SAP IoT Services SDK Documentation

Release 1.0

Philipp Steinroetter

Apr 12, 2019

CONTENTS:

CHAPTER

ONE

SAP_IOT_SERVICES_SDK

1.1 sap_iot_services_sdk package

1.1.1 Submodules

1.1.2 sap_iot_services_sdk.about module

class sap_iot_services_sdk.about.**AboutService**(*instance*, *user*, *password*)
Bases: *sap_iot_services_sdk.iot_service.IoTService*

get_information() → sap_iot_services_sdk.iot_service.Response
The endpoint returns information about the service and its configuration parameters.

Returns: Response – Response object

1.1.3 sap_iot_services_sdk.capability module

class sap_iot_services_sdk.capability.**CapabilityService**(*instance*, *user*, *password*)
Bases: *sap_iot_services_sdk.iot_service.IoTService*

create_capability(*alternate_id: str*, *name: str*, *properties: list*) → sap_iot_services_sdk.iot_service.Response
This endpoint is used to create a capability.

Arguments: alternate_id {str} – Alternate ID of the capability name {str} – Name of the capability properties {list} – List of dicts describing the properties

Returns: Response – Response object

delete_capability(*capability_id: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint is used to delete the capability associated to the given id.

Arguments: capability_id {str} – Unique identifier of a capability

Returns: Response – Response object

get_capabilities(*filters=None*, *orderby=None*, *asc=True*, *skip=None*, *top=None*) → sap_iot_services_sdk.iot_service.Response
The endpoint returns a list of capabilities.

Keyword Arguments: filters {list} – This parameter allows clients to filter the collection for attributes of a capability. The filters must be provided as a list of strings, e.g. [“name eq ‘my-name’”, “id eq ‘111’”] (default: {None}) orderby {str} – The attribute to order by. (default: {None}) asc {bool} – Only considered if orderby is not none. Defines if the values should be ordered asc or desc. (default: {True}) skip {int} – This parameter specifies the number of items in the queried collection which will be skipped and therefore not included in the result set (default: {None}) top {int} – This parameter restricts the maximum number of items which will be returned by the request (default: {None})

Returns: Response – Response object

get_capability(*capability_id: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint returns the capability associated to the given id.

Arguments: capability_id {str} – Unique identifier of a capability.

Returns: Response – Response object

update_capability(*capability_id: str*, *alternate_id: str*, *name: str*) → sap_iot_services_sdk.iot_service.Response
This endpoint is used to update the capability associated to the given id with details specified in the request body.

Arguments: capability_id {str} – Unique identifier of a capability alternate_id {str} – Alternate identifier of a capability name {str} – Name of a capability

Returns: Response – Response object

1.1.4 sap_iot_services_sdk.device module

class sap_iot_services_sdk.device.**DeviceService**(*instance*, *user*, *password*)
Bases: *sap_iot_services_sdk.iot_service.IoTService*

add_custom_property_to_device(*device_id: str*, *key: str*, *value: str*) → sap_iot_services_sdk.iot_service.Response
Used to add a custom property to the device associated to the given id.

Arguments: device_id {str} – Unique identifier of a device key {str} – Key of the custom property value {str} – Value of the custom property

Returns: Response – Response object

create_device(*gateway_id: str*, *name: str*, *as_router=False*, *custom_properties=[]*) → sap_iot_services_sdk.iot_service.Response
This endpoint is used to create a device.

Arguments: gateway_id {str} – Unique identifier of a gateway name {str} – Unique identifier of a name custom_properties {list} – List of dicts with the keys 'key' and 'value' specifying the custom properties

Returns: Response – Response object

delete_custom_property(*device_id: str*, *key: str*) → sap_iot_services_sdk.iot_service.Response
Delete a custom property from the device associated to the given id.

Arguments: device_id {str} – Unique identifier of a device key {str} – Key of the custom property

Returns: Response – Response object

delete_device(*device_id: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint is used to delete the device associated to the given id.

Arguments: device_id {str} – Unique identifier of a device

Returns: Response – Response object

get_device(*device_id: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint returns the device associated to the given id.

Arguments: device_id {str} – Unique identifier of a device

Returns: Response – Response object

get_device_p12(*device_id: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint is used to download device specific p12 file for authentication.

Arguments: device_id {str} – Unique identifier of a device

Returns: Response – Response object

get_device_pem(*device_id: str*) → sap_iot_services_sdk.iot_service.Response
Used to download a device specific private key and certificate in PEM format for authentication.

Arguments: device_id {str} – Unique identifier of a device

Returns: Response – Response object

get_devices(*filters=None*, *orderby=None*, *asc=True*, *skip=None*, *top=None*) → sap_iot_services_sdk.iot_service.Response
The endpoint returns a list of devices.

