24 August 2017
EMA/241514/2016

Substance, Product, Organisation, Referentials (SPOR)

SPOR API Specification
Version 1.11

Table of Contents

1. This document Purpose

The purpose of this document is to present the description of the intended Application Programming Interface (API) for SPOR.

2. Context

The requirements for implementing the API are expressed in the following documents:-

1. "Referentials Management System - Business Requirements". Copy available on request.
2. "Organisation Management System - Business Requirements". Copy available on request.

The specifications for the SPOR API are a first step towards implementing the system supporting the API.

3. Scope

The scope is defined as the new SPOR API only.

4. Introduction

4.1. *Definitions*

An API can be defined in various ways; the definitions below form a good start for this:

1. "It is a set of **routines**, **protocols**, and tools for building software applications."
2. "It expresses a **software component** in terms of its **operations**, **inputs**, **outputs**, and **underlying types**."

These definitions were taken from Wikipedia http://en.wikipedia.org/wiki/Application_programming_interface.

If we take the second definition we can expand on the terms used in order to make it more particular to the problem at hand:

Element	Description
Software component	System hosted at EMA
Operations	Create, read, update, and delete
Inputs	Search terms, documents, metadata attributes
Outputs	Documents, metadata attributes
Underlying types	Lists, Terms, Translations, Change Requests, Documents, user defined Tags, Subscriptions, Organisations, Locations.

4.2. *What the API is not*

It should also be noted that there are misconceptions and fallacies about an API, so an API is not:

- A software component that you install on a computer

- A process that automates human activities

- An end-to-end system between the NCAs and EMA

4.3. *Flexibility and constraints*

The definition of the API must be such that it addresses concerns of all the stakeholders as opposed to a small number of stakeholders. This is the trade-off between genericity and specificity, and in order to be able to specify an API, the following points must be taken into account:

1. The API must meet the requirements.

2. The stakeholders have different needs as they have different business processes, IT infrastructures and budgets.

3. ***There will only be one API***.

4. It is important to draw the line between generic features, usable by all stakeholders, vs. specific features, usable just by 1 or a few stakeholders only.

5. Features that appear to be specific to 1 or a few stakeholder must be implemented on the client side and are out of the scope of the API definition.

5. Specification

The specification for the API is based on the RESTful style API. The same style of API was adopted for the PSUR API and for the Common Repository; it will also be the style for the other components of SPOR. This is for the sake of consistency but also for its clarity, ease of use with minimal infrastructure and its clear separation between resources and the operations that can be applied on those resources.

5.1. *Authentication and authorisation*

All SPOR services require authentication, unless explicitly stated otherwise in the service definition (see *6. REST Services* for details).

Authentication will be HTTP Basic Authentication over SSL.

Authorisation will be RBAC with roles assigned during user registration.

5.2. *Service versioning*

Each endpoint URL will be prefixed with /v{version}, where version is the service version number. The service URL is case sensitive.
i.e. `GET /v1/lists`

5.3. *XML schemas versioning*

XML schemas representing resources will be versioned using a 'Major.Minor' number scheme:
- Production XML schema versions will have a Minor increment (1.1, 1.2 …) when a change will not cause the previous version of the schema to be invalidated, this ensures backward compatibility.
- Production XML schema versions will have a Major increment (2.0, 3.0 …) when a change will cause the previous version to fail schema validation, this means that there is no backward compatibility and is used for fundamental structural changes. Major resources conforming to different major XML Schema versions will be exposed using different service versions (see Section 5.2.)

XML document instances representing resources will include a mandatory schema-version attribute to indicate XML schema version they conform to (see *5.9. Metadata*).

5.4. *Errors*

The API will make use of a number of HTTP status codes where applicable accompanied by more user friendly messages in the payload. The complete error catalogue for the API will be defined separately.

5.5. *HTTP methods*

The API makes use of the standard HTTP methods such as `GET` and `POST` to read and write respectively from and to RMS.

5.6. *Resources and representations*

For an API with a RESTful style, a resource is anything that can be identified and manipulated by a set of HTTP verbs. The list of resources for this API is:

- lists
- terms
- translations
- change-requests
- documents
- tags
- subscriptions
- organisations
- locations

Those resources can be expressed using various representations depending on the need of the user and the nature of the resource. In the context of this API, the representations for resources are, according to their media type defined by IANA:

- **application/xml** - used to indicate that the resource is represented by xml data.
- **application/json** - used to indicate that data is represented using the JavaScript Object Notation. It is a programming language independent data format which expresses information in the form of key-value pairs.

Additionally some resource would be represented as:

- **application/xml;application/zip** – used to indicate that the resource is represented as a zipped file of xml data
- **application/zip** – used to indicate that the resource is represented as a zipped file of csv data

The default resource representation is application/xml and it is the client's responsibility to indicate if application/json is required. For this purpose, the client must make use of the `Accept` header field in the HTTP request.

If the representation requested is not supported by the server then an appropriate error is returned by the server to the client (see section 5.8. HTTP status codes).

Examples:

- Request for a resource representation in xml format: (may be omitted as default)
  `Accept: application/xml`

- Request for a resource representation in JSON format:
  `Accept: application/json`

See *6. REST Services* for the Accept Headers supported by each service.

5.6.1. Encoding

All SPOR resources are UTF-8 encoded, unless explicitly stated otherwise in the service definition (see *6. REST Services* for details).

5.7. *Request parameters*

For this API specification, the parameters for a request can be provided in number of ways to the server:

- **Path**: `/v1/lists/{parameter}`
  where the single parameter is the List identifier

- **Query string**: /v1/lists/10001?pagesize=10&page=1
  where the List identifier is 10001 followed by browser pagination information

- **Header of the request**: `Accept: application/json` which is used by the server to determine which representation will be return to the client.

All of the above can be used jointly in the same request to the server. The service URL is case-sensitive.

5.7.1. Parameter characteristics/behaviours

The following characteristics/behaviours apply to all path and query parameters in SPOR REST Services, unless explicitly stated otherwise in the service definitions (see *6. REST Services* for details).

- All path parameters are mandatory.

- Query parameters can be mandatory or optional (see MD column in *6. REST Services* for details).

- Search Service query parameters operate on a restrictive basis. That is, they are used to restrict/filter resources that are returned by the service. If query parameters are not provided then all available resources are returned – subject to the application of default values (see Default column in *6. REST Services* for details).

- Some parameters support wildcarding (see WC column in *6. REST Services* for details). If a "begins with search" is required then the supplied parameter value must be suffixed with * (e.g. london*). If a "contains search" search is required then the supplied parameter value must be prefixed and suffixed with * (e.g. *london*).

- All query parameter names (path and query) are case-sensitive.

- All query parameter values are case-insensitive. That is, passing ?name=john will return the same results as ?name=John.

- Query parameters are treated as AND conditions. That is, a resource must match all provided parameters in order for it to be returned. For example, the below query will only return resources that have a name equal to acme AND a status equal to ACTIVE:

?name=acme&status=ACTIVE

Sample data returned:

name=acme, status=ACTIVE, city=London
name=acme, status=ACTIVE, city=Paris

- Some parameters support multi-value tilde (~) separated values. If multi-values are provided for a parameter they are treated as OR conditions. That is, the resource only needs to match one of the provided multi-values in order to be treated as a match. For example, the below query will only return resources that have a name equal to acme AND a status equal to either ACTIVE OR INACTIVE:

?name=acme&status=ACTIVE~INACTIVE

Sample data returned:

name=acme, status=ACTIVE, city=London
name=acme, status=ACTIVE, city=Paris
name=acme, status=INACTIVE, city=Leeds

See 'MV' column in *6. REST Services* for details of parameters that accept multi-values.

5.8. *HTTP status codes*

The API will make use of a number of HTTP status codes where applicable accompanied by more user friendly messages in the payload. The complete error catalogue for the API will be defined separately.

Code	Name	Description
200	OK	The request has succeeded.
201	Created	The request has been fulfilled and resulted in a new resource being created.
400	Bad Request	The request could not be understood by the server due to malformed syntax. The client SHOULD NOT repeat the request without modifications.
401	Unauthorized	The request requires user authentication. The response MUST include a WWW-Authenticate header field containing a challenge applicable to the requested resource.
403	Forbidden	The server understood the request, but is refusing to fulfil it. Authorization will not help and the request SHOULD NOT be repeated.
404	Not Found	The server has not found anything matching the Request-URI.
405	Method Not Allowed	The client tried to use a method on a resource which is not allowed by the server.
406	Not Acceptable	The resource identified by the request is only capable of generating response entities which have content characteristics not acceptable according to the accept headers sent in the request.
413	Request Entity Too Large	The server is refusing to process a request because the request entity is larger than the server is willing or able to process.
415	Unsupported	The server is refusing to service the request because the entity of the

	Media Type	request is in a format not supported by the requested resource for the requested method.
500	Server Error	The server encountered an unexpected condition which prevented it from fulfilling the request.

5.9. *Metadata*

The following metadata is returned by each service:

Attribute	Description	Mandatory
query-link	Link to a service call/query that generated this response	No
query-timestamp	Time stamp of a service call/query that generated this response	No
total-items	Number of records generated in this response	No
next-page	Link to a next page of this service response (if pagination used)	No
previous-page	Link to a previous page of this service response (if pagination used)	No
schema-version	Version of the XML Schema to which this element conforms	Yes

5.10. *Standards*

- All dates/times returned by resources are expressed in YYYY-MM-DDThh:mm:ssZ format (XSD 1.0, timezone UTC).

- The API supports a maximum URL size of 2048 characters – including the hostname, resource path and query parameters. This limit is subject to ongoing technical investigations.

6. REST Services

Resource Summary

A non-exhaustive list of resources for this API is:

Resource	Description	Resource Details	SPOR Domain[1]
list	List is a collection of related terms, e.g. countries list, units of measure list. List has additional properties like description, list owner and other.	*7.1. List*	**R**
term	Term is a word or phrase used to describe a thing or to express a concept. All RMS terms have a name in English and possible translations of this name to other languages. Terms have other attributes and names some of which are also translatable into other languages. Related terms are grouped into	*7.2. Term*	**R**

---

[1] O=OMS specific, R=RMS specific

	lists.		
translation	Translation is a representation of the term's name or other translatable names in other languages than English.	*7.3. Translation*	**R**
change request rms	Change request provides details and additional justification and information of requested change to create and update terms and lists.	*7.4. Change Request RMS*	**R**
document	Documents provide additional information for lists and change requests. Also provide general information and help.	*7.5. Document*	**RO**
search query	Search query stores parameters of a user defined query that can be reused in the future. It does not store query results.	*7.6. Search Query*	**R**
subscription	Subscription is a user's arrangement to receive notifications on modifications of lists.	*7.7. Subscription*	**R**
tag	Tag is a user created label that can be attached to terms.	*7.9. Tag*	**R**
Preferred name	Preferred name is a one of the term's names selected by a user as her/his preferred name. A term must have an English name, may have at most one name in any other language, may have at most one short name in any language and zero or more other names (aliases) in any language. User can select one of these names as her/his preferred name.		**R**
organisation	Organisation is a representation of a legal entity (e.g. a business, government department, regulatory body). Organisations may have a name, acronym, alternative names and classifications (e.g. organisation type). They also have optional communication details such as phone numbers, email addresses and postal addresses. Organisations have a collection of locations (addresses) associated with them. An active organisation must have at least one active location associated with it.	*7.10. Organisation*	**O**
location	Location is a representation of an address/site within an organisation. Each location is associated with one – and only one – parent organisation. Each location has a primary postal address, along with optional additional communication details such as phone numbers, email addresses and alternative	*7.11. Location*	**O**

	postal addresses.		
Change request oms	Change request provides details and additional justification and information of requested changes to create and update organisations and locations.	*7.12. Change Request OMS*	**O**

Service Summary

Service	Description	Details	SPOR Domain[2]
List Service	This service enables the user to view Lists which have been previously created and allows the user to search for lists based on the provided search criteria	*6.1. List Service*	**R**
Term Service	This service enables the user to view Terms that have previously created based on the provided search criteria.	*6.2. Term Service*	**R**
Translation Service	This service enables the user to create and update Translations for terms and to view Translations that they have previously created.	*6.3. Translation Service*	**R**
Change Request RMS Service	This service enables the user to create new Change Requests, to view Change Requests that they have previously created, and to remove any Change Requests they no longer want. Services are not provided to create/update/delete Lists and Terms directly. All such operations are facilitated via change requests services. Each change request must include a Request Reason. This informs the Data Steward the type of change being requested (e.g. a change to a term, the deletion of a term, the creation of a new list). The justification attribute can be used to provide more information about the reason for the change request, and to provide any supplementary information.	*6.4. Change Request RMS Service*	**R**
Document Service	This service enables the user to view Documents that they have previously created, remove any documents & meta-data that are no longer required, and to create new documents & meta-data.	*6.5. Document Service*	**RO**

---

[2] O=OMS specific, R=RMS specific

Service	Description	Reference	R/O
Search Query Service	This service enables the user and to create a new SearchQuery (search criteria), to view a SearchQuery that they have previously created and to remove a SearchQuery they no longer want.	*6.6. SearchQuery Service*	**R**
Subscription Service	This service enables the user to create new subscriptions, to view subscriptions that they have previously created, to remove any subscriptions they no longer want.	*6.7. Subscription Service*	**R**
Tag Service	This service enables the user to create values to use as tags, see the terms that have been associated with a particular tag value and choose to remove a tag from one or more terms.	*6.8. Tag Service*	**R**
Preferred Name Service	This service enables the user to create / update / delete / view his preferred term name	*6.9. Preferred Name Service*	**R**
Organisation Service	This service enables the user to retrieve a single organisation via one of its identifiers, or a collection of organisations based on provided search criteria.	*6.10. Organisation Service*	**O**
Location Service	This service enables the user to retrieve a single location via one of its identifiers, or a collection of locations based on provided search criteria.	*6.11. Location Service*	**O**
Change Request OMS Service	This service enables the user to retrieve a single change request via one of its identifiers, or a collection of all the change requests that the user has created. A user is only able to retrieve their own change requests; they are not able retrieve change requests raised by other users. This service also enables the user to create a new change request to request the creation/update of an organisation and/or location. Services are not provided to create/update/delete organisations and locations directly. All such operations are facilitated via change requests services. Each change request must include a Request Reason. This informs the Data Steward the type of change being requested (e.g. a change to location details, the deletion of a location, moving a location to a different parent organisation). The justification	*6.12. Change Request OMS Service*	**O**

	attribute can be used to provide more information about the reason for the change request, and to provide any supplementary information.		

6.1. *List Service*

This service enables the user to view lists and allows the user to search for lists based on the provided search criteria.

See Annex II – "Navigation between Lists & Terms" for details of interaction between List & Term Services.

6.1.1. (EP11) Search lists

Use this operation to return a collection of lists, based on provided search criteria and supports server side paging

Resource Information

End Point	GET /v{version}/lists
Request	
Accept	application/xml application/json
Body	n/a
Content-Type	n/a
Response	
Body	<list-of-lists>

Path Parameters

Name	Description
version	Service version number **Example value**: 1

Query Parameters

Name	Description	MD	WC	MV	Default
domain	Term code for domain term **Example Value:** 100000000014	N	N	Y	N/A
status	A short List status code Possible values[3]: CURRENT NON_CURRENT PROVISIONAL NULLIFIED UNDER_CONSULTATION **Example Value:** CURRENT	N	N	Y	N/A
name	List name **Example Values**: - name* (begins with search) - *name* (contains search) - name (exact search)	N	Y	N	N/A
short-name	List short name **Example Values:** - name* (begins with search) - *name* (contains search) - name (exact search)	N	Y	N	N/A

---

[3] These values are defined in the SPOR glossary of terms, which is published as a separate document.

