

IUCLID 6 Public API

Legal Notice

The information in this document does not constitute legal advice. Usage of the information remains under the sole responsibility of the user. The European Chemicals Agency does not accept any liability with regard to the use that may be made of the information contained in this document.

Title: IUCLID 6 Public API

Issue date: October 2020

Language: en

IUCLID 6 is developed by the European Chemicals Agency in association with the OECD.

© European Chemicals Agency, 2020

Reproduction is authorised provided the source is fully acknowledged in the form

“Source: European Chemicals Agency, <http://echa.europa.eu/>”, and provided written notification is given to the ECHA Communication Unit (publications@echa.europa.eu).

If you have questions or comments in relation to this document, please send them to ECHA via the information request form at the address below, quoting the reference and issue date given above:

<http://echa.europa.eu/contact>

European Chemicals Agency

Mailing address: P.O. Box 400, FI-00121 Helsinki, Finland

Visiting address: Telakkakatu 6, FI-00150 Helsinki, Finland

Contents

1. Overview	5
1.1. Media Types	5
1.1.1. IUCLID Document media types	5
1.2. HTTP methods and status codes	6
1.2.1. Semantics of HTTP methods	6
1.2.2. Semantics of HTTP status codes	6
1.3. Authorization	8
1.4. Base URL	9
1.5. About	9
2. Definitions	11
2.1. Introduction	11
2.2. Legislation providers	11
2.3. Submission types (obsolete since v5.0.0)	12
2.4. Working contexts (since v5.0.0)	15
2.5. Trees (since v5.0.0)	20
2.5.1. Listing Trees	20
2.5.2. Loading a single tree	21
2.6. Documents	25
2.7. Phrase-groups	30
2.8. Phrases	31
2.9. Text templates	32
3. Search.....	34
3.1. Results	35
3.1.1. Document Secured Representation	38
3.1.2. Ordering	39
3.2. Available queries	39
3.2.1. byDate	40
3.2.2. byName	41
3.2.3. byContact	41
3.2.4. bySubstance	42
3.2.5. byMixture	43
3.2.6. byLegalEntity	43
3.2.7. byReferenceSubstance	44
3.2.8. byTemplate	45
3.2.9. byLiterature	45
3.2.10. byAnnotation	46

3.2.11. bySite.....	46
3.2.12. byAdditive	47
3.2.13. byImpurity	48
3.2.14. byConstituent	49
3.2.15. byGhs	49
3.2.16. byDsd	50
3.2.17. byJointSubmission.....	51
3.2.18. byCategory (since v2.0.0).....	51
3.2.19. SubstanceRelatedCategories	52
3.2.20. SubstanceRelatedMixtures	52
4. Document handling.....	53
4.1. Datatypes.....	53
4.1.1. JsonDocumentEnvelope	53
4.1.2. Response types.....	60
4.2. Semantics of HTTP status codes	61
4.2.1. Success codes	61
4.2.2. Error codes	61
4.3. Load	62
4.3.1. Raw data	62
4.3.2. Dossier.....	64
4.4. Write raw data	66
4.4.1. Entities.....	67
4.4.2. Composite entities.....	70
4.4.3. Sections.....	73
4.4.4. Attachments.....	74
4.5. Create Dossier.....	74
4.6. Delete Dossier.....	78
4.7. Links (since v3.0.0)	78
4.7.1. Annotations	78
4.7.2. Templates	81
5. Entity access.....	84
5.1. Introduction	84
5.2. Datatypes.....	84
5.3. Semantics of HTTP status codes	85
5.3.1. Success Codes.....	85
5.3.2. Error codes	85
5.4. Retrieve data access.....	85
5.5. Modify data access	86

6. Engines.....	88
6.1. Supported Types.....	88
6.1.1. Entities.....	88
6.1.2. Dossiers.....	88
6.2. Export.....	89
6.2.1. Export entity.....	90
6.2.2. Export dossier.....	91
6.2.3. Customization of the export operation.....	92
6.3. Import.....	94
6.4. Validation assistant.....	105
6.4.1. Raw entities.....	106
6.4.2. Dossiers.....	109
6.5. Dossier comparison (since v3.0.0).....	109
6.6. Report generation (since v3.0.0).....	110
6.6.1. Document Printing.....	110
6.6.2. CSR Report.....	112
6.7. Dossier filtering (since v3.0.0).....	113
7. Inventories.....	118
7.1. Results.....	118

1. Overview

The IUCLID 6 Public REST API defines a unified and uniform URI space to access and manipulate the abstract resources comprising the IUCLID system.

This means, in practice, that each resource is identified by a single URI, and that behaviour and representation are determined by standard HTTP mechanisms, such as the HTTP method and the media-type based content negotiation.

For example, the same document URI may be used to retrieve a JSON representation of a document, export it in i6d format, print it in HTML format through a simple GET with a different Accept header.

In addition, new representations and behaviour can be added to the IUCLID Public REST API through a well defined extension mechanism. The primary extension point of the API is through the extension media type. The extension media type serves as a marker to convey alternative representations of the IUCLID data and trigger custom handlers for each of the supported endpoints.

In some situation, such as custom queries or reports, the URI space makes room for accessing named extension components within the semantic boundaries of the general API.

The IUCLID Public REST API strives to expose a more business-oriented view of the IUCLID resources, avoiding shortcuts that could be enabled by the technical characteristics of the IUCLID data model.

In particular, the hierarchical part of the URIs reflects the parent-child and ownership relations of the resources and their components, when a more flat URI space could have been supported, given the pervasive use of UUID across the IUCLID domain model. Similarly, different business entities, such as for example substances and mixtures, have their own sub-trees in the hierarchical structure, even though very little distinguishes them from a technical perspective thanks to the flexibility of the IUCLID model.

1.1. Media Types

Media types play a central role in the IUCLID Public REST API. Leveraging standard HTTP content negotiation, they affect both behaviour and representation, and support the extension mechanism of the API.

As such, the IUCLID Public REST API makes use of a wide range of media types.

Note that not all media types are available for all resources or all the methods, as each media types may only support a subset of the API semantics.

1.1.1. IUCLID Document media types

The IUCLID Document media types correspond to the standard representation of IUCLID documents, and other associated entities.

JSON	application/vnd.iuclid6.ext+json;type=iuclid6.Document
HTML	text/html

1.2. HTTP methods and status codes

The IUCLID Public REST API is tightly integrated with the underlying HTTP protocol. This implies that the semantics of the functional operations supported on resource + media type combinations are mapped to appropriate HTTP methods and respect the specification of these methods <https://tools.ietf.org/html/rfc7231>. Similarly, the status codes used by the API are aligned with the HTTP specification and map the functional semantics of the IUCLID Public REST API execution results.

1.2.1. Semantics of HTTP methods

API function	HTTP method	Scope
<i>list</i>	GET	Collections of resources. Collections can be filtered using named filters, and paged.
<i>search</i>	GET	Named query resources.
<i>read</i>	GET	Individual resources (however coarse or fine grained)
<i>create / add</i>	POST	Collections of resources. A POST maps to an addition to the collection i.e. a creation
<i>update / edit</i>	PUT	Individual resource. A PUT maps to an update of the resource
<i>delete / remove</i>	DELETE	Individual resource. Deletion of collections is not supported
<i>import</i>	POST	Collections of resources. Import is a variant of creation or update
<i>export</i>	GET	Individual resource. Export is a variant of load
<i>print</i>	GET	Individual resources. The print format is determined by the media type requested by the client.

1.2.2. Semantics of HTTP status codes

1.2.2.1. Success codes

HTTP Status Code	Usage
200 OK	Success of data retrieval operations such as <i>read</i> , <i>list</i> , <i>search</i> , <i>export</i> , <i>print</i>
200 OK	Success of update operations, such as <i>update</i> , <i>edit</i> , <i>import</i>
204 No Content	Success of update or deletion operations, when no entity is included in the response, e.g. <i>delete</i> , <i>remove</i>

201 Created	Success of creation operations such as <i>create</i> , <i>add</i> , <i>import</i> (if the import results in the creation of a new entity)
202 Accepted	POST operations may return this status code if a background job was launched as a result of the request, e.g. for <i>import</i> or <i>create dossier</i> . See below for polling.
308 Permanent Redirect	Used for generic URI that don't change, such as <code>/dossier/{uuid}/subject</code>
303 See Other	Result of a background job can be accessed through GET at the URL specified in the Location header.
307 Temporary Redirect	Used for generic URI that may evolve, such as <code>/definition/document/{def_id}/latest</code>

1.2.2.2. Error codes

HTTP Status Code	Usage
400 Bad Request	The request entity or request parameters is not parsable by the server, either because it is malformed at the protocol level, or because it is malformed at the format level (XML, JSON, number or date representation).
403 Forbidden	The caller is attempting to perform an operation on a resource he is allowed to see (otherwise 404 would be appropriate) but hasn't sufficient privileges.
404 Not Found	If the resource identified by the URI path does not exist (at any depth), or a resource identified in the query string that is essential to the representation does not exist.
405 Method Not Allowed	If the method is not supported by the target resource, regardless of the privileges of the caller. E.g. update a dossier document, POST on a named query
406 Not Acceptable	If the requested media type is not supported for the resource or not known.
409 Conflict	To indicate a conflict of the request with the current state of the resource. E.g. update a document based on a version of it that is not the latest (the document has been updated in the meantime)
415 Unsupported Media Type	If the resource cannot handle the media type sent by the caller.
422 Unprocessable Entity	To indicate business validation errors on the received data. Unless those are already covered by 400, 403 or 404.
500 Internal Server Error	No particular business meaning can be associated to this code, as it is the result of an internal unexpected exception.

Usage of error status codes and response entities also take into account security considerations.

- 404 is preferred to 403 if the caller has insufficient privileges to even see the resource (i.e. perform a simple GET with any content type). In other words, for that particular user the system should behave as if the resource did not exist at all.
- The response entity in the case of an Internal Server Error (500) is kept completely general, and does not include any information about the reason of the failure, as this may disclose weaknesses of the server.

1.2.2.3. Polling

Access to some resources may require asynchronous processing on the server side, aka background jobs.

The IUCLID Public REST API does not specify exactly which calls may necessitate such processing.

Instead, it specifies a generic mechanism for informing the client, and the interaction patterns that a client should be prepared to handle.

- If the server chooses to launch a background job to satisfy a given request, it **MUST**
 - inform the client by returning a status code of 202 Accepted;
 - include a Location header (this part is non-standard) with the URI of the monitoring resource for the created job.
- The client should call the monitoring URI with a GET to retrieve progress information.
- When servicing the monitoring URI,
 - If the job is not completed, the server **MUST**
 - continue returning the 202 status code
 - include a response entity providing progress information. The format of the response entity may be adapted by extensions.
 - If the job is completed successfully, the server **MUST**
 - reply with a 200 if the job completed successfully and the result is included in the job status response
 - reply with a 201 if the job successfully created a new resource (e.g. import job) and include the URI of the created resource in the Location header
 - reply with a 303 if the job completed successfully and the result can be accessed from an other URI (e.g. downloaded)
 - If the job failed, the server **MUST**
 - reply with a 422, or 500 depending if the failure was due to business conditions or technical.

Using 201 Created as the status code of a GET request is quite uncommon, and seem to go against the HTTP specification. A 303 See Other may be more appropriate in all cases.

1.3. Authorization

To access most resources in the public API you have to be authenticated and authorized to do so.

To authenticate, each request should contain the following HTTP headers:

Header name	Description
IUCLID6-USER	The username of the user, who is to be authenticated
IUCLID6-PASS	The password of the user

If the authentication fails, the server will reply with **401 Unauthorized**.

The responses when authorization failures, vary according the nature of the REST resource, for example when a user tries to read a document she has no access to, **404 Not found** is returned,

however if she tries to modify a document she does have read access on, but not write access, then **403 Forbidden** is returned instead.

1.4. Base URL

Every REST resource of the public API is under:

Base url	/iuclid6-ext/api/ext/v1/
-----------------	--------------------------

1.5. About

On the **base url** you can find basic information about the system and its extensions, such the names of the various installed plugins along with their versions.

GET	/iuclid6-ext/api/ext/v1/
------------	--------------------------

example response

```
{
  "type": "Framework",
  "name": "IUCEF",
  "description": "",
  "version": "1.4.0",
  "components": [
    {
      "type": "Extension",
      "name": "iuclid6",
      "description": "",
      "version": "1.0.0",
      "components": [
        {
          "type": "ArchiveManifestHandler",
          "name": "ArchiveManifest",
          "description": "",
          "version": null,

```

```
    "components": []
  },
  {
    "type": "DocumentHandler",
    "name": "DocumentSecuredRepresentation",
    "description": "",
    "version": null,
    "components": []
  },
  ...
```

The format of the 'about' resource's respond is not finalized and is subject to change.

2. Definitions

2.1. Introduction

IUCLID6 is a highly extensible system, especially when it comes to the documents it can handle and how these are organized in datasets.

A legislation provider can, through a well defined mechanism, specify:

- document templates, that define the structure of a particular document,
- section-trees, that organize documents in *datasets* (e.g. 'Substance' dataset)
- submission types, that identify a specific section-tree, along with which dataset type applies to,
- phrases to be used, for example, as values in '*picklists*' fields.
- phrase-groups, that define a set of phrases that are available to a specific '*picklist*' field.
- text-templates,

The IUCLID6 Public REST API provide several resources to retrieve information about registered legislation providers, **no authentication** is needed to access these resources, since they convey publicly available information.

All the following resources respond with **application/json**.

Every REST resource in this document is under `/iuclid6-ext/api/ext/v1/`

2.2. Legislation providers

Registered legislation providers, with their version and localized name, are found under:

GET	<code>/definition/providers</code>
------------	------------------------------------

The response will be a JSON array of objects, each object will contain the following fields:

example response

```
[
  {
    "identifier": "bpr",
    "title": "EU_BPR",
    "version": "2.0"
  },
  {
    "identifier": "clp",
    "title": "EU_CLP",
    "version": "2.0"
  },
  {
    "identifier": "core",
    "title": "CORE",
```

```

    "version": "2.0"
  },
  {
    "identifier": "oecd",
    "title": "OECD",
    "version": "2.0"
  },
  {
    "identifier": "reach",
    "title": "EU_REACH",
    "version": "2.0"
  }
}

```

Name	Type	Description
identifier	string	The identification code of the legislation provider.
title	string	The localized name of the provider, in the requested language via the Accept-Language header. If the header is omitted, or no localization exists for the requested language, English is used.
version	string	The version of the provider.

The following section on Submission types is obsolete (since v5.0.0) and replaced by the working contexts

2.3. Submission types (obsolete since v5.0.0)

Each legislation provider can define its own submission types, and each submission type is accompanied by a section tree with the same identifier.