Keyword Arguments: filters {list} – This parameter allows clients to filter the collection for specific attributes. It is possible to filter by 'id', 'alternateId', 'gatewayId', 'name', 'description', and 'status'. The filters must be provided as a list of strings, e.q. ["name eq 'my-name'", "id eq '111'"]. (default: {None}) orderby {str} – The attribute to order by. (default: {None}) asc {bool} – Only considered if orderby is not none. Defines if the values should be ordered asc or desc. (default: {True}) skip {int} – This parameter specifies the number of items in the queried collection which will be skipped and therefore included in the result set (default: {None}) top {int} – This parameter restricts the maximum number of items which will be returned by the request (default: {None})

Returns: Response – Response object

get_measures(*device_id: str*, *filters=None*, *orderby=None*, *asc=True*, *skip=None*, *top=None*) → sap_iot_services_sdk.iot_service.Response
Returns a list of measures related to the device associated to the given id.

Arguments: device_id {str} – Unique identifier of a device

Keyword Arguments: filter {list} – This parameter allows clients to filter the collection for specific attributes. It is possible to filter by 'capabilityId' and 'timestamp'. When filtering by 'timestamp' the following binary operator are supported 'le', 'lt', 'ge', and 'gt'. The filters must be provided as a list of strings, e.q. ["name eq 'my-name'", "id eq '111'"] (default: {None}) orderby {str} – The attribute to order by. (default: {None}) asc {bool} – Only considered if orderby is not none. Defines if the values should be ordered asc or desc. (default: {True}) skip {int} – This parameter specifies the number of items in the queried collection which will be skipped and therefore included in the result set (default: {None}) top {int} – This parameter restricts the maximum number of items which will be returned by the request (default: {None})

Returns: Response – Response object

get_mqtt_client(*device_alternate_id: str*, *pemfile: str*, *secret: str*) → sap_iot_services_sdk.mqtt_client.MQTTClient
Returns MQTT client for specified router device

Arguments: device_alternate_id {str} – Alternate identifier of a device certfile_path {str} – Path to the certfile keyfile_path {str} – Path to the keyfile

Returns: MQTTClient – The MQTTClient object configured for the specified router device

get_rest_client(*device_alternate_id: str*, *pemfile: str*, *secret: str*) → sap_iot_services_sdk.rest_client.RestClient
Returns REST client for specified device

Arguments: device_alternate_id {str} – Alternate identifier of a device certfile_path {str} – Path to the certfile keyfile_path {str} – Path to the keyfile

Returns: RestClient – The RestClient object configured for the specified device

send_command_to_device(*device_id: str*, *capability_id: str*, *sensor_id: str*, *command: dict*) → sap_iot_services_sdk.iot_service.Response

Used to send the command specified in the request body to the device associated to the given id.

Arguments: device_id {str} – Unique identifier of a device capability_id {str} – Unique identifier of a capability sensor_id {str} – Unique identifier of a sensor command {dict} – Dict with additional properties and their respective values

Returns: Response – Response object

update_custom_property(*device_id: str*, *key: str*, *value: str*) → sap_iot_services_sdk.iot_service.Response

Updates a custom property of the device associated to the given id. The 'key' attribute cannot be modified.

Arguments: device_id {str} – Unique identifier of a device key {str} – Key of the custom property value {str} – The updated value of the custom property

Returns: Response – Response object

update_device(*device_id: str*, *name: str*) → sap_iot_services_sdk.iot_service.Response

This endpoint is used to update the device associated to the given id with details specified in the request body. This endpoint can only be used to modify a devices name. To update custom properties, sensors or authentications, use the respective APIs.

Arguments: device_id {str} – Unique identifier of a device name {str} – New device name

Returns: Response – Response object

1.1.5 sap_iot_services_sdk.gateway module

class sap_iot_services_sdk.gateway.**GatewayService**(*instance*, *user*, *password*)

Bases: *sap_iot_services_sdk.iot_service.IoTService*

add_custom_property(*gateway_id: str*, *key: str*, *value: str*) → sap_iot_services_sdk.iot_service.Response

The endpoint is used to add a custom property to the gateway associated to the given id.

Arguments: gateway_id {str} – Unique identifier of a gateway key {str} – Key of the custom property value {str} – Value of the custom property

Returns: Response – Response object

delete_custom_property(*gateway_id: str*, *key: str*) → sap_iot_services_sdk.iot_service.Response

This endpoint is used to delete a custom property from the gateway associated to the given id.

Arguments: gateway_id {str} – Unique identifier of a gateway key {str} – Key of the custom property

Returns: Response – Response object

delete_gateway(*gateway_id: str*) → sap_iot_services_sdk.iot_service.Response

The endpoint is used to delete the gateway associated to the given id.

Arguments: gateway_id {str} – Unique identifier of a gateway

Returns: Response – Response object

delete_osgi_bundle(*gateway_id: str*, *bundle_id: str*) → sap_iot_services_sdk.iot_service.Response

This endpoint is used to remove an OSGi bundle from the gateway associated to the given id.

Arguments: gateway_id {str} – Unique identifier of a gateway bundle_id {str} – Unique identifier of an OSGi bundle

Returns: Response – Response object

get_gateway(*gateway_id: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint returns the gateway associated to the given id.

