor_architecture = <sushy.resources.base.MappedField object>

The architecture of the processor

processor_id = <sushy.resources.system.processor.ProcessorIdField object>

The processor id

processor_type = <sushy.resources.base.MappedField object>

The type of processor

socket = <sushy.resources.base.Field object>

The socket or location of the processor

status = <sushy.resources.common.StatusField object>

The processor status

property sub_processors

A reference to the collection of Sub-Processors

total_cores = <sushy.resources.base.Field object>

The total number of cores contained in this processor

total_threads = <sushy.resources.base.Field object>

The total number of execution threads supported by this processor

class sushy.resources.system.processor.**ProcessorCollection**(*connector*, *path*, *redfish_version=None*, *registries=None*, *root=None*)

Bases: `ResourceCollectionBase`

property summary

Property to provide ProcessorSummary info

It is calculated once when the first time it is queried. On refresh, this property gets reset.

Returns

A namedtuple containing the `count` of processors in regards to logical CPUs, and their `architecture`.

class `sushy.resources.system.processor.`**ProcessorIdField**(**args*, ***kwargs*)

Bases: *CompositeField*

effective_family = <sushy.resources.base.Field object>

The processor effective family

effective_model = <sushy.resources.base.Field object>

The processor effective model

identification_registers = <sushy.resources.base.Field object>

The processor identification registers

microcode_info = <sushy.resources.base.Field object>

The processor microcode info

step = <sushy.resources.base.Field object>

The processor stepping

vendor_id = <sushy.resources.base.Field object>

The processor vendor id

class `sushy.resources.system.processor.`**ProcessorSummary**(*count*, *architecture*)

Bases: `tuple`

architecture

Alias for field number 1

count

Alias for field number 0

sushy.resources.system.secure_boot module

class `sushy.resources.system.secure_boot.`**ActionsField**(**args*, ***kwargs*)

Bases: *CompositeField*

reset_keys = <sushy.resources.system.secure_boot.ResetKeysActionField object>

Action that resets the UEFI Secure Boot keys.

class `sushy.resources.system.secure_boot.`**ResetKeysActionField**(**args*, ***kwargs*)

Bases: *ActionField*

allowed_values = <sushy.resources.base.Field object>

class sushy.resources.system.secure_boot.**SecureBoot**(*connector*, *path*, *redfish_version=None*, *registries=None*, *root=None*)

Bases: `ResourceBase`

current_boot = <sushy.resources.base.MappedField object>

The UEFI Secure Boot state during the current boot cycle.

property databases

A collection of secure boot databases.

It is set once when the first time it is queried. On refresh, this property is marked as stale (greedy-refresh not done). Here the actual refresh of the sub-resource happens, if stale.

Raises

MissingAttributeError if 'SecureBootDatabases/@odata.id' field is missing.

Returns

SimpleStorageCollection instance

description = <sushy.resources.base.Field object>

Human-readable description of the BIOS resource

enabled = <sushy.resources.base.Field object>

Whether the UEFI Secure Boot takes effect on next boot.

This property can be enabled in UEFI boot mode only.

get_allowed_reset_keys_values()

Get the allowed values for resetting the keys.

Returns

A set with the allowed values.

identity = <sushy.resources.base.Field object>

The Bios resource identity string

mode = <sushy.resources.base.MappedField object>

The current UEFI Secure Boot Mode.

name = <sushy.resources.base.Field object>

The name of the resource

reset_keys(*reset_type*)

Reset secure boot keys.

Parameters

reset_type – Reset type, one of *SECURE_BOOT_RESET_KEYS_** constants.

set_enabled(*enabled*)

Enable/disable secure boot.

Parameters

enabled – True, if secure boot is enabled for next boot.

sushy.resources.system.secure_boot_database module

class sushy.resources.system.secure_boot_database.**ActionsField**(**args*, ***kwargs*)

Bases: `CompositeField`

reset_keys = <sushy.resources.system.secure_boot_database.ResetKeysActionField object>

Action that resets the UEFI Secure Boot keys.

class sushy.resources.system.secure_boot_database.**ResetKeysActionField**(**args*, ***kwargs*)

Bases: `ActionField`

allowed_values = <sushy.resources.base.Field object>

class sushy.resources.system.secure_boot_database.**SecureBootDatabase**(*connector*, *path=''*, *redfish_version=No*
registries=None, *reader=None*, *json_doc=None*, *root=None*)

Bases: `ResourceBase`

database_id = <sushy.resources.base.MappedField object>

Standard UEFI database type.

description = <sushy.resources.base.Field object>

The system description

get_allowed_reset_keys_values()

Get the allowed values for resetting the keys.

Returns

A set with the allowed values.

identity = <sushy.resources.base.Field object>

The secure boot database identity string

name = <sushy.resources.base.Field object>

The secure boot database name

reset_keys(*reset_type*)

Reset secure boot keys.

Parameters

reset_type – Reset type, one of *SECURE_BOOT_RESET_KEYS_** constants.

class sushy.resources.system.secure_boot_database.**SecureBootDatabaseCollection**(*c*
p
re
fi
re
is
ro

Bases: *ResourceCollectionBase*

sushy.resources.system.simple_storage module

class sushy.resources.system.simple_storage.**DeviceListField**(**args, **kwargs*)

Bases: *ListField*

The storage device/s associated with SimpleStorage.

capacity_bytes = <sushy.resources.base.Field object>

The size of the storage device.

name = <sushy.resources.base.Field object>

The name of the storage device

status = <sushy.resources.common.StatusField object>

Describes the status and health of a storage device.

class sushy.resources.system.simple_storage.**SimpleStorage**(*connector, path='', redfish_version=None, registries=None, reader=None, json_doc=None, root=None*)

Bases: *ResourceBase*

This class represents a simple storage.

It represents the properties of a storage controller and its directly-attached devices. A storage device can be a disk drive or optical media device.

devices = <sushy.resources.system.simple_storage.DeviceListField object>

The storage devices associated with this resource.

identity = <sushy.resources.base.Field object>

The SimpleStorage identity string

name = <sushy.resources.base.Field object>

The name of the resource

class sushy.resources.system.simple_storage.**SimpleStorageCollection**(*connector*, *path*, *redfish_version=None*, *registries=None*, *root=None*)

Bases: *ResourceCollectionBase*

Represents a collection of simple storage associated with system.

property disks_sizes_bytes

Sizes of each Disk in bytes in SimpleStorage collection resource.

Returns the list of cached values until it (or its parent resource) is refreshed.

property max_size_bytes

Max size available (in bytes) among all enabled Disk resources.

Returns the cached value until it (or its parent resource) is refreshed.

sushy.resources.system.system module

class sushy.resources.system.system.**ActionsField**(**args*, ***kwargs*)

Bases: *CompositeField*

reset = <sushy.resources.common.ResetActionField object>

class sushy.resources.system.system.**BootField**(**args*, ***kwargs*)

Bases: *CompositeField*

allowed_values = <sushy.resources.base.Field object>

enabled = <sushy.resources.base.MappedField object>

http_boot_uri = <sushy.resources.base.Field object>

mode = <sushy.resources.base.MappedField object>

target = <sushy.resources.base.MappedField object>

class sushy.resources.system.system.**BootProgressField**(**args*, ***kwargs*)

Bases: *CompositeField*

last_boot_seconds_count = <sushy.resources.base.Field object>

The number of seconds the last boot took to reach OSRunning.

last_state = <sushy.resources.base.MappedField object>

The last recorded boot progress states.

last_state_updated_at = <sushy.resources.base.Field object>

The date-time value when the last state field was updated.

oem_last_state = <sushy.resources.base.Field object>

The OEM last state time to describe OEM specific state information.

class sushy.resources.system.system.**MemorySummaryField**(**args*, ***kwargs*)

Bases: `CompositeField`

health = <sushy.resources.base.Field object>

The overall health state of memory.

This signifies health state of memory along with its dependent resources.

size_gib = <sushy.resources.base.Field object>

The size of memory of the system in GiB.

This signifies the total installed, operating system-accessible memory (RAM), measured in GiB.

class sushy.resources.system.system.**System**(*connector*, *identity*, *redfish_version=None*, *registries=None*, *root=None*)

Bases: `ResourceBase`

asset_tag = <sushy.resources.base.Field object>

The system asset tag

property bios

Property to reference *Bios* instance

It is set once when the first time it is queried. On refresh, this property is marked as stale (greedy-refresh not done). Here the actual refresh of the sub-resource happens, if stale.

bios_version = <sushy.resources.base.Field object>

The system BIOS version

boot = <sushy.resources.system.system.BootField object>

A dictionary containing the current boot device, frequency and mode

boot_progress = <sushy.resources.system.system.BootProgressField object>

The last updated boot progress indicator

property chassis

A list of chassis where this system resides.