GET	/definition/submissiontypes
------------	-----------------------------

The resource supports query parameters for restricting the results by provider and by applicable entity type:

Parameter	Description
provider	The provider identification code

Parameter	Description
for	<p>The entity type the submission type must be applicable to.</p> <p>In general, the following IUCLID6 entity types are supported:</p> <ul style="list-style-type: none"> • SUBSTANCE • MIXTURE • TEMPLATE
dossier	<p>If true, submission types suitable for dossier creation.</p> <p>If false, submission types not suitable for dossier creation.</p>
tree	<p>If true, submission types suitable for tree visualization.</p> <p>If false, submission types not suitable for tree visualization.</p>
export	<p>If true, submission types suitable for export.</p> <p>If false, submission types not suitable for export.</p>

The submission type applies to specific entity type (i.e. SUBSTANCE, MIXTURE, TEMPLATE), in case a submission type applies to more than one entity type, multiple array entries will be returned:

example response

```
[
  {
    "provider": "core",
    "identifier": "COMPLETE",
    "title": "Complete table of contents",
    "applicableFor": "MIXTURE",
    "dossier": true,
    "tree": true,
    "export": true
  },
  {
    "provider": "bpr",
    "identifier": "BIOC_ACTIVE_SUBSTANCE_FOR_MIXTURES",
    "title": "BPR Active substance application (representative product)",
    "applicableFor": "MIXTURE",
    "dossier": true,
    "tree": true,
    "export": true
  },
  {

```

```

    "provider": "bpr",
    "identifier": "BIOC_BIOCIDAL_PRODUCT",
    "title": "BPR Biocidal product authorisation",
    "applicableFor": "MIXTURE",
    "dossier": true,
    "tree": true,
    "export": true
  },
  {
    "provider": "oecd",
    "identifier": "OECD_HT",
    "title": "OECD harmonised templates",
    "applicableFor": "MIXTURE",
    "dossier": true,
    "tree": true,
    "export": true
  },
  {
    "provider": "bpr",
    "identifier": "BIOC_BASIC_INFORMATION_MIXTURE",
    "title": "BPR Basic information (mixture)",
    "applicableFor": "MIXTURE",
    "dossier": false,
    "tree": true,
    "export": true
  },
  {
    "provider": "core",
    "identifier": "COMPLETE",
    "title": "Complete table of contents",
    "applicableFor": "SUBSTANCE",
    "dossier": true,
    "tree": true,
    "export": true
  },
  {
    "provider": "reach",
    "identifier": "R_A15_REST",
    "title": "REACH Annex XV - Restriction",
    "applicableFor": "SUBSTANCE",
    "dossier": true,
    "tree": true,
    "export": true
  }
]

```

The response will be a JSON array of objects, each object will contain the following fields:

Name	Type	Description
------	------	-------------

Name	Type	Description
provider	string	The identifier of the legislation provider for this submission type
identifier	string	The identification code of the submission type.
title	string	The localized name of the submission type, in the requested language via the Accept-Language header. If the header is omitted, or no localization exists for the requested language, English is used.
applicableFor	string	The entity type the submission type applies to.
dossier	boolean	If true the submission type is suitable for dossier creation
tree	boolean	If true the submission type is suitable for tree visualization
export	boolean	If true the submission type is suitable for export

2.4. Working contexts (since v5.0.0)

Each legislation provider can define its own working context. A working context is applicable for a single entity, in that entity is a complex entity (SUBSTANCE, MIXTURE, TEMPLATE) it defines the section tree for it. Additionally, if the working context supports dossier creation, it defines the definition of the dossier header document. Moreover it can optionally have a "viewAsMappings" section where alternative definitions for entities are defined, and also an optional "additionalTrees" section where a provides listing of section trees to be used for additional complex entities in a dataset.

GET	/definition/workingcontext
------------	----------------------------

The resource supports query parameters for restricting the results by provider and by applicable entity type:

Parameter	Description
provider	The provider identification code
for	The entity type the submission type must be applicable to. In general, the following IUCLID6 entity types are supported: <ul style="list-style-type: none"> • SUBSTANCE • MIXTURE • TEMPLATE

Parameter	Description
dossier	If true, submission types suitable for dossier creation. If false, submission types not suitable for dossier creation.

The submission type applies to specific entity type (i.e. SUBSTANCE, MIXTURE, TEMPLATE), in case a submission type applies to more than one entity type, multiple array entries will be returned:

example response

```
[
  {
    "provider": "clp",
    "identifier": "CLP_ALTERNATIVE_NAME",
    "title": "CLP Alternative name request",
    "applicableFor": "SUBSTANCE",
    "dossierHeader": "DOSSIER.CLP_ALTERNATIVE_NAME",
    "mainTreeld": "CLP_ALTERNATIVE_NAME.SUBSTANCE"
  },
  {
    "provider": "clp",
    "identifier": "CLP_ALTERNATIVE_NAME",
    "title": "CLP Alternative name request",
    "applicableFor": "TEMPLATE",
    "dossierHeader": "DOSSIER.CLP_ALTERNATIVE_NAME",
    "mainTreeld": "CLP_ALTERNATIVE_NAME.TEMPLATE"
  },
  {
    "provider": "clp",
    "identifier": "CLP_NOTIF",
    "title": "CLP Notification",
    "applicableFor": "SUBSTANCE",
    "dossierHeader": "DOSSIER.CLP_NOTIF",
    "mainTreeld": "CLP_NOTIF.SUBSTANCE"
  },
  {
    "provider": "clp",
    "identifier": "CLP_NOTIF",
    "title": "CLP Notification",
    "applicableFor": "TEMPLATE",
    "dossierHeader": "DOSSIER.CLP_NOTIF",
    "mainTreeld": "CLP_NOTIF.TEMPLATE"
  },
  {
    "provider": "clp",
    "identifier": "CLP_PCN",
    "title": "CLP Poison centres notification",
```



```

"applicableFor": "MIXTURE",
"dossierHeader": "DOSSIER.CLP_PCN",
"mainTreeId": "CLP_PCN.MIXTURE",
"additionalTrees": [
  {
    "treeName": "CLP_PCN_MIM",
    "activationClauses": [
      "MixtureInMixture"
    ]
  },
  {
    "treeName": "CLP_PCN_SUBSTANCE",
    "activationClauses": [
      "SubstanceInMixture"
    ]
  },
  {
    "treeName": "CLP_PCN_SFC",
    "activationClauses": [
      "StandardFormulaComponent"
    ]
  },
  {
    "treeName": "CLP_PCN_ICG",
    "activationClauses": [
      "InterchangeableComponentGroup"
    ]
  }
],
"viewAsMappings": [
  {
    "source": "CONTACT",
    "target": "CUSTOM_ENTITY.PCN_Contact_Person"
  },
  {
    "source": "MIXTURE",
    "target": "CUSTOM_ENTITY.PCN_Mixture"
  },
  {
    "source": "REFERENCE_SUBSTANCE",
    "target": "CUSTOM_ENTITY.PCN_Reference_Substance"
  },
  {
    "source": "SUBSTANCE",
    "target": "CUSTOM_ENTITY.PCN_Substance"
  }
]
},
{
  "provider": "clp",
  "identifier": "CLP_PCN",

```

```

"title": "CLP Poison centres notification",
"applicableFor": "TEMPLATE",
"dossierHeader": "DOSSIER.CLP_PCN",
"mainTreeId": "CLP_PCN.TEMPLATE",
"additionalTrees": [
  {
    "treeName": "CLP_PCN_MIM",
    "activationClauses": [
      "MixtureInMixture"
    ]
  },
  {
    "treeName": "CLP_PCN_SUBSTANCE",
    "activationClauses": [
      "SubstanceInMixture"
    ]
  },
  {
    "treeName": "CLP_PCN_SFC",
    "activationClauses": [
      "StandardFormulaComponent"
    ]
  },
  {
    "treeName": "CLP_PCN_ICG",
    "activationClauses": [
      "InterchangeableComponentGroup"
    ]
  }
],
"viewAsMappings": [
  {
    "source": "CONTACT",
    "target": "CUSTOM_ENTITY.PCN_Contact_Person"
  },
  {
    "source": "MIXTURE",
    "target": "CUSTOM_ENTITY.PCN_Mixture"
  },
  {
    "source": "REFERENCE_SUBSTANCE",
    "target": "CUSTOM_ENTITY.PCN_Reference_Substance"
  },
  {
    "source": "SUBSTANCE",
    "target": "CUSTOM_ENTITY.PCN_Substance"
  }
]
},
{
  "provider": "clp",

```

```

    "identifier": "CLP_CLH",
    "title": "CLP Regulation - CLH dossier",
    "applicableFor": "SUBSTANCE",
    "dossierHeader": "DOSSIER.CLP_CLH",
    "mainTreeId": "CLP_CLH.SUBSTANCE"
  },
  {
    "provider": "clp",
    "identifier": "CLP_CLH",
    "title": "CLP Regulation - CLH dossier",
    "applicableFor": "TEMPLATE",
    "dossierHeader": "DOSSIER.CLP_CLH",
    "mainTreeId": "CLP_CLH.TEMPLATE"
  },
  {
    "provider": "clp",
    "identifier": "CLP_CLH_CAT",
    "title": "CLP Regulation - CLH dossier - category",
    "applicableFor": "CATEGORY",
    "additionalTrees": [
      {
        "treeName": "CLP_CLH",
        "activationClauses": [
          "Substance"
        ]
      }
    ]
  }
]

```

The response will be a JSON array of objects, each object will contain the following fields:

Name	Type	Description
provider	string	The identifier of the legislation provider for this working context
identifier	string	The identification code of the working context.
title	string	The localized name of the working context, in the requested language via the Accept-Language header. If the header is omitted, or no localization exists for the requested language, English is used.
applicable For	string	The entity type the working context applies to.
dossierHeader	string	If the working context support dossier creation, the definition identifier of the dossier header (e.g. DOSSIER.CLP_ALTERNATIVE_NAME)

Name	Type	Description
mainTreeId	boolean	If the applicable entity is a complex one (SUBSTANCE, MIXTURE, TEMPLATE), the identifier of the section tree (e.g. CLP_ALTERNATIVE_NAME.SUBSTANCE)
additionalTrees	array	<p>an optional list of additional trees to be used for complex entities other than the main entity of the dataset.</p> <p>Each item is an object containing:</p> <ul style="list-style-type: none"> the "treeName", a string of the name of one tree (the name of a tree is the first part of the tree identifier. eg. CLP_ALTERNATIVE_NAME) "activationClauses", a list of string that enumerate the conditions an additional entity must satisfy for this tree to be used. The conditions are only referenced by name, the logic of them is specified externally: see https://pmo.trasys.be/confluence/pages/viewpage.action?spaceKey=IUCLID6&title=Activation+clauses)

2.5. Trees (since v5.0.0)

2.5.1. Listing Trees

You can retrieve a listing of all registered section tree at the following resource:

GET	/definition/tree
------------	------------------

Every listing resource the following query parameters:

Param	Type	Description	Default
formatter	String	<p>the extension name of a eu.echa.iuclid6.ext.api.SectionTreeHandler.</p> <p>Available extensions are:</p> <ul style="list-style-type: none"> standard.Void (it does not give any representation for the tree) standard.SectionNode (it get the same representation as /definition/tree/R_ABOVE_1000?for=SUBSTANCE endpoint) 	standard.Void
treeName	String	if specified, it filters the list by tree name	none

Param	Type	Description	Default
for	String	if specified, it only filters the list to keep only trees applicable for the specified entity type	none

The response is of the following format:

```
{
  "limit": 102,
  "totalCount": 102,
  "results": [
    {
      "uri": "i6def:/tree/R_A15_REST.SUBSTANCE",
      "representation": null //depends on the specified formatter
    },
    {
      "uri": "i6def:/tree/R_A15_REST.TEMPLATE",
      "representation": null //depends on the specified formatter
    },
    ...
  ]
}
```

2.5.2. Loading a single tree

You can retrieve a section tree with a specified tree name, applicable to a specific entity type at the following resources:

GET	/definition/tree/{tree name}?for={entity_type}
GET	/definition/tree/{tree name}.{entity_type}

The response type is not **application/json** as it should be, but **application/vnd.iuclid6.ext+json**

This is something that will be corrected in a subsequent version.

For example the to fetch the tree of the "REACH Registration above 1000 tonnes" submission type for substances:

GET	/definition/tree/R_ABOVE_1000?for=SUBSTANCE
------------	---

or equivalently:

GET	/definition/tree/R_ABOVE_1000.SUBSTANCE
------------	---

will return:

```
{
  "code": "R_ABOVE_1000",
  "title": "REACH Registration above 1000 tonnes",
  "sections": [
    {
      "code": "0",
      "title": "Related information",
      "sections": [],
      "documents": []
    },
    {
      "code": "1",
      "title": "General information",
      "sections": [
        {
          "code": "1.1",
          "title": "Identification",
          "sections": [],
          "documents": [
            {
              "code": "SUBSTANCE",
              "title": "Substance Identification",
              "access": "NO_ACCESS",
              "required": false,
              "single": true,
              "requirement": "NOT_APPLICABLE",
              "permissions": []
            }
          ]
        }
      ],
      "documents": [
        {
          "code": "1.2",
          "title": "Composition",
          "sections": [],
          "documents": [
            {
              "code": "FLEXIBLE_RECORD.SubstanceComposition",
              "title": "Composition",
              "access": "NO_ACCESS",
              "required": true,
              "single": false,
              "requirement": "REQUIRED"
            }
          ]
        }
      ]
    }
  ]
}
```

```

        "permissions": []
    }
  ],
},
{
  "code": "1.3",
  "title": "Identifiers",
  "sections": [],
  "documents": [
    {
      "code": "FIXED_RECORD.Identifiers",
      "title": "Identifiers",
      "access": "NO_ACCESS",
      "required": false,
      "single": true,
      "requirement": "NOT_REQUIRED",
      "permissions": []
    }
  ]
}
...

```

The response is an tree-like object containing the following:

Name	Type	Description
code	string	The code of the current node
title	string	The localized name of the current node, in the requested language via the Accept-Language header. If the header is omitted, or no localization exists for the requested language, English is used.
sections	array	An array of section node objects.
documents	array	An array of objects representing the allowed document types for the current node.

Each entry of the **documents** field above, contain the following attributes:

Name	Type	Description
code	string	The code of the current node

Name	Type	Description
title	string	<p>The localized name of the current node, in the requested language via the Accept-Language header.</p> <p>If the header is omitted, or no localization exists for the requested language, English is used.</p>
access	string	<p>If the request is authenticated, it contains the privileges of the authenticated user on the specific document type.</p> <p>Can be one of:</p> <ul style="list-style-type: none"> • NO_ACCESS • READ_ONLY • READ_WRITE • FULL_ACCESS <p>When the request is unauthenticated it will always contain 'NO_ACCESS'</p>
required	boolean	<p>True if the specific document type is required for the submission. <i>OBSOLETE check requirement value</i></p>
requirement	string	<p>Defines whether a document need to be included in dataset during create dossier and export operations.</p> <p>One of:</p> <ul style="list-style-type: none"> • REQUIRED, the document must be included • NOT_REQUIRED, the document can be optionally included, but it must be opted in for inclusion. • OPTIONAL, the document can be optionally excluded, but it must be opted in for exclusion. • NOT_APPLICABLE, the document must not be included • CONTROLLED_BY_REFERRER, the document can be included only if another document referring to it is included
single	boolean	<p>True if only one document of the specific type is allowed. e.g. The "Identification" of the substance</p>
permissions	array	<p>An array containing the <i>semantic</i> permissions a user has on the node, is derived from the <i>access</i> using the following logic:</p> <ul style="list-style-type: none"> • NO_ACCESS → empty array • READ → ["LIST"] • READ_WRITE → ["LIST", "CREATE"] • FULL_ACCESS → ["LIST", "CREATE"]

2.6. Documents

In IUCLID6, the structure of the contents of a document greatly depends on the document's type. The actual structure of a document of a particular is defined by the "document definition".

The document definition consists of "definition elements" of several types, each one of them defining a different data element in a document.

The various definition elements can be divided in two groups: the *fields* and the *block*.

Fields define data structures that can only contain values. In contrast, the block can be considered as a container of other definition elements. It defines a grouping for the contained definition elements, which can be either fields or blocks.

Each definition element has a name, that is unique within its container, and by aggregating the names of each definition element, starting from the document definition, up to a particular element, we derive the "**path**" to this particular element. This path can be used to uniquely identify an element within a document.

To help visualizing the aforementioned structure, follows an abstract of the definition of the "LEGAL_ENTITY" document:

```
{
  "identifier": "LEGAL_ENTITY",
  "version": "2.0",
  "provider": "domain",
  "@lang": "en",
  "contents": [
    {
      "type": "block",
      "name": "GeneralInfo",
      "title": "General information",
      "contents": [
        {
          "type": "text",
          "name": "LegalEntityName",
          "title": "Legal entity name",
          "required": true,
          "mimeType": "text/plain",
          "maxLength": 255
        },
        {
          "type": "picklist",
          "name": "LegalEntityType",
          "title": "Legal entity type",
          "phrasegroup": "N01"
        }
      ]
    }
  ]
}
```

```

    "type": "text",
    "name": "Remarks",
    "title": "Remarks",
    "mimeType": "text/plain",
    "maxLength": 32768
  },
  {
    "type": "block",
    "name": "OtherNames",
    "title": "Other names",
    "protectedBy": "LEGAL_ENTITY.GeneralInfo.OtherNames.DataProtection",
    "multiple": true,
    "contents": [
      {
        "type": "dataProtection",
        "name": "DataProtection",
        "title": "Flags"
      },
      {
        "type": "text",
        "name": "Name",
        "title": "Name",
        "mimeType": "text/plain",
        "maxLength": 255
      }
    ]
  }
]
},
...