Arguments: gateway_id {str} – Unique identifier of a gateway

Returns: Response – Response object

get_gateway_configuration(*gateway_id: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint is used to download the gateway specific configuration XML file.

Arguments: gateway_id {str} – Unique identifier of a gateway

Returns: Response – Response object

get_gateway_osgi_bundles(*gateway_id: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint returns a list of installed OSGi bundles.

Arguments: gateway_id {str} – Unique identifier of a gateway

Returns: Response – Response object

get_gateways(*filters=None*, *orderby=None*, *asc=True*, *skip=None*, *top=None*) → sap_iot_services_sdk.iot_service.Response
The endpoint returns a list of gateways.

Keyword Arguments: filters {list} – This parameter allows clients to filter the collection for specific attributes. The filters must be provided as a list of strings, e.q. [“name eq ‘my-name’”, “id eq ‘111’”] (default: {None}) orderby {str} – The attribute to order by. (default: {None}) asc {bool} – Only considered if orderby is not none. Defines if the values should be ordered asc or desc. (default: {True}) skip {int} – This parameter specifies the number of items in the queried collection which will be skipped and therefore not included in the result set (default: {None}) top {int} – This parameter restricts the maximum number of items which will be returned by the request (default: {None})

Returns: Response – Response object

get_osgi_bundle(*gateway_id: str*, *bundle_id: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint returns the OSGi bundle associated to the given id.

Arguments: gateway_id {str} – Unique identifier of a gateway bundle_id {str} – Unique identifier of an OSGi bundle

Returns: Response – Response object

install_gateway_osgi_bundle(*gateway_id: str*, *osgi_bundle: str*) → sap_iot_services_sdk.iot_service.Response
This endpoint is used to install an OSGi bundle on the gateway associated to the given id. Note that bundles with a file size over 128 MB will be rejected. The installation takes place asynchronously: the provided bundle is stored on the system (where it is kept up to 24 hours), then a request to download it is dispatched to Gateway. As soon as it receives the request, Gateway initiates the bundle download. The API returns immediately after the download request is dispatched to Gateway; in order to inspect the outcome of the bundle installation, get_gateway_osgi_bundles() should be used

Arguments: gateway_id {str} – Unique identifier of a gateway osgi_bundle {str} – Gateway OSGi Bundle

Returns: Response – Response object

start_gateway_osgi_bundle(*gateway_id: str*, *bundle_id: str*) → sap_iot_services_sdk.iot_service.Response
This endpoint is used to start the OSGi bundle of the gateway associated to the given id.

Arguments: gateway_id {str} – Unique identifier of a gateway bundle_id {str} – Unique identifier of an OSGi bundle

Returns: Response – Response object

stop_gateway_osgi_bundle(*gateway_id: str*, *bundle_id: str*) → sap_iot_services_sdk.iot_service.Response
This endpoint is used to stop the OSGi bundle of the gateway associated to the given id.

Arguments: gateway_id {str} – Unique identifier of a gateway bundle_id {str} – Unique identifier of an OSGi bundle

Returns: Response – Response object

update_custom_property(*gateway_id: str*, *key: str*, *value: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint is used to update a custom property of the gateway associated to the given id. The 'key' attribute cannot be modified.

Arguments: gateway_id {str} – Unique identifier of a gateway key {str} – Key of the custom property value {str} – Updates value of the custom property

Returns: Response – Response object

update_gateway_configuration(*gateway_id: str*, *xml: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint is used to update the gateway specific configuration by uploading a configuration XML file.

Arguments: gateway_id {str} – The endpoint is used to update the gateway specific configuration by uploading a configuration XML file. xml {str} – XML file as string containing the Gateway configuration

Returns: Response – Response object

update_gateway_name(*gateway_id: str*, *name: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint is used to update the gateway associated to the given id with details specified in the request body. To update custom properties, bundles or configuration, use the respective APIs.

Arguments: gateway_id {str} – Unique identifier of a gateway name {str} – Name for the gateway

Returns: Response – Response object

1.1.6 sap_iot_services_sdk.iot_service module

exception sap_iot_services_sdk.iot_service.**DeviceManagementAPIException**
Bases: Exception

class sap_iot_services_sdk.iot_service.**IoTService**(*instance: str*, *user: str*, *password: str*)
Bases: object

debug_requests_off()
Switches off logging of the requests module.

debug_requests_on()
Switches on logging of the requests module.

request_core(*method=None*, *service=None*, *headers=None*, *payload=None*, *accept_json=False*, *query=None*, *files=None*) → sap_iot_services_sdk.iot_service.Response
Fires a HTTP request to core services

Keyword Arguments: method {str} – HTTP method (default: {None}) service {str} – Service Path (default: {None}) headers {dict} – HTTP headers (default: {None}) payload {str} – Message payload (default: {None}) accept_json {bool} – If set to true, the response is parsed a JSON (default: {False}) query {str} – Query for filtering (default: {None}) files {str} – Path to files (default: {None})