Returns a list of *Chassis* objects representing the chassis or cabinets where this system is mounted.

Raises
MissingAttributeError if '@odata.id' field is missing.

Returns
A list of *Chassis* instances

description = <sushy.resources.base.Field object>

The system description

property ethernet_interfaces

Property to reference *EthernetInterfaceCollection* instance

It is set once when the first time it is queried. On refresh, this property is marked as stale (greedy-refresh not done). Here the actual refresh of the sub-resource happens, if stale.

get_allowed_reset_system_values()

Get the allowed values for resetting the system.

Returns
A set with the allowed values.

get_allowed_system_boot_source_values()

Get the allowed values for changing the boot source.

Returns
A set with the allowed values.

hostname = <sushy.resources.base.Field object>

The system hostname

identity = <sushy.resources.base.Field object>

The system identity string

indicator_led = <sushy.resources.base.MappedField object>

Whether the indicator LED is lit or off

maintenance_window = <sushy.resources.settings.MaintenanceWindowField object>

Indicates if a given resource has a maintenance window assignment for applying settings or operations

property managers

A list of managers for this system.

Returns a list of *Manager* objects representing the managers that manage this system.

Raises
MissingAttributeError if '@odata.id' field is missing.

Returns
A list of *Manager* instances

manufacturer = <sushy.resources.base.Field object>

The system manufacturer

memory_summary = <sushy.resources.system.system.MemorySummaryField object>

The summary info of memory of the system in general detail

name = <sushy.resources.base.Field object>

The system name

part_number = <sushy.resources.base.Field object>

The system part number

power_state = <sushy.resources.base.MappedField object>

The system power state

property processors

Property to reference *ProcessorCollection* instance

It is set once when the first time it is queried. On refresh, this property is marked as stale (greedy-refresh not done). Here the actual refresh of the sub-resource happens, if stale.

reset_system(*value*)

Reset the system.

Parameters

value – The target value.

Raises

InvalidParameterValueError, if the target value is not allowed.

property secure_boot

Property to reference *SecureBoot* instance

It is set once when the first time it is queried. On refresh, this property is marked as stale (greedy-refresh not done). Here the actual refresh of the sub-resource happens, if stale.

serial_number = <sushy.resources.base.Field object>

The system serial number

set_indicator_led(*state*)

Set IndicatorLED to the given state.

Parameters

state – Desired LED state, an IndicatorLED value.

Raises

InvalidParameterValueError, if any information passed is invalid.

set_system_boot_options(*target=None*, *enabled=None*, *mode=None*, *http_boot_uri=None*)

Set boot source and/or boot frequency and/or boot mode.

Set the boot source and/or boot frequency and/or boot mode to use on next reboot of the System.

Parameters

- **target** – The target boot source, a `sushy.BootSource` value. Optional.
- **enabled** – How long the override is enabled, a `sushy.BootSourceOverrideEnabled` value. Optional.
- **mode** – The boot mode, a `sushy.BootSourceOverrideMode` value. Optional.
- **http_boot_uri** – The requested HTTP Boot URI to transmit to the BMC. Only valid when BootSourceOverrideTarget is set to UefiHTTP, when utilizing the `target` parameter. If no value is supplied, and the target is set to UefiHTTP, then an empty value will be sent to the BMC to remove any prior setting, allowing the host to load configuration from DHCP. If not explicitly set, any value will be removed from a BMC when UefiHttp boot is not engaged.

Raises

InvalidParameterValueError, if any information passed is invalid.

set_system_boot_source(*target*, *enabled=BootSourceOverrideEnabled.ONCE*, *mode=None*)

Set boot source and/or boot frequency and/or boot mode.

Set the boot source and/or boot frequency and/or boot mode to use on next reboot of the System.

This method is obsoleted by *set_system_boot_options*.

Parameters

- **target** – The target boot source, a `sushy.BootSource` value.
- **enabled** – The frequency, whether to set it for the next a `sushy.BootSourceOverrideEnabled` value. Default is *ONCE*.
- **mode** – The boot mode, a `sushy.BootSourceOverrideMode` value.

Raises

InvalidParameterValueError, if any information passed is invalid.

property simple_storage

A collection of simple storage associated with system.

This returns a reference to *SimpleStorageCollection* instance. SimpleStorage represents the properties of a storage controller and its directly-attached devices.

It is set once when the first time it is queried. On refresh, this property is marked as stale (greedy-refresh not done). Here the actual refresh of the sub-resource happens, if stale.

Raises

MissingAttributeError if 'SimpleStorage/@odata.id' field is missing.

Returns

SimpleStorageCollection instance

sku = <sushy.resources.base.Field object>

The system stock-keeping unit

status = <sushy.resources.common.StatusField object>

The system status

property storage

A collection of storage subsystems associated with system.

This returns a reference to *StorageCollection* instance. A storage subsystem represents a set of storage controllers (physical or virtual) and the resources such as drives and volumes that can be accessed from that subsystem.

It is set once when the first time it is queried. On refresh, this property is marked as stale (greedy-refresh not done). Here the actual refresh of the sub-resource happens, if stale.

Raises

MissingAttributeError if 'Storage/@odata.id' field is missing.

Returns

StorageCollection instance

system_type = <sushy.resources.base.MappedField object>

The system type

uuid = <sushy.resources.base.Field object>

The system UUID

property virtual_media

Property to reference *VirtualMedia* instance

Returns

A *VirtualMediaCollection* instance.

class `sushy.resources.system.system.`**SystemCollection**(*connector*, *path*, *redfish_version=None*, *registries=None*, *root=None*)

Bases: `ResourceCollectionBase`

Module contents

sushy.resources.taskservice package

Submodules

sushy.resources.taskservice.constants module

class `sushy.resources.taskservice.constants.`**OverWritePolicy**(*value*)

Bases: `Enum`

Overwrite Policy constants

MANUAL = 'Manual'

Completed tasks are not automatically overwritten.

OLDEST = 'Oldest'

Oldest completed tasks are overwritten.

class `sushy.resources.taskservice.constants.`**TaskState**(*value*)

Bases: `Enum`

Task state related constants.

CANCELLED = 'Cancelled'

CANCELLING = 'Cancelling'

COMPLETED = 'Completed'

EXCEPTION = 'Exception'

INTERRUPTED = 'Interrupted'

KILLED = 'Killed'

NEW = 'New'

PENDING = 'Pending'

RUNNING = 'Running'

SERVICE = 'Service'

STARTING = 'Starting'

STOPPING = 'Stopping'

SUSPENDED = 'Suspended'

sushy.resources.taskservice.task module

class sushy.resources.taskservice.task.**Task**(*connector*, *identity*, *redfish_version=None*, *registries=None*, *json_doc=None*, *root=None*)

Bases: *ResourceBase*

description = <sushy.resources.base.Field object>

The Task description

end_time = <sushy.resources.base.Field object>

End time of the Task

identity = <sushy.resources.base.Field object>

The Task identity

property is_processing

Indicates if the Task is processing

messages = <sushy.resources.base.MessageListField object>

List of *MessageListField* with messages from the Task

name = <sushy.resources.base.Field object>

The Task name

parse_messages()

Parses the messages

percent_complete = <sushy.resources.base.Field object>

Percentage complete of the Task

start_time = <sushy.resources.base.Field object>

Start time of the Task

task_monitor = <sushy.resources.base.Field object>

An opaque URL that the client can use to monitor an asynchronous operation

task_state = <sushy.resources.base.MappedField object>

The Task state

task_status = <sushy.resources.base.MappedField object>

The Task status

class sushy.resources.taskservice.task.**TaskCollection**(*connector*, *path*, *redfish_version=None*, *registries=None*, *root=None*)

Bases: *ResourceCollectionBase*

property summary

Summary of task ids and corresponding state

Returns

dictionary in the format {'jid_123456789': sushy.TaskState.NEW, 'jid_123454321': sushy.TaskState.RUNNING}

sushy.resources.taskservice.taskservice module

class sushy.resources.taskservice.taskservice.**TaskService**(*connector*, *identity*, *redfish_version=None*, *registries=None*, *root=None*)

Bases: *ResourceBase*

event_on_task_state_change = <sushy.resources.base.Field object>

Whether a task state change sends an event

identity = <sushy.resources.base.Field object>

The task service identity

name = <sushy.resources.base.Field object>

The task service name

overwrite_policy = <sushy.resources.base.MappedField object>

The overwrite policy for completed tasks

service_enabled = <sushy.resources.base.Field object>

The status of whether this service is enabled

status = <sushy.resources.common.StatusField object>

The status of the task service

property tasks

Property to reference *TaskCollection* instance

It is set once when the first time it is queried. On refresh, this property is marked as stale (greedy-refresh not done). Here the actual refresh of the sub-resource happens, if stale.

