Meta-Data API Specification

Access to meta-data stored in accordance with the Standard Information Model for Meta-Data of SWITCH-ON (SIM) is provided through the SIP Expert Catalogue Management APIs and the Published Catalogue Access APIs. This document specifies the SWITCH-ON Meta-Data REST HTTP API, an implementation of the SIP Expert Catalogue Management API.

Figure 1 gives an overview on the different types of APIs for Expert Catalogue Management. The purpose of the different APIs is explained in the following subsections.

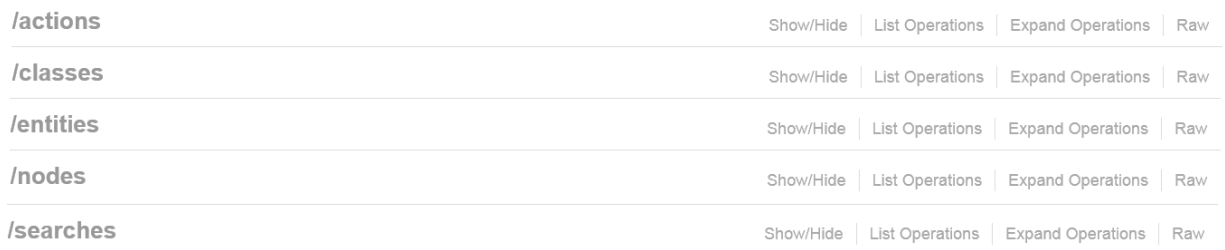

Figure 1: SIP Expert Catalogue Management REST HTTP APIs Overview

1 Actions

Actions are predefined tasks that are executed within the execution context of the SIP Backend, e.g. on the server or the virtual machine the SIP server components are deployed. Actions are implemented as Java classes in the cids Domain Server component and thus can perform long lasting and complex operations. Actions provide a simple mechanism to delegate business logic from the client to the server and can for example be used to directly call a local transformation script or initiate a data transformation on a remote server (e.g. WPS). Figure 2 gives an overview about the specification of the Actions API.

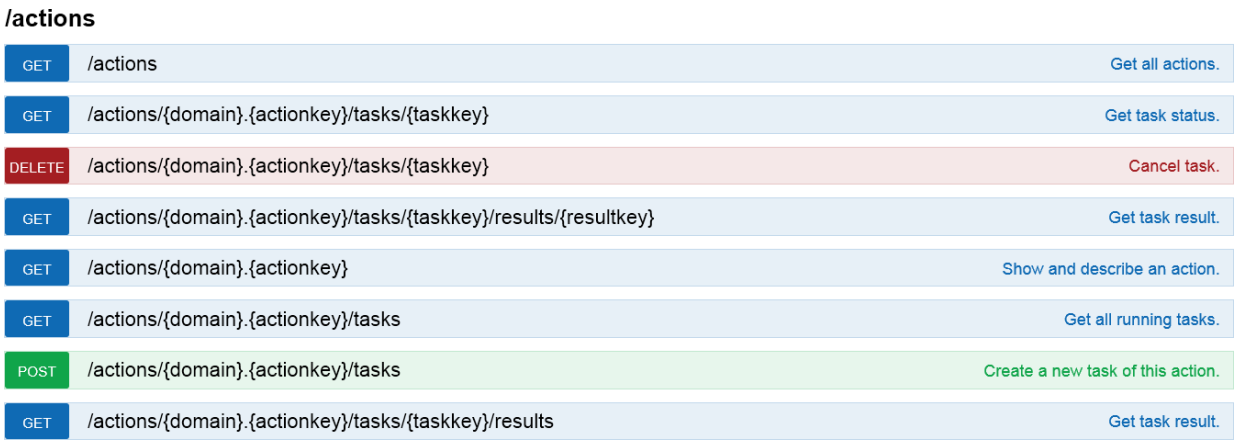

Figure 2: Actions API Specification

2 Classes

The Classes API provides information on the entity types (classes) supported by the Expert Catalogue APIs. The entity types are defined by relational model of the SIM. The Classes API delivers a JSON representation of the respective class. Instances of those classes can be retrieved via the Entities API. Figure 3 gives an overview about the specification of the Classes API.