Returns: Response – Response object

class sap_iot_services_sdk.iot_service.**Response**(*status_code*, *response*, *headers*)
Bases: object

Objects contain information received from API

get_headers() → str
Returns the header of the response

Returns: str – The header of the response message.

get_result() → str
Returns the result of the response, e.g. the body of the message

Returns: str – The body of the response message. Mostly in JSON formatting.

get_status_code() → int
Status code of the response

Returns: str – The status code of the HTTP communication

1.1.7 sap_iot_services_sdk.mqtt_client module

class sap_iot_services_sdk.mqtt_client.**MQTTClient**(*instance: str*, *device_alternate_id: str*, *pemfile: str*, *secret: str*)
Bases: *sap_iot_services_sdk.mqtt_client.PahoMQTT*

Wrapper around the Paho MQTT Client to simplify its usage with the IoTS Cloud Gateway

connect(*keepalive=60*)
Connects to the broker

Keyword Arguments: keepalive {int} – The number of seconds the connection should be kept alive (default: {60})

on_command
If implemented, called when the client has received a command message. Defined to allow command handling.

on_error
If implemented, called when the client has error. Defined to allow error handling.

publish(*capability_alternate_id: str*, *sensor_alternate_id: str*, *measures: list*, *device_alternate_id: str = None*, *timestamp: int = None*) → paho.mqtt.client.MQTTMessageInfo
Publishes measures to the IoT Services

Arguments: device_alternate_id {str} – Alternate ID of the device. If none, the device from the client id will be used. capability_alternate_id {str} – Alternate ID of the capability sensor_alternate_id {str} – Alternate ID of the sensor timestamp {int} – UNIX time in milliseconds. If None, current time will be used. measures {list} – List of key-value pairs containing the measures and their respective values

Returns: mqtt.MQTTMessageInfo – MQTT Message Info

simulate(*device_alternate_id: str*, *capability_alternate_id: str*, *sensor_alternate_id: str*, *measures: list*, *interval=1*, *runtime=60*)
Simulate measures for device

Arguments: device_alternate_id {str} – Alternate ID of the device you want to simulate data for capability_alternate_id {str} – Alternate ID of the capability you want to simulate data for sensor_alternate_id {str} – Alternate ID of the sensor you want to simulate data for measures {list} – List of dicts with one dict for each measure. You need to provide the field 'key' with the respective key of the measure you want to simulate. In 'dataType' you need to provide the respective data type. The function supports 'string', 'double', 'integer', 'boolean', and 'binary'. For data types 'double' and 'integer' you need

to provide 'min' and 'max'. For data types 'string' and 'binary' you need to provide a list of allowed strings in 'allowedStrings', from which is randomly chosen.

Keyword Arguments: interval {int} – The interval in seconds in which the data should be send (default: {1}) runtime {int} – Defines how long the simulation should be run in seconds (default: {60})

subscribe(*device_alternate_id: str) -> (<class 'str'>, <class 'str'>*)
Subscribe to a devices commands

Arguments: device_alternate_id {str} – The alternate id of the device

Returns: str – Result of the subscription str – Message ID

class sap_iot_services_sdk.mqtt_client.**PahoMQTT**(*client_id=''*, *clean_session=True*, *userdata=None*, *protocol=4*, *transport='tcp'*)
Bases: paho.mqtt.client.Client

Overwrites the mqtt.Client class to work with password protected pem files

tls_set(*ca_certs=None*, *pemfile=None*, *secret=None*, *tls_version=None*)
Configure network encryption and authentication options. Enables SSL/TLS support. ca_certs : a string path to the Certificate Authority certificate files that are to be treated as trusted by this client. If this is the only option given then the client will operate in a similar manner to a web browser. That is to say it will require the broker to have a certificate signed by the Certificate Authorities in ca_certs and will communicate using TLS v1, but will not attempt any form of authentication. This provides basic network encryption but may not be sufficient depending on how the broker is configured. By default, on Python 2.7.9+ or 3.4+, the default certification authority of the system is used. On older Python version this parameter is mandatory. pemfile is a string pointing to the PEM encoded client certificate. If the argument is not None it they will be used as client information for TLS based authentication. Support for this feature is broker dependent. Note that if the file in encrypted and needs a password to decrypt it, you will have to provide the secret, too. tls_version allows the version of the SSL/TLS protocol used to be specified. By default TLS v1 is used. Previous versions (all versions beginning with SSL) are possible but not recommended due to possible security problems. Must be called before connect() or connect_async().

1.1.8 sap_iot_services_sdk.protocol module

class sap_iot_services_sdk.protocol.**ProtocolService**(*instance*, *user*, *password*)
Bases: *sap_iot_services_sdk.iot_service.IoTService*

create_protocol(*protocol_id: str*) → sap_iot_services_sdk.iot_service.Response
he endpoint is used to create a protocol. The new protocol is visible for all tenants of the instance.

Arguments: protocol_id {str} – ID of the protocol that will be created

Returns: Response – Response object

delete_protocol(*protocol_id: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint is used to delete a protocol. Deleting is possible only if no tenant on the instance is using it anymore.