```

For example, the **path** of the "LegalEntityName" text element show above, is:
 "LEGAL_ENTITY.GeneralInfo.LegalEntityName"

Name	Type	Description
identifier	string	The document type of the definition. e.g. LEGAL_ENTITY, ENDPOINT_SUMMARY.ToxicityToBirds etc.
version	string	The version of the document definition
provider	string	The identifier of the legislation provider, providing this document definition
@lang	string	The locale used to localize the response
contents	array	An array of definition element objects
Name	Type	Description

Name	Type	Description
type	string	<p>The specific type of the element:</p> <p>Can be one of:</p> <ul style="list-style-type: none"> • block • dataProtection • text • boolean • date • numeric • picklist • quantity • range • docRef • blockRef • inventoryRef • definitionRef • attachment • address
name	string	The name of the element, which uniquely identifies it, within its direct container
title	string	<p>The localized name of the element, in the requested language via the Accept-Language header.</p> <p>If the header is omitted, or no localization exists for the requested language, English is used.</p>
protectedBy	string	The path to a "dataProtection" field, that impose confidentiality claims upon the field.
multiple	boolean	Whether the data content consists of a single or multiple values.
required	boolean	Whether the data element should be populated for the document to be valid.

Depending on the type, each element can support additional properties:

Element type	Name	Type	Description
block	contents	array	An array of definition element objects, contained in this block

Element type	Name	Type	Description
text	contentType	string	The mime-type of the text contents of the field. Can be either: text/plain or text/html
	maxLength	number	The maximum length the field support. <i>Omitted if unrestricted</i>
	textTemplates	array	The codes to relevant text-templates that can be used to populate the field
date	withTime	boolean	True, if the field contain time along with the date
numeric	numericType	string	<ul style="list-style-type: none"> • INTEGER • DOUBLE
	min	number	The minimum allowed value, if any
	max	number	The maximum allowed value, if any
picklist	phrasegroup	string	The code of the phrase-group, this field's allowed values can come from
	defaultPhrase	string	The default phrase, if any
	remarks	boolean	True, if additional text can be added to the field.
	remarksMaxLength	number	The maximum supported length of the additional text, if supported
quantity	unitPhrasegroup	string	The code of the phrase-group, containing the units for this field
range	numericType	string	<ul style="list-style-type: none"> • INTEGER • DOUBLE
	unitPhrasegroup	string	The code of the phrase-group, containing the units for this field
	isHalfBounded	boolean	True, if only one bound is supported.
	boundQualifiers	array	The allowed boundary qualifiers. Depending on
	lowerBoundQualifiers		

Element type	Name	Type	Description
	upperBoundQualifiers		<p>whether the field is half-bounded, either only the boundQualifiers or both of the lowerBoundQualifiers and upperBoundQualifiers, will be present.</p> <p>The allowed values are:</p> <ul style="list-style-type: none"> • ca. • < • <= • > • >=
docRef	referenceTypes	array	A list of the types of the documents, that the field can refer to.
blockRef	referenceTypes	array	A list of the relative path to the blocks, that the field can refer to
attachment	contentType	string	<p>The mime-type of the attachment, this field accepts.</p> <p>Can be one of the following:</p> <ul style="list-style-type: none"> • image/* • /*/*

Media type

The media type for the document definition object is **application/vnd.iuclid6.ext+json;type=iuclid6.Definition**

The document definition for a particular document type can be retrieved on the following resource:

GET /definition/document/{document_type}

A listing of every registered document definition can be obtained by the following resource:

GET /definition/documents

The above resource returns only the name of each document definition by default.

However you can obtain the full contents of each definition by setting the "formatter" query parameter to "iuclid6.Definition": /definition/documents?formatter=iuclid6.Definition

2.7. Phrase-groups

Phrase-groups organize several phrases together in a set identified by a code. Several iuclid fields (e.g. picklists, physical quantity, etc) use this code to indicate the phase codes they accept.

You can retrieve the phases of a particular phrase-group on the following resource:

GET	/definition/phrasegroup/{phrasegroup-code}/phrases
------------	--

The response is a JSON array, the entries of which contain the **phrase** object and a boolean indicating whether the phrase is **obsolete** and it should not be used.

example response for phrase-group N16

```
[
  {
    "phrase": {
      "code": "2820",
      "text": "final",
      "description": "",
      "open": false
    },
    "obsolete": false
  },
  {
    "phrase": {
      "code": "2793",
      "text": "draft",
      "description": "",
      "open": false
    },
    "obsolete": false
  },
  {
    "phrase": {
      "code": "1342",
      "text": "other:",
      "description": "",
      "open": true
    },
    "obsolete": false
  }
]
```

]

The **phrase** object contains the following attributes:

Name	Type	Description
code	string	The code of the phrase
text	string	The localized name of the phrase, in the requested language via the Accept-Language header. If the header is omitted, or no localization exists for the requested language, English is used.
description	string	Explanatory notes for the phrase.
open	boolean	True if the phrase can contain additional text (a.k.a "other text")

2.8. Phrases

The following resource, contains all registered phrases in the system, without any grouping:

GET	/definition/phrases
------------	---------------------

It returns an array of every phrase:

```
[
  {
    "code": "1",
    "text": "ISO/DIS 17556.2 (Plastics - Determination of the Ultimate Aerobic Biodegradability in Soil by Measuring the Oxygen Demand in a Respirometer or the Amount of Carbon Dioxide Evolved)",
    "description": "",
    "open": false
  },
  {
    "code": "10",
    "text": "#8",
    "description": "",
    "open": false
  },
  {
    "code": "100",
    "text": "Arenicola marina",
    "description": "",
    "open": false
  },
  ...
]
```

2.9. Text templates

The following resource contains all registered text templates in the system:

GET	/definition/texts
------------	-------------------

```

{
  "limit": null,
  "totalCount": 438,
  "results": [
    {
      "uri": null,
      "representation": {
        "code": "TT_1",
        "provider": "oecd",
        "header": "",
        "content": "PHYSICO-CHEMICAL PROPERTIES\n - Vapour pressure:\n - Water solubility:\n - log Pow: \n - pKa: \n - Base or acid catalysis of test material: \n - UV absorption:\n - Stability of test material at room temperature: \n \n OTHER PROPERTIES (if relevant for this endpoint)"
      }
    },
    {
      "uri": null,
      "representation": {
        "code": "TT_10",
        "provider": "oecd",
        "header": "",
        "content": "PHYSICO-CHEMICAL PROPERTIES\n - Vapour pressure:\n - Henry's Law constant: \n - log Pow: \n - pKa: \n - UV absorption:\n - Stability of test material at room temperature: \n \n OTHER PROPERTIES (if relevant for this endpoint)"
      }
    }
  ],
  ...

```

Each text template object contains the following:

Name	Type	Description
code	string	The identification code of the text template
provider	string	The identifier of the legislation provider, providing this text template
header	string	A title for this text template.
content	string	The actual template content.

In addition, you can retrieve a specific text template by its identification code on the following resource:

GET	/definition/texts/{code}
------------	--------------------------

3. Search

Every REST resource in this document is under /iuclid6-ext/api/ext/v1/

The IUCLID6 Rest API supports paged document searches by using predefined search queries, each accepting several parameters.

GET	/query/iuclid6/{query-name}
------------	-----------------------------

The **{query-name}** can obtain any of the following values:

- byDate
- byName
- byContact
- bySubstance
- byMixture
- byLegalEntity
- byReferenceSubstance
- byTemplate
- byLiterature
- byAnnotation
- bySite
- byAdditive
- byImpurity
- byConstituent
- byGhs
- byDsd
- byJointSubmission
- byCategory

Every search query supports the following query parameters:

Param	Type	Description	Default
count	boolean	include the total count of the results. <i>If the total count is not needed, omitting it improves performance.</i>	true
l	integer	the number of results (<i>limit</i>) per page.	10
o	integer	the offset of the first result in page. e.g. to fetch the 3rd page of 25 (l=25) results specify o=50 (o = l * zero_indexed_page_number)	0

For example, the following fetches the second page of legal entity documents with 20 items per page, including the total count of the legal entity documents.

GET	/query/iuclid6/byType?doc.type=SUBSTANCE&l=20&o=20&count=true
------------	---

3.1. Results

The search results are wrapped inside a standard payload that contains:

- the results of the search
- plus some additional information to support paging, such as
 - the limit
 - the offset
 - and the total count of the results.

Each result comprises of two values, the URI and the representation.

GET	/query/iuclid6/byType?doc.type=SUBSTANCE
------------	--

```
{
  "limit": 10,
  "offset": 0,
  "totalCount": 102,
  "results": [
    {
      "uri": "iuclid6:/0/SUBSTANCE/f200c268-7a94-4fb6-9d1d-6759d71f6894/SUBSTANCE/f200c268-7a94-4fb6-9d1d-6759d71f6894",
      "representation": null
    },
    {
      "uri": "iuclid6:/0/SUBSTANCE/d4e388de-aa35-4568-b25e-39a6538363a6/SUBSTANCE/d4e388de-aa35-4568-b25e-39a6538363a6",
      "representation": null
    },
    {
      "uri": "iuclid6:/0/SUBSTANCE/c61b4b3a-ffbc-46b4-a5dc-f3ccb512c11a/SUBSTANCE/c61b4b3a-ffbc-46b4-a5dc-f3ccb512c11a",
      "representation": null
    },
    {
      "uri": "iuclid6:/0/SUBSTANCE/db63a1af-6304-427f-aa1d-387b1f8feb5/SUBSTANCE/db63a1af-6304-427f-aa1d-387b1f8feb5",
      "representation": null
    },
    {
      "uri": "iuclid6:/0/SUBSTANCE/IUC5-f0958d98-4cb3-4058-80cd-bdc24e7d125d/SUBSTANCE/IUC5-f0958d98-4cb3-4058-80cd-bdc24e7d125d",
      "representation": null
    }
  ]
}
```

```

    },
    {
      "uri": "iuclid6:/0/SUBSTANCE/40374a6f-9175-4ac8-87c9-
d220c866997a/SUBSTANCE/40374a6f-9175-4ac8-87c9-d220c866997a",
      "representation": null
    },
    {
      "uri": "iuclid6:/0/SUBSTANCE/IUC4-f696abe6-513a-3771-b2ee-
76e4110e8103/SUBSTANCE/IUC4-f696abe6-513a-3771-b2ee-76e4110e8103",
      "representation": null
    },
    {
      "uri": "iuclid6:/0/SUBSTANCE/326ab136-d74c-4088-9425-
8d9d5642b96e/SUBSTANCE/326ab136-d74c-4088-9425-8d9d5642b96e",
      "representation": null
    },
    {
      "uri": "iuclid6:/0/SUBSTANCE/1ef71d52-ddc7-4b55-9c2f-
ea72efa55ddd/SUBSTANCE/1ef71d52-ddc7-4b55-9c2f-ea72efa55ddd",
      "representation": null
    },
    {
      "uri": "iuclid6:/0/SUBSTANCE/7cec4254-65f7-4d9a-9a6a-
2a74dd1c9b1c/SUBSTANCE/7cec4254-65f7-4d9a-9a6a-2a74dd1c9b1c",
      "representation": null
    }
  ]
}

```

The above shows a successful response back from the call. In this case SUBSTANCE documents were searched for and there are 102 results. The first 10 are contained in the response.

Since no formatter was specified, the representation is null.

The query results contain the following fields:

Name	Type	Description
limit	integer	the number of the results contained in the returned page.
offset	integer	the offset of the first result in page.
totalCount	integer	the total count of records that match the search. <i>Because of pagination, this is not necessarily the number of records in the response</i>
results	Object[]	The actual results of the search. See below.

To retrieve the results with the default IUCLID6 representation you need to set the "formatter" query parameter to "iuclid6.DocumentSecuredRepresentation".

GET	/query/iuclid6/byType?doc.type=SUBSTANCE?o=0&l=1&formatter=iuclid6.DocumentSecuredRepresentation
------------	--

Here 's an example of a response that could be potentially returned by the previous request:

QueryResults

```
{
  "limit": 1,
  "offset": 0,
  "totalCount": 102,
  "results": [
    {
      "uri": "iuclid6:/0/SUBSTANCE/f200c268-7a94-4fb6-9d1d-6759d71f6894/SUBSTANCE/f200c268-7a94-4fb6-9d1d-6759d71f6894",
      "representation": {
        "classtype": "SubstanceSecuredRepresentation",
        "accessRight": "READ_ONLY",
        "key": "f200c268-7a94-4fb6-9d1d-6759d71f6894/0",
        "name": "SUB WITH IMAGE",
        "createdOn": "2016-09-06T16:14:59.187Z",
        "modifiedOn": "2016-09-06T15:00:51.000Z",
        "publicName": "the public name",
        "legalEntityRepresentation": {
          "classtype": "LegalEntitySecuredRepresentation",
          "accessRight": "READ_ONLY",
          "key": "4f88bc7f-395c-4d0b-997b-14e8c9aef605/0",
          "name": "Predefined Legal entity",
          "createdOn": "2016-09-06T16:14:59.187X",
          "city": null,
          "country": {
            "code": null
          }
        },
        "definition": "LEGAL_ENTITY",
        "modifiedOn": "2016-09-01T15:40:27.000Z",
        "accessAllowed": true
      },
      "definition": "SUBSTANCE",
      "accessAllowed": true
    }
  ]
}
```

Each result contains the following two properties:

Name	Type	Description
uri	string	a URI of the document.
representation	object	an object containing the representation of the result. The exact format is governed by the "formatter" query param. If omitted, null is returned.

3.1.1. Document Secured Representation

The **DocumentSecuredRepresentation** defines a hierarchy of types, that represent the different entity types of IUCLID6.

Each result is represented, according to its document type, by the appropriate sub-type.


However, every sub-type include the following properties:

Name	Type	Description
classtype	String	the sub-type of the document representation. <i>e.g. SubstanceSecuredRepresentation</i>
key	String	the document key formatted as {uuid}/{snapshot}.
definition	String	the document definition identifier. <i>e.g. ENDPOINT_STUDY_RECORD.Partition</i>
name	String	the name of the document
accessRight	String	the access right the user has on the document. Possible values are: <ul style="list-style-type: none"> • NO_ACCESS • READ_ONLY • READ_WRITE • FULL_ACCESS

createdOn	String	the instant the document was created, in ISO 8601 format <i>e.g. 2016-09-06T13:14:59.187Z</i>
modifiedOn	String	the instant the document was last modified, in ISO 8601 format <i>e.g. 2016-09-01T07:20:11Z</i>
remarks	String	the optional, remarks of the document

3.1.2. Ordering

The results are ordered by default:

- by descending touched date 
- then by descending modification date
- and finally by ascending uuid.

Touched date

As touched date is considered the latest of the modification and creation date.

In case of import, the modification date can precede the creation date of a document, since as creation is the date when the import takes place, while the modification date is the respective field in the imported archive.

You can change the result ordering by specifying the ``order`` parameter. The available values for the **order** parameter are:

Parameter value	Description
touched	the latest of the modification and creation date
modified	the modification date
created	the creation date
uuid	the uuid of the document

To reverse order on a value, append the minus sign "-" to the end of the name.

For example the default ordering, would be specified as:

GET	/query/iuclid6/byType?doc.type=SUBSTANCE&order=touched-&modified-&uuid
------------	--

3.2. Available queries

The parameter values can be of the following types:

- **String**
- **Picklist**, formatted as code[:other_text] e.g. 55446 or 1342:other_text_value
- **Date**, formatted as yyyy-MM-dd e.g. 2016-07-24
- **Boolean**, true/false

The default type of each parameter is **String**, unless explicitly specified.

Every query supports the following parameters:

Parameter	Description
doc.type	<p>The type of the document.</p> <p>In general, the following IUCLID6 document types are supported:</p> <ul style="list-style-type: none"> • DOSSIER • SUBSTANCE • MIXTURE • TEMPLATE • CATEGORY • ANNOTATION • LEGAL_ENTITY • SITE • REFERENCE_SUBSTANCE • CONTACT • LITERATURE • TEST_MATERIAL_INFORMATION <p>However, some queries might restrict the accepted values.</p>
group	The name of group that the document should be shared to.
owner	<p>The owner of the document. Can be one of:</p> <ul style="list-style-type: none"> • ME • OTHER

The following queries are available:

3.2.1. byDate

byDate

Find entities by creation/modification date.

Parameter name	Description
created.after	Date. The document should be created from the specified date and afterwards, e.g. "2016-06-27"
created.before	Date. The document should be created until before the specified date. e.g. "2016-07-27"
modified.after	Date. The document should be last modified from the specified date and afterwards, e.g. "2016-06-27"
modified.before	Date. The document should be last modified until before the specified date. e.g. "2016-07-27"

3.2.2. byName

byName

Find entities of specific name.

Parameter name	Description
doc.name	The name of the document

3.2.3. byContact

byContact

Find contacts by various criteria.