Module contents

sushy.resources.updateservice package

Submodules

sushy.resources.updateservice.constants module

class sushy.resources.updateservice.constants.**UpdateTransferProtocolType**(*value*)

Bases: Enum

Transfer Protocol Type constants

CIFS = 'CIFS'

Common Internet File System (CIFS).

FTP = 'FTP'

File Transfer Protocol (FTP).

HTTP = 'HTTP'

Hypertext Transfer Protocol (HTTP).

HTTPS = 'HTTPS'

Hypertext Transfer Protocol Secure (HTTPS).

NFS = 'NFS'

Network File System (NFS).

NSF = 'NFS'

Network File System (NFS).

OEM = 'OEM'

A manufacturer-defined protocol.

SCP = 'SCP'

Secure Copy Protocol (SCP).

SFTP = 'SFTP'

Secure File Transfer Protocol (SFTP).

TFTP = 'TFTP'

Trivial File Transfer Protocol (TFTP).

sushy.resources.updateservice.softwareinventory module

class sushy.resources.updateservice.softwareinventory.**SoftwareInventory**(*connector, identity, redfish_version registries=None root=None*)

Bases: *ResourceBase*

identity = <sushy.resources.base.Field object>

The software inventory identity

lowest_supported_version = <sushy.resources.base.Field object>

The lowest supported version of the software

manufacturer = <sushy.resources.base.Field object>

The manufacturer of the software

name = <sushy.resources.base.Field object>

The software inventory name

related_item = <sushy.resources.base.Field object>

The ID(s) of the resources associated with the software inventory item

release_date = <sushy.resources.base.Field object>

Release date of the software

software_id = <sushy.resources.base.Field object>

The identity of the software

status = <sushy.resources.common.StatusField object>

The status of the software inventory

uefi_device_paths = <sushy.resources.base.Field object>

Represents the UEFI Device Path(s)

updateable = <sushy.resources.base.Field object>

Indicates whether this software can be updated by the update service

version = <sushy.resources.base.Field object>

The version of the software

class sushy.resources.updateservice.softwareinventory.**SoftwareInventoryCollectio**

Bases: *ResourceCollectionBase*

description = <sushy.resources.base.Field object>

The software inventory collection description

name = <sushy.resources.base.Field object>

The software inventory collection name

sushy.resources.updateservice.updateservice module

class sushy.resources.updateservice.updateservice.**ActionsField**(**args*, ***kwargs*)

Bases: `CompositeField`

simple_update = <sushy.resources.common.ActionField object>

class sushy.resources.updateservice.updateservice.**UpdateService**(*connector*, *identity*, *redfish_version=None*, *registries=None*, *root=None*)

Bases: `ResourceBase`

property firmware_inventory

Property to reference FirmwareInventory collection instance

get_allowed_transfer_protocols()

Get the allowed values for transfer protocol.

Returns

A set of allowed values.

Raises

MissingAttributeError, if Actions/#UpdateService.SimpleUpdate attribute not present.

get_task_monitor(*task_monitor*)

Used to retrieve a TaskMonitor.

Deprecated: Use sushy.Sushy.get_task_monitor :returns: A task monitor.

http_push_uri = <sushy.resources.base.Field object>

The URI used to perform an HTTP or HTTPS push update to the Update Service

http_push_uri_targets = <sushy.resources.base.Field object>

http_push_uri_targets_busy = <sushy.resources.base.Field object>

identity = <sushy.resources.base.Field object>

The update service identity

name = <sushy.resources.base.Field object>

The update service name

service_enabled = <sushy.resources.base.Field object>

The status of whether this service is enabled

simple_update(*image_uri*, *targets=None*, *transfer_protocol=UpdateTransferProtocolType.HTTP*)

Simple Update is used to update software components.

Returns
A task monitor.

property software_inventory
Property to reference SoftwareInventory collection instance

status = <sushy.resources.common.StatusField object>
The status of the update service

Module contents

Submodules

sushy.resources.base module

class `sushy.resources.base.`**AbstractDataReader**
Bases: `object`

abstract get_data()
Based on data source get data and parse to JSON

set_connection(*connector*, *path*)
Sets mandatory connection parameters

Parameters

- **connector** – A Connector instance
- **path** – path of the resource

class `sushy.resources.base.`**CompositeField**(**args*, ***kwargs*)
Bases: `Mapping`, *Field*

Base class for fields consisting of several sub-fields.

class `sushy.resources.base.`**DictionaryField**(**args*, ***kwargs*)
Bases: *Field*

Base class for fields consisting of dictionary of several sub-fields.

class `sushy.resources.base.`**Field**(*path*, *required=False*, *default=None*, *adapter=<function Field.<lambda>>*)
Bases: `object`

Definition for fields fetched from JSON.

class `sushy.resources.base.`**FieldData**(*status_code*, *headers*, *json_doc*)
Bases: `object`

Contains data to be used when constructing Fields

property headers
The headers

property json_doc

The parsed JSON body

property status_code

The status code

class sushy.resources.base.**JsonArchiveReader**(*archive_file*)

Bases: *AbstractDataReader*

Gets the data from JSON file in archive

get_data()

Gets JSON file from archive. Currently supporting ZIP only

class sushy.resources.base.**JsonDataReader**

Bases: *AbstractDataReader*

Gets the data from HTTP response given by path

get_data()

Gets JSON file from URI directly

class sushy.resources.base.**JsonPackagedFileReader**(*resource_package_name*)

Bases: *AbstractDataReader*

Gets the data from packaged file given by path

get_data()

Gets JSON file from packaged file denoted by path

class sushy.resources.base.**JsonPublicFileReader**

Bases: *AbstractDataReader*

Loads the data from the Internet

get_data()

Get JSON file from full URI

class sushy.resources.base.**LinksField**(**args*, ***kwargs*)

Bases: *CompositeField*

Reference to linked resources.

oem_vendors = <sushy.resources.base.Field object>

class sushy.resources.base.**ListField**(**args*, ***kwargs*)

Bases: *Field*

Base class for fields consisting of a list of several sub-fields.

class sushy.resources.base.**MappedField**(*field*, *mapping*, *required=False*, *default=None*)

Bases: *Field*

Field taking real value from a mapping.

class sushy.resources.base.**MappedListField**(*field*, *mapping*, *required=False*, *default=None*)

Bases: *Field*

Field taking a list of values with a mapping for the values

Given JSON {'field':['xxx', 'yyy']}, a sushy resource definition and mapping {'xxx':'a', 'yyy':'b'}, the sushy object to come out will be like resource.field = ['a', 'b']

class sushy.resources.base.**MessageListField**(**args*, ***kwargs*)

Bases: *ListField*

List of messages with details of settings update status

message = <sushy.resources.base.Field object>

Human readable message, if provided

message_args = <sushy.resources.base.Field object>

List of message substitution arguments for the message referenced by *message_id* from the message registry

message_id = <sushy.resources.base.Field object>

The key for this message which can be used to look up the message in a message registry

resolution = <sushy.resources.base.Field object>

Used to provide suggestions on how to resolve the situation that caused the error

severity = <sushy.resources.base.MappedField object>

Severity of the error

class sushy.resources.base.**MutableResourceCollectionBase**(*connector*, *path*, *redfish_version=None*, *registries=None*, *root=None*)

Bases: *ResourceCollectionBase*

delete_member(*identity*)

Delete the given member of the collection.

class sushy.resources.base.**ResourceBase**(*connector*, *path=''*, *redfish_version=None*, *registries=None*, *reader=None*, *json_doc=None*, *root=None*)

Bases: object

clone_resource(*new_resource*, *path=''*)

Instantiate given resource using existing BMC connection context

get_oem_extension(*vendor*)

Get the OEM extension instance for this resource by OEM vendor

Parameters

vendor – the OEM vendor string which is the vendor-specific extensibility identifier. Examples are 'Contoso', 'Hpe'. Possible value can be got from oem_vendors attribute.

Returns
the Redfish resource OEM extension instance.