Arguments: protocol_id {str} – Unique identifier of a protocol.

Returns: Response – Response object

get_protocols(*skip: int*, *top: int*) → sap_iot_services_sdk.iot_service.Response
The endpoint returns a list of protocols available on the instance.

Arguments: skip {int} – This parameter specifies the number of items in the queried collection which will be skipped and therefore included in the result top {int} – This parameter restricts the maximum number of items which will be returned by the request

Returns: Response – Response object

1.1.9 sap_iot_services_sdk.rest_client module

class sap_iot_services_sdk.rest_client.**RESTGatewayAdapter**(**args*, ***kwargs*)
Bases: requests.adapters.HTTPAdapter

init_poolmanager(**args*, ***kwargs*)
Initializes a urllib3 PoolManager.

This method should not be called from user code, and is only exposed for use when subclassing the HTTPAdapter.

Parameters

- **connections** – The number of urllib3 connection pools to cache.
- **maxsize** – The maximum number of connections to save in the pool.
- **block** – Block when no free connections are available.
- **pool_kwargs** – Extra keyword arguments used to initialize the Pool Manager.

proxy_manager_for(**args*, ***kwargs*)
Return urllib3 ProxyManager for the given proxy.

This method should not be called from user code, and is only exposed for use when subclassing the HTTPAdapter.

Parameters

- **proxy** – The proxy to return a urllib3 ProxyManager for.
- **proxy_kwargs** – Extra keyword arguments used to configure the Proxy Manager.

Returns ProxyManager

Return type urllib3.ProxyManager

exception sap_iot_services_sdk.rest_client.**RESTGatewayException**
Bases: Exception

class sap_iot_services_sdk.rest_client.**RestClient**(*instance: str*, *device_alternate_id: str*, *pemfile: str*, *secret: str*)
Bases: object

post_batched_measures(*messages: list*) → sap_iot_services_sdk.iot_service.Response
Post batched measures over rest gateway

Arguments: messages {list} – List of dicts, each containing sensorAlternateId, capabilityAlternateId and an array of measures as key-value pairs. If the device should be onboarded automatically, the sensorTypeAlternateId must also be provided.

Returns: Response – Response object

post_command(*capability_alternate_id: str*, *sensor_alternate_id: str*, *command: dict*) → sap_iot_services_sdk.iot_service.Response
Post commands over rest gateway for specified device

Arguments: capability_alternate_id {str} – Alternate ID for capability sensor_alternate_id {str} – Alternate ID for sensor command {dict} – Dict with the keys and respective values of the desired commands

Returns: Response – Response object

post_measures(*capability_alternate_id: str*, *sensor_alternate_id: str*, *measures: list*, *sensor_type_alternate_id: int = None*, *timestamp: int = None*) → sap_iot_services_sdk.iot_service.Response

Post measures over rest gateway for specified device

Arguments: capability_alternate_id {str} – Alternate ID for capability sensor_alternate_id {str} – Alternate ID for sensor measures {list} – List of key-value pairs with the keys and respective values of the desired measures sensor_type_alternate_id {int} – (Optional) If this parameter is set, the device will be auto-onboarded if it does not exist yet. Note: The alternate id of the sensor type must be numeric. timestamp {int} – UNIX time in milliseconds. If None, current time will be used.

Returns: Response – Response object

1.1.10 sap_iot_services_sdk.sensor module

class sap_iot_services_sdk.sensor.**SensorService**(*instance*, *user*, *password*)

Bases: *sap_iot_services_sdk.iot_service.IoTService*

create_sensor(*device_id: str*, *alternate_id: str*, *name: str*, *sensor_type_id: str*) → sap_iot_services_sdk.iot_service.Response

This endpoint is used to create a sensor.

Arguments: device_id {str} – Respective device ID for the sensor alternate_id {str} – Alternate ID for the sensor name {str} – Name for the sensor sensor_type_id {str} – ID of the respective sensor type

Returns: Response – Response object

delete_sensor(*sensor_id: str*) → sap_iot_services_sdk.iot_service.Response

The endpoint is used to delete the sensor associated to the given id.

Arguments: sensor_id {str} – Unique identifier of a sensor

Returns: Response – Response object

get_sensor(*sensor_id: str*) → sap_iot_services_sdk.iot_service.Response

The endpoint returns the sensor associated to the given id.

Arguments: sensor_id {str} – Unique identifier of a sensor

Returns: Response – Response object

get_sensors(*filters=None*, *orderby=None*, *asc=True*, *skip=None*, *top=None*) → sap_iot_services_sdk.iot_service.Response

The endpoint returns a list of sensors.