Parameter	Description				
description	List description **Example Values:** - legal basis* (begins with search) - *legal basis* (contains search) - legal basis (exact search)	N	Y	N	N/A
modified-after	If specified, only lists modified on or after the supplied datetime are returned. Format = YYYY-MM-DDThh:mm:ssZ **Example value:** 2016-05-09T11:58:00Z	N	N	N	N/A
modified-before	If specified, only lists modified on or before the supplied datetime are returned. Format = YYYY-MM-DDThh:mm:ssZ **Example value:** 2016-05-09T11:58:00Z	N	N	N	N/A
list-owner	List owner. Possible values: CVMP EDQM EUTCT EVMPD EVMPD Team EVMPD Vet Team EudraCT TIG ISO MSSO SIAMED SIAMED Team WHO WHO Collaborating Centre for Drug Statistics Methodology eSubmission (TIGes) **Example values:** list-owner=EDQ* (begins with search) list-owner=*EDQ* (contains search) list-owner=EDQM (exact search)	N	Y	N	N/A
oid	OID code **Example Value:** oid=AB123	N	N	N	N/A
anytext	Allows for google-style search in list-code, name, short-name Note: These search criteria can't be combined with any other criteria. If it is include with other search criteria it would be ignored. **Example value:** anytext=lint	N	N	N	N/A
pagesize	Number of items per page. Possible value: • specific number (i.e. 10) • unlimited **Example value:** pagesize=10	N	N	N	20
page	Page to return. **Example value:** page=7	N	N	N	1
searchtoken	This parameter is used for multipage requests. The initial request includes the search criteria and its response would include a searchtoken, which needs to be included in subsequent requests. Subsequent requests to retrieve next / previous	N	N	N	N/A

pages should not contain a search criteria but only searchtoken (and page, pagesize, if needed)
Example value:
searchtoken=a234sb44

MD - Mandatory

WC – Support wildcards

MV – Supports multiple values (tilde delimited)

Example request

```
GET /v1/lists?name=*name*&status=NON_CURRENT
```

6.1.2. (EP12) Get list (detailed)

Use this operation to return metadata information for a specific list, identified by its list-id. No terms information returned.

Resource Information

End Point	GET /v{version}/lists/{list-id}
Request	
Accept	application/xml application/json application/zip
Body	n/a
Content-Type	n/a
Response	
Body	<list-details>

Path Parameters

Name	Description
list-id	A 12 digit unique list identifier **Example value:** 100000000004
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default

Example Request

```
GET /v1/lists/100000000004
```

6.1.3. (EP13) Get list terms(summarised)

Use this operation to return information for a specific list, identified by its list-id. Only summarised information is returned for terms contained in the list.

Resource Information

End Point	GET /v{version}/lists/{list-id}/term-summaries
Request	
Accept	application/xml application/json
Body	n/a
Content-Type	n/a
Response	
Body	<controlled-terms-list-summary>

Path Parameters

Name	Description
list-id	Unique list identifier within RMS **Example value:** 100000001902
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default
pagesize	Response page size - how many term-summary items per page are to be included. Possible values: • Specific number (i.e.10) • Unlimited Note: pagesize is not supported, if lang<> specific language **Example value:** 20	N	N	N	20
page	Page to return. Note: page is not supported, if lang<> specific language **Example value:** 2	N	N	N	1
lang-country	Include translatable attributes in the specified language-country. Possible value: (default en if not provided) • a specific lang (i.e. fr or fr-ca). If the language is different than en, en would also be included implicitly • a list of languages, separated by tilde (~) (i.e. fr~de~en-us) • all **Example value:** fr	N	N	Y	N/A
sortby	Attributes to sort by. Possible values: • id, • term-name, • short-name, • status Default order is ascending order. Prefix with "-" for descending order **Example value:** -status	N	N	N	id

| parent | Include terms depending on their position in parent-child hierarchy.

Possible values:
• root - returns all top level terms (terms that don't have a parent)
• {term-id} - the id of a term. Returns the immediate children of the specified term
• all - returns all terms regardless of their position in the hierarchy (i.e. return both root and child terms).

Examples:
  ?parent=root
  ?parent=all
  ?parent=100000075923 | N | N | N | all |
| searchtoken | Provided in returned message as next/prev page attributes. | N | N | N | N/A |

Example Request

```
GET /v1/lists/100000000004/term-summaries?parent=1000000321321
```

6.1.4. (EP14) Search terms

Use this operation to return a collection of up to 1000 terms (term summaries only), based on provided search criteria. The operation supports searches across multiple lists and supports server side paging.

Resource Information

End Point	GET /v{version}/lists/search-terms
Request	
Accept	application/xml application/json
Body	n/a
Content-Type	n/a
Response	
Body	<controlled-terms-collection> - collection of term summaries only

Path Parameters

Name	Description
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default
domain	Term code for domain term **Example Value:** 100000000014	N	N	Y	N/A
status	A short term status code Possible values[4]:	N	N	Y	N/A

---

[4] These values are defined in the SPOR glossary of terms, which is published as a separate document.

	CURRENT *NON_CURRENT* *PROVISIONAL* *NULLIFIED* *UNDER_CONSULTATION* **Example Value:** CURRENT				
name	Term name. **Example Values**: - ethics* (begins with search) - *committee* (contains search) - ethics committee (exact search)	N	Y	N	N/A
short-name	Term short name. **Example Values:** - iec* (begins with search) - *iec* (contains search) - iec (exact search)	N	Y	N	N/A
description	Term description. **Example Values:** - legal basis* (begins with search) - *legal basis* (contains search) - legal basis (exact search)	N	Y	N	N/A
modified-after	If specified, only terms modified on or after the supplied datetime are returned. The latest version of the term(s) is returned. Format = YYYY-MM-DDThh:mm:ssZ **Example value:** 2016-05-09T11:58:00Z	N	N	N	N/A
modified-before	If specified, terms modified on or before the supplied datetime are returned. The latest version of the term(s) is returned. Format = YYYY-MM-DDThh:mm:ssZ **Example value:** 2016-05-09T11:58:00Z	N	N	N	N/A
lists	list-codes. Search only in these lists. If this parameter is not provided, search in all lists **Example Value:** lists=100000075860~100000073347	N	N	Y	N/A
other-name	Other (alternative) term name. **Example Values:** - legal basis* (begins with search) - *legal basis* (contains search) - legal basis (exact search)	N	Y	N	N/A
source	Term code for the 'Source of information' term **Example Value:** source = 100000072041	N	N	N	N/A
source-term-name	Source term names. **Example Value:** - random* (begins with search) - *randomize* (contains search) - randomize (exact search)	N	Y	N	N/A

source-term-id	This is the external identifier of a term present within RMS. **Example Value:** source-term-id=AR~123RD	N	N	Y	N/A
tag	Search on user created tag **Example Value:** tag=frequently+used	N	N	N	N/A
ext-attributes	Search on list specific extended attributes and/or relationships. In order for a search on extended attribute to be used, parameter 'lists' must be included in the search query and its' value must be one list-code only. The value of 'ext-attributes' may be one or multiple key-value pairs, where 'key' is the extended attribute name to search on; 'value' is one of more search criteria values, separated by '~' (tilde). Multiple key value pairs should be separated by '&' (ampersand). Relationship represents a relationship to another list in RMS and the value is term-code for a term in the related list. Extended attributes may be of type: 'String', 'Number', 'Date', 'Boolean'. - If type is 'String' the search allows for 'begins with', 'contains' and exact search. Hence the value may be of type: - random* (begins with search) - *randomize* (contains search) - randomize (exact search) - If type is 'Number' a valid number may be provided as search value - If the type is 'Date' a valid datetime in the format YYYY-MM-DDThh:mm:ssZ may be provided as search value. Time is optional. - If the type is 'Boolean' a Y/N value may be provided as search value Note: the value of parameter 'ext-attributes' must be enclosed in double quotes (""). **Example Value:** ext-attributes="COUNTRY_GROUPING=100000093501~100000066601&START_DATE=2015-02-20&ISREADY=Y&WEIGHT=100&GENDER=mal*"	N	Y	Y	N/A
lang-filter	Language-Country to filter on. This is not a search criterion but allows control of which languages are included in the response (for translatable attributes like name). The default value is en. The value of 'lang-filter' may be: - a specific language (and optionally country) i.e. fr, or fr-ca	N	N	Y	N/A

	- a tilde separated list of languages (And optionally countries) i.e. fr-ca~de - all languages i.e. all Note: whenever one or more languages (that are not English) English translations would also be included. Example Value: Lang-filter=fr				
anytext	Allows for google-style search in term-code, name, short-name, other-name, source-term-name, source-term-id across all lists and all languages. Results are returned by relevance and not sortable. Note: These search criteria cannot be combined with any other search criteria. If it is include with other search criteria it would be ignored. Google-style search would allow for pagination so parameters pagesize, page, searchtoken are supported. **Example value:** anytext=lint	N	N	N	N/A
lang	2 letter ISO 639-1 language code. Searches for results only in the specified language. If this parameter is provided then name-type and translation-status parameters must also be provided. **Example value:** en	N	N	N	N/A
country	2 letter ISO 3166-1 language country code Searches for results only in the specified language country. If this parameter is provided then lang, name-type and translation-status parameters must also be provided. **Example value:** ca	N	N	N	N/A
translation-status	Translation status Possible Values[5]: • CURRENT • PROVISIONAL • NON_CURRENT • NULLIFIED • *MISSING* *Missing is not a stored status, it is a special processing instruction to the service to add a partial row for the end user to populate*	N	N	N	N/A

[5] These values are defined in the SPOR glossary of terms, which is published as a separate document.

	If this parameter is provided then name-type and lang parameters must also be provided. **Example value:** MISSING				
name-type	Name types. Possible values: • NAME • SHORT_NAME • OTHER_NAME • DESCRIPTION • LIST_NAME If this parameter is provided then translation-status and lang parameters must also be provided. **Example value:** NAME~DESCRIPTION	N	N	Y	N/A
pagesize	Number of items per page. Possible value: • specific number (i.e. 10) • unlimited **Example value:** pagesize=10	N	N	N	20
page	Page to return. **Example value:** page=7	N	N	N	1
sortby	Determines how the result will be ordered. The assumed order is ascending. For descending order prefix with '-'. The possible values are: • id • term-name • short-name • status • list-name **Example value:** sortby=-status (order by status in descending order)	N	N	N	id
searchtoken	This parameter is used for multipage requests. The initial request includes the search criteria and its response would include a searchtoken, which needs to be included in subsequent requests. Subsequent requests to retrieve next / previous pages should not contain a search criteria but only searchtoken (and page, pagesize, sortby, if needed) Example value: searchtoken=a234sb44	N	N	N	N/A

Example Request

```
GET /v1/lists/search-terms?name=*republic*&status=non_current&lists=100000000002&ext-
attributes="COUNTRY_GROUPING=100000093501~START_DATE=2015-02-20"
```

6.1.5. (EP15) Export terms

Use this operation to export a collection of up to 1000 term details, based on provided search criteria. The operation supports searches across multiple lists.

For an authenticated user it will return details of user defined tags and preferred names for each returned term (if present). For unauthenticated users tags and preferred names will not be populated. The ZIP output will consist of a multiple CSV files, one per list.

The requirements to export up to 1000 terms exceeds the capabilities of a GET call, therefore the service has been implemented as a POST with a payload.

Resource Information

End Point	POST /v{version}/lists/export-terms
Request	
Accept	application/xml application/zip
Body	<identifiers>
Content-Type	application/xml application/json
Response	
Body	<controlled-terms-collection> - collection of term details

Path Parameters

Name	Description
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default
lang-country	2 letter ISO language code followed by 2 letter ISO country code (if appropriate) If this parameter is provided then translations in the specified language are included in the export. Note, if the specified language is not en, en translations are also included. If a term does not have any translations for the specified language then an empty place-holder row is included. If this parameter is not provided then translations in all languages are included. **Example value:** fr	N	N	N	

Example Request

```
POST /v1/lists/export-terms
```

6.1.6. (EP16) Associate a list and document

Use this operation to associate a document with a list.

This service provides the functionality intentionally omitted from the 'document' interface, the ability to create the relationship between the list and a document. While PUT /documents indirectly has this capability it will be blocked behind the interface.

Resource Information

End Point	PUT /v{version}/lists/{list-id}/documents
Request	
Accept	application/xml application/json
Body	<document>
Content-Type	application/xml application/json
Response	
Body	<document> - metadata only

Path Parameters

Name	Description
list-id	A 12 digit unique list identifier **Example value:** 100000000004
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default

Example Request

```
PUT /v1/lists/{list-id}/documents
```

6.1.7. (EP17) Get list terms (detailed)

Use this operation to return a list with a collection of its term details as XML or zipped csv / xml file.

Resource Information

End Point	GET /v{version}/lists/{list-id}/terms
Request	
Accept	application/xml application/xml;application/zip application/zip
Body	n/a
Content-Type	n/a
Response	
Body	<controlled-terms-list>

Path Parameters

Name	Description
list-id	A 12 digit unique list identifier **Example value:** 100000000004
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default

6.2. *Term Service*

This service enables the user to view Terms.

See Annex II – "Navigation between Lists & Terms" for details of interaction between List & Term Services.

6.2.1. (EP21) Get term

Use this operation to return information for a specific term, identified by its term-id.

Resource Information

End Point	GET /v{version}/lists/{list-id}/terms/{term-id}
Request	
Accept	application/xml application/json
Body	n/a
Content-Type	n/a
Response	
Body	<term-details>

Path Parameters

Name	Description
list-id	Unique list identifier within RMS **Example value:** 100000000004
term-id	Unique term identifier within RMS **Example value:** 100000012300
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default
show-translation-ids	true/false	N	N	N	false

versions	If true, all previous versions of the term's data are returned.
	
	Possible values:
	      true
	      false
	
	Example value: true
		N	N	N	false

version-number	If set, return a specified version number of the term. Note, if a version-number parameter is provided then the versions parameter is ignored. Example value: 12	N	N	N	N/A

| version-timestamp | If specified, a representation of the term as it stood at the timestamp is returned.

Note, if a version-timestamp parameter is provided
then the versions parameter is ignored. That is, a <versions> collection is not returned.

If both version-number and version-timestamp parameters are provided then the version-number parameter takes precedence. That is, the version-timestamp parameter is ignored.

Format = YYYY-MM-DDThh:mm:ssZ

Example value: 2016-05-09T11:58:00Z | N | N | N | N/A |

Example Request

```
GET /v1/lists/100000000004/terms/100000012300
```

6.2.2. Create term

Not supported; In order to suggest new term to be created submit a new term Change

`See Change Request: – 6.4.3.` (EP43) Create change request

6.2.3. Update term

Not supported; In order to suggest update to a term submit an update term Change Request

`See Change Request: – 6.4.4.` (EP44) Update change request

6.2.4. (EP24) Get term mappings

Use this operation to translate an external term identifier into a collection of RMS identifiers. Identifiers only and unconstrained by data access policies.

Resource Information

End Point	GET /v{version}/lists/{list-id}/mappings
Request	
Accept	application/xml application/json
Body	n/a
Content-Type	n/a
Response	
Body	<mappings>

Path Parameters

Name	Description
list-id	Unique list identifier within RMS **Example value:** 100000000004

version	Service version number				
	Example value: 1				

Query Parameters

Name	Description	MD	WC	MV	Default
source-term-id	External identifier of a term present within RMS. **Example Value:** source-term-id=AR~123RD	Y	N	Y	N/A
source-id	RMS identifier of the source vocabulary in which the external identifier is defined.	N	N	N	N/A

Example Request

```
GET /v1/lists/100000000004/mappings?source-term-id=abc123~xyz123
```

6.3. *Translation Service*

This service enables the user to create and update Translations for terms and to view Translations that they have previously created.

6.3.1. (EP31) Get values for translatable term attributes

Use this operation to return English values for translatable term attributes and the translations of those attributes for a specific term and language.