Raises
OEMExtensionNotFoundError

`invalidate` (*force_refresh=False*)

Mark the resource as stale, prompting refresh() before getting used.

If `force_refresh` is set to True, then it invokes `refresh()` on the resource.

Parameters
`force_refresh` – will invoke refresh on the resource, if set to True.

Raises
ResourceNotFoundError

Raises
ConnectionError

Raises
HTTPError

`property json`

`links = <sushy.resources.base.LinksField object>`

`property oem_vendors`

`property path`

`redfish_version = None`

The Redfish version

`refresh` (*force=True*, *json_doc=None*)

Refresh the resource

Freshly retrieves/fetches the resource attributes and invokes `_parse_attributes()` method on successful retrieval. It is recommended not to override this method in concrete ResourceBase classes. Resource classes can place their refresh specific operations in `_do_refresh()` method, if needed. This method represents the template method in the paradigm of Template design pattern.

Parameters

- **`force`** – if set to False, will only refresh if the resource is marked as stale, otherwise neither it nor its subresources will be refreshed.
- **`json_doc`** – parsed JSON document in form of Python types.

Raises
ResourceNotFoundError

Raises
ConnectionError

Raises
HTTPError

property registries

property resource_name

property root

class `sushy.resources.base.`**ResourceCollectionBase**(*connector*, *path*, *redfish_version=None*, *registries=None*, *root=None*)

Bases: `ResourceLinksBase`

members_identities = <sushy.resources.base.Field object>

A tuple with the members identities

name = <sushy.resources.base.Field object>

The name of the collection

class `sushy.resources.base.`**ResourceLinksBase**(*connector*, *path*, *redfish_version=None*, *registries=None*, *root=None*)

Bases: `ResourceBase`

get_member(*identity*)

Given the identity return a `_resource_type` object

Parameters

identity – The identity of the `_resource_type`

Returns

The `_resource_type` object

Raises

ResourceNotFoundError

get_members()

Return a list of `_resource_type` objects present in collection

Returns

A list of `_resource_type` objects

abstract property members_identities

A sequence with members identities

`sushy.resources.base.`**get_reader**(*connector*, *path*, *reader=None*)

Create and configure the reader.

Parameters

- **connector** – A Connector instance
- **path** – sub-URI path to the resource.
- **reader** – Reader to use to fetch JSON data.

Returns

the reader

sushy.resources.common module

class sushy.resources.common.**ActionField**(**args*, ***kwargs*)

Bases: *CompositeField*

operation_apply_time_support = <sushy.resources.common.OperationApplyTimeSupportField object>

target_uri = <sushy.resources.base.Field object>

class sushy.resources.common.**IdRefField**(**args*, ***kwargs*)

Bases: *CompositeField*

Reference to the resource odata identity field.

resource_uri = <sushy.resources.base.Field object>

The unique identifier for a resource

class sushy.resources.common.**IdentifiersListField**(**args*, ***kwargs*)

Bases: *ListField*

This type describes any additional identifiers for a resource.

durable_name = <sushy.resources.base.Field object>

This indicates the world wide, persistent name of the resource.

durable_name_format = <sushy.resources.base.MappedField object>

This represents the format of the DurableName property.

class sushy.resources.common.**InitializeActionField**(**args*, ***kwargs*)

Bases: *ActionField*

allowed_values = <sushy.resources.base.Field object>

class sushy.resources.common.**OperationApplyTimeSupportField**

Bases: *CompositeField*

maintenance_window_duration_in_seconds = <sushy.resources.base.Field object>

The expiry time of maintenance window in seconds

maintenance_window_start_time = <sushy.resources.base.Field object>

The start time of a maintenance window

mapped_supported_values = <sushy.resources.base.MappedListField object>

The types of apply times that the client is allowed request when performing a create, delete, or action operation returned as a mapped list

supported_values = <sushy.resources.base.Field object>

The types of apply times that the client is allowed request when performing a create, delete, or action operation returned as an unmapped list

Deprecated: Use *mapped_supported_values*.

class sushy.resources.common.**ResetActionField**(**args*, ***kwargs*)

Bases: *ActionField*

allowed_values = <sushy.resources.base.Field object>

class sushy.resources.common.**StatusField**(**args*, ***kwargs*)

Bases: *CompositeField*

This Field describes the status of a resource and its children.

This field shall contain any state or health properties of a resource.

health = <sushy.resources.base.MappedField object>

Represents health of resource w/o considering its dependent resources

health_rollup = <sushy.resources.base.MappedField object>

Represents health state of resource and its dependent resources

state = <sushy.resources.base.MappedField object>

Indicates the known state of the resource, such as if it is enabled.

sushy.resources.constants module

class sushy.resources.constants.**ApplyTime**(*value*)

Bases: Enum

Apply time constants

AT_MAINTENANCE_WINDOW_START = 'AtMaintenanceWindowStart'

Apply during a maintenance window as specified by an administrator.

IMMEDIATE = 'Immediate'

Apply immediately.

IN_MAINTENANCE_WINDOW_ON_RESET = 'InMaintenanceWindowOnReset'

Apply after a reset but within maintenance window as specified by an administrator.

ON_RESET = 'OnReset'

Apply on a reset.

class sushy.resources.constants.**DurableNameFormat**(*value*)

Bases: Enum

Durable name format constants

EUI = 'EUI'

The IEEE-defined 64-bit Extended Unique Identifier (EUI).

FC_WWN = 'FC_WWN'

The Fibre Channel (FC) World Wide Name (WWN).

NAA = 'NAA'

The Name Address Authority (NAA) format.

NGUID = 'NGUID'

The Namespace Globally Unique Identifier (NGUID).

NQN = 'NQN'

The NVMe Qualified Name (NQN).

NSID = 'NSID'

The NVM Namespace Identifier (NSID).

UUID = 'UUID'

The Universally Unique Identifier (UUID).

iQN = 'iQN'

The iSCSI Qualified Name (iQN).

class sushy.resources.constants.**Health**(*value*)

Bases: Enum

Health related constants.

CRITICAL = 'Critical'

A critical condition requires immediate attention.

OK = 'OK'

Normal.

WARNING = 'Warning'

A condition requires attention.

class sushy.resources.constants.**IndicatorLED**(*value*)

Bases: Enum

Indicator LED Constants

BLINKING = 'Blinking'

The Indicator LED is blinking

LIT = 'Lit'

The Indicator LED is lit

OFF = 'Off'

The Indicator LED is off

UNKNOWN = 'Unknown'

The state of the Indicator LED cannot be determine

class sushy.resources.constants.**PowerState**(*value*)

Bases: Enum

System PowerState constants

OFF = 'Off'

The resource is powered off, although some components may continue to have AUX power such as management controller

ON = 'On'

The resource is powered on

PAUSED = 'Paused'

The resource is paused.

POWERING_OFF = 'PoweringOff'

A temporary state between On and Off. The power off action can take time while the OS is in the shutdown process

POWERING_ON = 'PoweringOn'

A temporary state between Off and On. This temporary state can be very short

class sushy.resources.constants.**Protocol**(*value*)

Bases: `Enum`

Protocol type constants

AHCI = 'AHCI'

Advanced Host Controller Interface (AHCI).

DISPLAY_PORT = 'DisplayPort'

DisplayPort.

DVI = 'DVI'

DVI.

ETHERNET = 'Ethernet'

Ethernet.

FC = 'FC'

Fibre Channel.

FCP = 'FCP'

Fibre Channel Protocol for SCSI.

FCoE = 'FCoE'

Fibre Channel over Ethernet (FCoE).

FICON = 'FICON'

FIbre CONnection (FICON).

FTP = 'FTP'

File Transfer Protocol (FTP).

GEN_Z = 'GenZ'

GenZ.

HDMI = 'HDMI'

HDMI.

HTTP = 'HTTP'

Hypertext Transport Protocol (HTTP).

HTTPS = 'HTTPS'

Hypertext Transfer Protocol Secure (HTTPS).

I2C = 'I2C'

Inter-Integrated Circuit Bus.

INFINI_BAND = 'InfiniBand'

InfiniBand.

MULTI_PROTOCOL = 'MultiProtocol'
Multiple Protocols.

NFSv3 = 'NFSv3'
Network File System (NFS) version 3.

NFSv4 = 'NFSv4'
Network File System (NFS) version 4.

NVLINK = 'NVLink'
NVLink.

NVMe = 'NVMe'
Non-Volatile Memory Express (NVMe).

NVMe_OVER_FABRICS = 'NVMeOverFabrics'
NVMe over Fabrics.