Keyword Arguments: filters {list} – This parameter allows clients to filter the collection for specific attributes. It is possible to filter by 'id', 'deviceId', 'name', and 'alternateId'. The filters must be provided as a list of strings, e.q. ["name eq 'my-name'", "id eq '111'"]. (default: {None}) orderby {str} – The attribute to order by. (default: {None}) asc {bool} – Only considered if orderby is not none. Defines if the values should be ordered asc or desc. (default: {True}) skip {int} – This parameter specifies the number of items in the queried collection which will be skipped and therefore included in the result set. (default: {None}) top {int} – This parameter restricts the maximum number of items which will be returned by the request. (default: {None})

Returns: Response – Response object

update_sensor(*sensor_id: str*, *name: str*, *sensor_type_id: str*) → sap_iot_services_sdk.iot_service.Response

This endpoint is used to update a sensor associated to the given id with details specified in the request body.

Arguments: sensor_id {str} – Unique identifier of a sensor name {str} – Name of the sensor sensor_type_id {str} – Respective sensor type ID

Returns: Response – [description]

1.1.11 sap_iot_services_sdk.sensor_type module

class sap_iot_services_sdk.sensor_type.**SensorTypeService**(*instance*, *user*, *password*)

Bases: *sap_iot_services_sdk.iot_service.IoTService*

add_capability(*sensor_type_id: str*, *capability_id: str*, *capability_type: str*) → sap_iot_services_sdk.iot_service.Response

This endpoint is used to add a capability. Note that it is not supported to add a capability to a sensor type which is already associated with a sensor.

Arguments: sensor_type_id {str} – Unique identifier of a sensorType capability_id {str} – ID of the capability that will be added capability_type {str} – Type of the capability that will be added. Can be 'measure' or 'command'

Returns: Response – Response object

create_sensor_type(*alternate_id: str*, *name: str*, *capabilities: list*) → sap_iot_services_sdk.iot_service.Response

This endpoint is used to create a sensor type.

Arguments: alternate_id {str} – Alternate ID of the sensor type name {str} – Name of the sensor type capabilities {list} – List of dicts each containing key-value pairs for 'id' and 'type'

Returns: Response – Response object

delete_sensor_type(*sensor_type_id: str*) → sap_iot_services_sdk.iot_service.Response

The endpoint is used to delete the sensor type associated to the given id.

Arguments: sensor_type_id {str} – Unique identifier of a sensor type

Returns: Response – Response object

get_sensor_type(*sensor_type_id: str*) → sap_iot_services_sdk.iot_service.Response

The endpoint returns the sensor type associated to the given id.

Arguments: sensor_type_id {str} – Unique identifier of a sensor type

Returns: Response – Response object

get_sensor_types(*filters=None*, *orderby=None*, *asc=True*, *skip=None*, *top=None*) → sap_iot_services_sdk.iot_service.Response

The endpoint returns a list of sensor types.

Keyword Arguments: filters {list} – This parameter allows clients to filter the collection for attributes of a sensorType. The filters must be provided as a list of strings, e.g. ["name eq 'my-name'", "id eq '111'"] (default: {None}) orderby {str} – The attribute to order by. (default: {None}) asc {bool} – Only considered if orderby is not none. Defines if the values should be ordered asc or desc. (default: {True}) skip {int} – This parameter specifies the number of items in the queried collection which will be skipped and therefore not included in the result set. (default: {None}) top {int} – This parameter restricts the maximum number of items which will be returned by the request. (default: {None})

Returns: Response – Response object

remove_capability(*sensor_type_id: str*, *capability_id: str*) → sap_iot_services_sdk.iot_service.Response

The endpoint is used to remove the capability associated to the given id. Note that it is not supported to delete a capability in a sensor type which is already associated with a sensor.

Arguments: sensor_type_id {str} – Unique identifier of a sensorType capability_id {str} – Unique identifier of a capability

Returns: Response – Response object

update_capability(*sensor_type_id: str*, *capability_id: str*, *capability_type: str*) → sap_iot_services_sdk.iot_service.Response
This endpoint is used to update the capability associated to the given id with details specified in the request body. Note that it is not supported to modify the type of a capability in a sensor type if it is already associated with a sensor.

Arguments: sensor_type_id {str} – Unique identifier of a sensorType capability_id {str} – ID of the capability that will be updated capability_type {str} – Type of the capability that will be added. Can be 'measure' or 'command'

Returns: Response – Response object

update_sensor_type(*sensor_type_id: str*, *alternate_id: str*, *name: str*) → sap_iot_services_sdk.iot_service.Response
This endpoint is used to update the sensor type associated to the given id with details specified in the request body. To update capabilities, use the respective API.

Arguments: sensor_type_id {str} – Unique identifier of a sensor type alternate_id {str} – Alternate identifier of the sensor type name {str} – Name of the sensor type

Returns: Response – Response object

1.1.12 sap_iot_services_sdk.tenant module

class sap_iot_services_sdk.tenant.**TenantService**(*instance*, *user*, *password*)
Bases: *sap_iot_services_sdk.iot_service.IoTService*

add_custom_property(*tenant_id: str*, *key: str*, *value: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint is used to add a custom property to the tenant associated to the given id.