Resource Information

End Point	GET /v{version}/lists/{list-id}/terms/{term-id}/translations/{lang-country}
Request	
Accept	application/xml application/json
Body	n/a
Content-Type	n/a
Response	
Body	<translation>

Path Parameters

Name	Description
list-id	Unique list identifier **Example value:** 100000000004
term-id	Unique term identifier **Example value:** 100000000102
lang-country	2 letter ISO language code followed by 2 letter ISO country code (if appropriate) Include translatable attributes in the specified language-country. Possible value: (default en if not provided) a specific lang (i.e. fr or fr-ca). If the language is different than en, en would also be included implicitly **Example values:** en-us, de-de
version	Service version number

Example value: 1	

Query Parameters

Name	Description	MD	WC	MV	Default

Example Request

```
GET /v1/lists/10000000004/terms/100000000014/translations/fr-ca
```

6.3.2. (EP32) Update the translatable term attributes for a specific list and language/country (via file)

Use this operation to update the translatable term attributes for terms in a specific list and for specified language/country.
Allows a csv file encoded as base64 to be uploaded. The file contains English values and translations for translatable attributes for terms in a specific list.
Validation errors returned as per standard HTML extended error code <error-notification>

Resource Information

End Point	PUT /v{version}/lists/{list-id}/translations/{lang-country}
Request	
Accept	n/a
Body	text/csv;application/base64
Content-Type	CSV
Response	
Body	n/a – base64 file upload

Path Parameters

Name	Description
list-id	Unique 12-digit list identifier **Example value:** 100000000004
lang-county	A 2 letter ISO language code followed by a 2 letter ISO county code. The language code is mandatory, but the country code is not. Include translatable attributes in the specified language-country. Possible value: (default en if not provided) • a specific lang (i.e. fr or fr-ca). If the language is different than en, en would also be included implicitly **Example value:** fr-ca, bg
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default

```
PUT /v1/lists/100000000004/translations/fr-ca
```

6.3.3. (EP33) Update the translatable term attributes for a specific term and language/country

Use this operation to update the translatable term attributes for a specific term and language/country.

Resource Information

End Point	PUT /v{version}/lists/{list-id}/terms/{term-id}/translations/{lang-country}
Request	
Accept	application/xml
	application/json
Body	<translation>
Content-Type	application/xml
	application/json
Response	
Body	<translation>

Path Parameters

Name	Description
list-id	Unique list identifier
	Example value: 100000000004
term-id	Unique list identifier
	Example value: 100000000102
lang-country	2 letter ISO language code followed by 2 letter ISO country code (if appropriate)
	Include translatable attributes in the specified language-country.
	Possible value: (default en if not provided)
	• a specific lang (i.e. fr or fr-ca). If the language is different than en, en would also be included implicitly
	Example values: en, de-de
version	Service version number
	Example value: 1

Query Parameters

Name	Description	MD	WC	MV	Default

Example Request

```
PUT /v1/lists/10000000004/terms/100000000014/translations/de
```

6.4. *Change Request RMS Service*

This service enables the user to create and update RMS Change Requests, to view RMS Change Requests that they have previously created, to remove any RMS Change Requests they no longer want.

6.4.1. (EP41) Search change requests rms

Use this operation to return a collection of change-requests, based on provided search criteria.

Resource Information

End Point	GET /v{version}/change-requests-rms
Request	
Accept	application/xml application/json application/zip
Body	n/a
Content-Type	n/a
Response	
Body	<change-requests-rms>

Path Parameters

Name	Description
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default
cr-name	Change Request name **Example value:** "RoAdm-Other use - EudraCT request"	N	Y	N	N/A
name	Term name in RMS **Example value:** Rotherdam	N	Y	N	N/A
cr-id	Change Request id. **Example value:** RRQ-1000000002	N	Y	Y	N/A
list	List Identifier If not provided search for change request related to any list **Example value:** 100000072057	N	N	N	N/A

status	Search only for change requests with these statuses. If this parameter is not provided, search all change requests.
	
	Possible values[6]:
	    SAVED
	    SUBMITTED
		N	N	Y	N/A

---

[6] These values are defined in the SPOR glossary of terms, which is published as a separate document.

	VALID *INVALID* *RETURNED* *APPROVED* *APPROVED_WC* *REJECTED* blank (default value) **Example value:** SUBMITTED~SAVED				

type	Search only for change requests with these types. If this parameter is not provided, search all change requests.
	
	Possible values:
	    blank (default value)
	    ADD_TERM
	    UPD_TERM
	    DEL_TERM
	    ADD_LIST
	    UPD_LIST
	
	Example value: *ADD-TERM*
		N	N	Y	N/A

date-submitted-after	If specified, only change requests submitted on or after the supplied datetime are returned. Format = YYYY-MM-DDThh:mm:ssZ **Example value:** 2016-05-09T11:58:00	N	N	N	N/A
date-submitted-before	If specified, only change requests submitted on or before the supplied datetime are returned. Format = YYYY-MM-DDThh:mm:ssZ **Example value:** 2016-05-09T11:58:00Z	N	N	N	N/A

owned	If true return only change requests that have been created by the user
	
	Possible values:
	    True
	    false
	
	Example value: true
		N	N	N	false

pagesize	Number of change requests to return per page **Example value:** 5	N	N	N	20
page	The page number to return. If missing return all change requests depending on privileges **Example value:** 2	N	N	N	N/A
sortby	Determines how the result will be ordered. The assumed order is ascending. For descending order prefix with '-'. The possible values are: • id • name • type • requestor-user-id • date-submitted • status • modified-on	N	N	N	id

	Example value: sortby=-status (order by status in descending order)				
searchtoken	This parameter is used for multipage requests. The initial request includes the search criteria and its response would include a searchtoken, which needs to be included in subsequent requests. Subsequent requests to retrieve next / previous pages should not contain a search criteria but only searchtoken (and page, pagesize, sortby, if needed) Example value: searchtoken=a234sb44		N	N	N/A

summary	Indicates that the change request will have minimal, summary level data populated. Possible values: true/false
	
	Attributes :
	    (CR) identifier, link, name, type, status, change-requester, date-submitted
	    (list-reference **)** id, name, short-name, href, rel
		N	N	N	true

Example Request

```
GET /v1/change-requests-rms?cr-
name=republic&status=SUBMITTED~REJECTED&list=100000000002
```

6.4.2. (EP42) Get change request rms

Use this operation to return information for a specific change request, identified by its change-request-id.

Resource Information

End Point	GET /v{version}/change-requests-rms/{change-request-id}
Request	
Accept	application/xml application/json
Body	n/a
Content-Type	n/a
Response	
Body	<change-request-rms>

Path Parameters

Name	Description
change-request-id	Change Request Identifier **Example value:** RRQ-100000123
version	Service version number **Example value:** 1

Name	Description	MD	WC	MV	Default

Example Request

```
GET /v1/change-requests-rms/RRQ-100000123
```

6.4.3. (EP43) Create change request rms

Use this operation to create a new change request.

A change request may be of the following types: "add term", "update term", "delete term", "add list", "update list".
Change requests relating to lists (i.e. "add list", "update list") have only change request attributes and details of associated documents.
Change requests relating to terms (i.e. "add term", "update term", "delete term") have change request attributes, optionally details of associated documents and controlled term.
For "delete term" change requests, an alternative/replacement term(s) may be specified for the term being deleted.

Change requests may or may not contain a number of elements depending on the type of request.
Those conditional elements are change-request-rms.draft-term, change-request-rms.draft-term.term-id and change-request-rms.list-ref. The following table shows when they are required:

Change Request Type	change-request-rms.list-ref	change-request-rms.draft-term	change-request-rms.draft_term.term-id
Add list	No	No	No
Update list	Yes	No	No
Add term	Yes	Yes	No
Update term	Yes	Yes	Yes
Delete term	Yes	Yes	Yes

Once a change request creation is processed by the server, the response shall contain the change-request-rms.request-id assigned to such change request. It is the responsibility of the implementing system to store this request-id in order to provide it again in any update action taken onto the created change request.

In the case of term related change requests, the change-request-rms payload that must be passed back to system must retain a number of identifiers that belong to the existing term. The implementing system is responsible to obtain them before creating an update or delete change request, and provide them back in the payload of the change request. Such identifiers are part of the returned controlled-term structure and must be provided as part of the draft-term structure. For an exhaustive list of identifiers to store, please see below 7.4. Change Request RMS.

Note: Services are not provided to create/update/delete Lists and Terms directly. All such operations are facilitated via change requests services.

Resource Information

End Point	POST /v{version}/change-requests-rms
Request	
Accept	application/xml application/json

Body	<change-request-rms>
Content-Type	application/xml
	application/json
Response	
Body	<change-request-rms>

Path Parameters

Name	Description
version	Service version number
	Example value: 1

Query Parameters

Name	Description	MD	WC	MV	Default

Example Request

```
POST /v1/change-requests-rms
```

6.4.4. (EP44) Update change request rms

Use this operation to update an existing change request.

A change request may be of the following types: "add term", "update term", "delete term", "add list", "update list".

An update on any type of change request must contain its request-id. Otherwise the change request shall be treated as new.

Change requests relating to lists (i.e. "add list", "update list") have only change request attributes and details of associated documents.

Change requests relating to terms (i.e. "add term", "update term", "delete term") have change request attributes plus optional details of associated documents and controlled term. When an update on an already created change request relating to terms is executed, the relevant draft-term identifiers previously obtained in the change request creation must be retained, that is, provided again. Any new element within the change request must render no identifier. However, once the response is obtained, the new identifiers assigned by the system to those elements must also be stored in order to provide them back in the case of a new change request update. For an exhaustive list of identifiers to store, please see below 7.4. Change Request RMS.

For "delete term" change requests an alternative/replacement term(s) may be specified for the term being deleted.

Note: This operation does not allow updating documents uploaded with the previous version of the change request. It allows only deleting an existing and adding a new document to the change request.

Note: Services are not provided to create/update/delete Lists and Terms directly. All such operations are facilitated via change requests services.

Resource Information

End Point	PUT /v{version}/change-requests-rms/{change-request-id}
Request	
Accept	application/xml
	application/json

Body	<change-request-rms>
Content-Type	application/xml application/json
Response	
Body	<change-request-rms>

Path Parameters

Name	Description
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default

Example Request

```
PUT /v1/change-requests-rms/RRQ-100000123
```

6.4.5. (EP45) Delete change request rms

Use this operation to delete an existing change request.

Deletion is only allowed if the Change Request has status 'Saved'. Once the change request is submitted (status is different than 'Saved'), the change request cannot be deleted.
"Status" refers to the where the Change Request is in the saved/submitted/approve|rejectlife-cycle

Resource Information

End Point	DELETE /v{version}/change-requests-rms/{change-request-id}
Request	
Accept	n/a
Body	n/a
Content-Type	n/a
Response	
Body	n/a

Path Parameters

Name	Description
change-request-id	Unique identifier of change request Example value: RRQ-1000000001
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default

Example Request

```
DELETE /v1/change-requests-rms/RRQ-1000000001
```

6.5. *Document Service*

This service enables the user to view Documents that they have previously created, remove any documents & meta-data that are no longer required, and to create new documents & meta-data.

6.5.1. (EP51) Create document

Use this operation to create (upload) a new document. The method allows the content of the file to be uploaded as base64 and creates the file and its' metadata.

Resource Information

End Point	POST /v{version}/documents
Request	
Accept	application/xml application/json
Body	<document>
Content-Type	application/xml application/json
Response	
Body	<document>

Path Parameters

Name	Description
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default
app-domain	SPOR domain the document relates to **Example value:** RMS	Y	N	N	N/A
doc-type	Document type. i.e. GENERAL, HELP, TECHNICAL , NCA, REQUEST, LIST-INFO **Example value:** HELP	Y	N	N	N/A

Example Request

```
POST /v1/documents?app-domain=RMS&doc-type=GENERAL
```

6.5.2. (EP52) Update document

Use this operation to update an existing document.

The operation updates the document metadata and replaces the document file. The content of the document file is uploaded as base64. Note, this operation replaces the existing document file, multiple versions of the document file are not maintained.

While PUT /documents indirectly has the capability to associate the document with a list it will be blocked behind the interface as it is provided for in `PUT /lists/{list-id}/documents`.

Resource Information

End Point	PUT /v{version}/documents/{document-id}
Request	
Accept	application/xml application/json
Body	<document>
Content-Type	application/xml application/json
Response	
Body	<document>

Path Parameters

Name	Description
document-id	Unique document identifier **Example value:** 123
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default
app-domain	SPOR domain the document relates to **Example value:** RMS	Y	N	N	N/A
doc-type	Document type. i.e. GENERAL, HELP, TECHNICAL , NCA, REQUEST, LIST-INFO **Example value:** HELP	Y	N	N	N/A

Example Request

`PUT /v1/documents/123?app-domain=RMS&doc-type=GENERAL`

6.5.3. (EP53) Delete document

Use this operation to delete an existing document and its associated metadata.

Resource Information

End Point	DELETE /v{version}/documents/{document-id}
Request	
Accept	n/a

Body	n/a
Content-Type	n/a
Response	
Body	n/a

Path Parameters

Name	Description
document-id	Unique document identifier **Example value:** 123
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default
app-domain	SPOR domain the document relates to **Example value:** RMS	Y	N	N	N/A
doc-type	Document type. i.e. GENERAL, HELP, TECHNICAL , NCA, REQUEST, LIST-INFO **Example value:** HELP	Y	N	N	N/A

Example Request URL

```
DELETE /v1/documents/123?app-domain=RMS&doc-type=GENERAL
```

6.5.4. (EP54) Get document

Use this operation to return information for a specific document, identified by its document-id. The operation can be used to return document metadata or document content.

Resource Information

End Point	GET /v{version}/documents/{document-id}
Request	
Accept	**Document Metadata Representations** application/xml application/json **Document Content Representations** application/base64 application/*xxx* Where *xxx* is the document's native mime type (e.g. pdf).
Body	n/a
Content-Type	n/a
Response	
Body	<document>

Path Parameters

Name	Description
document-id	Unique document identifier **Example value**: 123

version	Service version number
	Example value: 1

Query Parameters

Name	Description	MD	WC	MV	Default
app-domain	SPOR domain the document relates to **Example value:** RMS	Y	N	N	N/A
doc-type	Document type. i.e. GENERAL, HELP, TECHNICAL , NCA, REQUEST, LIST-INFO **Example value:** HELP	Y	N	N	N/A

Example Request

```
GET /v1/document/123?app-domain=RMS&doc-type=GENERAL
```

6.5.5. (EP55) Search documents

Use this operation to return a collection of documents, based on provided search criteria.

Resource Information

End Point	GET /v{version}/documents
Request	
Accept	application/xml application/json
Body	n/a
Content-Type	n/a
Response	
Body	<documents>

Path Parameters

Name	Description
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default
app-domain	SPOR domain the document relates to **Example value:** RMS	Y	N	N	N/A
doc-type	Document type. i.e. GENERAL, HELP, TECHNICAL , NCA, REQUEST, LIST-INFO **Example value:** HELP	Y	N	N	N/A
latest	Include the document with the latest publication date of the requested doctype. **Example value:** true	N	N	N	false

Example Request

```
GET /v1/documents?app-domain=RMS&doc-type=HELP&latest=true
```

6.6. *SearchQuery Service*

This service enables users to create a new SearchQuery (search criteria), to view a SearchQuery that they have previously created and update it and to remove a SearchQuery they no longer want.

6.6.1. (EP61) Get all search queries

Use this operation to return all search queries that the user has created.