OEM = 'OEM'
OEM-specific.

PCIe = 'PCIe'
PCI Express.

RoCE = 'RoCE'
RDMA over Converged Ethernet Protocol.

RoCEv2 = 'RoCEv2'
RDMA over Converged Ethernet Protocol Version 2.

SAS = 'SAS'
Serial Attached SCSI.

SATA = 'SATA'
Serial AT Attachment.

SFTP = 'SFTP'
SSH File Transfer Protocol (SFTP).

SMB = 'SMB'
Server Message Block (SMB). Also known as the Common Internet File System (CIFS).

TCP = 'TCP'
Transmission Control Protocol (TCP).

TFTP = 'TFTP'
Trivial File Transfer Protocol (TFTP).

UDP = 'UDP'
User Datagram Protocol (UDP).

UHCI = 'UHCI'
Universal Host Controller Interface (UHCI).

USB = 'USB'
Universal Serial Bus (USB).

VGA = 'VGA'
VGA.

iSCSI = 'iSCSI'
Internet SCSI.

iWARP = 'iWARP'
Internet Wide Area RDMA Protocol (iWARP).

class `sushy.resources.constants.`**ResetType**(*value*)
Bases: `Enum`

Reset action constants

FORCE_OFF = 'ForceOff'
Turn off the unit immediately (non-graceful shutdown).

FORCE_ON = 'ForceOn'
Turn on the unit immediately.

FORCE_RESTART = 'ForceRestart'
Shut down immediately and non-gracefully and restart the system.

GRACEFUL_RESTART = 'GracefulRestart'
Shut down gracefully and restart the system.

GRACEFUL_SHUTDOWN = 'GracefulShutdown'
Shut down gracefully and power off.

NMI = 'Nmi'
Generate a diagnostic interrupt, which is usually an NMI on x86 systems, to stop normal operations, complete diagnostic actions, and, typically, halt the system.

ON = 'On'
Turn on the unit.

PAUSE = 'Pause'
Pause execution on the unit but do not remove power. This is typically a feature of virtual machine hypervisors.

POWER_CYCLE = 'PowerCycle'
Power cycle the unit. Behaves like a full power removal, followed by a power restore to the resource.

PUSH_POWER_BUTTON = 'PushPowerButton'
Simulate the pressing of the physical power button on this unit.

RESUME = 'Resume'
Resume execution on the paused unit. This is typically a feature of virtual machine hypervisors.

SUSPEND = 'Suspend'
Write the state of the unit to disk before powering off. This allows for the state to be restored when powered back on.

`sushy.resources.constants.`**Severity**
alias of *`Health`*

class sushy.resources.constants.**State**(*value*)

Bases: `Enum`

State related constants.

ABSENT = 'Absent'

This function or resource is either not present or detected.

DEFERRING = 'Deferring'

The element does not process any commands but queues new requests.

DISABLED = 'Disabled'

This function or resource is disabled.

ENABLED = 'Enabled'

This function or resource is enabled.

IN_TEST = 'InTest'

This function or resource is undergoing testing, or is in the process of capturing information for debugging.

QUALIFIED = 'Qualified'

The element quality is within the acceptable range of operation.

QUIESCED = 'Quiesced'

The element is enabled but only processes a restricted set of commands.

STANDBY_OFFLINE = 'StandbyOffline'

This function or resource is enabled but awaits an external action to activate it.

STANDBY_SPARE = 'StandbySpare'

This function or resource is part of a redundancy set and awaits a failover or other external action to activate it.

STARTING = 'Starting'

This function or resource is starting.

UNAVAILABLE_OFFLINE = 'UnavailableOffline'

This function or resource is present but cannot be used.

UPDATING = 'Updating'

The element is updating and might be unavailable or degraded.

sushy.resources.ipaddresses module

class sushy.resources.ipaddresses.**AddressState**(*value*)

Bases: `Enum`

An enumeration.

DEPRECATED = 'Deprecated'

This address is currently within its valid lifetime but is now outside its RFC4862-defined preferred lifetime.

FAILED = 'Failed'

This address has failed Duplicate Address Detection (DAD) testing, as defined in RFC4862, section 5.4, and is not currently in use.

PREFERRED = 'Preferred'

This address is currently within both its RFC4862-defined valid and preferred lifetimes.

TENTATIVE = 'Tentative'

This address is currently undergoing Duplicate Address Detection (DAD) testing, as defined in RFC4862, section 5.4.

class sushy.resources.ipaddresses.**IPv4AddressOrigin**(*value*)

Bases: Enum

An enumeration.

BOOTP = 'BOOTP'

A BOOTP service-provided address.

DHCP = 'DHCP'

A DHCPv4 service-provided address.

LINK_LOCAL = 'IPv4LinkLocal'

The address is valid for only this network segment, or link.

STATIC = 'Static'

A user-configured static address.

class sushy.resources.ipaddresses.**IPv6AddressOrigin**(*value*)

Bases: Enum

An enumeration.

DHCP = 'DHCPv6'

A DHCPv6 service-provided address.

LINK_LOCAL = 'LinkLocal'

The address is valid for only this network segment, or link.

SLAAC = 'SLAAC'

A stateless autoconfiguration (SLAAC) service-provided address.

STATIC = 'Static'

A static user-configured address.

sushy.resources.settings module

class sushy.resources.settings.**MaintenanceWindowField**(**args*, ***kwargs*)

Bases: *CompositeField*

maintenance_window_duration_in_seconds = <sushy.resources.base.Field object>

The expiry time of maintenance window in seconds

maintenance_window_start_time = <sushy.resources.base.Field object>

The start time of a maintenance window

sushy.resources.settings.**NO_UPDATES = 4**

No updates made

class sushy.resources.settings.**SettingsApplyTimeField**

Bases: *CompositeField*

apply_time = <sushy.resources.base.Field object>

When the future configuration should be applied

apply_time_allowable_values = <sushy.resources.base.Field object>

The list of allowable ApplyTime values

maintenance_window_duration_in_seconds = <sushy.resources.base.Field object>

The expiry time of maintenance window in seconds

maintenance_window_start_time = <sushy.resources.base.Field object>

The start time of a maintenance window

class sushy.resources.settings.**SettingsField**

Bases: *CompositeField*

The settings of a resource

Represents the future state and configuration of the resource. The field is added to resources that support future state and configuration.

This field includes several properties to help clients monitor when the resource is consumed by the service and determine the results of applying the values, which may or may not have been successful.

commit(*connector*, *value*)

Commits new settings values

The new values will be applied when the system or a service restarts.

Parameters

- **connector** – A Connector instance
- **value** – Value representing JSON whose structure is specific to each resource and the caller must format it correctly

Returns

Response object

get_status(*registries*)

Determines the status of last update based

Uses message id-s and severity to determine the status.

Parameters

registries – registries to use to parse message

Returns
SettingsUpdate object containing status and any messages

property maintenance_window

MaintenanceWindow field

Indicates if a given resource has a maintenance window assignment for applying settings or operations

messages = <sushy.resources.base.MessageListField object>

Represents the results of the last time the values of the Settings resource were applied to the server

property operation_apply_time_support

OperationApplyTimeSupport field

Indicates if a client is allowed to request for a specific apply time of a create, delete, or action operation of a given resource

property resource_uri

time = <sushy.resources.base.Field object>

Indicates the time the settings were applied to the server

class sushy.resources.settings.**SettingsUpdate**(*status*, *messages*)

Bases: `object`

Contains Settings update status and details of the update

property messages

List of *MessageListField* with messages from the update

property status

The status of the update

sushy.resources.settings.**UPDATE_FAILURE = 2**

Update encountered errors

sushy.resources.settings.**UPDATE_PENDING = 3**

Update waiting for being applied

sushy.resources.settings.**UPDATE_SUCCESS = 1**

Update was successful

sushy.resources.settings.**UPDATE_UNKNOWN = 0**

Update status unknown

Module contents

Submodules

sushy.auth module

class sushy.auth.**AuthBase**(*username=None*, *password=None*)

Bases: object

authenticate()

Perform authentication.

Raises

RuntimeError

abstract can_refresh_session()

Method to assert if session based refresh can be done.

close()

Shutdown Redfish authentication object

Undoes whatever should be undone to cancel authenticated session.

set_context(*root_resource*, *connector*)

Set the context of the authentication object.

Parameters

- **root_resource** – Root sushy object
- **connector** – Connector for http connections

class sushy.auth.**BasicAuth**(*username=None*, *password=None*)

Bases: *AuthBase*

Basic Authentication class.