Note: The doc.type parameter should be omitted

Parameter name	Description
contact.first_name	The first name of the contact
contact.last_name	The last name of the contact
contact.organisation	The organization of the contact

3.2.4. bySubstance**bySubstance**

Find entities, being themselves or referring to, 'Substances' satisfying certain criteria.

Note: the doc.type parameter can be one of:

- SUBSTANCE
- MIXTURE
- DOSSIER

Depending on the value of doc.type parameter, the following criteria are applied:

- in case of SUBSTANCE, directly to the substance document
- in case of MIXTURE, to the linked substance of the MixtureComposition of a mixture
- in case of DOSSIER, to any dossier component of type SUBSTANCE

Parameter name	Description
sub.chemical	The substance chemical name
sub.owner	The name of the legal entity owner
sub.other	The other names
ref_sub.name	The name of the reference substance that is linked through field: SUBSTANCE.ReferenceSubstance.ReferenceSubstance.
ref_sub.inv.cas_number	The cas number in the chemical inventory of the linked reference substance.
ref_sub.inv.name	The name in the chemical inventory of the linked reference substance.
ref_sub.inv.number	The number in the chemical inventory of the linked reference substance.
ref_sub.cas_number	The cas number of the linked reference substance.
ref_sub.cas_name	The cas name of the linked reference substance.
ref_sub.iupac_name	The iupac name of the linked reference substance.
ref_sub.smiles_notation	The smiles notation of the linked reference substance.
ref_sub.mol_formula	The molecular formula of the linked reference substance.
ref_sub.in_chl	The inChI of the linked reference substance.
sub.reg_prog_id	The regulatory programme identifier of the FIXED_RECORD.Identifiers child of the SUBSTANCE.

sub.reg_prog	Picklist. The regulatory programme of the FIXED_RECORD.Identifiers child of the SUBSTANCE. Should be one of the phase codes in Phasegroup N12
sub.it_system_id	The IT System Identifier of the FIXED_RECORD.Identifiers child of the SUBSTANCE.

3.2.5. byMixture

byMixture

Find entities, being themselves or referring to, 'Mixtures' satisfying certain criteria.

Note: the doc.type parameter can be one of:

- MIXTURE
- DOSSIER

Depending on the value of doc.type parameter, the following criteria are applied:

- in case of MIXTURE, directly to the mixture document
- in case of DOSSIER, to any dossier component of type MIXTURE

Parameter name	Description
mix.name	The mixture name.
mix.other_name	The name in the 'Other_names'
sub.reg_prog_id	The regulatory programme identifier of the FIXED_RECORD.Identifiers child of the MIXTURE.
sub.reg_prog	Picklist. The regulatory programme of the FIXED_RECORD.Identifiers child of the MIXTURE. Should be one of the phase codes in Phasegroup N12
sub.it_system_id	The IT System Identifier of the FIXED_RECORD.Identifiers child of the MIXTURE.

3.2.6. byLegalEntity

byLegalEntity

Find entities, being themselves or referring to, 'legal entities' satisfying certain criteria.

Note: the doc.type parameter can be one of:

- LEGAL_ENTITY
- SUBSTANCE
- MIXTURE
- TEMPLATE

- CATEGORY
- DOSSIER

Depending on the value of doc.type parameter, the following criteria are applied:

- in case of LEGAL_ENTITY, directly to the legal entity document
- in case of DOSSIER, to the submitting legal entity of the dossier.
- the owner legal entity, for the rest.

Parameter name	Description
le.name	The legal entity name.
le.country	Picklist. The country of the legal entity. Should be one of the phase codes in Phasegroup N03
le.town	The town of the legal entity.

3.2.7. byReferenceSubstance

byReferenceSubstance

Find entities, being themselves or referring to, 'Reference substances' satisfying certain criteria.

Note: the doc.type parameter can be one of:

- REFERENCE_SUBSTANCE
- SUBSTANCE
- DOSSIER

Depending on the value of doc.type parameter, the following criteria are applied:

- in case of REFERENCE_SUBSTANCE, directly to the reference substance document
- in case of DOSSIER, to the reference substance linked to the dossier subject.
- in case of SUBSTANCE, the reference substance linked to the substance

Parameter name	Description
ref_sub.name	The name of the reference substance.
ref_sub.inv.cas_number	The cas number in the chemical inventory.
ref_sub.inv.name	The name in the chemical inventory.
ref_sub.inv.number	The number in the chemical inventory.
ref_sub.cas_number	The cas number.
ref_sub.cas_name	The cas name.
ref_sub.iupac_name	The iupac name.
ref_sub.smiles_notation	The smiles notation.

ref_sub.mol_formula	The molecular formula.
ref_sub.in_chl	The inChI of the linked reference substance.

3.2.8. byTemplate

byTemplate

Find 'Templates' satisfying certain criteria.

Note: the doc.type parameter should be omitted.

Parameter name	Description
doc.name	The template name.
le.name	The legal entity name.
le.country	Picklist. The country of the legal entity. Should be one of the phase codes in Phasegroup N03
le.town	The town of the legal entity.

3.2.9. byLiterature

byLiterature

Find 'Literature references' satisfying certain criteria.

Note: the doc.type parameter should be omitted.

Parameter name	Description
lit.title	The literature title.
lit.type	Picklist. The reference type of the literature. Should be one of the phase codes in Phasegroup Z31
lit.author	The author of the literature.
lit.ref_year	The year of the literature.
lit.source	The bibliographic source.
lit.test_lab	The testing laboratory
lit.owner	The company owner.
lit.owner_study_no	The company study no.
lit.report_no	The report no.

lit.report.after	Date. The report should be from the specified date and afterwards, e.g. "2016-06-27"
lit.report.before	Date. The document should be created until before the specified date. e.g. "2016-07-27"

3.2.10. byAnnotation

byAnnotation

Find annotations by various criteria.

Note: the doc.type parameter should be omitted.

Parameter name	Description
annotation.status	Picklist. The status of the annotation Should be one of the phase codes in Phasegroup N16
annotation.authority	The name of the authority/organisation of the annotation.
annotation.agreement	Boolean. Is the agreement with the applicant's summary checked.
annotation.data_waiver	Picklist. The data waiver acceptable of the literature. Should be one of the phase codes in Phasegroup N19
annotation.reliability	Picklist. The reliability of the literature. Should be one of the phase codes in Phasegroup N17

3.2.11. bySite

bySite

Find entities, being themselves or referring to, 'Sites' satisfying certain criteria.

Note: the doc.type parameter can be one of:

- SITE
- SUBSTANCE
- DOSSIER

Depending on the value of doc.type parameter, the following criteria are applied:

- in case of SITE, directly to the site document
- in case of SUBSTANCE, the SITE linked to a FLEXIBLE_RECORD.Sites of a substance
- in case of DOSSIER, to any dossier component of type SITE.

Parameter name	Description
site.name	The name of the site.
site.country	Picklist. The country of the site. Should be one of the phase codes in Phasegroup N03
site.town	The town of the site.
site.owner	The name of the legal entity onwer of the site.

3.2.12. byAdditive

byAdditive

Find entities having specific reference substances as additives.

Note: the doc.type parameter can be one of:

- SUBSTANCE
- MIXTURE
- DOSSIER

Depending on the value of doc.type parameter, the following criteria are applied:

- in case of SUBSTANCE, the linked reference substance through the additives field of a SubstanceComposition of a substance
- in case of MIXTURE, the linked reference substance through the additives field of a MixtureComposition of a mixture
- in case of DOSSIER, the linked reference substance through the additives field of a SubstanceComposition or MixtureComposition of the dossier's subject

Parameter name	Description
ref_sub.name	The name of the reference substance that is linked through field: SUBSTANCE.ReferenceSubstance.ReferenceSubstance.
ref_sub.inv.cas_number	The cas number in the chemical inventory of the linked reference substance.
ref_sub.inv.name	The name in the chemical inventory of the linked reference substance.
ref_sub.inv.number	The number in the chemical inventory of the linked reference substance.
ref_sub.cas_number	The cas number of the linked reference substance.
ref_sub.cas_name	The cas name of the linked reference substance.

ref_sub.iupac_name	The iupac name of the linked reference substance.
ref_sub.smiles_notation	The smiles notation of the linked reference substance.
ref_sub.mol_formula	The molecular formula of the linked reference substance.
ref_sub.in_chl	The inChI of the linked reference substance.

3.2.13. byImpurity

byImpurity

Find entities having specific reference substances as impurities.

Note: the doc.type parameter can be one of:

- SUBSTANCE
- MIXTURE
- DOSSIER

Depending on the value of doc.type parameter, the following criteria are applied:

- in case of SUBSTANCE, the linked reference substance through the impurities field of a SubstanceComposition of a substance
- in case of MIXTURE, the linked reference substance through the impurities field of a MixtureComposition of a mixture
- in case of DOSSIER, the linked reference substance through the impurities field of a SubstanceComposition or MixtureComposition of the dossier's subject

Parameter name	Description
ref_sub.name	The name of the reference substance that is linked through field: SUBSTANCE.ReferenceSubstance.ReferenceSubstance.
ref_sub.inv.cas_number	The cas number in the chemical inventory of the linked reference substance.
ref_sub.inv.name	The name in the chemical inventory of the linked reference substance.
ref_sub.inv.number	The number in the chemical inventory of the linked reference substance.
ref_sub.cas_number	The cas number of the linked reference substance.
ref_sub.cas_name	The cas name of the linked reference substance.
ref_sub.iupac_name	The iupac name of the linked reference substance.
ref_sub.smiles_notation	The smiles notation of the linked reference substance.
ref_sub.mol_formula	The molecular formula of the linked reference substance.
ref_sub.in_chl	The inChI of the linked reference substance.

3.2.14. byConstituent

byConstituent

Find entities having specific reference substances as constituent.

Note: the doc.type parameter can be one of:

- SUBSTANCE
- MIXTURE
- DOSSIER

Depending on the value of doc.type parameter, the following criteria are applied:

- in case of SUBSTANCE, the linked reference substance through the constituents field of a SubstanceComposition of a substance
- in case of MIXTURE, the linked reference substance through the **components** field of a MixtureComposition of a mixture
- in case of DOSSIER, the linked reference substance through the constituents field of a SubstanceComposition or the **components** field MixtureComposition of the dossier's subject

Parameter name	Description
ref_sub.name	The name of the reference substance that is linked through field: SUBSTANCE.ReferenceSubstance.ReferenceSubstance.
ref_sub.inv.cas_number	The cas number in the chemical inventory of the linked reference substance.
ref_sub.inv.name	The name in the chemical inventory of the linked reference substance.
ref_sub.inv.number	The number in the chemical inventory of the linked reference substance.
ref_sub.cas_number	The cas number of the linked reference substance.
ref_sub.cas_name	The cas name of the linked reference substance.
ref_sub.iupac_name	The iupac name of the linked reference substance.
ref_sub.smiles_notation	The smiles notation of the linked reference substance.
ref_sub.mol_formula	The molecular formula of the linked reference substance.
ref_sub.in_chl	The inChI of the linked reference substance.

3.2.15. byGhs

byGhs

Find entities containing GHS sections with certain criteria.

Note: the doc.type parameter can be one of:

- SUBSTANCE

- MIXTURE
- DOSSIER

Depending on the value of doc.type parameter, the following criteria are applied:

- in case of SUBSTANCE or MIXTURE, to the FLEXIBLE_RECORD.Ghs of a substance
- in case of DOSSIER, to the FLEXIBLE_RECORD.Ghs of the dossier's subject

Parameter name	Description
ghs.not_classified	Boolean. Is the 'Not classified' field checked?
ghs.hazard_statement	Picklist. The hazard statement (<i>path: FLEXIBLE_RECORD.Ghs.Labelling.HazardStatementsBlock.HazardStatements.HazardStatement</i>) Should be one of the phase codes in Phasegroup GHS65
ghs.hazard_pictogram	Picklist. The hazard pictogram (<i>path: FLEXIBLE_RECORD.Ghs.Labelling.HazardPictogramBlock.HazardPictogram.Code</i>) Should be one of the phase codes in Phasegroup DM02
ghs.precaution_statement	Picklist. The precautionary statement (<i>path: FLEXIBLE_RECORD.Ghs.Labelling.PrecautionaryStatementsBlock.PrecautionaryStatements.PrecautionaryStatement</i>) Should be one of the phase codes in Phasegroup GHS66

3.2.16. byDsd

byDsd

Find entities containing DSD sections with certain criteria.

Note: the doc.type parameter can be one of:

- SUBSTANCE
- MIXTURE
- DOSSIER

Depending on the value of doc.type parameter, the following criteria are applied:

- in case of SUBSTANCE or MIXTURE, to the FLEXIBLE_RECORD.DsdDpd of a substance
- in case of DOSSIER, to the FLEXIBLE_RECORD.DsdDpd of the dossier's subject

Parameter name	Description
----------------	-------------

dsd.not_classified	Boolean. Is the 'Not classified' field checked?
dsd.risk	Picklist. The risk phrases (<i>path: FLEXIBLE_RECORD.DsdDpd.Labelling.RiskPhrases.Risks</i>) Should be one of the phase codes in Phasegroup N29
dsd.danger	Picklist. The indication of danger (<i>path: FLEXIBLE_RECORD.DsdDpd.Labelling.IndicationsOfDangerBlock.IndicationsOfDanger</i>) Should be one of the phase codes in Phasegroup N30
dsd.safety	Picklist. The safety phrases (<i>path: FLEXIBLE_RECORD.DsdDpd.Labelling.SafetyPhrases.Safeties.Code</i>) Should be one of the phase codes in Phasegroup N31

3.2.17. byJointSubmission

byJointSubmission

Find entities containing 'Joint submission' sections with certain criteria.

Note: the doc.type parameter can be one of:

- SUBSTANCE
- MIXTURE
- DOSSIER

Depending on the value of doc.type parameter, the following criteria are applied:

- in case of SUBSTANCE or MIXTURE, to the FLEXIBLE_RECORD.JointSubmission of a substance
- in case of DOSSIER, to the FLEXIBLE_RECORD.JointSubmission of the dossier's subject

Parameter name	Description
joint_submission.name	The name of the joint submission
joint_submission.leader	The name of the leader legal entity.
joint_submission.member	The name of the member legal entity.

3.2.18. byCategory (since v2.0.0)

byCategory

Find CATEGORY entities by various criteria.

Note: The doc.type parameter should be omitted

Parameter name	Description
doc.name	The category name
cat.public_name	The public name of the category
sub.uuid	The uuid of the substance category member
le.name	The legal entity name.
le.country	Picklist. The country of the legal entity. Should be one of the phase codes in Phasegroup N03
le.town	The town of the legal entity.

3.2.19. SubstanceRelatedCategories

byCategory

Find CATEGORY entities that are related to a Substance. A Category is considered related to a given Substance, if its 'Substances' field references the Substance.

Parameter name	Description
key	The document key of the substance

3.2.20. SubstanceRelatedMixtures

byCategory

Find MIXTURE entities that are related to a Substance. A Mixture is considered related to a given Substance, if it has a Mixture Composition document that has the Substance as component, impurity or additive.

Parameter name	Description
key	The document key of the substance

4. Document handling

Every REST resource in this document is under `/iuclid6-ext/api/ext/v1/`

4.1. Datatypes

4.1.1. JsonDocumentEnvelope

A IUCLID6 document is represented by a `JsonDocumentEnvelope` object, that is a JSON array of two elements, the `JsonDocumentHeader` being the first, followed by the `JsonDocumentContent`.

Media type

The media type for the `JsonDocumentEnvelope` object is `application/vnd.iuclid6.ext+json;type=iuclid6.Document`

To give an example, the following snippet represents a `SubstanceComposition` section document:

JsonDocumentEnvelope

```
[
  {
    "key": "7bcaf4db-9f2c-403f-aa79-c2cadedf74a0e/0",
    "definition": "FLEXIBLE_RECORD.SubstanceComposition",
    "parentKey": "9915ba13-ffa7-405b-b024-c06296b43820/0",
    "parentDefinition": "SUBSTANCE",
    "order": 1,
    "name": "Composition.001",
    "attachments": [
      "e1c921f6-5c85-4e0a-a5c5-33fabade0a2f/0",
      "cd166edf-1d7f-49eb-b1c0-d7424c28de1f/0"
    ],
    "createdOn": "2016-09-13T17:41:43.220Z",
    "modifiedOn": "2016-09-13T17:42:25.609Z"
  },
  {
    "GeneralInformation": {
      "Name": "substance-composition-1",
      "TypeOfComposition": {
        "code": "61423"
      }
    }
  }
],
```

```

"Constituents": {
  "Constituents": [
    {
      "uuid": "e67542c8-094b-422e-8bbb-d65d6bd66d22",
      "ReferenceSubstance": "b0c168d2-c97c-4cc7-83d0-9b2716759030/0",
      "ProportionTypical": {
        "lowerQualifier": ">",
        "lowerValue": 1,
        "unit": {
          "code": "2098"
        }
      }
    }
  ]
}
]

```

Note, that to make the response more compact, not every field of the SubstanceComposition document is included, only the ones containing data.