Resource Information

End Point	GET /v{version}/search-queries
Request	
Accept	application/xml
	application/json
Body	n/a
Content-Type	n/a
Response	
Body	<search-queries>

Path Parameters

Name	Description
version	Service version number
	Example value: 1

Query Parameters

Name	Description	MD	WC	MV	Default
None					

Example Request

```
GET /v1/search-queries
```

6.6.2. (EP62) Create search query

Use this operation to create a new search query.

Resource Information

End Point	POST /v{version}/search-queries
Request	
Accept	application/xml
	application/json
Body	<search-query>
Content-Type	application/xml
	application/json
Response	
Body	<search-query>

Path Parameters

Name	Description
version	Service version number
	Example value: 1

Query Parameters

Name	Description	MD	WC	MV	Default

Example Request

```
POST /v1/search-queries
```

6.6.3. (EP63) Get search query

Use this operation to return information for a specific search query, identified by its search-query-id.

Resource Information

End Point	GET /v{version}/search-queries/{search-query-id}
Request	
Accept	application/xml application/json
Body	n/a
Content-Type	n/a
Response	
Body	<search-query>

Path Parameters

Name	Description
search-query-id	Search query unique identifier **Example value:**123
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default

Example Request

```
GET /v1/search-queries/123
```

6.6.4. (EP64) Update search query

Use this operation to update an existing search query.

Resource Information

End Point	PUT /v{version}/search-queries/{search-query-id}
Request	
Accept	application/xml application/json
Body	<search-query>
Content-Type	application/xml application/json
Response	

Body	<search-query>

Path Parameters

Name	Description
search-query-id	Search query unique identifier **Example value:** 123
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default

Example Request

```
PUT /v1/search-queries/{123}
```

6.6.5. (EP65) Delete search query

Use this operation to delete an existing search query.

Resource Information

End Point	DELETE /v{version}/search-queries/{search-query-id}
Request	
Accept	n/a
Body	n/a
Content-Type	n/a
Response	
Body	n/a

Path Parameters

Name	Description
search-query-id	Search query unique identifier **Example value:** 123
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default

Example Request

```
DELETE /v1/search-queries/123
```

6.7. *Subscription Service*

This service enables the user to create new subscriptions, to view subscriptions that they have previously created, to remove any subscriptions they no longer want.

A subscription to a specific list will indicate whether the user is to be notified of all changes or only major changes to that list. The user can also choose to be notified of any new lists that are created.

6.7.1. (EP71) Get all Subscriptions

Use this operation to return all the lists subscriptions that the user has created. As a system call all user subscriptions are returned ordered by last-notification-time.

Resource Information

End Point	GET /v{version}/subscriptions
Request	
Accept	application/xml application/json
Body	n/a
Content-Type	n/a
Response	
Body	<subscription>

Path Parameters

Name	Description
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default
pagesize optional	Number of items per page. Possible value: • specific number (i.e. 10) • unlimited **Example value:** pagesize=10	N	N	N	N
Pageno optional	Page to return. **Example value:** page=7	N	N	N	N

Example Request

```
GET /v1/subscriptions
GET /v1/subscriptions?pagesize=10&pageno=1
```

6.7.2. (EP72) Get all Subscriptions from a specific list

Use this operation to return all subscriptions for a specific list, identified by its list-id.

Resource Information

End Point	GET /v{version}/lists/{list-id}/subscriptions
Request	
Accept	application/xml application/json

Body	n/a
Content-Type	n/a
Response	
Body	<subscriptions> Sub-element list-subscriptions would only contain one element (for the specified list)

Path Parameters

Name	Description
list-id	Unique list identifier **Example value:** 100000000002
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default

Example Request

```
GET /v1/lists/100000000002/subscriptions
```

6.7.3. (EP73) Create / update / delete user subscriptions

Use this operation to create / update / delete subscriptions for the user that sends the request. While not strictly RESTful, this operation has been designed to merge user list subscription changes into the database as a set to minimise individual calls.

Resource Information

End Point	PUT /v{version}/subscriptions
Request	
Accept	application/xml application/json
Body	<subscriptions>
Content-Type	application/xml application/json
Response	
Body	<subscriptions>

Path Parameters

Name	Description
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default

Example Request

```
PUT /v1/subscriptions
```

6.8. *Tag Service*

This service enables the user to tags a collection of terms with a unique name (per user) and to see the terms associated with a particular tag names.
A tag may be removed one or more terms. An empty tag with no associated terms will be deleted. A tag with terms cannot be deleted

6.8.1. (EP81) Get all tags

Use this operation to return all tags that the user has created, optionally by term-id and tag-name

Resource Information

End Point	GET /v{version}/tags
Request	
Accept	application/xml application/json
Body	n/a
Content-Type	n/a
Response	
Body	<tags>

Path Parameters

Name	Description
version	Service version number **Example value: 1**

Query Parameters

Name	Description	MD	WC	MV	Default
term-id optional	term id **Example value:** 10000000123	N	N	N	N/A
tag-name optional	Search on user created tag **Example Value:** tag=frequently used	N	N	N	N/A

Example Request

```
GET /v1/tags?term-id=10000000123
```

6.8.2. (EP82) Get tag

Use this operation to return information for a tag, identified by its tag-id.

Resource Information

End Point	GET /v{version}/tags/{tag-id}
Request	
Accept	application/xml application/json

Body	n/a
Content-Type	n/a
Response	
Body	<tag>

Path Parameters

Name	Description
tag-id	Unique tag identifier **Example value**: 123
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default

Example Request

```
GET /v1/tags/123
```

6.8.3. (EP83) Create tag

Use this operation to create a new tag.

Resource Information

End Point	POST /v{version}/tags
Request	
Accept	application/xml application/json
Body	<tag>
Content-Type	application/xml application/json
Response	
Body	<tag>

Path Parameters

Name	Description
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default

Example Request

```
POST /v1/tags
```

6.8.4. (EP84) Update tag

Use this operation to update an existing tag.

Resource Information

End Point	PUT /v{version}/tags/{tag-id}	
Request		
Accept	application/xml application/json	
Body	<tag>	
Content-Type	application/xml application/json	
Response		
Body	<tag>	

Path Parameters

Name	Description
version	Service version number **Example value:** 1
tag-id	Tag id **Example value:** 123

Query Parameters

Name	Description	MD	WC	MV	Default

Example Request

```
PUT /v1/Tags/123
```

6.8.5. (EP85) Delete tag

Use this operation to delete an existing Tag.

Resource Information

End Point	DELETE /v{version}/tags/{tag-id}
Request	
Accept	n/a
Body	n/a
Content-Type	n/a
Response	
Body	n/a

Path Parameters

Name	Description
tag-id	Tag id **Example value:** 123
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default

Example Request

```
DELETE /v1/Tags/123
```

6.8.6. Add a new Tag to a specific Term

Use *(EP83) Create tag* operation and populate collection of related term identifiers.

6.8.7. Add an existing Tag to a specific Term

Use *(EP84) Update tag* operation and populate collection of related term identifiers.

6.9. *Preferred Name Service*

This service enables the process preferred term name.

6.9.1. (EP90) Get all preferred names

Use this operation to get the user's preferred names for terms across all lists.

Resource Information

End Point	GET /v{version}/preferred-names
Request	
Accept	application/xml application/json
Body	n/a
Content-Type	n/a
Response	
Body	<preferred-names>

Path Parameters

Name	Description
term-id	Term identifier Example value: 100000000023
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default

Example Request

6.9.2. (EP91) Get preferred name

Use this operation to get a user's preferred name for a specific term.

Resource Information

End Point	GET /v{version}/lists/{list-id}/terms/{term-id}/preferred-name
Request	
Accept	application/xml application/json
Body	n/a
Content-Type	n/a
Response	
Body	<preferred-name>

Path Parameters

Name	Description
term-id	Term identifier

	Example value: 100000000023
list-id	Unique list identifier **Example value:** 100000000002
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default

Example Request

```
GET /v1/lists/100000000002/terms/100000000023/preferred-name
```

6.9.3. (EP92) Create preferred name

Use this operation to create a user's preferred name for a specific term.

Resource Information

End Point	POST /v{version}/lists/{list-id}/terms/{term-id}/preferred-name
Request	
Accept	application/xml application/json
Body	<preferred-name>
Content-Type	application/xml application/json
Response	
Body	<preferred-name>

Path Parameters

Name	Description
term-id	Term identifier **Example value:** 100000000023
list-id	Unique list identifier **Example value:** 100000000002
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default

Example Request

```
POST /v1/lists/100000000002/terms/100000000023/preferred-name
```

6.9.4. (EP93) Update preferred name

Use this operation to update a user's preferred name for a specific term.

Resource Information

End Point	PUT /v{version}/lists/{list-id}/terms/{term-id}/preferred-name
Request	
Accept	application/xml application/json
Body	<preferred-name>
Content-Type	application/xml application/json
Response	
Body	<preferred-name>

Path Parameters

Name	Description
term-id Mandatory	Term identifier **Example value:** 100000000023
list-id Mandatory	Unique list identifier **Example value:** 100000000002
version mandatory	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default

Example Request

```
PUT /v1/lists/100000000002/terms/100000000023/preferred-name
```

6.9.5. (EP94) Delete preferred name

Use this operation to delete a user's preferred name for a specific term.

Resource Information

End Point	DELETE /v{version}/lists/{list-id}/terms/{term-id}/preferred-name
Request	
Accept	n/a
Body	n/a
Content-Type	n/a
Response	
Body	n/a

Path Parameters

Name	Description
term-id	Term identifier **Example value:** 100000000023

list-id	Unique list identifier
	Example value: 100000000002
version	Service version number
	Example value: 1

Query Parameters

Name	Description	MD	WC	MV	Default

Example Request

```
DELETE /v1/lists/100000000002/terms/100000000023/preferred-name
```

6.10. *Organisation Service*

6.10.1. (EP101) Search organisations

Use this operation to return a collection of organisations, based on provided search criteria.

Resource Information

End Point	GET /v{version}/organisations
Request	
Accept	application/xml application/json application/zip
Body	n/a
Content-Type	n/a
Response	
Body	<organisations>

Path Parameters

Name	Description
version	Service version number
	Example value: 1

Query Parameters

Name	Description	MD	WC	MV	Default
pagesize	Number of organisations to return per page (maximum = 200). 0 indicates that all organisations should be returned. Note, pagesize=0 cannot be provided in conjunction with any other query parameters except versions=true or versions=false. **Example value:** 5	N	N	N	20
page	The page number of results to return. **Example value:** 2	N	N	N	1

Parameter	Description				Default
sortby	Indicates which attribute results are to be sorted by. Supported values are: *org-id* *org-name* Default order is ascending order. Prefix sortby with "-" for descending order. **Example value:** -org-name	N	N	N	org-id
searchtoken	This parameter is used for multipage searches. The initial request includes the search criteria and its response would include a searchtoken, which needs to be included in subsequent requests. Subsequent requests to retrieve next / previous pages should not contain a search criteria but only the searchtoken (and page, pagesize, sortby, if needed) **Example value:** a234sb44	N	N	N	N/A
org-id	If specified, only organisations that have matching Organisation IDs are returned. **Example value:** ORG-1002*	N	Y	N	N/A
org-name	If specified, only organisations that have matching names are returned. The org-name parameter is matched against the following types of organisation names[7]: *Name* *Acronym* *Alternative Names* Both current and history names are matched. Accented characters are ignored when matching. **Example value:** acme limited	N	Y	N	N/A
org-modified-after	If specified, only organisations modified on or after the supplied datetime are returned. Format = YYYY-MM-DDThh:mm:ssZ **Example value:** 2016-05-09T11:58:00	N	N	N	N/A
org-status	A tilde (~) separated list of statuses. If specified, only organisations with matching statuses are returned. Possible values: ACTIVE INACTIVE **Example value:** ACTIVE~INACTIVE	N	N	Y	ACTIVE
versions	If true, all previous versions of the organisation's data are returned. Previous versions are represented in the <versions> collection.	N	N	N	false

---

[7] These values are defined in the SPOR glossary of terms, which is published as a separate document.

	Possible values:					
	true					
	false					
	Example value: true					
mapping-code-system **mapping-code**	Used to request an organisation via one of its mapping codes. If used, both mapping-code-system and mapping-code parameters must be provided. mapping-code-system Used to specify which system the mapping code relates to. mapping-code Used to specify the mapping code required. mapping-code-system/mapping-code and org-id parameters are mutually exclusive. If both parameters are provided then mapping-code-system/mapping-code parameters are ignored. **Example value:** ?mapping-code-system=100000167431&mapping-code= 12345	N	N	N	N/A	
summary	If true then only summary information is returned for each matching organisation (see *7.10. Organisation* for details). Possible values: true false versions and summary parameters are mutually exclusive. If both parameters are provided then the versions parameter is ignored. That is, the summary parameter takes precedence. **Example value:** true	N	N	N	false	

Example Request

```
GET /v1/organisations?org-name=acme&org-status=ACTIVE~INACTIVE
```

6.10.2. (EP102) Get organisation

Use this operation to return information for a specific organisation, identified by one of its identifiers (see id parameter for supported identifier types).

Resource Information

End Point	GET /v{version}/organisations/{id}
Request	
Accept	application/xml application/json application/zip
Body	n/a
Content-Type	n/a
Response	

Body	<organisation>

Path Parameters

Name	Description
id	Unique identifier for organisation
	The service returns the organisation that has an identifier matching the id parameter.
	The id parameter is matched against the following identifier types:
	Organisation ID
	Request ID – Organisations can be retrieved via the Request ID that was used to create the organisation. They cannot be retrieved via Request IDs that were used to update the organisation.
	Example value: ORG-100000123
version	Service version number
	Example value: 1

Query Parameters

Name	Description	MD	WC	MV	Default
versions	If true, all previous versions of the organisation's data are returned. Previous versions are represented in the <versions> collection. Possible values: true false **Example value:** true	N	N	N	false
version-timestamp	If specified, a representation of the organisation as it stood at the timestamp is returned. If not specified, the current representation of the organisation is returned (i.e. based on current data). Note, if a version-timestamp parameter is provided then the versions parameter is ignored. That is, a <versions> collection is not returned. Format = YYYY-MM-DDThh:mm:ssZ **Example value:** 2016-05-09T11:58:00Z	N	N	N	N/A

Example Request

```
GET /v1/organisations/ORG-100000123?version-timestamp=2016-01-01
```

6.11. *Location Service*

6.11.1. (EP111) Search locations

Use this operation to return a collection of locations, based on provided search criteria.

Resource Information

End Point	GET /v{version}/locations
Request	
Accept	application/xml application/json application/zip
Body	n/a
Content-Type	n/a
Response	
Body	<locations>

Path Parameters

Name	Description
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default
pagesize	Number of locations to return per page (maximum = 200). 0 indicates that all locations should be returned. Note, pagesize=0 cannot be provided in conjunction with any other query parameters except versions=true or versions=false. **Example value:** 5	N	N	N	20
page	The page number of results to return. **Example value:** 2	N	N	N	1
sortby	Indicates which attribute results are to be sorted by. Supported values[8]: *id* *org-name* *loc-country* *loc-city* *loc-postalcode* *loc-status* *loc-modified-on* Default order is ascending order. Prefix sortby with "-" for descending order. **Example value:** -org-name	N	N	N	id

---

[8] These values are defined in the SPOR glossary of terms, which is published as a separate document.

searchtoken	This parameter is used for multipage searches. The initial request includes the search criteria and its response would include a searchtoken, which needs to be included in subsequent requests. Subsequent requests to retrieve next / previous pages should not contain a search criteria but only searchtoken (and page, pagesize, sortby, if needed) **Example value:** a234sb44	N	N	N	N/A
org-id	If specified, only locations that have parent organisations with matching Organisation IDs are returned. **Example value:** ORG-1002*	N	Y	N	N/A
org-location-id	If specified, only locations that have matching identifiers (Org Location IDs) are returned. **Example value:** LOC-1002*	N	Y	N	N/A

org-name	If specified, only organisations that have matching names are returned.
	