Arguments: tenant_id {str} – Unique identifier of a tenant key {str} – Key of the custom property value {str} – Value of the custom property

Returns: Response – Response object

add_user(*tenant_id: str*, *role: str*, *user_id: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint is used to add the user specified in the request body to the tenant associated to the given id.

Arguments: tenant_id {str} – Unique identifier of a tenant role {str} – Role of the user user_id {str} – ID of the user

Returns: Response – Response object

create_tenant(*name: str*, *custom_properties=[]*) → sap_iot_services_sdk.iot_service.Response
The endpoint is used to create a tenant.

Arguments: name {str} – Name of the tenant custom_properties {list} – Custom properties of the tenant as a list of dicts, each with the key-value pairs 'key' and 'value'

Returns: Response – Response object

delete_custom_property(*tenant_id: str*, *custom_property_key: str*) → sap_iot_services_sdk.iot_service.Response
This endpoint is used to delete a custom property from the tenant associated to the given id.

Arguments: tenant_id {str} – Unique identifier of a tenant custom_property_key {str} – Key of the custom property

Returns: Response – Response object

delete_tenant(*tenant_id: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint is used to delete the tenant associated to the given id.

Arguments: tenant_id {str} – Unique identifier of a tenant

Returns: Response – Response object

delete_user(*tenant_id: str*, *user_id: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint is used to remove the user from the tenant associated to the given id.

Arguments: tenant_id {str} – Unique identifier of a tenant user_id {str} – Unique identifier of a user

Returns: Response – Response object

get_tenant(*tenant_id: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint returns the tenant associated to the given id.

Arguments: tenant_id {str} – Unique identifier of a tenant

Returns: Response – Response object

get_tenants(*filters=None*, *orderby=None*, *asc=True*, *skip=None*, *top=None*) → sap_iot_services_sdk.iot_service.Response
The endpoint returns a list of tenants.

Keyword Arguments: filters {list} – This parameter allows clients to filter the collection for attributes of a tenant. The filters must be provided as a list of strings, e.q. [“name eq ‘my-name’”, “id eq ‘111’”]. (default: {None}) orderby {str} – The attribute to order by. (default: {None}) asc {bool} – Only considered if orderby is not none. Defines if the values should be ordered asc or desc. (default: {True}) skip {int} – This parameter specifies the number of items in the queried collection which will be skipped and therefore not included in the result set. (default: {None}) top {int} – This parameter restricts the maximum number of items which will be returned by the request. (default: {None})

Returns: Response – Response object

get_trusted_ca_certificates(*tenant_id: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint is used to download tenant specific trusted CA certificates for authentication.

Arguments: tenant_id {str} – Unique identifier of a tenant

Returns: Response – Response object

get_user(*tenant_id: str*, *user_id: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint is used to return the tenant user associated to the given id.

Arguments: tenant_id {str} – Unique identifier of a tenant user_id {str} – Unique identifier of a user

Returns: Response – Response object

get_users(*tenant_id: str*, *orderby=None*, *asc=True*, *skip=None*, *top=None*) → sap_iot_services_sdk.iot_service.Response
The endpoint returns a list of users assigned to the tenant associated to the given id.

Arguments: tenant_id {str} – Unique identifier of a tenant

Keyword Arguments: orderby {str} – The attribute to order by. (default: {None}) asc {bool} – Only considered if orderby is not none. Defines if the values should be ordered asc or desc. (default: {True}) skip {int} – This parameter specifies the number of items in the queried collection which will be skipped and therefore not included in the result set. (default: {None}) top {int} – This parameter restricts the maximum number of items which will be returned by the request. (default: {None})

Returns: Response – Response object

update_custom_property(*tenant_id: str*, *key: str*, *value: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint is used to update a custom property of the tenant associated to the given id. The 'key' attribute cannot be modified.

Arguments: tenant_id {str} – Unique identifier of a tenant key {str} – Key of the custom property value {str} – Value of the custom property

Returns: Response – Response object

update_tenant(*tenant_id: str*, *name: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint is used to update the tenant associated to the given id with details specified in the request body. To update custom properties or users, use the respective APIs.

Arguments: tenant_id {str} – Unique identifier of a tenant name {str} – Name of the tenant

Returns: Response – Response object

update_user(*tenant_id: str*, *user_id: str*, *role: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint is used to update the tenant user associated to the given id with details specified in the request body.

Arguments: tenant_id {str} – Unique identifier of a tenant user_id {str} – Unique identifier of a user role {str} – Role of the user

Returns: Response – Response object

1.1.13 sap_iot_services_sdk.user module

class sap_iot_services_sdk.user.**UserService**(*instance*, *user*, *password*)
Bases: *sap_iot_services_sdk.iot_service.IoTService*

add_custom_property(*user_id: str*, *key: str*, *value: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint is used to add a custom property to the user associated to the given id.

Arguments: user_id {str} – Unique identifier of a user key {str} – Key of the custom property that will be created value {str} – Value of the custom property that will be created

Returns: Response – Response object

add_role(*user_id: str*, *role: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint is used to add a role to user associated to the given id. The role is valid across the instance.