This is a class used to encapsulate a basic authentication session.

Parameters

- **username** – User account with admin/server-profile access privilege.
- **password** – User account password.

can_refresh_session()

Method to assert if session based refresh can be done.

class sushy.auth.**SessionAuth**(*username=None*, *password=None*)

Bases: *AuthBase*

Session Authentication class.

This is a class used to encapsulate a redfish session.

can_refresh_session()

Method to assert if session based refresh can be done.

close()

Close the Redfish Session.

Attempts to close an established RedfishSession by deleting it from the remote Redfish controller.

get_session_key()

Returns the session key.

Returns
The session key.

get_session_resource_id()

Returns the session resource id.

Returns
The session resource id.

refresh_session()

Method to refresh a session to a Redfish controller.

This method is called to create a new session after a session that has already been established has timed-out or expired.

Raises
MissingXAuthToken

Raises
ConnectionError

Raises
AccessError

Raises
HTTPError

reset_session_attrs()

Reset active session related attributes.

class sushy.auth.**SessionOrBasicAuth**(*username=None*, *password=None*)

Bases: *SessionAuth*

refresh_session()

Method to refresh a session to a Redfish controller.

This method is called to create a new RedfishSession if we have previously established a RedfishSession and the previous session has timed-out or expired. If we did not previously have an established session, we simply return our BasicAuthentication requests.Session.

sushy.connector module

class sushy.connector.**Connector**(*url*, *username=None*, *password=None*, *verify=True*, *response_callback=None*, *server_side_retries=0*, *server_side_retries_delay=0*)

Bases: `object`

check_retry_on_exception(*exception_msg*)

Checks whether retry on exception is required.

close()

Close this connector and the associated HTTP session.

delete(*path=''*, *data=None*, *headers=None*, *blocking=False*, *timeout=60*, ***extra_session_req_kwargs*)

HTTP DELETE method.

Parameters

- **path** – Optional sub-URI path to the resource.
- **data** – Optional JSON data.
- **headers** – Optional dictionary of headers.
- **blocking** – Whether to block for asynchronous operations.
- **timeout** – Max time in seconds to wait for blocking async call.
- **extra_session_req_kwargs** – Optional keyword argument to pass requests library arguments which would pass on to requests session object.

Returns

The response object from the requests library.

Raises

ConnectionError

Raises

HTTPError

get(*path=''*, *data=None*, *headers=None*, *blocking=False*, *timeout=60*, ***extra_session_req_kwargs*)

HTTP GET method.

Parameters

- **path** – Optional sub-URI path to the resource.
- **data** – Optional JSON data.
- **headers** – Optional dictionary of headers.
- **blocking** – Whether to block for asynchronous operations.
- **timeout** – Max time in seconds to wait for blocking async call.
- **extra_session_req_kwargs** – Optional keyword argument to pass requests library arguments which would pass on to requests session object.

Returns
The response object from the requests library.

Raises
ConnectionError

Raises
HTTPError

patch (*path=''*, *data=None*, *headers=None*, *etag=None*, *blocking=False*, *timeout=60*, ***extra_session_req_kwargs*)

HTTP PATCH method.

Parameters

- **path** – Optional sub-URI path to the resource.
- **data** – Optional JSON data.
- **headers** – Optional dictionary of headers.
- **etag** – Optional eTag string.
- **blocking** – Whether to block for asynchronous operations.
- **timeout** – Max time in seconds to wait for blocking async call.
- **extra_session_req_kwargs** – Optional keyword argument to pass requests library arguments which would pass on to requests session object.

Returns
The response object from the requests library.

Raises
ConnectionError

Raises
HTTPError

post (*path=''*, *data=None*, *headers=None*, *blocking=False*, *timeout=60*, ***extra_session_req_kwargs*)

HTTP POST method.

Parameters

- **path** – Optional sub-URI path to the resource.
- **data** – Optional JSON data.
- **headers** – Optional dictionary of headers.
- **blocking** – Whether to block for asynchronous operations.
- **timeout** – Max time in seconds to wait for blocking async call.
- **extra_session_req_kwargs** – Optional keyword argument to pass requests library arguments which would pass on to requests session object.

Returns
The response object from the requests library.

Raises
ConnectionError

Raises
HTTPError

put (*path=''*, *data=None*, *headers=None*, *blocking=False*, *timeout=60*, ***extra_session_req_kwargs*)

HTTP PUT method.

Parameters

- **path** – Optional sub-URI path to the resource.
- **data** – Optional JSON data.
- **headers** – Optional dictionary of headers.
- **blocking** – Whether to block for asynchronous operations.
- **timeout** – Max time in seconds to wait for blocking async call.
- **extra_session_req_kwargs** – Optional keyword argument to pass requests library arguments which would pass on to requests session object.

Returns
The response object from the requests library.

Raises
ConnectionError

Raises
HTTPError

set_auth (*auth*)

Sets the authentication mechanism for our connector.

set_http_basic_auth (*username*, *password*)

Sets the http basic authentication information.

set_http_session_auth (*session_auth_token*)

Sets the session authentication information.

sushy.exceptions module

exception `sushy.exceptions.`**AccessError** (*method*, *url*, *response*)

Bases: *HTTPError*

exception `sushy.exceptions.`**ArchiveParsingError** (*message=None*, ***kwargs*)

Bases: *SushyError*

message = 'Failed parsing archive "%(path)s": %(error)s'

exception `sushy.exceptions.`**BadRequestError** (*method*, *url*, *response*)

Bases: *HTTPError*

exception `sushy.exceptions.`**ConnectionError** (*message=None*, ***kwargs*)

Bases: *SushyError*

message = 'Unable to connect to %(url)s. Error: %(error)s'

exception sushy.exceptions.**ExtensionError**(*message=None*, ***kwargs*)

Bases: *SushyError*

message = 'Sushy Extension Error: %(error)s'

exception sushy.exceptions.**HTTPError**(*method*, *url*, *response*)

Bases: *SushyError*

Basic exception for HTTP errors

body = None

Error JSON body, if present.

code = 'Base.1.0.GeneralError'

Error code defined in the Redfish specification, if present.

detail = None

Error message defined in the Redfish specification, if present.

extended_info = None

Extended information provided in the response.

message = 'HTTP %(method)s %(url)s returned code %(code)s. %(error)s Extended information: %(ext_info)s'

property related_properties

List of properties related to the error.

status_code = None

HTTP status code.

exception sushy.exceptions.**InvalidParameterValueError**(*message=None*, ***kwargs*)

Bases: *SushyError*

message = 'The parameter "%(parameter)s" value "%(value)s" is invalid. Valid values are: %(valid_values)s'

exception sushy.exceptions.**MalformedAttributeError**(*message=None*, ***kwargs*)

Bases: *SushyError*

message = 'The attribute %(attribute)s is malformed in the resource %(resource)s: %(error)s'

exception sushy.exceptions.**MissingActionError**(*message=None*, ***kwargs*)

Bases: *SushyError*

message = 'The action %(action)s is missing from the resource %(resource)s'

exception sushy.exceptions.**MissingAttributeError**(*message=None*, ***kwargs*)

Bases: *SushyError*

message = 'The attribute %(attribute)s is missing from the resource %(resource)s'

exception sushy.exceptions.**MissingHeaderError**(*message=None*, ***kwargs*)

Bases: *SushyError*

message = 'Response to %(target_uri)s did not contain a %(header)s header'

exception sushy.exceptions.**MissingXAuthToken**(*method*, *url*, *response*)

Bases: *HTTPError*

message = 'No X-Auth-Token returned from remote host when attempting to establish a session. Error: %(error)s'

exception sushy.exceptions.**NotAcceptableError**(*method*, *url*, *response*)

Bases: *HTTPError*

exception sushy.exceptions.**OEMExtensionNotFoundError**(*message=None*, ***kwargs*)

Bases: *SushyError*

message = 'No %(resource)s OEM extension found by name "%(name)s".'

exception sushy.exceptions.**ResourceNotFoundError**(*method*, *url*, *response*)

Bases: *HTTPError*

message = 'Resource %(url)s not found'

exception sushy.exceptions.**ServerSideError**(*method*, *url*, *response*)

Bases: *HTTPError*

exception sushy.exceptions.**SushyError**(*message=None*, ***kwargs*)

Bases: Exception

Basic exception for errors raised by Sushy

message = None

exception sushy.exceptions.**UnknownDefaultError**(*message=None*, ***kwargs*)

Bases: *SushyError*

message = 'Failed at determining default for "%(entity)s": %(error)s'

sushy.exceptions.**raise_for_response**(*method*, *url*, *response*)

Raise a correct error class, if needed.

sushy.main module

class sushy.main.**LazyRegistries**(*service_root*)

Bases: MutableMapping

Download registries on demand.