	The org-name parameter is matched against the following types of organisation names:
	    Name
	    Acronym
	    Alternative Names
	
	Both current and history names are matched. Accented characters are ignored when matching.
	
	Example value: acme limited
		N	Y	N	N/A

loc-address	If specified, only locations with matching addresses are returned.
	
	The loc-address parameter is matched against the following elements of location addresses:
	
	    address-line-1
	    address-line-2
	    address-line-3
	    address-line-4
	    state
	    region
	    county
	
	All language versions of the address are considered when matching.
	
	Accented characters are ignored when matching.
	
	Example value: *the high street*
		N	Y	N	N/A

loc-city	If specified, only locations with matching cities are returned. All language versions are considered when matching.	N	Y	N	N/A

	Accented characters are ignored when matching. **Example value:** *london*				
loc-postalcode	If specified, only locations with matching postal-codes are returned. **Example value:** me1*	N	Y	N	N/A
loc-country	A tilde (~) separated list of 2 character ISO country codes. If specified, only locations in matching countries are returned. **Example value:** FI~NO~SE~DK~IS	N	N	Y	N/A
loc-modified-after	If specified, only locations modified on or after the supplied datetime are returned. Format = YYYY-MM-DDThh:mm:ssZ **Example value:** 2016-05-09T11:58:00Z	N	N	N	N/A
loc-status	If specified, only locations with matching statuses are returned. Possible values[9]: ACTIVE INACTIVE **Example value:** ACTIVE~INACTIVE	N	N	Y	ACTI VE
versions	If true, all previous versions of the location's data are returned. Previous versions are represented in the \<versions\> collection. Possible values: true false **Example value:** true	N	N	N	false
mapping-code-system **mapping-code**	Used to request a location via one of its mapping codes. mapping-code-system Used to specify which system the mapping code relates to. mapping-code Used to specify the mapping code required. mapping-code-system/mapping-code and org-id parameters are mutually exclusive. If both parameters are provided then mapping-code-system/mapping-code parameters are ignored. mapping-code-system/mapping-code and org-location-id parameters are mutually exclusive. If	N	N	N	N/A

---

[9] These values are defined in the SPOR glossary of terms, which is published as a separate document.

	both parameters are provided then mapping-code-system/mapping-code parameters are ignored. **Example value:** ?mapping-code-system=100000167431&mapping-code= 03-563-9413				
summary	If true then only summary information is returned for each matching location (see *7.11. Location* for details). Possible values: true false versions and summary parameters are mutually exclusive. If both parameters are provided then the versions parameter is ignored. That is, the summary parameter takes precedence. **Example value:** true	N	N	N	false
organisation-details	If true, an organisations.csv is included in the returned locations.zip containing details of the returned locations' parent organisations. Please note the following: 1) The organisation-details parameter is only valid for application/zip representations. It is not possible to request organisation details with application/xml and application/json representations. 2) The value passed in the versions parameter is used to determine if previous versions of parent organisation data is included in the returned organisations.csv. Possible values: true false **Example value:** true	N	N	N	false

Example Request

```
GET /v1/locations?org-name=acme&loc-status=ACTIVE~INACTIVE
```

6.11.2. (EP112) Get location

Use this operation to return information for a specific location, identified by one of its identifiers (see id parameter for supported types).

Resource Information

End Point	GET /v{version}/locations/{id}
Request	
Accept	application/xml

	application/json application/zip
Body	n/a
Content-Type	n/a
Response	
Body	<location>

Path Parameters

Name	Description
id	Unique identifier for location

The service returns the location that has an identifier matching the id parameter.

The id parameter is matched against the following identifier types:

Org Location ID

Request ID - Locations can be retrieved via the Request ID that was used to create the location. They cannot be retrieved via Request IDs that were used to update the location.

Example value: LOC-100000123 |
| **version** | Service version number

Example value: 1 |

Query Parameters

Name	Description	MD	WC	MV	Default
versions	If true, all previous versions of the location's data are returned. Previous versions are represented in the <versions> collection.

Possible values:
    true
    false

Example value: true | N | N | N | false |
| **version-timestamp** | If specified, a representation of the location as it stood at the timestamp is returned. If not specified, the current representation of the location is returned (i.e. based on current data).

Note, if a version-timestamp parameter is provided then the versions parameter is ignored. That is, a <versions> collection is not returned.

Format = YYYY-MM-DDThh:mm:ssZ

Example value: 2016-05-09T11:58:00Z | N | N | N | N/A |
| **organisation-details** | If true, an organisation.csv is included in the returned location.zip containing details of the returned location's parent organisation.

Please note the following: | N | N | N | false |

	1) The organisation-details parameter is only valid for application/zip representations. It is not possible to request organisation details with application/xml and application/json representations.
	
	2) The value passed in the versions parameter is used to determine if previous versions of parent organisation data is included in the returned organisation.csv.
	
	Possible values:
	        true
	        false
	
	Example value: true			

Example Request

```
GET /v1/locations/LOC-100000123?version-timestamp=2016-01-01
```

6.12. *Change Request OMS Service*

6.12.1. (EP121) Search change requests oms

Use this operation to return a collection of OMS change requests, based on provided search criteria.

Resource Information

End Point	GET /v{version}/change-requests-oms
Request	
Accept	application/xml application/json application/zip
Body	n/a
Content-Type	n/a
Response	
Body	<change-requests-oms>
Content-Type	application/xml application/json application/zip

Path Parameters

Name	Description
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default
pagesize	Number of change requests to return per page. 0 indicates that all matching change requests should be returned. **Example value:** 5	N	N	N	0

page	The page number of results to return. Example value: 2	N	N	N	1
searchtoken	This parameter is used for multipage searches. The initial request includes the search criteria and its response would include a searchtoken, which needs to be included in subsequent requests. Subsequent requests to retrieve next / previous pages should not contain a search criteria but only searchtoken (and page, pagesize, sortby, if needed) Example value: a234sb44	N	N	N	N/A
summary	If true then only summary information is returned for each matching location (see *7.12. Change Request OMS* for details). Possible values: true false The following properties are not returned when summary information is requested. Example value: true	N	N	N	false

Example Request

```
GET /v1/change-requests-oms?summary=true
```

6.12.2. (EP122) Get change request oms

Use this operation to return information for a specific OMS change request, identified by its change-request-id.

Resource Information

End Point	GET /v{version}/change-requests-oms/{change-request-id}
Request	
Accept	application/xml application/json application/zip
Body	n/a
Content-Type	n/a
Response	
Body	<change-request-oms>

Path Parameters

Name	Description
id	Unique identifier for change request The service returns the change request that has an identifier matching the id parameter. Example value: ORQ-100000123

version	Service version number
	Example value: 1

Query Parameters

Name	Description	MD	WC	MV	Default

Example Request

```
GET /v1/change-requests-oms/ORQ-100000123
```

6.12.3. (EP123) Create change request oms

Use this operation to create a new change request.

A change request may be one of the following types:

- ADD-ORGANISATION
- UPD-ORGANISATION
- ADD-LOCATION
- UPD-LOCATION
- UPD-ORG-AND-LOCATION

It must be noted that a change request at a given element will block the creation of new change requests for that same element. That is, if an active change request for an organisation is present in the system, no new change request for such organisation can be submitted (any change request referring to locations of the organisation shall be accepted); alternatively, if an active change request for a location is present in the system, no new change request for such location can be submitted (any change request referring to the organisation or other locations shall be accepted).

Change requests contain data for the organisation (<draft-organisation>) and location (<draft-location>) being created/changed. The following table indicates which elements (draft-organisation/draft-location) are required by change request type.

Change Request Type	draft-organisation required	draft-location required
ADD-ORGANISATION	Yes	Yes
UPD-ORGANISATION	Yes	No
ADD-LOCATION	No	Yes
UPD-LOCATION	No	Yes
UPD-ORG-AND-LOCATION	Yes	Yes

The presence of a number of fields in the change request for OMS is conditional to the type of request. Additionally there are also cases in which a given field must not be provided for a given type, but it is defined in the XSD as mandatory. The way to deal with such fields is giving them any valid value they can accept. This whole behaviour is described in the following table (all elements from root change-request-oms):

Key:
M (Mandatory)
O (Optional)
- (Must not be provided)

	Change request type					
Element	ADD-ORGANISATION	ADD-LOCATION	UPD-ORGANISATION	UPD-LOCATION	UPD-ORG-AND-LOCATION	Dummy value

request-id.id	-	-	-	-	-	
request-id.link.href	-	-	-	-	-	
request-id.link.rel	-	-	-	-	-	
name	M	M	M	M	M	
type	M	M	M	M	M	
status	-	-	-	-	-	"-1"
status-comments	-	-	-	-	-	
requestor-user-id	-	-	-	-	-	"-1"
requestor-organisation-id	O	O	O	O	O	
requestor-email	M	M	M	M	M	
requestor-phone	O	O	O	O	O	
date-submitted	-	-	-	-	-	
justification	O	O	O	O	O	
status-changes.status-change.status	-	-	-	-	-	
status-changes.status-change.status-comments	-	-	-	-	-	
status-changes.status-change.modified-on	-	-	-	-	-	
documents.document.operational-attributes.created-by	-	-	-	-	-	
documents.document.operational-attributes.created-on	-	-	-	-	-	Current datetime
documents.document.operational-attributes.modified-by	-	-	-	-	-	
documents.document.operational-attributes.modified-on	-	-	-	-	-	
documents.document.document-id.id	-	-	-	-	-	"-1"
documents.document.document-id.link.href	-	-	-	-	-	"http://"
documents.document.document-id.link.rel	-	-	-	-	-	
documents.document.application-domain	-	-	-	-	-	"-1"
documents.document.type	-	-	-	-	-	"-1"
documents.document.name	-	-	-	-	-	"-1"
documents.document.description	-	-	-	-	-	
documents.document.version	-	-	-	-	-	
documents.document.published-date	-	-	-	-	-	
documents.document.file-name	M (only if mime-type or file-base64 provided)	M (only if mime-type or file-base64 provided)	M (only if mime-type or file-base64 provided)	M (only if mime-type or file-base64 provided)	M (only if mime-type or file-base64 provided)	
documents.document.mime-type	M (only if file-name or file-base64 provided)	M (only if file-name or file-base64	M (only if file-name or file-base64 provided)	M (only if file-name or file-base64	M (only if file-name or file-base64 provided)	

		provided)		provided)		
documents.document.file-base64	M (only if file-name or mime-type provided)	M (only if file-name or mime-type provided)	M (only if file-name or mime-type provided)	M (only if file-name or mime-type provided)	M (only if file-name or mime-type provided)	
documents.document.resource.id	-	-	-	-	-	
documents.document.resource.link.href	-	-	-	-	-	
documents.document.resource.link.rel	-	-	-	-	-	
documents.document.representations.representation.href	-	-	-	-	-	
documents.document.representations.representation.content-type	-	-	-	-	-	
address-language	-	-	-	M	M	"-1" when not M
request-reason.code-system	-	-	-	-	-	
request-reason.code-system-name	-	-	-	-	-	
request-reason.code	M	M	M	M	M	
request-reason.display-name	-	-	-	-	-	"-1"
request-reason.code-version	-	-	-	-	-	
draft-organisation.organisation-id.id	-	-	M	-	M	
draft-organisation.organisation-id.link.href	-	-	-	-	-	"http://"
draft-organisation.organisation-id.link.rel	-	-	-	-	-	
draft-organisation.category-classifications.category-classification.category-type.code-system	-	-	-	-	-	
draft-organisation.category-classifications.category-classification.category-type.code-system-name	-	-	-	-	-	
draft-organisation.category-classifications.category-classification.category-type.code	O	-	O	-	O	
draft-organisation.category-classifications.category-classification.category-type.display-name	-	-	-	-	-	"-1"

draft-organisation.category-classifications.category-classification.category-type.code-version	-	-	-	-	-	
draft-organisation.category-classifications.category-classification.category.code-system	-	-	-	-	-	
draft-organisation.category-classifications.category-classification.category.code-system-name	-	-	-	-	-	
draft-organisation.category-classifications.category-classification.category.code	O	-	O	-	O	
draft-organisation.category-classifications.category-classification.category.display-name	-	-	-	-	-	"-1"
draft-organisation.category-classifications.category-classification.category.code-version	-	-	-	-	-	
draft-organisation.category-classifications.category-classification.valid-from	-	-	-	-	-	
draft-organisation.category-classifications.category-classification.valid-to	-	-	-	-	-	
draft-organisation.name	M	-	M	-	M	
draft-organisation.acronym	O	-	O	-	O	
draft-location.location-id.id	-	-	-	M	M	
draft-location.location-id.href	-	-	-	-	-	"http://"
draft-location.location-id.rel	-	-	-	-	-	
draft-location.mappings.mapping.code-system	M (only if code provided)	M (only if code provided)	-	M (only if code provided)	M (only if code provided)	
draft-location.mappings.mapping.code-system-name	-	-	-	-	-	"-1"
draft-location.mappings.mapping.code	M (only if code-system provided)	M (only if code-system provided)	-	M (only if code-system provided)	M (only if code-system provided)	
draft-location.mappings.mapping.valid-from	-	-	-	-	-	
draft-location.mappings.mapping.valid-to	-	-	-	-	-	
draft-location.is-	-	-	-	-	-	

headquarters						
draft-location.organisation-id.id	-	M	-	M	M	
draft-location.organisation-id.href	-	-	-	-	-	"-1"
draft-location.organisation-id.rel	-	-	-	-	-	
draft-location.address.po-box	-	-	-	-	-	
draft-location.address.postal-code	O	O	-	O	O	
draft-location.address.country.code-system	-	-	-	-	-	
draft-location.address.country.code-system-name	-	-	-	-	-	
draft-location.address.country.code	M	M	-	M	M	
draft-location.address.country.display-name	-	-	-	-	-	
draft-location.address.country.code-version	-	-	-	-	-	
draft-location.address.gps-location	-	-	-	-	-	
draft-location.address.address-details.address-detail.lang	-	-	-	-	-	
draft-location.address.address-details.address-detail.address-line-1	M	M	-	M	M	
draft-location.address.address-details.address-detail.address-line-2	O	O	-	O	O	
draft-location.address.address-details.address-detail.address-line-3	O	O	-	O	O	
draft-location.address.address-details.address-detail.address-line-4	O	O	-	O	O	
draft-location.address.address-details.address-detail.city	O	O	-	O	O	
draft-location.address.address-details.address-detail.state	O	O	-	O	O	
draft-location.address.address-details.address-detail.county	O	O	-	O	O	
draft-location.address.address-details.address-	O	O	-	O	O	

detail.region					
draft-location.communication-details.communication-detail.communication-details-type	-	-	-	-	-
draft-location.communication-details.communication-detail.email-addresses.email-address.	O	O	-	O	O
draft-location.communication-details.communication-detail.phone-numbers.phone-number.country-prefix	O	O	-	O	O
draft-location.communication-details.communication-detail.phone-numbers.phone-number.number	M (only if country-prefix provided)	M (only if country-prefix provided)	-	M (only if country-prefix provided)	M (only if country-prefix provided)
draft-location.communication-details.communication-detail.phone-numbers.phone-number.extension	O	O	-	O	O

Note: Services are not provided to update or delete change-requests, they can only be created.

Resource Information

End Point	POST /v{version}/change-requests-oms
Request	
Accept	application/xml application/json
Body	<change-request-oms>
Content-Type	application/xml application/json
Response	
Body	<change-request-oms>

Path Parameters

Name	Description
version	Service version number **Example value:** 1

Query Parameters

Name	Description	MD	WC	MV	Default

Example Request

```
POST /v1/change-requests-oms
```

7. Resources

7.1. *List*

List is a collection of related terms, e.g. countries list, units of measure list. List has additional properties like description, list owner and other. RMS manages multiple lists.