Arguments: user_id {str} – Unique identifier of a user role {str} – Specification of the role that will be created

Returns: Response – Response object

create_user(*name: str*, *password: str*, *custom_properties=[]*) → sap_iot_services_sdk.iot_service.Response
The endpoint is used to create a user. Note: This function only supports basic authentication method.

Arguments: name {str} – Name of the user password {str} – Password of the user custom_properties {list} – Custom properties for the user given as list of dicts each with the key-value pairs 'key' and 'value'

Returns: Response – Response object

delete_custom_property(*user_id: str*, *key: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint is used to delete a custom property from the user associated to the given id.

Arguments: user_id {str} – Unique identifier of a user key {str} – Key of the custom property

Returns: Response – Response object

delete_role(*user_id: str*, *role: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint is used to delete the role from user associated to the given id. The role is valid across the instance.

Arguments: user_id {str} – Unique identifier of a user role {str} – Name of the role

Returns: Response – Response object

delete_user(*user_id: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint is used to delete the user associated to the given id.

Arguments: user_id {str} – Unique identifier of a user

Returns: Response – Response object

get_p12_certificate(*user_id: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint is used to download a user specific p12 file for authentication.

Arguments: user_id {str} – Unique identifier of a user

Returns: Response – Response object

get_pem_certificate(*user_id: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint is used to download a user specific private key and certificate in PEM format for authentication.

Arguments: user_id {str} – Unique identifier of a user

Returns: Response – Response object

get_user(*user_id: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint returns the user associated to the given id.

Arguments: user_id {str} – Unique identifier of a user

Returns: Response – Response object

get_users(*filters=None*, *orderby=None*, *asc=True*, *skip=None*, *top=None*) → sap_iot_services_sdk.iot_service.Response
The endpoint returns a list of users.

Keyword Arguments: filters {list} – This parameter allows clients to filter the collection for attributes of a user. The filters must be provided as a list of strings, e.g. [“name eq ‘my-name’”, “id eq ‘111’”]. (default: {None}) orderby {str} – The attribute to order by. (default: {None}) asc {bool} – Only considered if orderby is not none. Defines if the values should be ordered asc or desc. (default: {True}) skip {int} – This parameter specifies the number of items in the queried collection which will be skipped and therefore not included in the result (default: {None}) top {int} – This parameter restricts the maximum number of items which will be returned by the request (default: {None})

Returns: Response – Response object

update_custom_property(*user_id: str*, *key: str*, *value: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint is used to update a custom property of the user associated to the given id. The ‘key’ attribute cannot be modified.

Arguments: user_id {str} – Unique identifier of a user key {str} – Key of the custom property value {str} – Value of the custom property

Returns: Response – Response object

update_password(*user_id: str*, *password: str*) → sap_iot_services_sdk.iot_service.Response
The endpoint is used to update the password of the user associated to the given id.

Arguments: user_id {str} – Unique identifier of a user password {str} – Password that will be updated

Returns: Response – Response object

1.1.14 sap_iot_services_sdk.utils module

sap_iot_services_sdk.utils.**build_query**(*filters=None*, *orderby=None*, *asc=True*, *skip=None*, *top=None*) → str

Builds query string

Keyword Arguments: filters {list} – The filters must be provided as a list of strings, e.q. ["name eq 'myname'", "id eq '111'"]. (default: {None}) orderby {str} – The attribute to order by. (default: {None}) asc {bool} – Only considered if orderby is not none. Defines if the values should be ordered ascending or descending. (default: {True}) skip {int} – This parameter specifies the number of items in the queried collection which will be skipped and therefore included in the result set. (default: {None}) top {int} – This parameter restricts the maximum number of items which will be returned by the request. [description] (default: {None})

Returns: str – Query string

sap_iot_services_sdk.utils.**current_milli_time**()

1.1.15 sap_iot_services_sdk.vendor module

class sap_iot_services_sdk.vendor.**VendorService**(*instance*, *user*, *password*)

Bases: *sap_iot_services_sdk.iot_service.IoTService*

create_vendor(*vendor_id: str*) → sap_iot_services_sdk.iot_service.Response

The endpoint is used to create a new vendor. The new vendor is visible for all tenants of the instance.

Arguments: vendor_id {str} – ID of the vendor that will be created

Returns: Response – Response object

delete_vendor(*vendor_id: str*) → sap_iot_services_sdk.iot_service.Response

The endpoint is used to delete a vendor. Deleting is possible only if no tenant on the instance is using it anymore.

Arguments: vendor_id {str} – Unique identifier of a vendor

Returns: Response – Response object

get_vendors(*skip=None*, *top=None*) → sap_iot_services_sdk.iot_service.Response

The endpoint returns a list of vendors, shared among all tenants of the instance.

Keyword Arguments: skip {int} – This parameter specifies the number of items in the queried collection which will be skipped and therefore included in the result (default: {None}) top {int} – This parameter restricts the maximum number of items which will be returned by the request (default: {None})

Returns: Response – Response object

1.1.16 Module contents

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

S

INDEX