/classes

GET	/classes/{domain}.{classkey}	Get a certain class.
GET	/classes	Get all classes.
GET	/classes/{domain}.{classkey}/{attributekey}	Get a certain class.

Figure 3: Classes API Specification

A complete example of the JSON Class Definition of the “Representation” Class is shown below:


```
{
            "$self": "/SWITCHON.REPRESENTATION",
            "configuration": {
                "name": "representation",
                "policy": "STANDARD",
                "attributePolicy": "STANDARD",
                "pK Field": "id"
            },
            "attributes": {
                "id": {
                    "$self": "/SWITCHON.REPRESENTATION/id",
                    "name": "id",
                    "position": 0,
                    "javaclassname": "java.lang.Integer"
                },
                "name": {
                    "$self": "/SWITCHON.REPRESENTATION/name",
                    "visible": true,
                    "name": "name",
                    "position": 1,
                    "javaclassname": "java.lang.String"
                },
                "description": {
                    "$self": "/SWITCHON.REPRESENTATION/description",
                    "visible": true,
                    "optional": true,
                    "name": "description",
                    "position": 2,
                    "javaclassname": "java.lang.String"
                },
                "tags": {
                    "$self": "/SWITCHON.REPRESENTATION/tags",
                    "visible": true,
                    "array": true,
                    "optional": true,
                    "name": "tags",
                    "position": 3,
                    "referenceType": "/SWITCHON.TAG"
                },
                "contenttype": {
                    "$self": "/SWITCHON.REPRESENTATION/contenttype",
                    "visible": true,
```

```
                    "optional": true,
                    "name": "contenttype",
                    "position": 4,
                    "javaclassname": "java.lang.String"
                },
                "contentlocation": {
                    "$self": "/SWITCHON.REPRESENTATION/contentlocation",
                    "visible": true,
                    "optional": true,
                    "name": "contentlocation",
                    "position": 5,
                    "javaclassname": "java.lang.String"
                },
                "content": {
                    "$self": "/SWITCHON.REPRESENTATION/content",
                    "visible": true,
                    "optional": true,
                    "name": "content",
                    "position": 6,
                    "javaclassname": "java.lang.String"
                }
            }
        }
```


The most interesting part of this JSON Class definition shown in the example above is the list of attributes of the class. The type of the attribute is determined by the “referenceType” and by “javaclassname” attributes as shown in Figure 4.

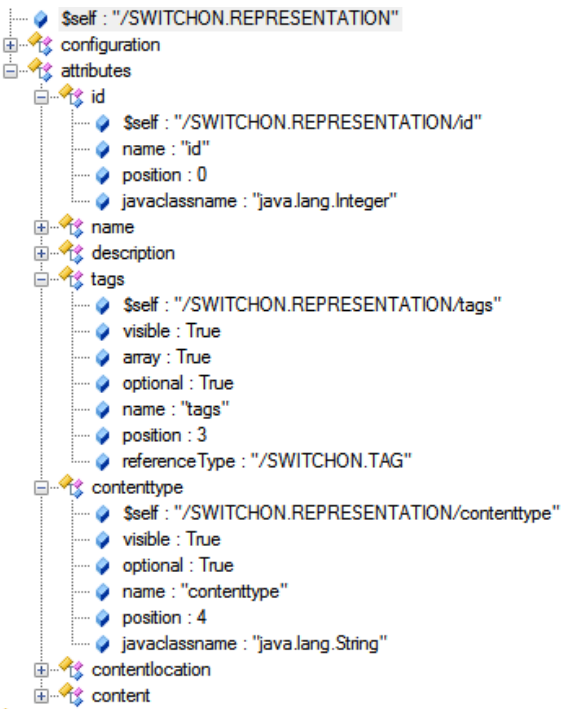

Figure 4: JSON Structure of the “Representation” Class

As comparison, the Relational Structure of the “Representation” Class as defined in the relational model of the SIM is shown in Figure 5 below.