List schema definitions are in the attached XML Schema.

This schema is authoritative source of information on resource properties for this API.

List properties can be represented as:
Collection of summary list information   Global element: \<list-of-list>
Detailed information on a single list   Global element: \<list-details>

Please see Annex II on guidance on how to navigate between above elements.

Summary list information has following properties (non-exhaustive, see schema for details):

Property	Card.	Type	Description
\<list-id>	[1]	Complex	List identifier and a link to list detailed information Format: nnnnnnnnnnnn Example: 100000155052
\<list-name>	[1]	String	Name of the list
\<list-short-name>	[0..1]	String	Short name of the list
\<list-status>	[1]	Enumeration	Status of the list
\<terms-link>	[1]	Complex	Link to a complete collection of terms for a given list
\<terms-summaries-link>	[1]	Complex	Link to a collection of terms summaries for a given list
\<visibility>	[1]	Enumeration	Data visibility classification of the list
\<domains>	[0..1]	Collection	Domains for which the list is defined
\<list-operational-attributes>	[1]	Complex	Operational and lifecycle attributes of the list
\<list-documents>	[0..1]	Complex	Collection of documents associated with the list
\<list-description>	[0..1]	String	Description of the list
\<owner>	[0..1]	String	The entity owning the vocabulary and its values
\<version-provided-by-owner>	[0..1]	String	The version under the owner published a given list

Detailed list information extends the above by (non-exhaustive, see schema for details):

Property	Card.	Type	Description
\<oid>	[0..1]	anyURI	Globally unique object identifier defined for the list
\<list-level-definition>	[0..1]	Complex	Definitions of named list levels for a given list. Note: individual terms will be assigned to lists levels using list-level assignment element in controlled-term
\<extended-relationship-definitions>	[0..1]	Complex	Collection of definitions of relationships to other lists
\<extended-attributes-definitions>	[0..1]	Complex	Collection of extended attribute definitions defined for a given list

7.2. *Term*

Term is a word or phrase used to describe a thing or to express a concept. All RMS terms have a name in English and possible translations of this name to other languages. Related terms are grouped into lists.

Term schema definitions are in the attached XML Schema.

This schema is authoritative source of information on resource properties for this API.

Term properties can be represented as:

Collection of summary terms information from a single list. Contains additional information on	Global element: <controlled-terms-list-summary>
Collection of controlled terms details from a single list	Global element: <controlled-terms-list>
Collection of controlled term details or summaries from multiple lists	Global element: <controlled-terms-collection>
Details of an individual controlled term	Global element: <term-details>

Please see Annex II on guidance on how to navigate between above elements.

Summary term information has the following properties (non-exhaustive, see schema for details):

Property	Card.	Type	Description
<term-id>	[1]	Complex	Agreed European Identifier, this will be a unique sequential number for each term. Format: nnnnnnnnnnnn Example: 100000155052
<term-names>	[1]	Collection	Term names in different languages. Max. one name per language.
<short-names>	[0..1]	Collection	Short term names in different languages. Max. one name per language.
<status>	[1]	Enumeration	The status of the term
<children-link>	[0..1]	Collection	Link to a collection of direct children of this term.

Detailed term information extends the above by (non-exhaustive, see schema for details):

Property	Card.	Type	Description
<operational-attributes>	[1]	Complex	Operational and life cycle attributes
<visibility>	[1]	Enumeration	Domains for which the term is defined (e.g. human, veterinary)
<domains>	[0..1]	Collection	Domains for which the term is defined
<other-names>	[0..1]	Collection	Other term names in different languages. Multiple names allowed per language
<term-descriptions>	[0..1]	Collection	Term description in different languages. Max. one description per language
<term-comments>	[0..1]	String	Comments to the term
<current-term-ids>	[0..1]	Collection	References to current terms replacing a term with the status NULLIFIED or NON_CURRENT. The implementing system is responsible to update its local records in order to start using the current terms instead of the deactivated one.
<parents>	[0..1]	Collection	References to the parents of this controlled term in the hierarchical structure
<children>	[0..1]	Collection	References to the children of this controlled term in the hierarchical structure
<term-paths>	[0..1]	Collection	Paths from this term to a root of its hierarchical structure. For some lists multiple paths are possible.

<related-terms>	[0..*]	Collection	Other terms related to this term as defined in the extended relationships of the list definition.
<extended-attributes>	[0..1]	Collection	Extended attributes of this term as defined in the list definition.
<mappings>	[0..1]	Collection	Mappings of this controlled term to codes and identifiers in systems other than RMS.
<term-applicability>	[0..*]	Complex	Applicability of this term to some specific context, e.g. applicable to a particular country or applicable to a particular IT system.
<list-level-assignment>	[0..1]	Complex	Information to which named list level a given term is assigned. Note: definition of named list levels defined for a list are in list-level-definition element of the list-information.
<symbols>	[0..1]	Collection	Term symbols
<unit-of-measure-conversions>	[0..1]	Collection	Conversion formulas between different units of measure. Applicable only for units of measure (UoM)
<user-preferences>	[0..1]	Complex	User specific preferences for this term (e.g. tag assignments or user preferred name).
<terms-link>	[0..1]	Complex	Reference to the collection of term details in a given list
<terms-summaries-link>	[0..1]	Complex	Reference to the collection of term summaries in a given list
<list-reference>	[0..1]	Complex	Reference to the list details
<versions>	[0..*]	Complex	Contains a collection of historical representations of the term. A historical representation is created whenever a term is modified, where the historical representation represents the state of the organisation prior to the modification. Note, the collection includes all historical representations (previous versions) and the current representation (current version).

7.3. *Translation*

Translation is a representation of the term's name or other translatable names/descriptions in other languages than English.

Translation schema definitions are in the attached XML Schema.

This schema is authoritative source of information on resource properties for this API.

Translation properties can be represented as:
Collection of translations   Global element: <translations>
Individual translation details   Global element: <translation>

Property	Card.	Type	Description
<term-ref>	[1]	Complex	Reference to a term being translated.
<list-ref>	[0..1]	Complex	Reference to a list containing the term being translated.
<translation-language>	[0..1]	String	Language of the translation
<term-name>	[0..*]	Complex	Collection of translated term names.
<short-name>	[0..*]	Complex	Collection of translated term short names.
<other-name>	[0..*]	Complex	Collection of translated term other names.
<description>	[0..*]	Complex	Collection of translated term descriptions.

7.4. *Change Request RMS*

Change Request provides details and additional justification and information of requested change to create and update lists and to create, update and delete terms.

Change Request schema definitions are in the attached XML Schema.

This schema is authoritative source of information on resource properties for this API.

Change Request properties can be represented as:

Collection of Change Requests	Global element: <change-requests>
Individual Change Request details	Global element: <change-request>

Property	Card.	Type	Description
<request-id>	[0..1]	Complex	Unique change request identifier and link to the change request resource. Format: RRQ-nnnnnnnnn Example: RRQ-100000123
<name>	[1]	String	Name of the change request.
<type>	[1]	Enumeration	Type of the change request, e.g. add new term, or update existing term.
<status>	[1]	Enumeration	Current status of the change request
<status-comments>	[0..1]	String	Comments related to the current status.
<requestor-user-id>	[1]	String	Identifier of a user who made the change request
<requestor-organisation-id>	[0..1]	String	Identifier of an organisation to which the request belongs on of which behalf (s)he is making the request.
<requestor-email>	[1]	String	E-mail address of the requestor.
<requestor-phone>	[0..1]	String	Phone number of the requestor.
<date-submitted>	[0..1]	Date/time	Date and time when the request was submitted.
<justification>	[0..1]	String	Change request justification
<status-change>	[0..*]	Collection	History information on change request status changes over time.
<document>	[0..*]	Collection	Collection of documents attached to the change request.
<list-ref>	[0..1]	Complex	Reference to a list for which the change request has been defined.
<draft-term>	[0..1]	Complex	Details of the term that is subject of the change request.

7.4.1. Change Request RMS: identifier retention

As described in 6.4. Change Request RMS Service, any Update or Delete Term Change Request for RMS must retain the identifiers of any structure within controlled-term as obtained from the system. These identifiers must be provided back in the draft-term structure of the change request. Otherwise, the provided data shall be treated as new and may cause the request to be refused as incorrect. Specifically, the identifiers to keep are rowid and translation-id.

The detailed list of elements whose rowid must be kept is described here. Note that it may refer to a collection of repeatable items, each of which must retain its own rowid. All properties can be found under the root element change-request-rms.draft-term:

Property
rowid
status.rowid
domains.domain.rowid
current-term-ids.current-term-id.rowid

Property
parents.parent-term.rowid
symbols.symbol.rowid
mappings.mapping.rowid
term-applicability.applicable-to.rowid
extended-attributes.extended-attribute.rowid
unit-of-measure-conversions.unit-of-measure-conversion.rowid
related-terms.related-term.rowid

The detailed list of elements whose translation-id must be kept is described here. Note that it may refer to a collection of repeatable items, each of which must retain its own rowid. All properties can be found under the root element change-request-rms.draft-term:

Property
term-names.term-name.translation-id
short-name.short-name.translation-id
other-names.other-name.translation-id
term-descriptions.term-description.translation-id

7.5. *Document*

Documents provide additional information on lists, change requests, and general information and help.

Document schema definitions are in the attached XML Schema.

This schema is authoritative source of information on resource properties for this API.

Document properties can be represented as:

Collection of documents	Global element: \<documents\>
Individual document details	Global element: \<document\>

Property	Card.	Type	Description
\<document-id\>	[0..1]	Complex	Unique document identifier and link to the document resource.
\<operational-attributes\>	[0]	Complex	Operational attributes of the document (creation and modification dates, etc.)
\<application-domain\>	[1]	Enumeration	Indicates for which SPOR application domain the document is valid, e.g. RMS, OMS
\<type\>	[1]	Enumeration	Type of document, e.g. GENERAL, TECHNICAL
\<name\>	[1]	String	Name of the document
\<description\>	[0..1]	String	Description of the document
\<version\>	[0..1]	String	Business version as provided by the document creator
\<published-date\>	[0..1]	String	Official publication date as provided by the document creator
\<file-name\>	[1]	String	Name of the file related to the document.
\<mime-type\>	[1]	String	MIME resource type of the document, e.g. application/pdf
\<file-base64\>	[0..1]	Base 64 binary	Content of the document encoded as Base64 binary format. Only required for POST and PUT operations. Not returned for GET operation.
\<resource\>	[0..1]	Complex	Reference to a resource to which the document is related to. The resource type is determined by the document \<type\>, e.g. list, change request.
\<representation\>	[0..*]	Complex	Contains all available representations of the resource. Will contain representations of the resource as: • application/[mime-type] - The document

			content, where mime-type is document.mime-type, e.g. application/pdf
			• application/base64 - Document content, base64 encoded
			• application/xml and application/json will contain only metadata about the document

7.6. *Search Query*

Search Query stores parameters of a user defined search query that can be reused in the future. It does not store query results.

Search Query schema definitions are in the attached XML Schema.

This schema is authoritative source of information on resource properties for this API.

Search Query properties can be represented as:
Collection of search queries   Global element: <search-queries>
Individual search query details   Global element: <search-query>

Property	Card.	Type	Description
<search-query-id>	[0..1]	Complex	Unique search query identifier and link to the search query resource.
<user-id>	[1]	String	Identifier of a user who defined a search query.
<name>	[1]	String	Name of a search query
<search-type>	[1]	Enumeration	Type and target of the search query: • TERM • LIST
<search-criterion>	[0..*]	Collection	Collection of search criteria with parameter names and values.
<operational-attributes>	[0]	Complex	Operational attributes of the search query (creation and modification dates, etc.)

7.7. *Subscription*

Subscription is user's arrangement to receive notifications on modifications of lists.

Subscription schema definitions are in the attached XML Schema.

This schema is authoritative source of information on resource properties for this API.

Subscription properties can be represented as:
Collection of subscriptions   Global element: <subscriptions>
Individual subscription details   Global element: <subscription>

Property	Card.	Type	Description
<subscription-id>	[0..1]	Complex	Unique subscription identifier and link to the subscription resource.
<subscription-email>	[1]	String	Email address to which subscription notifications should be delivered.
<subscribe-to-new-lists>	[1]	Boolean	Indicates that the user wishes to be notified about creation of new lists.
<user-id>	[1]	String	Identifier of a user who created a subscription.
<subscribed-list>	[0..*]	Collection	Collection of lists for which subscription is defined. For each subscribed lists it is possible to indicate if notifications should be only for major or for all

			changes to the list.

7.8. *Preferred name*

Preferred name is a one of the term's names selected by a user as her/his preferred name.

A term must have an English name, may have at most one name in any other language, may have at most one short name in any language and zero or more other names (aliases) in any language. User can select one of these names as her/his preferred name.

Tag properties can be represented as:
Preferred name details   Global element: <preferred-name>

Property	Card.	Type	Description
<term-ref>	[1]	Complex	Reference to a term for which the preferred name is defined.
<user-id>	[0..1]	String	Identifier of a user who defined the preferred name.
<translation-id>	[1]	String	Reference to a particular name translation used for preferred name. It can refer to either term name, short name or other name

7.9. *Tag*

Tag is a user created label that can be attached to terms.
Tag schema definitions are in the attached XML Schema.

This schema is authoritative source of information on resource properties for this API.

Tag properties can be represented as:
Collection of tags   Global element: <tags>
Individual tag details   Global element: <tag>

Property	Card.	Type	Description
<tag-id>	[0..1]	Complex	Unique tag identifier and link to the tag resource.
<tag-name>	[1]	String	Tag text name.
<user-id>	[0]	String	Identifier of a user who defined a tag.
<term-refs>	[0..*]	Complex	References to terms that have been tagged with this tag.

7.10. *Organisation*

Organisation is a representation of a legal entity (e.g. a business, government department, regulatory body).

Organisation schema definitions are in the attached XML Schema.

This schema is authoritative source of information on resource properties for this API.

Organisation properties can be represented as:
Collection of organisations   Global element: <organisations>
Individual organisation details   Global element: <organisation>

Summary organisation information contains the following elements:

Property	Card.	Type	Description

Property	Card.	Type	Description
\<operational-attributes>	[1]	Complex	Operational attributes of the organisation (creation and modification dates, etc.)
\<organisation-id>	[1]	String	Organisation's unique identifier (Org ID). Format: ORG-nnnnnnnnn Example: ORG-100000123 Note, organisations may have multiple Org IDs associated with them if they have been subject to consolidation/merging. In this situation the oldest Org ID is exposed. If the implementing system is tracking the organisation by one of the other identifiers (in the \<identifiers> location), all actions will keep on working based on that identifier for compatibility reasons, but the system is strongly encouraged to start using this \<organisation-id> instead as soon as it has knowledge of it.
\<active-request>	[1]	Boolean	Indicates if a change request is currently active (in progress) for the organisation. Note, an organisation may only have one active change request at any one time.
\<identifiers>	[1..*]	Complex	Organisation's identifiers. Includes all Org IDs that are associated with the organisation, including the Org ID that has been exposed as the \<organisation-id>. The identifiers collection can also contain Request IDs that relate to the organisation (i.e. Change Requests that have been used to create/update the organisation). A Request ID is only included in the identifiers collection if it was used in the organisation request. For example: `GET /v1/organisations/ORQ-100000123`
\<status>	[1]	Enumeration	The status of the organisation. Possible values: ACTIVE INACTIVE
\<languages>	[0..*]	Complex	Organisation's languages.
\<name>	[1]	String	Organisation's name.
\<short-name>	[0..1]	String	Organisation's short name.
\<acronym>	[0..1]	String	Organisation's acronyms.
\<alternative-names>	[0..*]	Complex	Organisation's alternative names.
\<locations>	[1.*]	Complex	References to the locations associated with the organisation.