4.1.1.1. JsonDocumentHeader

The **JsonDocumentHeader** contains **metadata about the document**, such as the document key, the definition identifier, etc.

In detail it includes the following fields:

Name	Type	Description
key	String	The document key, formatted as {uuid}/{snapshot}
definition	String	The document definition identifier. <i>e.g. FLEXIBLE_RECORD.SubstanceComposition</i>
parentKey	String	Only if the document is a section. The document key of the parent, formatted as {uuid}/{snapshot}
parentDefinition	String	Only if the document is a section. The document definition identifier of the parent. <i>e.g. SUBSTANCE</i>
order	Number	Only if the document is a section. The order of the document within the group of documents of the same definition.
name	String	The document name, up to 255 characters.

Name	Type	Description
attachments	Array	An array of attachment keys formatted as {uuid}/{snapshot}
createdOn	Date	The instant the document was created, in ISO 8601 format. <i>e.g. 2016-09-06T13:14:59.187Z</i>
modifiedOn	Date	The instant the document was last modified, in ISO 8601 format. <i>e.g. 2016-09-06T13:14:59.187Z</i>
templates	Array	An array of template keys formatted as {uuid}/{snapshot}
creationTool	String	The name of the application that created the entity, e.g. 'IUC6'
snapshotCreationTool	String	The name of the application that created the dossier, e.g. 'IUC6'

4.1.1.2. JsonDocumentContent

The **JsonDocumentContent** contains the **actual data of the document**, the structure of this object varies according to the structure definition of each document, which can be found in **I.TE.M. (Iuclid Template Manager)**

In general, it follows a tree-like structure where each node can either be a container or a leaf node.

Container nodes can contain leaf nodes and other container nodes, while leaf nodes contain field values.

In the lists below you can find how each datatype of I.TE.M. is represented in the json format of the **JsonDocumentContent**.

4.1.1.3. Container types

I.TE.M Type	Type	Description
Header	Object	<p>It defines a grouping of fields and other container elements</p> <p>A JSON object with properties for every contained element.</p> <p>e.g. The "GeneralInformation" header of the "SubstanceComposition" document</p> <pre>"GeneralInformation": { "Name": "substance-composition-1", "TypeOfComposition": { "code": "61423" } }</pre>

Repeatable Set	Array	It defines a grouping of fields and other container elements, that can be repeated.
Repeatable List		<p>Both varieties are converted to the same json representation</p> <p>A JSON array containing JSON objects for every entry of the repeatable set/list..</p> <p>Each object, in addition to its other contents, will include the uuid for that particular entry.</p> <p>e.g. The "Constituents" repeatable set of the "SubstanceComposition" document</p> <pre> "Constituents": [{ "uuid": "e67542c8-094b-422e-8bbb-d65d6bd66d22", "ReferenceSubstance": "b0c168d2-c97c-4cc7-83d0-9b2716759030/0", "ProportionTypical": { "lowerQualifier": ">", "lowerValue": 1, "unit": { "code": "2098" } } }] </pre>

4.1.1.4. Field types

I.TE.M Type	Type	Description
Date	string	A date in ISO 8601 format. e.g. 2016-09-06
TextField255	string	Text up to 255 characters.
MultiLineText2000	string	Text up to 2000 characters.
TextArea32768	string	Text up to 32786 characters.
TextTemplate	string	Text up to 32786 characters.
RichTextArea	string	Text up to 32786 characters. Can contain html
CheckBox	boolean	A true/false value
Integer	number	A integer value
Decimal	number	A decimal value

PickList	object	<p>An object containing the 'code' string field, and optionally a 'other' string field if the phrase is '<i>open</i>'</p> <p>The allowed values of the 'code' field vary with the 'phrase-group' defined for this element in I.TE.M</p> <p>e.g.</p> <pre>{ "code": "1342", "other": "other" }</pre>
PickListWithRemarks	object	<p>An object containing the 'code' string field, and optionally a 'other' string field if the phrase is <i>open</i> and a 'remarks' string field.</p> <p>e.g.</p> <pre>{ "code": "1342", "other": "other", "remarks": "remarks" }</pre>
PickListWithRemarks2000		
PickListWithRemarks32000		
UnitMeasure	object	<p>An object containing a 'value' number and a 'unit' object that is a picklist</p> <p>e.g.</p> <pre>{ "value": 1.3, "unit":{ "other": "2493" } }</pre>

<p>RangeDecimalWithPickList</p>	<p>object</p>	<p>An object containing the following:</p> <table border="1"> <thead> <tr> <th data-bbox="727 275 930 327">field</th> <th data-bbox="930 275 1052 327">type</th> <th data-bbox="1052 275 1430 327">description</th> </tr> </thead> <tbody> <tr> <td data-bbox="727 327 930 558">lowerQualifier</td> <td data-bbox="930 327 1052 558">string</td> <td data-bbox="1052 327 1430 558"> one of: <ul style="list-style-type: none"> • ca. • > • >= </td> </tr> <tr> <td data-bbox="727 558 930 611">lowerValue</td> <td data-bbox="930 558 1052 611">number</td> <td data-bbox="1052 558 1430 611">the lower numeric boundary</td> </tr> <tr> <td data-bbox="727 611 930 842">upperQualifier</td> <td data-bbox="930 611 1052 842">string</td> <td data-bbox="1052 611 1430 842"> one of: <ul style="list-style-type: none"> • ca. • < • <= </td> </tr> <tr> <td data-bbox="727 842 930 936">upperValue</td> <td data-bbox="930 842 1052 936">number</td> <td data-bbox="1052 842 1430 936">the upper numeric boundary</td> </tr> <tr> <td data-bbox="727 936 930 989">unit</td> <td data-bbox="930 936 1052 989">object</td> <td data-bbox="1052 936 1430 989">a picklist object.</td> </tr> </tbody> </table>	field	type	description	lowerQualifier	string	one of: <ul style="list-style-type: none"> • ca. • > • >= 	lowerValue	number	the lower numeric boundary	upperQualifier	string	one of: <ul style="list-style-type: none"> • ca. • < • <= 	upperValue	number	the upper numeric boundary	unit	object	a picklist object.
field	type	description																		
lowerQualifier	string	one of: <ul style="list-style-type: none"> • ca. • > • >= 																		
lowerValue	number	the lower numeric boundary																		
upperQualifier	string	one of: <ul style="list-style-type: none"> • ca. • < • <= 																		
upperValue	number	the upper numeric boundary																		
unit	object	a picklist object.																		
<p>RangeDecimal</p>	<p>object</p>	<p>same as 'RangeDecimalWithPickList' without the 'unit' field.</p>																		
<p>HalfBounded</p>	<p>object</p>	<p>An object containing the following:</p> <table border="1"> <thead> <tr> <th data-bbox="727 1173 930 1226">field</th> <th data-bbox="930 1173 1052 1226">type</th> <th data-bbox="1052 1173 1430 1226">description</th> </tr> </thead> <tbody> <tr> <td data-bbox="727 1226 930 1562">lowerQualifier</td> <td data-bbox="930 1226 1052 1562">string</td> <td data-bbox="1052 1226 1430 1562"> one of: <ul style="list-style-type: none"> • <ul style="list-style-type: none"> ○ ca. ○ < ○ <= ○ > ○ >= </td> </tr> <tr> <td data-bbox="727 1562 930 1656">lowerValue</td> <td data-bbox="930 1562 1052 1656">number</td> <td data-bbox="1052 1562 1430 1656">the lower numeric boundary</td> </tr> <tr> <td data-bbox="727 1656 930 1709">unit</td> <td data-bbox="930 1656 1052 1709">object</td> <td data-bbox="1052 1656 1430 1709">a picklist object.</td> </tr> </tbody> </table>	field	type	description	lowerQualifier	string	one of: <ul style="list-style-type: none"> • <ul style="list-style-type: none"> ○ ca. ○ < ○ <= ○ > ○ >= 	lowerValue	number	the lower numeric boundary	unit	object	a picklist object.						
field	type	description																		
lowerQualifier	string	one of: <ul style="list-style-type: none"> • <ul style="list-style-type: none"> ○ ca. ○ < ○ <= ○ > ○ >= 																		
lowerValue	number	the lower numeric boundary																		
unit	object	a picklist object.																		

Confidentiality	object	An object containing the following:		
		field	type	description
		confidentiality	object	a picklist object with codes from phase-group N64 : <ul style="list-style-type: none"> • 2732 (<i>confidential business information</i>) • 2859 (<i>intellectual property</i>) • 3441 (<i>not publicly available</i>)
		legislations	array	an array of picklist objects. The phrasegroup of each picklist is N78
justification	string	text up to 32786 characters. Can be populated with text template TT_501		
MultiSelect	array	An array of picklist objects with/without remarks.		
MultiSelectWithRemarks		e.g.		
MultiSelectWithRemarks2000		[
MultiSelectWithRemarks32000		<pre> { "code": "60224", "remarks": "remarks-1" }, { "code": "60225", "remarks": "remarks-2" }, { "code": "60230" }] </pre>		
EntityReferenceField	string	a document key, formatted as {uuid}/{snapshot}		
EndpointReferenceField		e.g. "4f88bc7f-395c-4d0b-997b-14e8c9aef605/0"		
SingleFileAttachment	string	an attachment key, formatted as {uuid}/{snapshot}		
Image		e.g. "4f88bc7f-395c-4d0b-997b-14e8c9aef605/0"		
EntityReferenceList	array	an array of strings, that represent keys, either of		

EndpointReferenceList		documents or attachments.
AttachmentsList		The keys are formatted as {uuid}/{snapshot}
LiteratureReferenceList		e.g. ["77748886-6911-4a41-af93-7d5a86fc075f/0", "f9d571cd-d0ed-403d-b944-21e8d26c7c26/0"]

4.1.2. Response types

4.1.2.1. Links

The **Links** object is returned from resources having create semantics. Typically these resources will respond with **HTTP CODE 201 (created)** and will include the Links object in the response payload.

It contains the following fields:

Name	Type	Description
source	string	The URI of the created entity/document.
links	object	For future use. Currently empty.

4.1.2.2. Error

The **Error** object is returned when something goes wrong, and it contains more detailed information about the source of the problem.

It contains the following fields:

Name	Type	Description
requestId	string	Optional. The request identifier. As sent by the client or generated by the server if not sent.
uri	string	The I6Uri of the request resource.
subjectKey	string	Optional. The key of the entity that is related with the error.
code	string	The specific code of the error. <i>Not to be confused with the HTTP code.</i>

Name	Type	Description
message	string	Optional. The error message.
info	object	Optional. An object that varies with the specific error and contains detailed information about the error.

4.2. Semantics of HTTP status codes

4.2.1. Success codes

HTTP Status Code	Usage
200 OK	Success of data retrieval operations such as <i>read, list, search, export, print, ...</i>
200 OK	Success of update operations, such as <i>update, edit, import</i>
204 No Content	Success of update or deletion operations, when no entity is included in the response, e.g. <i>delete, remove</i>
201 Created	Success of creation operations
308 Permanent Redirect	Used for generic URI that don't change, such as <code>/dossier/{uuid}/subject</code>

4.2.2. Error codes

HTTP Status Code	Usage
400 Bad Request	The request entity or request parameters is not parsable by the server, either because it is malformed at the protocol level, or because it is malformed at the format level (XML, JSON, number or date representation).
403 Forbidden	The caller is attempting to perform an operation on a resource he is allowed to see (otherwise 404 would be appropriate) but hasn't sufficient privileges.
404 Not Found	If the resource identified by the URI path does not exist (at any depth), or a resource identified in the query string that is essential to the representation does not exist.
405 Method Not Allowed	If the method is not supported by the target resource, regardless of the privileges of the caller. E.g. update a dossier document, create a dossier for an unsupported entity type
406 Not Acceptable	If the requested media type is not supported for the resource or not known.

HTTP Status Code	Usage
415 Unsupported Media Type	If the resource cannot handle the media type sent by the caller.
422 Unprocessable Entity	To indicate business validation errors on the received data. Unless those are already covered by 400, 403 or 404.
500 Internal Server Error	No particular business meaning can be associated to this code, as it is the result of an internal unexpected exception.

4.3. Load

Loading of IUCLID6 documents is supported by a **HTTP GET** call on the appropriate REST resources, and requesting the JsonDocumentEnvelope via the **Accept** header:

Accept	application/vnd.iuclid6.ext+json;type=iuclid6.Document
---------------	---

4.3.1. Raw data

Raw documents are the documents that their content can be **modified**. These documents can be either **entities**, or **sections**.

4.3.1.1. Entities

Entities in IUCLID6 are the top-level documents (without a parent), that may or may not contain children.

They are categorized by their entity-type:

- SUBSTANCE
- MIXTURE
- TEMPLATE
- CATEGORY
- ANNOTATION
- LEGAL_ENTITY
- SITE
- REFERENCE_SUBSTANCE
- CONTACT
- LITERATURE
- TEST_MATERIAL_INFORMATION

More entity-types might be added in the future

To load an entity, two pieces of information are needed, its **entity-type** and its **uuid**:

```
GET /raw/{entity_type}/{uuid}
```

So, for example, to load the legal entity with uuid 4f88bc7f-395c-4d0b-997b-14e8c9aef605 execute the following:

```
curl --request GET \
  --url http://localhost:8080/iuclid6-ext/api/ext/v1/raw/LEGAL_ENTITY/4f88bc7f-395c-4d0b-997b-14e8c9aef605 \
  --header 'accept: application/vnd.iuclid6.ext+json; type=iuclid6.Document' \
  --header 'iuclid6-user: SuperUser' \
  --header 'iuclid6-pass: *****'
```

4.3.1.2. Sections

Section documents are identified by a **uuid** and a **document definition**, same as entities, **however** they exist in the context of their parent entity, therefore they can only be accessed through the sub-tree of their parent.

The following, obtains a listing of every section contained in a given entity:

```
GET /raw/{entity_type}/{uuid}/documents
```

To get only the sections of a specific definition;

```
GET /raw/{entity_type}/{uuid}/document/{doc_def}
```

To call any of the previous two resources you need to specify **application/vnd.iuclid6.ext+json** as the **Accept** header. These resources return a wrapper object containing the URIs of the sections, see the searching documentation for more details.

Notice the change from plural **documents** to singular **document** in the previous resources.

By default the above two resources do not list inherited documents, however this can be altered by setting the query param '**inherited**' to true

To retrieve a specific section:

```
GET /raw/{entity_type}/{uuid}/document/{doc_def}/{doc_uuid}
```

For example, to load the `SubstanceComposition` of a `Substance`:

```
curl --request GET \
  --url http://localhost:8080/iuclid6-ext/api/ext/v1/raw/SUBSTANCE/9915ba13-ffa7-405b-b024-
c06296b43820/document/FLEXIBLE_RECORD.SubstanceComposition/1aca8956-fbfe-4f2e-
80f7-54182bc72454 \
  --header 'accept: application/vnd.iuclid6.ext+json; type=iuclid6.Document' \
  --header 'iuclid6-user: SuperUser' \
  --header 'iuclid6-pass: *****'
```

4.3.1.3. Attachments

You can download the content of an attachment on the following resource:

```
GET /raw/attachment/{attachment_uuid}
```

The attachment content will be contained in the response body and information about the attachment such as size, mime type and md5 hash will be in the relevant standard HTTP headers:

Header	Description
Content-Type	the mime type of the attachment content
Content-Length	The size in bytes of the attachment content
Content-MD5	The md5 hash of the attachment content
Content-Disposition	The attachment filename will be contained in the 'filename' parameter

4.3.2. Dossier

The IUCLID6 **dossier**, is a structured set of **non-modifiable**, documents related with each other.

Every dossier includes:

- a **header document**

- a **dossier subject** entity along with its section documents
- a **submitting legal entity**
- a list of other entities referenced by the subject entity or its sections.