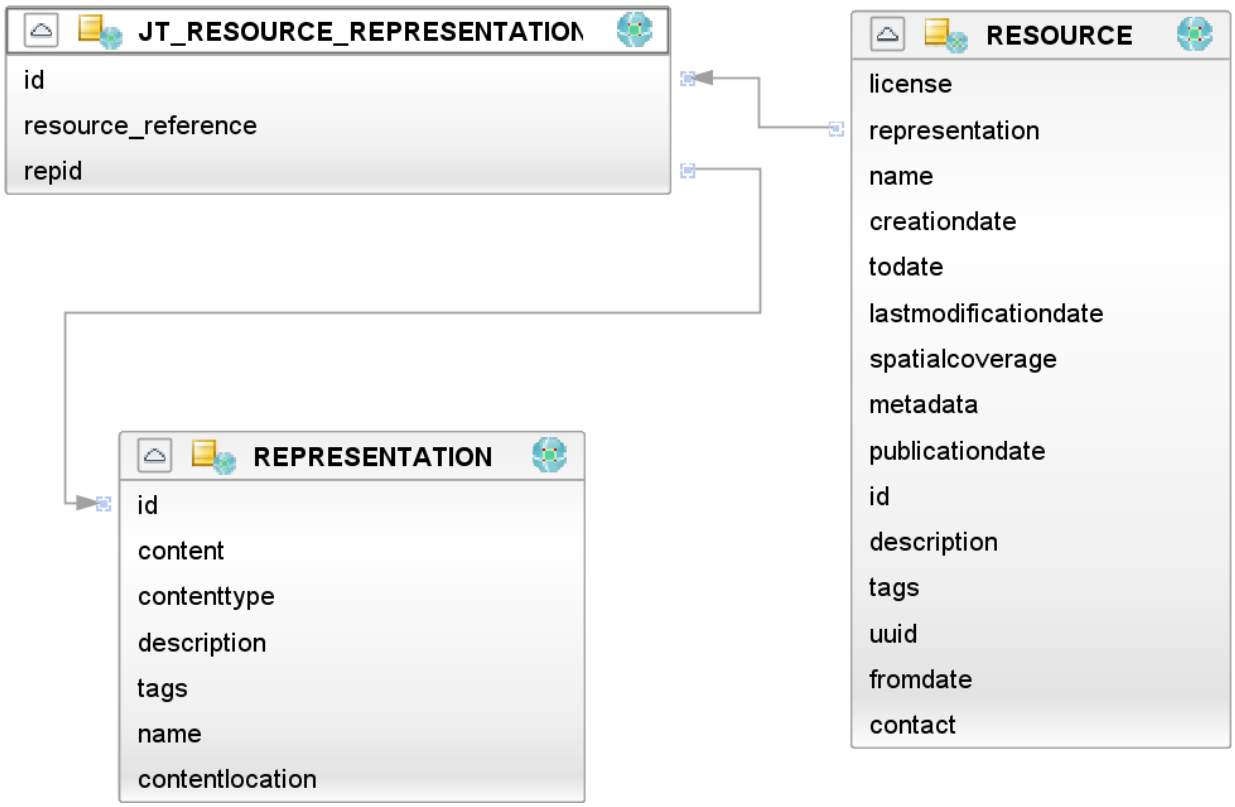

Figure 5: Relational Structure of the “Representation” Class

2.1.1.1 Entities

The Entities API is the most important API. It provides access to the actual object instances stored in the Meta-Data Repository. Thereby, the API offers basic CRUD functionality as shown in Figure 4.

/entities

GET	/{domain}.{class}/{objectid}	Get a certain object by its id.
PUT	/{domain}.{class}/{objectid}	Update or creates an object.
DELETE	/{domain}.{class}/{objectid}	Delete a certain object.
POST	/{domain}.{class}	Create a new object.
GET	/{domain}.{class}	Get all objects of a certain class.
GET	/{domain}.{class}/emptyInstance	Get an empty instance of a certain class.

Figure 6: Entities API Specification

The JSON Class definitions of all supported entities can be obtained via the Classes API that is described in section 2. Below is an example of the structure (Figure 7) and the JSON representation of a resource.