Detailed organisation information is only returned if summary=false. Extends the above by:

Property	Card.	Type	Description
\<mappings>	[0..*]	Complex	Identifiers from other systems/providers associated with the organisation (e.g. ECD IDs).
\<category-classifications>	[0..*]	Complex	Classifications that have been applied to the organisations. For example, "Organisation Type".
\<communication-details>	[0.*]	Complex	Communication details for the organisation.

Property	Card.	Type	Description
			Including:
			Email Addresses Phone Numbers Fax Numbers Websites Alternative postal addresses
<versions>	[0.*]	Complex	Contains a collection of historical representations of the organisation. A historical representation is created whenever an organisation is modified, where the historical representation represents the state of the organisation prior to the modification. Note, the collection includes all historical representations (previous versions) and the current representation (current version). <timestamp-from> and <timestamp-to> indicate the period that the representation was applicable. In the case of the current representation <timestamp-to> will be blank.
<version-timestamp>	[1]	Date/time	Only returned for individual organisation (<organisation>). Not returned in organisation collection (<organisations>). The date/time that the organisation's data is represented at. This will generally be the data/time the resource was requested, although can be a date/time in the past if the resource was requested via a timestamp parameter. For example: `GET /v1/organisations/ORG-100000123?timestamp=2016-01-01`

7.11. *Location*

Location is a representation of an address/site within an organisation. Each location is associated with one – and only one – parent organisation.

Location schema definitions are in the attached XML Schema.

This schema is authoritative source of information on resource properties for this API.

Location properties can be represented as:
Collection of locations   Global element: <locations>
Individual location details   Global element: <location>

Summary location information contains the following elements:

Property	Card.	Type	Description
<operational-attributes>	[1]	Complex	Operational attributes of the location (creation and modification dates, etc.)
<location-id>	[1]	String	Location's unique identifier (Org Location ID). Format: LOC-nnnnnnnnn Example: LOC-100000123

Property	Card.	Type	Description
			Note, locations may have multiple Org Location IDs associated with them if they have been subject to consolidation/merging. In this situation the oldest Org Location ID is exposed.
<active-request>	[1]	Boolean	Indicates if a change request is currently active (in progress) for the location. Note, a location may only have one active change request at any one time.
<identifiers>	[1..*]	Complex	Location's identifiers. Includes all Org Location IDs that are associated with the location, including the Org Location ID that has been exposed as the <location-id>. The identifiers collection can also contain Request IDs that relate to the location (i.e. Change Requests that have been used to create/update the location). A Request ID is only included in the identifiers collection if it was used in the location request. For example: `GET /v1/locations/ORQ-100000123`
<status>	[1]	Enumeration	The status of the location. Possible values: ACTIVE INACTIVE
<is-headquarters>	[0..1]	Boolean	Indicates if the location is the organisation's headquarters.
<organisation>	[1]	Complex	Summarised information relating to the location's parent organisation.
<address>	[1]	Complex	The location's primary postal address. English representation only. Postal addresses may be represented in multiple languages, subject to the languages supported by the local postal authority. There will always be an English representation of the address.

Detailed location information is only returned if summary=false. Extends the above by:

Property	Card.	Type	Description
<mappings>	[0..*]	Complex	Identifiers from other systems/providers associated with the organisation (e.g. DUNS IDs, GS1 IDs).
<address>	[1]	Complex	The location's primary postal address. All representations. Postal addresses may be represented in multiple languages, subject to the languages supported by the local postal authority. There will always be an English representation of the address.
<communication-details>	[0.*]	Complex	Additional communication details for the location. Including: Email Addresses Phone Numbers Fax Numbers Websites Alternative postal addresses

Property	Card.	Type	Description
<versions>	[0.*]	Complex	Contains a collection of historical representations of the location. A historical representation is created whenever a location is modified, where the historical representation represents the state of the organisation prior to the modification. Note, the collection includes all historical representations (previous versions) and the current representation (current version). <timestamp-from> and <timestamp-to> indicate the period that the representation was applicable. In the case of the current representation <timestamp-to> will be blank.
<version-timestamp>	[1]	Date/time	Only returned for individual location (<location>). Not returned in location collection (<locations>). The date/time that the location's data is represented at. This will generally be the data/time the resource was requested, although can be a date/time in the past if the resource was requested via a timestamp parameter. For example: `GET /v1/locations/LOC-100000123?timestamp=2016-01-01`

7.12. *Change Request OMS*

Change Request provides details and additional justification and information of requested changes to create and update organisations and locations.

Change Request schema definitions are in the attached XML Schema.

This schema is authoritative source of information on resource properties for this API.

Change Request properties can be represented as:
Collection of change requests   Global element: <change-requests>
Individual change request details   Global element: <change-request>

Summary change request information contains the following elements:

Property	Card.	Type	Description
<request-id>	[0..1]	Complex	Request's unique identifier. Format: ORQ-nnnnnnnnn Example: ORQ-100000123
<name>	[1]	String	Name of the change request.
<type>	[1]	Enumeration	Type of the change request. Possible values: ADD-ORGANISATION UPD-ORGANISATION ADD-LOCATION UPD-LOCATION UPD-ORG-AND-LOCATION
<status>	[1]	Enumeration	Current status of the change request. Possible values:

Property	Card.	Type	Description
			SUBMITTED VALID INVALID RETURNED APPROVED APPROVED WC REJECTED ON_HOLD
<status-comments>	[0..1]	String	Comments related to the current status.
<requestor-user-id>	[1]	String	Identifier of a user who made the change request
<requestor-organisation-id>	[0..1]	String	Identifier of an organisation to which the request belongs on of which behalf (s)he is making the request.
<requestor-email>	[1]	String	E-mail address of the requestor.
<requestor-phone>	[0..1]	String	Phone number of the requestor.
<date-submitted>	[0..1]	Date/time	Date and time when the request was submitted.
<justification>	[0..1]	String	Change request justification
<status-change>	[0..*]	Collection	History information on change request status changes over time.
<documents>	[0..*]	Collection	Collection of documents attached to the change request.
<address-language>	[1]	String	Language selected for the location address.
<request-reason>	[1]	Complex	Reason for the change request.
<draft-organisation>	[1]	Complex	Summary of the organisation that is the subject of the change request. Contains the following child elements: organisation-id name
<draft-location>	[1]	Complex	Summary of the location that is the subject of the change request. Contains the following child elements: location-id address

Detailed change request information is only returned if summary=false. Extends the above by:

Property	Card.	Type	Description
<draft-organisation>	[1]	Complex	Details of the organisation that is the subject of the change request. Contains the following child elements: organisation-id category-classifications name acronym
<draft-location>	[1]	Complex	Details of the location that is the subject of the change request. Contains the following child elements: location-id mappings organisation address communication-details

8. About this Document

8.1. *Definitions, Acronyms, and Abbreviations*

Acronym/Abbreviation	Description
API	Application Programming Interface
EUTCT	European Union Telematics Controlled Terms
HTTP	Hypertext Transfer Protocol
IANA	Internet Assigned Numbers Authority
JSON	JavaScript Object Notation
MIME	Multipurpose Internet Mail Extensions
OMS	Organisations Management System
SPOR	Substances, Products, Organisations, Referentials
RMS	Referentials Management System
REST	Representational State Transfer
XML	Extensible Mark-up Language

8.2. *Open Issues*

- Use of ETags for caching resources.

- Polling for updates vs notifications.

8.3. *Referenced documents*

Doc ID	Title	Location
EMA/399050/ 2009	EUTCT REST API Interfaces	http://eutct.emea.eu.int/eutct/iop/EUTCT_REST_Interfaces.pdf

8.4. *Document Approval*

Date	Version	Submitted by	Approved by	Approver Role

8.5. *Document history*

Version	Who	What
1.0	EMA	Creation for RMS consultation
1.1	EMA	Incorporate OMS & rename to SPOR
1.2	EMA	OMS services added. Metadata section added. Parameter characteristics section added.
1.3	EMA	Incorporate RMS consultation feedback Reformatting and standardisation.
1.4	EMA	Creation for OMS consultation
1.5	EMA	Incorporate OMS consultation feedback
1.6	EMA	Incorporate further OMS consultation feedback
1.7	**EMA**	Include additional information on: • rowid handling in RMS change requests • conditional fields in OMS/RMS change requests • OMS identifiers in a merge • OMS change request status compatibility Export terms by lang-country now supported Additional sorting options for search-terms internal-documents utility service added (EP511). Search-terms (EP14) domain parameter now multi-value lists (EP11) domain parameter now multi-value. EP101 – pagesize default=20, maximum = 200 EP111 – pagesize default=20, maximum = 200 EP501 – status parameter added
1.8	EMA	EP512 Created sortby parameter added to EP41
1.9	**EMA**	RMS Change Request identifier retention description has been refined
1.10	EMA	EP101 & EP102 – application/zip representations added EP111 & EP112 – organisation-details parameters added
1.11	EMA	EP41 – cr-name, cr-id and name parameters now support wildcards

9. Annexes

9.1. *Annex I – Record versioning in RMS and OMS*

RMS and OMS are implemented based on a commercial master data management solution (MDM), which comes with several off the shelf features for managing data. One of them is keeping a full track of all the changes made to the mastered data and ability to reconstruct the state of a data record at any time point during this record life time.

This model assumes that data record versions are based on timestamps and makes it possible to query a version of a record at any timestamp in the past. This approach works naturally with other important MDM feature, which is merging of separate records into one master record. For example, if two separate records A and B got merged into a new master record AB at a timestamp T1, it is relatively easy to query their different versions based on timestamps: all queries for versions of the record A before the timestamp T1 will return an appropriate version of record A; but all queries for versions of the record A after the timestamp T1 will return an appropriate version of the merged record AB.

A versioning model alternative to timestamps can be based on numeric versions, e.g. v. 1, v. 2, v. 3, etc. However, this model will not work well in scenarios where different records with own version numbers are merged into one new master record. For example, if two separate records like a record A in version 2 and a record B in version 19 are merged into a new record AB, what should the version of this new record be? Should it be version 1, since a new record was created? Should it be version 20, calculated as the maximum version number of merged records increased by 1? This leaves open questions about how to query the particular numeric version of a record. There also can be anomalies, like sudden increases of version number by more than 1.

RMS and OMS, by default, version all their mastered records based on timestamps. This approach works particularly well with OMS, where a large rate of records merges are expected. On top of timestamp based versions, RMS also offers versions based on numeric values. This is mainly for backward compatibility with an earlier system EUTCT. It is possible, because RMS business process does not allow for merge of different terms, but rather invalidates old terms and creates a new current term. Thus, the anomalies described above will not occur.

When referencing any record in RMS and OMS, a client has two options:

- Reference only record identifier (i.e. term identifier, organisation identifier or location identifier) – in this case any call back to RMS or OMS will always return the latest version of the referenced term.

- Reference a record identifier (i.e. term identifier, organisation identifier or location identifier) and a timestamp when the reference was made. Thanks to MDM features described above is will always be possible to extract the exact version of referenced record at this timestamp.

- In case of RMS only, a client can reference a term identifier and its version number.

It is up to the client implementation and its use case and business process to decide which model of referencing RMS and OMS records is used.

9.2. *Annex II – RMS Synchronisation Mechanism*

There two models of working with RMS possible:

- A client can use API calls directly to RMS to populate its local forms and data fields with codes and names from RMS. This approach does require frequent calls to remote RMS API endpoints.

- A client can make a local copy of relevant RMS lists and terms, and subsequently populate its local forms and data fields from this copy. This approach requires that the local copy periodically gets synchronised with the master data in RMS.

This section describes a synchronisation mechanism for the latter scenario.

1. The local copy of the RMS list has been created by copying the complete list content at a timestamp T1.

2. A follow up synchronisation process is triggered by the client at the timestamp T2 to receive all the terms that have been modified in the RMS in the period between timestamps T1 and T2.

3. Each subsequent synchronisation at the timestamp Tn+1 receives terms modified since the timestamp Tn, $n \geqslant 1$.

Activity diagram with relevant service calls and data flows is presented below:

**act RMS Synchronisation Using Timestamps**

Assumption:

Local copy already completely synchronised with RMS at the timestamp Tn.

This process synchronises only changes done to master data since last synchronisation, i.e. between timestamps Tn and Tn+1.

Initiate synchronisation
at timestamp Tn+1

e.g.
GET /lists/search_terms?modified-after=<Tn (last-sync-dateTime)>
&lists-<list-id>

(EP14) Search terms modified since last sync

<controlled-terms-collection>

Term summaries only

<controlled-terms-collection>

Extract identifiers of changed terms

<term-ids>

<term-ids>

(EP15) Export terms modified since last sync

<controlled-terms-collection>

Term details

<controlled-terms-collection>

Receive and process details of terms updated since last sync

End synchronisation
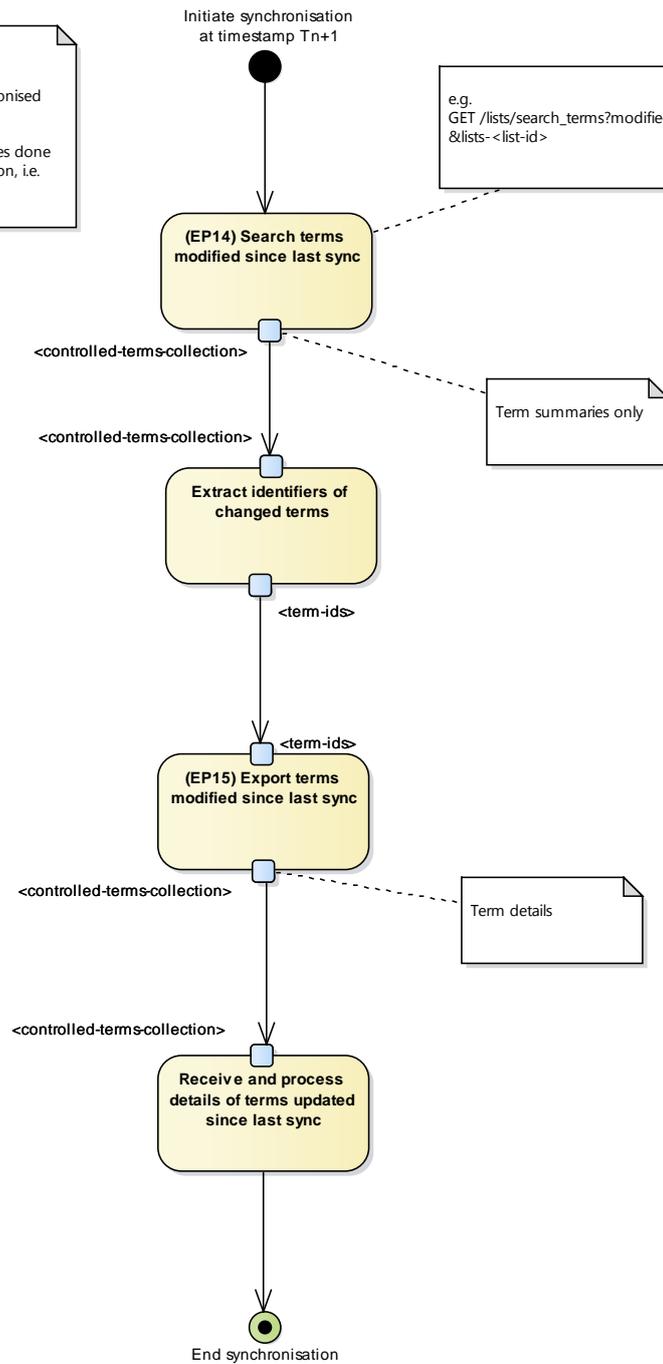
9.3. *Annex III – OMS Synchronisation Mechanism*

There two models of working with OMS possible:

- A client can use API calls directly to OMS to populate its local forms and data fields with organisations and locations from OMS. This approach does require frequent calls to remote OMS API endpoints.