4.3.2.1. Header document

You access the contents of the header document by:

GET	/dossier/{dossier_uuid}/header
------------	--------------------------------

4.3.2.2. Subject

The dossier subject is available on:

GET	/dossier/{dossier_uuid}/subject
------------	---------------------------------

4.3.2.3. Sections of subject

The section documents of the dossier's subject entity can be accessed in a similar fashion as in the [raw sections](#), for example:

GET	/dossier/{dossier_uuid}/subject/documents
------------	---

To access a specific document included in the dossier's subject entity:

GET	/dossier/{dossier_uuid}/subject/document/{doc_def}/{document_uuid}
------------	--

To access a specific document included in a dossier's entity:

GET	/dossier/{uuid}/{entity_type}/{entity_uuid}/document/{doc_def}/{doc_uuid}
------------	---

4.3.2.4. Submitter

The submitting legal entity of the dossier is accessible through:

GET	/dossier/{dossier_uuid}/submitter
------------	-----------------------------------

4.3.2.5. Dossier components

Dossier components are considered all the entities in a dossier, **including** the **subject** and **submitter**, **except** the dossier header.

Listing every component of a dossier, without specifying its type, is not supported in this version of the API

You can access any of them via the following resource:

GET	/dossier/{uuid}/{entity_type}/{entity_uuid}
------------	---

As with the section documents of the subject entity, the sections of a dossier component can be accessed in a similar fashion as in the [raw sections](#).

4.3.2.6. Dossier attachments

Similarly with raw attachments, you can access the content of a dossier attachment on the following resource:

GET	/dossier/{uuid}/attachment/{attachment_uuid}
------------	--

The attachment content will be contained in the response body and information about the attachment such as size, mime type and md5 hash will be in the relevant standard HTTP headers:

Header	Description
Content-Type	the mime type of the attachment content
Content-Length	The size in bytes of the attachment content
Content-MD5	The md5 hash of the attachment content
Content-Disposition	The attachment filename will be contained in the 'filename' parameter

4.4. Write raw data

Writing operations, namely:

- **creation**
- **modification**
- and **deletion**

of documents, are supported by using the relevant HTTP verbs POST, PUT and DELETE respectively. Wherever content needs to be sent, its type is expected to be:

Content-Type	application/vnd.iuclid6.ext+json;type=iuclid6.Document
---------------------	---

4.4.1. Entities

4.4.1.1. Create

The creation of entities is supported by posting the `JsonDocumentEnvelope` to create, to the relative entity's collection resource:

POST	<code>/raw/{entity_type}</code>
-------------	---------------------------------

For instance, the following HTTP request creates a new literature reference document:

```
curl -X POST -H "IUCLID6-USER: SuperUser" -H "IUCLID6-PASS: *****" -H "Content-Type: application/vnd.iuclid6.ext+json; type=iuclid6.Document" -d '[
{
  "definition": "LITERATURE",
  "name": "A demo literature reference"
},
{
  "GeneralInfo": {
    "LiteratureType": {
      "code": "1586"
    },
    "Name": "A demo literature reference",
    "Author": "Unknown",
    "ReferenceYear": 2016,
    "Source": "unknown source",
    "TestLab": "unknown lab",
    "ReportNo": "xx-xxxx-x",
    "CompanyOwner": "unknown owner",
    "CompanyOwnerStudyNo": "xx-xxxx-x",
    "ReportDate": "2016-09-20",
    "Remarks": "nothing remarkable"
  }
}
]' "http://localhost:8080/iuclid6-ext/api/ext/v1/raw/LITERATURE"
```

It is possible to create a new document with a specific key, by including the **key** property in the posted `JsonDocumentHeader`, be prepared however to handle the **409** (conflict) HTTP error code, in case the specified key already exists.

If the entity is successfully created, **201** (Created) is returned along with the [Links](#) object.

Error responses

HTTP Status	Payload	Reason
-------------	---------	--------

code		
409	e.g. <pre>{ "requestId": "f1017ae7-ab4e-426e-a597-c699d6819a34", "uri": "iuclid6:/0/SUBSTANCE", "subjectKey": "a97d1b21-1f58-44ae-a2dc-58781ebb4abb/0", "code": "DOC409", "message": "Duplicate document key: a97d1b21-1f58-44ae-a2dc-58781ebb4abb/0" }</pre>	A document with the same key already exists.
422	e.g. <pre>{ "requestId": "015737f7-8b5c-410f-9218-fb6be90dddac", "uri": "iuclid6:/0/SUBSTANCE", "code": "SYS400", "message": "Field 'SUBSTANCE.TypeOfSubstance.Composition': Phrase '2915123' is not a valid value for phrasegroup N08." }</pre>	The document content is invalid. This can be either due to incorrect structure, or value content.

4.4.1.2. Update

You can modify the contents of an entity, by using **PUT** on the entity's resource:

PUT	/raw/{entity_type}/{entity_uuid}
------------	----------------------------------

To change, for example, the author on the previously created literature:

```
curl -X PUT -H "IUCLID6-USER: SuperUser" -H "IUCLID6-PASS: *****" -H "Content-Type: application/vnd.iuclid6.ext+json; type=iuclid6.Document" -d '[
  {
    "definition": "LITERATURE"
  },
  {
    "GeneralInfo": {
      "LiteratureType": {
        "code": "1586"
      },
      "Name": "A demo literature reference",
      "Author": "John Doe",
      "ReferenceYear": 2016,
      "Source": "unknown source",
      "TestLab": "unknown lab",
```

```

    "ReportNo": "xx-xxxx-x",
    "CompanyOwner": "unknown owner",
    "CompanyOwnerStudyNo": "xx-xxxx-x",
    "ReportDate": "2016-09-20",
    "Remarks": "nothing remarkable"
  }
}
] "http://localhost:8080/iuclid6-ext/api/ext/v1/raw/LITERATURE/fe9dbc97-1c00-4f4f-875f-906c7b864443"

```

If the request succeeds, **HTTP CODE 204** will be returned.

Error responses

Same as in create scenario.

4.4.1.3. Delete

Entity deletions can be achieved using the **DELETE** verb on the entity resource:

DELETE	/raw/{entity_type}/{entity_uuid}
---------------	----------------------------------

If the request succeeds, **HTTP CODE 204** will be returned.

Error responses

HTTP Status code	Payload	Reason
409	e.g. <pre> { "requestId": "c53935bb-0002-4669-9030-2c5340419074", "uri": "iuclid6:/LEGAL_ENTITY/4f88bc7f-395c-4d0b-997b-14e8c9aef605", "subjectKey": "4f88bc7f-395c-4d0b-997b-14e8c9aef605/0", "code": "DOC502", "message": "Document is referred by other: 4f88bc7f-395c-4d0b-997b-14e8c9aef605/0 ", "info": { "referralKeys": [] } } </pre>	There are documents that refer to the entity you are trying to delete.

4.4.2. Composite entities

4.4.2.1. Sections reordering (since v3.0.0)

It is possible to reorder section documents in a dataset using the public API. In summary, the caller needs to specify the following instructions:

- the entity which contains the section documents to be reordered
- a list of "DocumentMoves", with each one describing the movement of a single section document from one position to another

A DocumentMove is an object composed of the following properties:

Name	Type	Description
uri	string	The I6URI of the section document to be reordered
from	integer	The current position of the section document (1-based). Taken only into account in case of validation
to	integer	The new desired position of the section document (1-based)

Validations performed

- The URI given in each DocumentMove is valid, in that the respective document exists
- The 'from' position given in each DocumentMove is the same as the actual position of the record to be moved

Notes:

- Validation can be relaxed by passing the query parameter 'validate' and setting its value to 'false'. In that case, the 'from' position of each DocumentMove is not validated against the actual position of the document to be moved.
- Reordering of multiple sections at the same time is supported. This means that one may pass a list of DocumentMoves that refer to documents belonging to different sections.

In order to **reorder** section documents of an entity, one can perform the following call:

```
PUT /raw/{ENTITY_TYPE}/{uuid}
```

where ENTITY_TYPE is the type of the entity (e.g., SUBSTANCE) and uuid is the identifier of the entity

The **Content-Type** header needs to be set as follows:

Content-Type	application/vnd.iuclid6.ext+json;type=standard.reorder
---------------------	--

The **Accept** header needs to be set as follows:

Accept	application/vnd.iuclid6.ext+json
---------------	----------------------------------

To specify the section documents to be reordered as well as the respective moves from one position to the other a list of DocumentMoves need to be provided in the HTTP request body

Body	<pre> [{ "uri": "iuclid6:/0/{ENTITY_TYPE}/{uuid}/{SUB_TYPE}/{doc-uuid}", "from": {from_position}, "to": {from_position} }, ...]</pre>
-------------	---

The following optional query parameter can also be used to control the reordering operation:

Query parameter name	Default value	Description
validate	true	Whether to skip validation on the current position of sections as provided through the 'from' property

For example, in order to reorder two section documents in one invocation, the following call can be performed:

```
curl --request PUT \
--url http://localhost:8080/iuclid6-ext/api/ext/v1/raw/SUBSTANCE/sub-uuid \
--header 'content-type: application/vnd.iuclid6.ext+json;type=standard.reorder' \
--header 'accept: application/vnd.iuclid6.ext+json' \
--header 'iuclid6-user: username' \
--header 'iuclid6-pass: *****' \
--data '{\
"uri": "iuclid6:/0/SUBSTANCE/sub-
uuid/ENDPOINT_STUDY_RECORD.Hydrolysis/doc-uuid-1", \
"from": 1, \
"to": 2 \
}, {\
"uri": "iuclid6:/0/SUBSTANCE/sub-
uuid/ENDPOINT_STUDY_RECORD.Hydrolysis/doc-uuid-2", \
"from": 2, \
"to": 1 \
}]'
```

4.4.2.2. Copy sections to target entity (since v3.0.0)

In order to **copy a template** into a dataset, one can perform the following call:

POST	/system/script
-------------	----------------

To specify the template to be copied as well as the dataset to which it will be copied, the following information needs to be provided in the HTTP request body:

Body	{ "sourceEntity": "iuclid6:/0/TEMPLATE/{templ-uuid}", "targetEntity": "iuclid6:/0/{ENTITY_TYPE}/{e-uuid}" }
-------------	--

where "templ-uuid" is the identifier of the template, while "ENTITY_TYPE" and "e-uuid" are the entity type and its identifier of the dataset respectively.

The **Content-Type** header needs to be set as follows:

Content-Type	application/vnd.iuclid6.ext+json;type=iuclid6.copy
---------------------	--

The **Accept** header needs to be set as follows:

Accept	application/vnd.iuclid6.ext+json;type=iuclid6.CopyResult
---------------	--

The result of this call is an object which contains a list of document I6URIs, which correspond to the newly created documents that have been copied from the template.

```
{
  "documents": [
    "iuclid6:/0/{ENTITY_TYPE}/{e-uuid}/{doc-def-id}/{doc-uuid}",
    ...
  ]
}
```

For example, in order to copy a template with uuid "templ-uuid" into a substance dataset with uuid "sub-uuid" we may perform the following call:

```
curl --request POST \
  --url http://localhost:8080/iuclid6-ext/api/ext/v1/system/script \
  --header 'content-type: application/vnd.iuclid6.ext+json;type=iuclid6.copy' \
  --header 'accept: application/vnd.iuclid6.ext+json;type=iuclid6.CopyResult' \
  --header 'iuclid6-user: username' \
  --header 'iuclid6-pass: *****' \
  --data '{"sourceEntity":"iuclid6:/0/TEMPLATE/templ-uuid","targetEntity":"iuclid6:/0/SUBSTANCE/sub-uuid"}'
```

4.4.3. Sections

The modification operations on section documents are very similar to those on entities. The only difference is the resources that you make the requests.

Briefly:

4.4.3.1. Create	POST	/raw/{entity_type}/{entity_uuid}/document/{doc_def}
4.4.3.2. Update	PUT	/raw/{entity_type}/{entity_uuid}/document/{doc_def}/{doc_uuid}
4.4.3.3. Delete	DELETE	/raw/{entity_type}/{entity_uuid}/document/{doc_def}/{doc_uuid}

The modification operations on section documents are nearly identical to those on entities. The only difference is the resources that you make the requests.

4.4.4. Attachments

You can upload binary content that will be attached to iuclid documents by posting the content to the following resource:

POST	/raw/attachment
-------------	-----------------

You can specify a filename for the uploaded content by the Content-Disposition request header, for example:

Content-Disposition: attachment;filename*=utf-8"my-attachment-filename.txt"

Currently omitting the Content-Disposition header leads to 500 error, this will be fixed in a subsequent release

If the attachment upload is successful, the server will respond with **201 Created** and the response body will contain information about the created attachment:

```
{
  "snapshotUuid": "0",
  "uuid": "a4fdfe60-9852-4a38-929a-50415b8bebd1",
  "filename": "thor.jpg",
  "md5": "8a3aca8c73ecb6f9338d88d73b57a841",
  "size": 62208,
  "mediaType": "image/jpeg"
}
```

Name	Type	Description
snapshotUuid	string	The snapshot uuid of the created attachment. Will always be '0'
uuid	string	The uuid of the created attachment
filename	string	The filename of the attachment
md5	string	The md5 hash of the attachment content
size	string	The size in bytes of the attachment content
mediaType	object	The mime type of the attachment content

4.5. Create Dossier

To create a dossier of a particular subject you can **POST** a JsonDocumentEnvelope of the header document of the dossier to be created on the following resource:

POST	/raw/{entity_type}/{entity_uuid}/dossiers/{submission_type}
-------------	---

For example, to create a **REACH Inquiry** dossier of a specific substance:

```
curl -X POST -H "IUCLID6-USER: SuperUser" -H "IUCLID6-PASS: *****" -H "Content-Type: application/vnd.iuclid6.ext+json; type=iuclid6.Document" -d '[
  {
    "definition": "DOSSIER.R_INQUIRY",
    "name": "demo dossier"
  },
  {
    "DossierSpecificInformation": {
      "Remarks": "demo remarks"
    }
  }
]' "http://localhost:8080/iuclid6-ext/api/ext/v1/raw/SUBSTANCE/5307a86a-afd4-4da9-9c01-4ac12b1bc9c0/dossiers/R_INQUIRY"
```

The above POST request will create a dossier with the default settings, that will include every document and field that is present in the subject entity dataset.

To alter the default behavior, and specify exclusions, instead of sending only the header document, you can send a **DossierCreationDTO** object that, apart of the dossier header, can contain configuration parameters for the dossier creation process.

Content-Type: application/vnd.iuclid6.ext+json; type=iuclid6.DossierCreationDTO

```
{
  "excludeLegalEntity": true,
  "excludeNotRequiredDocuments": true,
  "excludeAnnotations": false,
  "excludeConfidentialFields": false,
  "excludeDetailFields": false,
  "filterCategoryMemberChildren": false,
  "excludeConfidentialityRestrictions": [],
  "excludeRegulationRestrictions": [],
  "excludeDocuments": [],
  "header": [
    {
      "definition": "DOSSIER.R_INQUIRY",
      "name": "demo dossier"
    },
    {
      "DossierSpecificInformation": {
        "Remarks": "demo remarks"
      }
    }
  ]
}
```

The **DossierCreationDTO** object contains the following properties:

Name	Type	Description	Default value
excludeLegalEntity	boolean	Exclude the legal entity of the subject.	true
excludeNotRequiredDocuments	boolean	Exclude not required sections	true
excludeAnnotations	boolean	Exclude annotations	false
excludeConfidentialFields	boolean	Exclude fields marked as confidential	false
excludeDetailFields	boolean	Exclude fields marked as detailed	false
filterCategoryMemberChildren	boolean	Exclude the section documents of category member substances	false
excludeConfidentialityRestrictions	array	Exclude fields with the specified confidentiality claims	empty array
excludeRegulationRestrictions	array	Exclude fields with the specified regulatory programme	empty array
excludeDocuments	array	Exclude specific documents by document key.	empty array
header	array	The header for the dossier as a JsonDocumentEnvelope	

For example, to create a **REACH Inquiry** dossier of a specific substance:

```
curl -X POST -H "IUCLID6-USER: SuperUser" -H "IUCLID6-PASS: *****" -H "Content-Type: application/vnd.iuclid6.ext+json; type=iuclid6.DossierCreationDTO" -d '
```

```
{
  "excludeLegalEntity": true,
  "excludeNotRequiredDocuments": true,
  "excludeAnnotations": false,
  "excludeConfidentialFields": false,
  "excludeDetailFields": false,
  "filterCategoryMemberChildren": false,
  "excludeConfidentialityRestrictions": [],
  "excludeRegulationRestrictions": [],
  "excludeDocuments": [],
  "header": [
    {
      "definition": "DOSSIER.R_INQUIRY",
      "name": "demo dossier"
    }
  ],
  "DossierSpecificInformation": {
    "Remarks": "demo remarks"
  }
}
```

```

    }
  }
}
} "http://localhost:8080/iuclid6-ext/api/ext/v1/raw/SUBSTANCE/5307a86a-afd4-4da9-9c01-4ac12b1bc9c0/dossiers/R_INQUIRY"

```

Note that each submission type has a corresponding definition for the header document of the dossier, trying to create a dossier with submission type and header definition that not match, will fail.