Redfish message registries can be very large. On top of that, they are not used frequently. Thus, let's not pull them off the BMC unless the consumer is actually trying to use them.

Parameters

service_root (`sushy.main.Sushy`) – Redfish service root object

property registries

class `sushy.main.`**`ProtocolFeaturesSupportedField`**(**args*, ***kwargs*)

Bases: *`CompositeField`*

`excerpt_query = <sushy.resources.base.Field object>`

The excerpt query parameter is supported

`expand_query = <sushy.resources.base.Field object>`

The expand query parameter is supported

`filter_query = <sushy.resources.base.Field object>`

The filter query parameter is supported

`only_member_query = <sushy.resources.base.Field object>`

The only query parameter is supported

`select_query = <sushy.resources.base.Field object>`

The select query parameter is supported

class `sushy.main.`**`Sushy`**(*base_url*, *username=None*, *password=None*, *root_prefix='/redfish/v1/'*, *verify=True*, *auth=None*, *connector=None*, *public_connector=None*, *language='en'*, *server_side_retries=10*, *server_side_retries_delay=3*)

Bases: *`ResourceBase`*

`create_session`(*username=None*, *password=None*)

Creates a session without invoking SessionService.

For use when a new connection is to be established. Removes prior Session and authentication data before making the request.

Parameters

- **username** – The username to utilize to create a session with the remote endpoint.
- **password** – The password to utilize to create a session with the remote endpoint.

Returns

A session key and uri in the form of a tuple

Raises

MissingXAuthToken

Raises

ConnectionError

Raises

AccessError

Raises

HTTPError

Raises

MissingAttributeError

`get_certificate_service`()

Get the CertificateService object

Returns
The CertificateService object

`get_chassis`(*identity=None*)

Given the identity return a Chassis object

Parameters
`identity` – The identity of the Chassis resource. If not given, sushy will default to the single available chassis or fail if there appear to be more or less then one Chassis listed.

Raises
UnknownDefaultError if default system can't be determined.

Returns
The Chassis object

`get_chassis_collection`()

Get the ChassisCollection object

Raises
MissingAttributeError, if the collection attribute is not found

Returns
a ChassisCollection object

`get_composition_service`()

Get the CompositionService object

Raises
MissingAttributeError, if the composition service attribute is not found

Returns
The CompositionService object

`get_event_service`()

Get the EventService object

Raises
MissingAttributeError, if the EventService is not found

Returns
The EventService object

`get_fabric`(*identity*)

Given the identity return a Fabric object

Parameters
`identity` – The identity of the Fabric resource

Returns
The Fabric object

`get_fabric_collection`()

Get the FabricCollection object

Raises
MissingAttributeError, if the collection attribute is not found

Returns
a FabricCollection object

get_manager(*identity=None*)

Given the identity return a Manager object

Parameters
identity – The identity of the Manager resource. If not given, sushy will default to the single available Manager or fail if there appear to be more or less then one Manager listed.

Returns
The Manager object

get_manager_collection()

Get the ManagerCollection object

Raises
MissingAttributeError, if the collection attribute is not found

Returns
a ManagerCollection object

get_session(*identity*)

Given the identity return a Session object

Parameters
identity – The identity of the session resource

Returns
The Session object

get_session_service()

Get the SessionService object

Raises
MissingAttributeError, if the collection attribute is not found

Returns
as SessionCollection object

get_sessions_path()

Returns the Sessions url

get_system(*identity=None*)

Given the identity return a System object

Parameters
identity – The identity of the System resource. If not given, sushy will default to the single available System or fail if there appear to be more or less then one System listed.

Raises
UnknownDefaultError if default system can't be determined.

Returns
The System object

get_system_collection()

Get the SystemCollection object

Raises
MissingAttributeError, if the collection attribute is not found

Returns
a SystemCollection object

get_task_monitor(*task_monitor_uri*)

Used to retrieve a TaskMonitor by task monitor URI.

Parameters
task_monitor_uri – Task monitor URI

Returns
A task monitor.

get_task_service()

Get the TaskService object

Returns
The TaskService object

get_update_service()

Get the UpdateService object

Returns
The UpdateService object

identity = <sushy.resources.base.Field object>

The Redfish root service identity

property lazy_registries

Gets and combines all message registries together

Fetches all registries if any provided by Redfish service and combines together with packaged standard registries.

Returns
dict of combined message registries where key is Registry_name.Major_version.Minor_version and value is registry itself.

name = <sushy.resources.base.Field object>

The Redfish root service name

product = <sushy.resources.base.Field object>

The product associated with this Redfish service

protocol_features_supported = <sushy.main.ProtocolFeaturesSupportedField object>

The information about protocol features supported by the service

property registries

Gets and combines all registries together

Fetches all registries if any provided by Redfish service and combines together with packaged standard registries. Both message and attribute registries are supported from the Redfish service.

Returns

dict of combined registries keyed by both the registry name (Registry_name.Major_version.Minor_version) and the registry file identity, with the value being the actual registry itself.

uuid = <sushy.resources.base.Field object>

The Redfish root service UUID

sushy.taskmonitor module

class `sushy.taskmonitor.`**TaskMonitor**(*connector*, *task_monitor_uri*, *redfish_version=None*, *registries=None*, *response=None*)

Bases: `object`

property cancellable

The amount of time to sleep before retrying

Returns

A Boolean indicating if the Task is cancellable.

property check_is_processing

Refreshes task and check if it is still processing

Returns

A boolean indicating if the task is still processing.

static from_response(*conn*, *response*, *target_uri*, *redfish_version=None*, *registries=None*)

Construct TaskMonitor instance from received response.

Response

Unprocessed response

Target_uri

URI used to initiate async operation

Redfish_version

Redfish version. Optional when used internally.

Registries

Redfish registries. Optional when used internally.

Returns

TaskMonitor instance

Raises

MissingHeaderError if Location is missing in response

get_task()

Construct Task instance from task monitor URI.

Returns
Task instance.

property is_processing

Indicates if the task is still processing

Returns
A boolean indicating if the task is still processing.

refresh()

Refresh the Task

Freshly retrieves/fetches the Task. :raises: ResourceNotFoundError :raises: ConnectionError :raises: HTTPError

property response

Unprocessed response.

Intended to be used internally. :returns: Unprocessed response.

property sleep_for

Seconds the client should wait before querying the operation status

Defaults to 1 second if Retry-After not specified in response.

Returns
The number of seconds to wait

property task

The executing task

Returns
The Task being executed.

property task_monitor_uri

The TaskMonitor URI

Returns
The TaskMonitor URI.

wait(*timeout_sec*)

Waits until task is completed or it times out.

Parameters
timeout_sec – Timeout to wait

Raises
ConnectionError when times out

sushy.utils module

sushy.utils.**bool_or_none**(*x*)

Given a value x this method returns either a bool or None

Parameters
x – The value to transform and return

Returns
Either None or x cast to a bool

sushy.utils.**cache_clear**(*res_selfie*, *force_refresh*, *only_these=None*)

Clear some or all cached values of the resource.

If the cache variable refers to a resource instance then the `invalidate()` method is called on that. Otherwise it is set to None. Should there be a need to force refresh the resource and its sub-resources, "cascading refresh", `force_refresh` is to be set to True.

This is the complimentary method of `cache_it` decorator.

Parameters

- **res_selfie** – the resource instance.
- **force_refresh** – force_refresh argument of `invalidate()` method.
- **only_these** – expects a sequence of specific method names for which the cached value/s need to be cleared only. When None, all the cached values are cleared.

sushy.utils.**cache_it**(*res_accessor_method*)

Utility decorator to cache the return value of the decorated method.

This decorator is to be used with any Sushy resource class method. This will internally create an attribute on the resource namely `_cache_<decorated_method_name>`. This is referred to as the "caching attribute". This attribute will eventually hold the resultant value from the method invocation (when method gets first time called) and for every subsequent calls to that method this cached value will get returned. It expects the decorated method to contain its own logic of evaluation.

This also assigns a variable named `_cache_attr_names` on the resource. This variable maintains a collection of all the existing "caching attribute" names.