- A client can make a local copy of relevant OMS data, and subsequently populate its local forms and data fields from this copy. This approach requires that the local copy periodically gets synchronised with the master data in OMS.

This section describes a synchronisation mechanism for the latter scenario.

1. The local copy of the OMS list has been created by copying the complete organisation/location data at a timestamp T1.

2. A follow up synchronisation process is triggered by the client at the timestamp T2 to receive all the organisations/locations that have been modified in the OMS in the period between timestamps T1 and T2.

3. Each subsequent synchronisation at the timestamp Tn+1 receives organisations/locations modified since the timestamp Tn, n≥1.

Activity diagram with relevant service calls and data flows is presented below:

act OMS Synchronisation Using Timestamps

Initiate synchronisation at timestamp Tn+1

●

Assumption:

Local copy already completely synchronised with OMS at the timestamp Tn.

This process synchronises only changes done to master data since last synchronisation, i.e. between timestamps Tn and Tn+1.

e.g.
GET /v{version}/organisations?org-modified-after=<Tn (last-sync-dateTime)>

(EP101) Search Organisations modified since last sync

<organisations>

<organisations>

Receive and process details of organisations updated since last sync
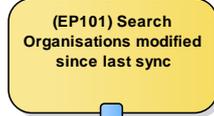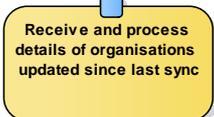
e.g.
GET /v{version}/locations?loc-modified-after=<Tn (last-sync-dateTime)>

(EP111) Search Locations modified since last sync

<locations>

<locations>

Receive and process details of locations updated since last sync
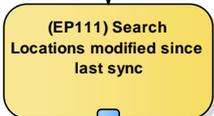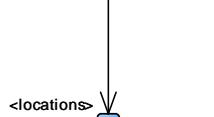
◉
End
syncronisation

9.4. *Annex IV – Navigation between Lists & Terms*
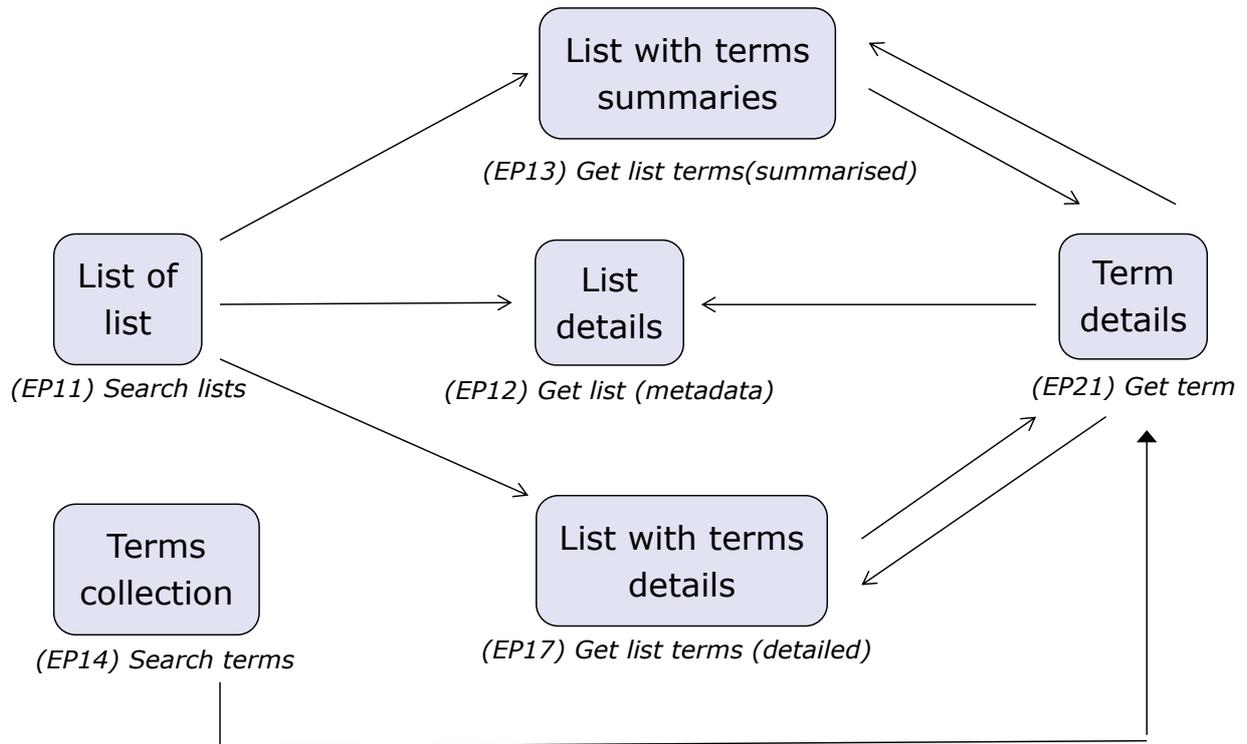
9.5. *Annex V – API Endpoint Summary*

Summary	Description	URL	SPOR Domain[10]
(EP11) Search lists	Use this operation to return a collection of lists, based on provided search criteria and supports server side paging	GET /v{version}/lists	**R**
(EP12) Get list (detailed)	Use this operation to return metadata information for a specific list, identified by its list-id. No terms information returned.	GET /v{version}/lists/{list-id}	**R**
(EP13) Get list terms(summarised)	Use this operation to return information for a specific list, identified by its list-id. Only summarised information is returned for terms contained in the list.	GET /v{version}/lists/{list-id}/term-summaries	**R**
(EP14) Search terms	Use this operation to return a collection of up to 1000 terms (term summaries only), based on provided search criteria. The operation supports searches across multiple lists and supports server side paging.	GET /v{version}/lists/search-terms	**R**
(EP15) Export terms	Use this operation to export a collection of up to 1000 term details, based on provided search criteria. The operation supports searches across multiple lists. For an authenticated user it will return details of user defined tags and preferred names for each returned term (if present). For unauthenticated users tags and preferred names will not be populated. The ZIP output will consist of a multiple CSV files, one per list. The requirements to export up to 1000 terms exceeds	POST /v{version}/lists/export-terms	**R**

---

[10] O=OMS specific, R=RMS specific

	the capabilities of a GET call, therefore the service has been implemented as a POST with a payload.		
(EP16) Associate a list and document	Use this operation to associate a document with a list. This service provides the functionality intentionally omitted from the 'document' interface, the ability to create the relationship between the list and a document. While PUT /documents indirectly has this capability it will be blocked behind the interface.	PUT /v{version}/lists/{list-id}/documents	**R**
(EP17) Get list terms (detailed)	Use this operation to return a list with a collection of its term details as XML or zipped csv / xml file.	GET /v{version}/lists/{list-id}/terms	**R**
(EP21) Get term	Use this operation to return information for a specific term, identified by its term-id.	GET /v{version}/lists/{list-id}/terms/{term-id}	**R**
(EP24) Get term mappings	Use this operation to translate an external term identifier into a collection of RMS identifiers.	GET /v{version}/lists/{list-id}/mappings	**R**
(EP31) Get values for translatable term attributes	Use this operation to return English values for translatable term attributes and the translations of those attributes for a specific term and language.	GET /v{version}/lists/{list-id}/terms/{term-id}/translations/{lang-country}	**R**
(EP32) Update the translatable term attributes for a specific list and language/country (via file)	Use this operation to update the translatable term attributes for terms in a specific list and for specified language/country. Allows a csv file encoded as base64 to be uploaded. The file contains English values and translations for translatable attributes for terms in a specific list. Validation errors returned as per standard HTML extended error code <error-notification>	PUT /v{version}/lists/{list-id}/translations/{lang-country}	**R**
(EP33) Update the translatable term attributes for a specific term and language/country	Use this operation to update the translatable term attributes for a specific term and language/country.	PUT /v{version}/lists/{list-id}/terms/{term-id}/translations/{lang-country}	**R**

(EP41) Search change requests rms	Use this operation to return a collection of change-requests, based on provided search criteria.	GET /v{version}/change-requests-rms	**R**
(EP42) Get change request rms	Use this operation to return information for a specific change request, identified by its change-request-id.	GET /v{version}/change-requests-rms/{change-request-id}	**R**
(EP43) Create change request rms	Use this operation to create a new change request. A change request may be of the following types: "add term", "update term", "delete term", "add list", "update list". Change requests relating to lists (i.e. "add list", "update list") have only change request attributes and details of associated documents. Change requests relating to terms (i.e. "add term", "update term", "delete term") have change request attributes, optionally details of associated documents and controlled term. For "delete term" change requests, an alternative/replacement term(s) may be specified for the term being deleted. Change requests may or may not contain a number of elements depending on the type of request. Those conditional elements are change-request-rms.draft-term, change-request-rms.draft-term.term-id and change-request-rms.list-ref. The following table shows when they are required:	POST /v{version}/change-requests-rms	**R**

Change Request Type	change-request-rms.list-ref
Add list	No
Update list	Yes
Add term	Yes
Update term	Yes
Delete term	Yes

	Once a change request creation is processed by the server, the response shall contain the change-request-rms.request-id assigned to such change request. It is the responsibility of the implementing system to store this request-id in order to provide it again in any update action taken onto the created change request.		
	In the case of term related change requests, the change-request-rms payload that must be passed back to system must retain a number of identifiers that belong to the existing term. The implementing system is responsible to obtain them before creating an update or delete change request, and provide them back in the payload of the change request. Such identifiers are part of the returned controlled-term structure and must be provided as part of the draft-term structure. For an exhaustive list of identifiers to store, please see below 7.4. Change Request RMS.

Note: Services are not provided to create/update/delete Lists and Terms directly. All such operations are facilitated via change requests services. | | |
| (EP44) Update change request rms | Use this operation to update an existing change request.

A change request may be of the following types: "add term", "update term", "delete term", "add list", "update list".

An update on any type of change request must contain its request-id. Otherwise the change request shall be treated as new.

Change requests relating to lists (i.e. "add list", "update list") have only change request attributes and details of associated documents. | PUT /v{version}/change-requests-rms/{change-request-id} | **R** |

	Change requests relating to terms (i.e. "add term", "update term", "delete term") have change request attributes plus optional details of associated documents and controlled term. When an update on an already created change request relating to terms is executed, the relevant draft-term identifiers previously obtained in the change request creation must be retained, that is, provided again. Any new element within the change request must render no identifier. However, once the response is obtained, the new identifiers assigned by the system to those elements must also be stored in order to provide them back in the case of a new change request update. For an exhaustive list of identifiers to store, please see below 7.4. Change Request RMS.

For "delete term" change requests an alternative/replacement term(s) may be specified for the term being deleted.

Note: This operation does not allow updating documents uploaded with the previous version of the change request. It allows only deleting an existing and adding a new document to the change request.

Note: Services are not provided to create/update/delete Lists and Terms directly. All such operations are facilitated via change requests services. | | |
| (EP45) Delete change request rms | Use this operation to delete an existing change request.

Deletion is only allowed if the Change Request has status 'Saved'. Once the change request is submitted (status is different than 'Saved'), the change request cannot be deleted.
"Status" refers to the where the Change Request is in the saved/submitted/approve\|rejectlife-cycle | DELETE /v{version}/change-requests-rms/{change-request-id} | **R** |

(EP51) Create document	Use this operation to create (upload) a new document. The method allows the content of the file to be uploaded as base64 and creates the file and its' metadata.	POST /v{version}/documents	**OR**
(EP52) Update document	Use this operation to update an existing document. The operation updates the document metadata and replaces the document file. The content of the document file is uploaded as base64. Note, this operation replaces the existing document file, multiple versions of the document file are not maintained. While PUT /documents indirectly has the capability to associate the document with a list it will be blocked behind the interface as it is provided for in `PUT /lists/{list-id}/documents.`	PUT /v{version}/documents/{document-id}	**OR**
(EP53) Delete document	Use this operation to delete an existing document and its associated metadata.	DELETE /v{version}/documents/{document-id}	**OR**
(EP54) Get document	Use this operation to return information for a specific document, identified by its document-id. The operation can be used to return document metadata or document content.	GET /v{version}/documents/{document-id}	**OR**
(EP55) Search documents	Use this operation to return a collection of documents, based on provided search criteria.	GET /v{version}/documents	**OR**
(EP61) Get all search queries	Use this operation to return all search queries that the user has created.	GET /v{version}/search-queries	**R**
(EP62) Create search query	Use this operation to create a new search query.	POST /v{version}/search-queries	**R**

(EP63) Get search query	Use this operation to return information for a specific search query, identified by its search-query-id.	GET /v{version}/search-queries/{search-query-id}	**R**
(EP64) Update search query	Use this operation to update an existing search query.	PUT /v{version}/search-queries/{search-query-id}	**R**
(EP65) Delete search query	Use this operation to delete an existing search query.	DELETE /v{version}/search-queries/{search-query-id}	**R**
(EP71) Get all Subscriptions	Use this operation to return all the lists subscriptions that the user has created. As a system call all user subscriptions are returned ordered by last-notification-time.	GET /v{version}/subscriptions	**R**
(EP72) Get all Subscriptions from a specific list	Use this operation to return all subscriptions for a specific list, identified by its list-id.	GET /v{version}/lists/{list-id}/subscriptions	**R**
(EP73) Create / update / delete user subscriptions	Use this operation to create / update / delete subscriptions for the user that sends the request. While not strictly RESTful, this operation has been designed to merge user list subscription changes into the database as a set to minimise individual calls.	PUT /v{version}/subscriptions	**R**
(EP81) Get all tags	Use this operation to return all tags that the user has created, optionally by term-id and tag-name	GET /v{version}/tags	**R**
(EP82) Get tag	Use this operation to return information for a tag, identified by its tag-id.	GET /v{version}/tags/{tag-id}	**R**
(EP83) Create tag	Use this operation to create a new tag.	POST /v{version}/tags	**R**
(EP84) Update tag	Use this operation to update an existing tag.	PUT /v{version}/tags/{tag-id}	**R**
(EP85) Delete tag	Use this operation to delete an existing Tag.	DELETE /v{version}/tags/{tag-id}	**R**
(EP91) Get preferred name	Use this operation to get a user's preferred name for a specific term.	GET /v{version}/lists/{list-id}/terms/{term-id}/preferred-name	**R**

(EP92) Create preferred name	Use this operation to create a user's preferred name for a specific term.	POST /v{version}/lists/{list-id}/terms/{term-id}/preferred-name	**R**
(EP93) Update preferred name	Use this operation to update a user's preferred name for a specific term.	PUT /v{version}/lists/{list-id}/terms/{term-id}/preferred-name	**R**
(EP94) Delete preferred name	Use this operation to delete a user's preferred name for a specific term.	DELETE /v{version}/lists/{list-id}/terms/{term-id}/preferred-name	**R**
(EP101) Search organisations	Use this operation to return a collection of organisations, based on provided search criteria.	GET /v{version}/organisations	**O**
(EP102) Get organisation	Use this operation to return information for a specific organisation, identified by one of its identifiers (see id parameter for supported identifier types).	GET /v{version}/organisations/{id}	**O**
(EP111) Search locations	Use this operation to return a collection of locations, based on provided search criteria.	GET /v{version}/locations	**O**
(EP112) Get location	Use this operation to return information for a specific location, identified by one of its identifiers (see id parameter for supported types).	GET /v{version}/locations/{id}	**O**
(EP121) Search change requests oms	Use this operation to return a collection of OMS change requests, based on provided search criteria.	GET /v{version}/change-requests-oms	**O**
(EP122) Get change request oms	Use this operation to return information for a specific OMS change request, identified by its change-request-id.	GET /v{version}/change-requests-oms/{change-request-id}	**O**
(EP123) Create change request oms	Use this operation to create a new change request.	POST /v{version}/change-requests-oms	**O**