By convention, the header document definition identifiers are in the form:
 "DOSSIER.{submission_type}"

Error responses

HTTP Status code	Payload	Reason
405	e.g. <pre> { "requestId": "8394333f-9748-49fd-b0f0-af800a1ce953", "uri": "iuclid6:/0/LEGAL_ENTITY/4f88bc7f-395c-4d0b-997b-14e8c9aef605", "subjectKey": "LEGAL_ENTITY:4f88bc7f-395c-4d0b-997b-14e8c9aef605/0", "code": "DOS405", "message": "Invalid subject for dossier: LEGAL_ENTITY 4f88bc7f-395c-4d0b-997b-14e8c9aef605/0" } </pre>	You are trying to create a dossier for an entity, that can not be a subject of a dossier.
422	e.g. <pre> { "requestId": "530d1d96-0a0a-416d-9903-10e6204bc495", "uri": "iuclid6:/0/SUBSTANCE/2f4c245c-bcd9-43c2-add1-cfe199e7cc20", "subjectKey": "iuclid6.Document: DOSSIER.COMPLETE.foreign-field", "code": "EXT422", "message": "Validation failure for: DOSSIER.COMPLETE.foreign-field ", "info": { "componentType": "Module", "extensionName": "iuclid6", "componentName": "Document", "errorDetails": {} } } </pre>	The document content is invalid. This can be either due to incorrect structure, or value content.

4.6. Delete Dossier

Dossier deletions can be achieved using the **DELETE** verb on the dossier resource:

DELETE	/dossier/{dossier_uuid}
---------------	-------------------------

If the request succeeds, **HTTP CODE 204** will be returned.

4.7. Links (since v3.0.0)

4.7.1. Annotations

4.7.1.1. Link annotation to document

In order to **link** an annotation to a document we may perform the following call:

POST	/raw/ANNOTATION/{ann-uuid}/links/annotates
-------------	--

where ann-uuid corresponds to the identifier of the annotation entity.

The target document can be specified by the corresponding I6URI and passed through the body of the HTTP request:

Body	{i6Uri}
-------------	---------

The **Content-Type** header needs to be set as follows:

Content-Type	application/json
---------------------	------------------

For example, in order to annotate a dossier with uuid "dossier-uuid" with the annotation with uuid "ann-uuid" we may perform the following call:

```
curl --request POST \
  --url http://localhost:8080/iuclid6-ext/api/ext/v1/raw/ANNOTATION/ann-uuid/links/annotates \
  --header 'content-type: application/json' \
  --header 'iuclid6-user: username' \
  --header 'iuclid6-pass: *****' \
  --data "iuclid6:/dossier-uuid"
```

Valid resources that can be annotated are entities, dossiers or single documents within a composite entity. Some example I6URIs can be found below:

- Dossier: "iucld6:/<dossier-uuid>"
- Substance: "iucld6:/0/SUBSTANCE/<substance-uuid>"
- Substance Composition document: "iucld6:/0/SUBSTANCE/<substance-uuid>/FLEXIBLE_RECORD.SubstanceComposition/<composition-doc-uuid>"

4.7.1.2. Unlink annotation from document

In order to **unlink** an annotation from a document we may perform the following call:

DELETE	/raw/ANNOTATION/{ann-uuid}/links/annotates/{doc-uri}
---------------	--

where "ann-uuid" corresponds to the identifier of the annotation entity, and "doc-uri" to the "reduced" I6URI (without the prefix "iucld6:") of the annotated document .

The **Content-Type** header needs to be set as follows:

Content-Type	application/json
---------------------	------------------

For example, in order to unlink the annotation with uuid "ann-uuid" from the raw substance with uuid "sub-uuid" we may perform the following call:

```
curl --request DELETE \
  --url http://localhost:8080/iucld6-ext/api/ext/v1/raw/ANNOTATION/ann-
  uuid/links/annotates/0/SUBSTANCE/sub-uuid \
  --header 'content-type: application/json' \
  --header 'iucld6-user: username' \
  --header 'iucld6-pass: *****'
```

Note: Unlinking an annotation from a document does only affect the relation between the two resources. Neither the annotation or the document are deleted after this operation

4.7.1.3. List documents linked to an annotation

In order to list the documents that are annotated by an annotation we may perform the following call:

GET	/raw/ANNOTATION/{ann-uuid}/links/annotates
------------	--

where "ann-uuid" corresponds to the identifier of the annotation entity.

A formatter can be optionally provided as a query parameter for the returned documents to have a specific representation format.

For example, in order to retrieve the documents annotated by the annotation with uuid "ann-uuid" we may perform the following call:

```
curl --request GET \
  --url http://localhost:8080/iuclid6-ext/api/ext/v1/raw/ANNOTATION/ann-uuid/links/annotates \
  --header 'iuclid6-user: username' \
  --header 'iuclid6-pass: *****'
```

4.7.1.4. List annotations linked to a document

In order to list the annotations that are linked to a document we may perform any of the following calls, depending on the type of the document:

Annotations of any document within a dataset

GET /raw/{ENTITY_TYPE}/{e-uuid}/links/annotatedBy

Annotations of a specific document within a dataset

GET /raw/{ENTITY_TYPE}/{e-uuid}/document/{doc-def-id}/{doc-uuid}/links/annotatedBy

Annotations of any document within a dossier

GET /dossier/{dossier-uuid}/links/annotatedBy

Annotations of the header of a dossier

GET /dossier/header/{dossier-uuid}/links/annotatedBy

Annotations of any document within a dataset of a dossier

GET /dossier/header/{dossier-uuid}/{ENTITY_TYPE}/{e-uuid}/links/annotatedBy

Annotations of a specific document within a dataset of a dossier

GET	/dossier/header/{dossier-uuid}/{ENTITY_TYPE}/{e-uuid}/document/{doc-def-id}/{doc-uuid}/links/annotatedBy
------------	--

where "ENTITY-TYPE" corresponds to the entity type of a dataset (e.g., SUBSTANCE), "e-uuid" to the identifier of the dataset, "doc-uuid" to the identifier of a document with the dataset and "dossier-uuid" to the identifier of a dossier.

A formatter can be optionally provided as a query parameter for the returned documents to have a specific representation format.

For example, in order to retrieve the annotations related to the raw substance with uuid "sub-uuid" we may perform the following call:

```
curl --request GET \
  --url http://localhost:8080/iuclid6-ext/api/ext/v1/raw/SUBSTANCE/sub-uuid/links/annotatedBy \
  --header 'iuclid6-user: username' \
  --header 'iuclid6-pass: *****'
```

4.7.2. Templates

4.7.2.1. Link template to dataset (inherit)

In order to **link** a template to a dataset we may perform the following call:

POST	/raw/{ENTITY_TYPE}/{e-uuid}/links/inherits
-------------	--

where "ENTITY_TYPE" is the entity type of the dataset (e.g., SUBSTANCE) and "e-uuid" corresponds to the identifier of the dataset.

The template to be inherited can be specified by the corresponding I6URI and passed through the body of the HTTP request:

Body	iuclid6:/0/TEMPLATE/{templ-uuid}
-------------	----------------------------------

where "templ-uuid" is the identifier of the template.

The **Content-Type** header needs to be set as follows:

Content-Type	application/json
---------------------	------------------

For example, in order for a SUBSTANCE with uuid "sub-uuid" to inherit a template with uuid "templ-uuid" we may perform the following call:

```
curl --request POST \
  --url http://localhost:8080/iuclid6-ext/api/ext/v1/raw/SUBSTANCE/sub-uuid/links/inherits \
  --header 'content-type: application/json' \
  --header 'iuclid6-user: username' \
  --header 'iuclid6-pass: *****' \
  --data "iuclid6:/0/TEMPLATE/templ-uuid"
```

4.7.2.2. Unlink template from dataset

In order to **unlink** a template from a dataset we may perform the following call:

```
DELETE /raw/{ENTITY_TYPE}/{e-uuid}/links/inherits/{templ-uri}
```

where "ENTITY_TYPE" is the entity type of the dataset (e.g., SUBSTANCE), "e-uuid" the the identifier of the dataset, and "templ-uri" the I6URI of the inherited template.

The **Content-Type** header needs to be set as follows:

```
Content-Type application/json
```

For example, in order to unlink the template with uuid "templ-uuid" from the raw substance with uuid "sub-uuid" we may perform the following call:

```
curl --request DELETE \
  --url http://localhost:8080/iuclid6-ext/api/ext/v1/raw/SUBSTANCE/sub-
  uuid/links/inherits/iuclid6:/0/TEMPLATE/templ-uuid/TEMPLATE/templ-uuid \
  --header 'content-type: application/json' \
  --header 'iuclid6-user: username' \
  --header 'iuclid6-pass: *****'
```

Note: Unlinking a template from a dataset does only affect the relation between the two resources. Neither the template or the dataset are deleted after this operation.

4.7.2.3. List datasets linked to a template (inheriting template)

In order to list the documents that are annotated by an annotation we may perform the following call:

```
GET /raw/TEMPLATE/{templ-uuid}/links/inheritedBy
```

where "templ-uuid" corresponds to the identifier of the template entity.

A formatter can be optionally provided as a query parameter for the returned documents to have a specific representation format.

For example, in order to retrieve the datasets which inherit the template with uuid "templ-uuid" we may perform the following call:

```
curl --request GET \
  --url http://localhost:8080/iuclid6-ext/api/ext/v1/raw/TEMPLATE/templ-uuid/links/inheritedBy \
  --header 'iuclid6-user: username' \
  --header 'iuclid6-pass: *****'
```

4.7.2.4. List templates linked to a dataset (inherited by dataset)

In order to list the templates that are inherited by a dataset we may perform the following call, depending on the type of the document:

Templates inherited by a raw a dataset

```
GET /raw/{ENTITY_TYPE}/{e-uuid}/links/inherits
```

Templates inherited by a dataset in a dossier

```
GET /dossier/header/{dossier-uuid}/{ENTITY_TYPE}/{e-uuid}/links/inherits
```

where "ENTITY-TYPE" corresponds to the entity type of a dataset (e.g., SUBSTANCE), "e-uuid" to the identifier of the dataset, and "dossier-uuid" to the identifier of a dossier.

A formatter can be optionally provided as a query parameter for the returned documents to have a specific representation format.

For example, in order to retrieve the templates inherited by the raw substance with uuid "sub-uuid" we may perform the following call:

```
curl --request GET \
  --url http://localhost:8080/iuclid6-ext/api/ext/v1/raw/SUBSTANCE/sub-uuid/links/inherits \
  --header 'iuclid6-user: username' \
  --header 'iuclid6-pass: *****'
```

5. Entity access

Every REST resource in this document is under `/iuclid6-ext/api/ext/v1/`

5.1. Introduction

IUCLID 6 can enforce access control at the entity level, if the installation is configured to do so .

A user "owns" the entities she creates, and by default, no other user can access these.

It is desirable, however, for other users to be able to access data created by others. In order to achieve this, in a controlled fashion, two concepts emerge: "groups" and "access rights":

- A "**group**" is nothing more than a name to identify a set of particular users
- while "**access right**" is the level of privileges someone has upon an entity.

Four such levels are defined:

- "**read only**" allowing read but not modification
- "**read/write**" allowing both read and modification, but not deletion
- "**full access**" allowing deletion on top of read and modification
- and finally "**no access**" which is effectively the absence of any privilege.

A user can control who and with what privileges, will be able to access an entity she owns, by sharing that entity with a "group" specifying the "access right" for this group. It is possible, of course, for an entity to be shared across multiple groups.

The public REST API provide a means to both retrieve the sharing information of an entity and change it.

5.2. Datatypes

The standard media type for exchanging data access information is: **application/vnd.iuclid6.ext+json;type=standard.access**

It is a json array, of simple objects containing to properties: "**group**" and "**access**":

```
[
  {
    "group": "group A",
    "access": "READ_ONLY"
  },
  {
    "group": "Common",
    "access": "NO_ACCESS"
  }
]
```

Name	Type	Description
group	String	The name of the group
access	String	The access right, can be one of: <ul style="list-style-type: none"> • NO_ACCESS • READ_ONLY • READ_WRITE • FULL_ACCESS

Note that both the **group** and **access** are case sensitive

5.3. Semantics of HTTP status codes

5.3.1. Success Codes

HTTP Status Code	Usage
200 OK	Success of data retrieval operations
204 No Content	Success of update or deletion operations, when no entity is included in the response, e.g. <i>delete</i> , <i>remove</i>

5.3.2. Error codes

HTTP Status Code	Usage
403 Forbidden	The caller is not allowed to share the entity
404 Not Found	If the resource identified by the URI path does not exist (at any depth).
422 Unprocessable Entity	To indicate business validation errors on the received data.

5.4. Retrieve data access

You can retrieve the groups an **entity** is currently shared with, in addition to the groups that can be shared with, on:

GET	/raw/{entity_type}/{entity_uuid}/access
------------	---

or, in the case of **dossiers**:

GET	/dossier/{uuid}/access
------------	------------------------

For both of the above resources you should specify the **Accept** header:

Accept	application/vnd.iuclid6.ext+json;type=standard.access
---------------	---

For example to see the groups that the dossier with uuid: 'uuid-1' is shared to:

```
curl --request GET \
  --url http://localhost:8080/iuclid6-ext/api/ext/v1/dossier/uuid-1/access \
  --header 'accept: application/vnd.iuclid6.ext+json;type=standard.access' \
  --header 'iuclid6-user: username' \
  --header 'iuclid6-pass: *****'
```

5.5. Modify data access

You can **share**, or "**un-share**", an **entity** by posting the changes to:

POST	/raw/{entity_type}/{entity_uuid}/access
-------------	---

or, in the case of **dossiers**:

POST	/dossier/{uuid}/access
-------------	------------------------

For both of the above resources you should specify the **Content-Type** header:

Content-Type	application/vnd.iuclid6.ext+json;type=standard.access
---------------------	---

For example, lets say that the substance with uuid "uuid-1" is currently shared to both "group-A" and "group-B" with full access, and we wish to lower the access privilege for "group-B" to "READ_ONLY"

```
curl --request POST \
  --url http://localhost:8080/iuclid6-ext/api/ext/v1/raw/SUBSTANCE/uuid-1/access \
  --header 'content-type: application/vnd.iuclid6.ext+json;type=standard.access' \
```

```
--header 'iuclid6-user: username' \  
--header 'iuclid6-pass: *****' \  
--data '{"group":"group-B","access":"READ_ONLY"}'
```

6. Engines

Every REST resource in this document is under /iuclid6-ext/api/ext/v1/

6.1. Supported Types

6.1.1. Entities

The IUCLID 6 Public API supports **both**

- **raw** entities export
- and **dossier** export capabilities.

All raw entity types (both composite types and simple types) are supported for export and can be one of the following:

- SUBSTANCE
- MIXTURE
- TEMPLATE
- REFERENCE_SUBSTANCE
- TEST_MATERIAL_INFORMATION
- CATEGORY
- LEGAL_ENTITY
- LITERATURE
- SITE
- CONTACT
- ANNOTATION
- ATTACHMENT

Export of section documents on their own (providing the uuid of a section document for export) is currently not supported from the public API.

6.1.2. Dossiers

The IUCLID6 dossier, is a structured set of non-modifiable (snapshot), related documents. Apart from the dossier header, the dossier includes a dossier subject and a list of documents with their sections.

It also may or may not contain a submitting legal entity, according to the dossier creation process (submitting legal entity is not included by default in the created dossier, unless the user selects to include it).