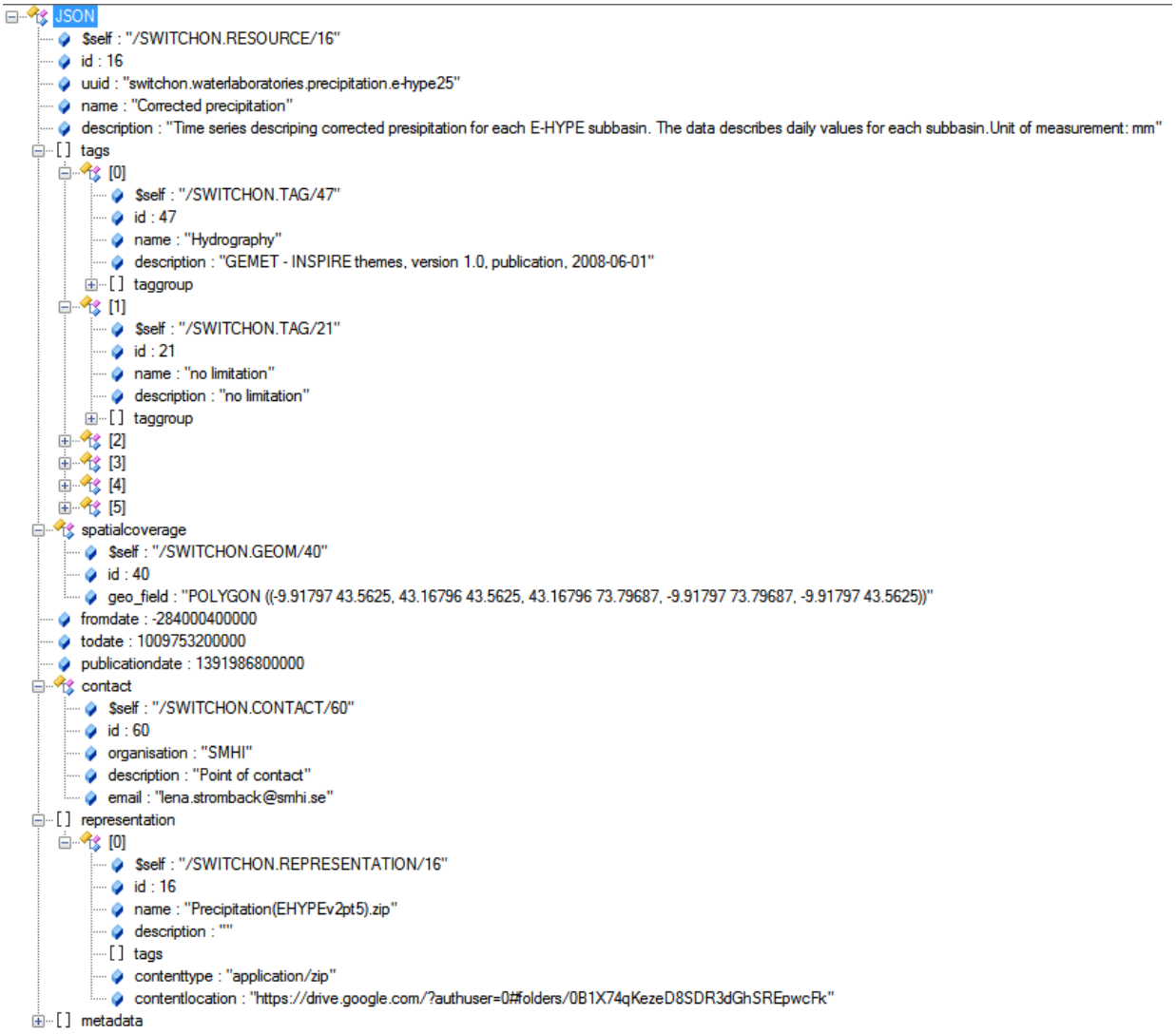

Figure 7: JSON Structure of a “Resource” Object


```
{
    "$self": "/SWITCHON.RESOURCE/16",
    "id": 16,
    "uuid": "switchon.waterlaboratories.precipitation.e-hype25",
    "name": "Corrected precipitation",
    "description": "Time series descriping corrected presipitation for each E-HYPE
subbasin. The data describes daily values for each subbasin.Unit of measurement: mm",
    "tags": [
        {
            "$ref": "/SWITCHON.TAG/47"
        },
        {
            "$ref": "/SWITCHON.TAG/21"
        },
        {
            "$ref": "/SWITCHON.TAG/12"
        },
        {
            "$ref": "/SWITCHON.TAG/68"
```

```
            },
            {
                "$ref": "/SWITCHON.TAG/95"
            },
            {
                "$ref": "/SWITCHON.TAG/118"
            }
        ],
        "spatialcoverage": {
            "$ref": "/SWITCHON.GEOM/40"
        },
        "fromdate": -284000400000,
        "todate": 1009753200000,
        "publicationdate": 1391986800000,
        "contact": {
            "$ref": "/SWITCHON.CONTACT/60"
        },
        "representation": [
            {
                "$ref": "/SWITCHON.REPRESENTATION/16"
            }
        ],
        "metadata": [
            {
                "$ref": "/SWITCHON.METADATA/40"
            }
        ]
    }
```


As can be seen in the JSON document above, properties referring to complex objects like tag, representation, metadata, etc. are only provided by reference. The behaviour whether JSON object obtained via the Entities API shall be expanded (contain all object values instead of object references) or not is controllable by “expand” and “level” parameters of API. Figure 8 gives an overview on all parameters of the GET Entities operation of the Entities API.

Parameter	Description	Parameter Type	Data Type
domain	**identifier (domainname) of the domain.**	path	string
class	**identifier (classkey) of the class.**	path	string
objectid	**identifier (objectkey) of the object.**	path	string