To invalidate or clear the cache use *cache_clear()*. Usage:

```
class SomeResource(base.ResourceBase):
  ...
  @cache_it
  def get_summary(self):
    # do some calculation and return the result
    # and this result will be cached.
    return result
  ...
  def _do_refresh(self, force):
    cache_clear(self, force)
```

If the returned value is a Sushy resource instance or a sequence whose element is of type Sushy resource it handles the case of calling the `refresh()` method of that resource. This is done to avoid unnecessary recreation of a new resource instance which got already created at the first place in contrast to fresh retrieval of the resource json data. Again, the `force` argument is deliberately set to False to do only the "light refresh" of the resource (only the fresh retrieval of resource) instead of doing the complete exhaustive "cascading refresh" (resource with all its nested subresources recursively).

```
class SomeResource(base.ResourceBase):
  ...
  @property
  @cache_it
  def nested_resource(self):
    return NestedResource(
      self._conn, "Path/to/NestedResource",
      redfish_version=self.redfish_version)
  ...
  def _do_refresh(self, force):
    # selective attribute clearing
    cache_clear(self, force, only_these=['nested_resource'])
```

Do note that this is not thread safe. So guard your code to protect it from any kind of concurrency issues while using this decorator.

Parameters

`res_accessor_method` – the resource accessor decorated method.

`sushy.utils.`**`camelcase_to_underscore_joined`**(*camelcase_str*)

Convert camelCase string to underscore_joined string

Parameters

`camelcase_str` – The camelCase string

Returns

the equivalent underscore_joined string

`sushy.utils.`**`get_members_identities`**(*members*)

Extract and return a tuple of members identities

Parameters

`members` – A list of members in JSON format

Returns

A tuple containing the members paths

`sushy.utils.`**`get_sub_resource_path_by`**(*resource*, *subresource_name*, *is_collection=False*)

Helper function to find the subresource path

Parameters

- **`resource`** – ResourceBase instance on which the name gets queried upon.
- **`subresource_name`** – name of the resource field to fetch the '@odata.id' from.
- **`is_collection`** – if *True*, expect a list of resources to fetch the '@odata.id' from.

Returns

Resource path (if *is_collection* is *False*) or a list of resource paths (if *is_collection* is *True*).

`sushy.utils.`**`int_or_none`**(*x*)

Given a value x it cast as int or None

Parameters

`x` – The value to transform and return

Returns

Either None or x cast to an int

`sushy.utils.`**`max_safe`**(*iterable*, *default=0*)

Helper wrapper over builtin max() function.

This function is just a wrapper over builtin max() w/o `key` argument. The `default` argument specifies an object to return if the provided `iterable` is empty. Also it filters out the None type values.

Parameters

- **`iterable`** – an iterable
- **`default`** – 0 by default

`sushy.utils.`**`process_apply_time_input`**(*payload*, *apply_time*, *maint_window_start_time*, *maint_window_duration*)

Validates apply time input for asynchronous operations

Parameters

- **`payload`** – Payload for which to process apply time settings
- **`apply_time`** – When to update the attribute. Optional. An `sushy.ApplyTime` value.
- **`maint_window_start_time`** – The start time of a maintenance window, datetime. Required when updating during maintenance window and default maintenance window not set by the system.
- **`maint_window_duration`** – Duration of maintenance time since maintenance window start time in seconds. Required when updating during maintenance window and default maintenance window not set by the system.

Raises

`ValueError` – When input apply time settings incorrect

Returns

Payload with adjusted apply time settings if valid

`sushy.utils.`**`revert_dictionary`**(*dictionary*)

Given a dictionary revert it's mapping

Parameters

`dictionary` – A dictionary to be reverted

Returns

A dictionary with the keys and values reverted

`sushy.utils.`**`sanitize`**(*item*)

Remove passwords from the item.

`sushy.utils.`**`setdefaultattr`**(*obj*, *name*, *default*)

Python's `dict.setdefault` applied on Python objects.

If name is an attribute with obj, return its value. If not, set name attribute with a value of default and return default.

Parameters

- **`obj`** – a python object
- **`name`** – name of attribute
- **`default`** – default value to be set

`sushy.utils.`**`synchronized`**(*wrapped*)

Simple synchronization decorator.

Decorating a method like so:

```
@synchronized
def foo(self, *args):
  ...
```

ensures that only one thread will execute the foo method at a time.

Module contents

`class` `sushy.`**`Sushy`**(*base_url*, *username=None*, *password=None*, *root_prefix='/redfish/v1/'*, *verify=True*, *auth=None*, *connector=None*, *public_connector=None*, *language='en'*, *server_side_retries=10*, *server_side_retries_delay=3*)

Bases: `ResourceBase`

`create_session`(*username=None*, *password=None*)

Creates a session without invoking SessionService.

For use when a new connection is to be established. Removes prior Session and authentication data before making the request.

Parameters

- **`username`** – The username to utilize to create a session with the remote endpoint.
- **`password`** – The password to utilize to create a session with the remote endpoint.

Returns

A session key and uri in the form of a tuple

Raises

MissingXAuthToken

Raises

ConnectionError

Raises

AccessError

Raises
HTTPError

Raises
MissingAttributeError

`get_certificate_service()`

Get the CertificateService object

Returns
The CertificateService object

`get_chassis(identity=None)`

Given the identity return a Chassis object

Parameters
`identity` – The identity of the Chassis resource. If not given, sushy will default to the single available chassis or fail if there appear to be more or less then one Chassis listed.

Raises
UnknownDefaultError if default system can't be determined.

Returns
The Chassis object

`get_chassis_collection()`

Get the ChassisCollection object

Raises
MissingAttributeError, if the collection attribute is not found

Returns
a ChassisCollection object

`get_composition_service()`

Get the CompositionService object

Raises
MissingAttributeError, if the composition service attribute is not found

Returns
The CompositionService object

`get_event_service()`

Get the EventService object

Raises
MissingAttributeError, if the EventService is not found

Returns
The EventService object

`get_fabric(identity)`

Given the identity return a Fabric object

Parameters
`identity` – The identity of the Fabric resource

Returns
The Fabric object

get_fabric_collection()

Get the FabricCollection object

Raises
MissingAttributeError, if the collection attribute is not found

Returns
a FabricCollection object

get_manager(*identity=None*)

Given the identity return a Manager object

Parameters
identity – The identity of the Manager resource. If not given, sushy will default to the single available Manager or fail if there appear to be more or less then one Manager listed.

Returns
The Manager object

get_manager_collection()

Get the ManagerCollection object

Raises
MissingAttributeError, if the collection attribute is not found

Returns
a ManagerCollection object

get_session(*identity*)

Given the identity return a Session object

Parameters
identity – The identity of the session resource

Returns
The Session object

get_session_service()

Get the SessionService object

Raises
MissingAttributeError, if the collection attribute is not found

Returns
as SessionCollection object

get_sessions_path()

Returns the Sessions url

get_system(*identity=None*)

Given the identity return a System object

Parameters
identity – The identity of the System resource. If not given, sushy will default

to the single available System or fail if there appear to be more or less then one System listed.

Raises
UnknownDefaultError if default system can't be determined.

Returns
The System object

get_system_collection()

Get the SystemCollection object

Raises
MissingAttributeError, if the collection attribute is not found

Returns
a SystemCollection object

get_task_monitor(*task_monitor_uri*)

Used to retrieve a TaskMonitor by task monitor URI.

Parameters
task_monitor_uri – Task monitor URI

Returns
A task monitor.

get_task_service()

Get the TaskService object

Returns
The TaskService object

get_update_service()

Get the UpdateService object

Returns
The UpdateService object

identity = <sushy.resources.base.Field object>

The Redfish root service identity

property lazy_registries

Gets and combines all message registries together

Fetches all registries if any provided by Redfish service and combines together with packaged standard registries.

Returns
dict of combined message registries where key is Registry_name.Major_version.Minor_version and value is registry itself.

name = <sushy.resources.base.Field object>

The Redfish root service name

product = <sushy.resources.base.Field object>

The product associated with this Redfish service

protocol_features_supported = <sushy.main.ProtocolFeaturesSupportedField object>

The information about protocol features supported by the service

property registries

Gets and combines all registries together

Fetches all registries if any provided by Redfish service and combines together with packaged standard registries. Both message and attribute registries are supported from the Redfish service.

Returns

dict of combined registries keyed by both the registry name (Registry_name.Major_version.Minor_version) and the registry file identity, with the value being the actual registry itself.

uuid = <sushy.resources.base.Field object>

The Redfish root service UUID

- genindex

PYTHON MODULE INDEX

INDEX