version	version of the object, 'current' version when not submitted	query	string
role	role of the user, 'default' role when not submitted	query	string
expand	a list of properties in the resulting objects that should be expanded	query	string
level	the level of expansion	query	string
fields	the fields of the resulting object, all fields when not submitted	query	string
profile	profile of the object, 'full' profile when not submitted and no fields are present	query	string
omitNullValues	Omit properties that have 'null' as value	query	boolean
deduplicate	if you don't want already expanded properties to be expanded again, set this parameter to true	query	boolean
Authorization	Basic Auth Authorization String	header	string

Figure 8: Parameters of the GET Entities Operation of the Entities API

2.1.1.2 Nodes

The Nodes API can be used to obtain a catalogue structure whereby the catalogue entries (nodes) can point to entities. Figure 9 gives an overview about the specification of the Nodes API.

/nodes

GET	/nodes/{domain}.{nodekey}	Get a certain node.
POST	/nodes/{domain}/children	Get the children of a certain node from the dynamicchildren section of the node.
GET	/nodes/{domain}.{nodekey}/children	Get the children of a certain node.
GET	/nodes	Get all Rootnodes.

Figure 9: Nodes API Specification

2.1.1.3 Searches

The Search API provides a convenient mechanism to search for specific entities. Similar to the Actions API (section 1), a custom search is implemented as a Java Class in SIP backend. Such a custom search can be parameterised and executed via the Search API. Search can be performed on relational (object entities) as well as on structured (value of “content” properties) meta-data. Apart from the default searches (free keywords, geospatial extent), other types of user defined searches, e.g. for aggregated mean values, can be implemented as required. Figure 9 gives an overview about the specification of the Searches API.

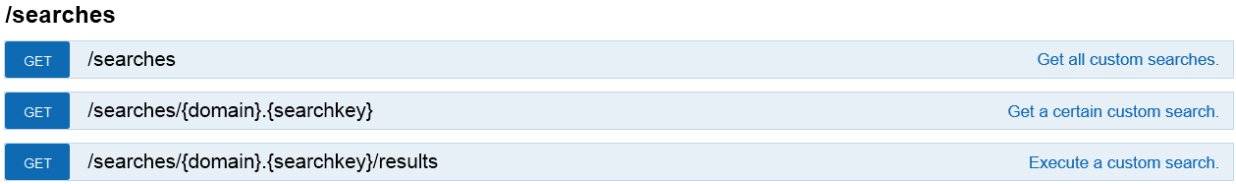

Figure 10: Searches API Specification