6.2. Export

Export of IUCLID6 entities / dossiers is supported by an **HTTP POST** call on the appropriate REST resources ([Entity Export Resource](#) / [Dossier Export Resource](#)) by providing as **content-type** the **FullExport** type and an **empty body** "{}" in the http request:

Content-Type	application/vnd.iuclid6.ext+json;type=iuclid6.FullExport
---------------------	---

The above request, will create the proper BackgroundJob for the export process, which will be persisted on the IUCLID 6 database with (among other) the following information:

- a **job id**;
- **status**;
- **name** (the uuid of the exported entity);
- and the **export result** in binary format (blob).

An HTTP GET request can then be applied on the corresponding Job REST resource by **providing the job id** which has been created as a result of the previous HTTP POST request.

ACCEPT	application/vnd.iuclid6.ext+json;type=iuclid6.Iuclid6Job
GET	/system/job/{job_uuid}

An example of the response after applying the above request on uuid 7652670e-6f28-4867-a084-065bd25203ad corresponding to a SUBSTANCE entity is shown below:

```
curl --request GET
      --url 'http://localhost:8080/iuclid6-ext/api/ext/v1/system/job/7652670e-6f28-4867-a084-065bd25203ad'
      --header 'accept: application/vnd.iuclid6.ext+json;type=iuclid6.Iuclid6Job'
      --header 'IUCLID6-USER: SuperUser'
      --header 'IUCLID6-PASS: root'
```

JobInfo

```
{
  "id": "7652670e-6f28-4867-a084-065bd25203ad",
  "status": "SUCCEEDED",
  "uri": "/system/jobs/7652670e-6f28-4867-a084-065bd25203ad",
  "representation": {
    "classtype": "Iuclid6Job",
    "group": null,
    "name": "IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055",
    "type": "EXPORT",
```

```

        "documentUri":null,
        "logs": [
            {
                "code":null,
                "level":"INFO",
                "value":"Export process succeeded",
                "documentUri":null,
                "params":[]
            }
        ]
    }

```

The **status** of the Background export Job can have either of the following values:

- **QUEUED**: Job is queued for execution => Non-final state
- **IN_PROGRESS**: Job is being processed by IUCLID => Non-final state
- **SUCCEEDED**: Job completed successfully => Final state
- **FAILED**: Job failed => Final state
- **CANCELED**: Job is canceled => Final state

The **status** of the background export job can be provided by applying an HTTP GET request on the corresponding job resource by providing only the **uuid** of the background job and setting the following **Accept Header**:

ACCEPT	text/plain
---------------	------------

Apparently, the binary output result of the export job, can also be downloaded in binary format from the user by providing the uuid of the background job in the following HTTP GET request:

GET	/system/job/{job_uuid}/result
------------	-------------------------------

6.2.1. Export entity

To create the export background job **for an entity (raw data)**, two pieces of information are needed:

- its **entity-type**;
- and **uuid**.

GET	/raw/{entity_type}/{uuid}/export
------------	----------------------------------

So, for example, to export the substance with uuid 7a96f12c-f7b3-445d-a19f-e096f9562646 the following request is required:

```
curl --request POST \
  --url http://localhost:8080/iuclid6-ext/api/ext/v1/raw/SUBSTANCE/7a96f12c-f7b3-445d-a19f-e096f9562646/export \
  --header 'accept: application/vnd.iuclid6.ext+json; type=iuclid6.Iuclid6Job' \
  --header 'content-type: application/vnd.iuclid6.ext+json; type=iuclid6.FullExport' \
  --header 'iuclid6-user: SuperUser' \
  --header 'iuclid6-pass: *****' \
  --data '{}'
```

To get the **status** of the job created from the previous request with job_uuid=0b30ad7c-decc-40ac-9edb-4e845879cc99:

```
curl --request GET \
  --url http://localhost:8080/iuclid6-ext/api/ext/v1/system/job/0b30ad7c-decc-40ac-9edb-4e845879cc99/status \
  --header 'accept: text/plain' \
  --header 'iuclid6-user: SuperUser' \
  --header 'iuclid6-pass: *****'
```

In order to download the binary output of the export procedure (i6z file) the following GET request could be applied:

```
curl --request GET \
  --url http://localhost:8080/iuclid6-ext/api/ext/v1/system/job/0b30ad7c-decc-40ac-9edb-4e845879cc99/result \
  --header 'iuclid6-user: SuperUser' \
  --header 'iuclid6-pass: *****'
```

6.2.2. Export dossier

To create the export background job **for a dossier**, the dossier **uuid** is required as shown in the below request:

GET	/dossier/{dossier_uuid}/export
------------	--------------------------------

So, for example, to export the dossier with uuid 1dc1301c-64eb-4984-8446-e73182e08764 the following request is required:

```
curl --request POST \
```

```
--url http://localhost:8080/iuclid6-ext/api/ext/v1/dossier/1dc1301c-64eb-4984-8446-
e73182e08764/export \
--header 'content-type: application/vnd.iuclid6.ext+json; type=iuclid6.FullExport' \
--header 'accept: application/vnd.iuclid6.ext+json; type=iuclid6.Iuclid6Job' \
--header 'iuclid6-user: SuperUser' \
--header 'iuclid6-pass: *****'
--data '{}'
```

The result of the above request would be the following:

JobInfo

```
{
  "id": "e83f86f6-e6f8-4823-8725-1750dfa82e5e",
  "status": "QUEUED",
  "uri": "/job/e83f86f6-e6f8-4823-8725-1750dfa82e5e",
  "representation": null
}
```

To get the **status** of the dossier export job created from the previous request with job_uuid=e83f86f6-e6f8-4823-8725-1750dfa82e5e:

```
curl --request GET \
--url http://localhost:8080/iuclid6-ext/api/ext/v1/system/job/e83f86f6-e6f8-4823-8725-
1750dfa82e5e/status \
--header 'accept: text/plain' \
--header 'iuclid6-user: SuperUser' \
--header 'iuclid6-pass: *****'
```

and the text/plain result of the above request could be:

SUCCEDED

In order to download the binary output of the dossier export procedure (i6z file) the following GET request could be applied:

```
curl --request GET \
--url http://localhost:8080/iuclid6-ext/api/ext/v1/system/job/e83f86f6-e6f8-4823-8725-
1750dfa82e5e/result \
--header 'iuclid6-user: SuperUser' \
--header 'iuclid6-pass: *****'
```

6.2.3. Customization of the export operation

The export operation can be controlled in terms of the contents of the exported archive. Specifically, it is possible to choose whether stylesheet files are included in the archive, as well as whether empty elements (elements with no value) are included in the XML files contained in the archive.

A raw dataset can be exported through the following call:

POST	/raw/{ENTITY_TYPE}/{e-uuid}/export
-------------	------------------------------------

where "ENTITY-TYPE" is the entity type of the dataset (e.g., SUBSTANCE) and "e-uuid" the identifier of the dataset.

A dossier can be similarly exported as follows:

POST	/dossier/{dossier-uuid}/export
-------------	--------------------------------

The **Content-Type** header needs to be set as follows:

Content-Type	application/vnd.iuclid6.ext+json;type=iuclid6.FullExport
---------------------	--

The body of the request should be an object with the following fields:

Name	Type	Default Value	Description
excludeXslStylesheets	boolean	false	If true, stylesheets are not included in the exported archive
excludeEmptyElements	boolean	true	If true, empty document elements are not included in the exported archive
Body	{ "excludeXslStylesheets": true/false, "excludeEmptyElements": true/false }		

Note: an empty object ({ }) is acceptable, in which case the default values are applied

For example, in order to export a dossier with uuid "dossier-uuid" without including the stylesheets, we may perform the following call:

```
curl --request POST \
  --url http://localhost:8080/iuclid6-ext/api/ext/v1/dossier/dossier-uuid/export \
  --header 'content-type: application/vnd.iuclid6.ext+json;type=iuclid6.FullExport' \
  --header 'iuclid6-user: username' \
  --header 'iuclid6-pass: *****' \
  --data '{"excludeXslStylesheets": true, "excludeEmptyElements": true}'
```

6.3. Import

You can import **both** iuclid6 archives (*.i6z) and iuclid5 (*.i5z) using the public REST API.

To do so, you issue a POST request to **/system/archives** resource, the binary content of the archive to be imported should be included in the body of the request, and the **Content-Type** header can be one of the following:

IUCLID6	application/vnd.iuclid6.archive
IUCLID5	application/vnd.iuclid5.archive

If the archive contains documents that are already present in the system, then by default they will only be updated if they are more recently modified than the existing version.

To change the default overwrite mode set the **overwrite** query parameter with one of the following values:

Value	Description
never	Do not update existing documents.
always	Update existing documents even if they are more recently modified
replace	In case the archive is of an entity with section documents (e.g. SUBSTANCE), then replace the existing dataset with the archive. That is, if a child section exist in the system, but is not contained in the archive it will be removed.

An example of an import request using curl:

```
curl --request POST \
  --url 'http://localhost:8080/iuclid6-ext/api/ext/v1/system/archives?overwrite=newer' \
  --data-binary <path_to_file>
```

```
--header 'content-type: application/vnd.iuclid5.archive' \  
--header 'iuclid6-user: SuperUser'--header 'iuclid6-pass: *****'
```

By using the returned job uuid, e.g., eb685af2-b9bc-45a0-a2b1-b470a0283bd8, we may fetch more information about the status of the job.

ACCEPT	application/vnd.iuclid6.ext+json;type=iuclid6.Iuclid6Job
GET	/system/job/{job_uuid}

```
curl --request GET  
--url 'http://localhost:8080/iuclid6-ext/api/ext/v1/system/job/eb685af2-b9bc-45a0-a2b1-  
b470a0283bd8'  
--header 'accept: application/vnd.iuclid6.ext+json;type=iuclid6.Iuclid6Job'  
--header 'IUCLID6-USER: SuperUser'  
--header 'IUCLID6-PASS: *****'
```

JobInfo

```
{  
  "id": "eb685af2-b9bc-45a0-a2b1-b470a0283bd8",  
  "status": "SUCCEEDED",  
  "uri": "/system/jobs/eb685af2-b9bc-45a0-a2b1-b470a0283bd8",  
  "representation": {  
    "classtype": "Iuclid6Job",  
    "group": null,  
    "name": "344446c5-4c9e-4bed-ab0c-9bf6a257a6cd_unnamed.i5z",  
    "type": "IMPORT",  
    "documentUri": "iuclid6:/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055",  
    "logs": [  
      {  
        "code": null,  
        "level": "INFO",  
        "value": "Started import of file: 344446c5-4c9e-4bed-ab0c-9bf6a257a6cd_unnamed.i5z",  
        "documentUri": null,  
        "params": [  
          "",  
          "",  
          "",  
          "",  
          ""  
        ]  
      },  
      {  
        "code": null,  
        "level": "INFO",
```

```

"value": "Overwrite mode: IF_NEWER_THAN_EXISTING",
"documentUri": null,
"params": [
  "",
  "",
  "",
  "",
  ""
]
},
{
"code": "MR0032linked",
"level": "WARN",
"value": "MR0032;Literature reference;IUC5-693459ee-92b5-43a0-9741-
bfacdf4cf136;IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055;Melting point/freezing
point.001;Missing mandatory fields;Literature reference was created with default values",
"documentUri": null,
"params": [
  "Literature reference",
  "IUC5-693459ee-92b5-43a0-9741-bfacdf4cf136",
  "IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055",
  "Melting point/freezing point.001",
  "Literature reference"
]
},
{
"code": null,
"level": "INFO",
"value": "Imported: CONTACT / Unique, Xenophon / cc2694c2-4341-3e5c-b4d2-
7a8da62fbeat/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055",
"documentUri": "iuclid6:/IUC5-e44ee572-3f11-4ad9-bdeb-
52160e442055/CONTACT/cc2694c2-4341-3e5c-b4d2-7a8da62fbeat",
"params": [
  "",
  "",
  "",
  "",
  ""
]
},
{
"code": null,
"level": "INFO",
"value": "Imported: LITERATURE / No information / f483d538-fa36-36cf-a4aa-
a4e396f5ef32/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055 linked to endpoint(s) with uuid:
IUC5-693459ee-92b5-43a0-9741-bfacdf4cf136",
"documentUri": "iuclid6:/IUC5-e44ee572-3f11-4ad9-bdeb-
52160e442055/LITERATURE/f483d538-fa36-36cf-a4aa-a4e396f5ef32",
"params": [
  "",
  ""
]
}

```



```
    ""  
    ,  
    ""  
    ,  
    ""  
  ]  
},  
{  
  "code": null,  
  "level": "INFO",  
  "value": "Imported: LEGAL_ENTITY / NewCompany / ECHA-59923ee8-3913-46b3-89c6-5cc696f567a6/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055",  
  "documentUri": "iuclid6:/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055/LEGAL_ENTITY/ECHA-59923ee8-3913-46b3-89c6-5cc696f567a6",  
  "params": [  
    ""  
    ,  
    ""  
    ,  
    ""  
    ,  
    ""  
    ,  
    ""  
  ]  
},  
{  
  "code": null,  
  "level": "INFO",  
  "value": "Imported: REFERENCE_SUBSTANCE / framycetin / ECB5-2fd4062c-cae1-4ddb-9f15-deecf3212f04/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055",  
  "documentUri": "iuclid6:/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055/REFERENCE_SUBSTANCE/ECB5-2fd4062c-cae1-4ddb-9f15-deecf3212f04",  
  "params": [  
    ""  
    ,  
    ""  
    ,  
    ""  
    ,  
    ""  
    ,  
    ""  
  ]  
},  
{  
  "code": null,  
  "level": "INFO",  
  "value": "Imported: REFERENCE_SUBSTANCE / schradan / ECB5-ddb83c51-3144-440b-92e9-c7afcbf76427/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055",  
  "documentUri": "iuclid6:/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055/REFERENCE_SUBSTANCE/ECB5-ddb83c51-3144-440b-92e9-c7afcbf76427",  
  "params": [  
    ""  
    ,  
    ""  
    ,  
    ""  
    ,  
    ""  
    ,  
    ""  
  ]  
},  
{
```

```

        "code": null,
        "level": "INFO",
        "value": "Imported: SITE / X site / IUC5-45e8bf7a-d928-4784-8d03-e590ef273eff/IUC5-
e44ee572-3f11-4ad9-bdeb-52160e442055",
        "documentUri": "iuclid6:/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055/SITE/IUC5-
45e8bf7a-d928-4784-8d03-e590ef273eff",
        "params": [
            "",
            "",
            "",
            "",
            ""
        ]
    },
    {
        "code": null,
        "level": "INFO",
        "value": "Imported: SUBSTANCE / RITMII-926_TC235_Reg1-
10_ReqUpdate_nonTCC_OCCOK_EoP_Automatic / IUC5-cea09f98-9007-410d-9c2f-
fe51f113297a/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055",
        "documentUri": "iuclid6:/IUC5-e44ee572-3f11-4ad9-bdeb-
52160e442055/SUBSTANCE/IUC5-cea09f98-9007-410d-9c2f-
fe51f113297a/SUBSTANCE/IUC5-cea09f98-9007-410d-9c2f-fe51f113297a",
        "params": [
            "",
            "",
            "",
            "",
            ""
        ]
    },
    {
        "code": null,
        "level": "INFO",
        "value": "Imported: FLEXIBLE_RECORD / RITMII-926_TC235_Reg1-
10_ReqUpdate_nonTCC_OCCOK_EoP_Automatic / d68a133c-54c3-3103-be2d-
72afbed85903/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055",
        "documentUri": "iuclid6:/IUC5-e44ee572-3f11-4ad9-bdeb-
52160e442055/SUBSTANCE/IUC5-cea09f98-9007-410d-9c2f-
fe51f113297a/FLEXIBLE_RECORD.DsdDpd/d68a133c-54c3-3103-be2d-72afbed85903",
        "params": [
            "",
            "",
            "",
            "",
            ""
        ]
    },
    {
        "code": null,
        "level": "INFO",

```

```
    "value": "Imported: FLEXIBLE_RECORD / RITMII-926_TC235_Reg1-10_ReqUpdate_nonTCC_OCCOK_EoP_Automatic / 34a747cd-7549-3c9a-b2ef-7926038352d5/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055",
    "documentUri": "iuclid6:/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055/SUBSTANCE/IUC5-cea09f98-9007-410d-9c2f-fe51f113297a/FLEXIBLE_RECORD.Ghs/34a747cd-7549-3c9a-b2ef-7926038352d5",
    "params": [
      ""
    ]
  },
  {
    "code": null,
    "level": "INFO",
    "value": "Imported: FLEXIBLE_RECORD / Sites.001 / ce5b8eba-295a-3307-a11c-1ed89cf0ce44/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055",
    "documentUri": "iuclid6:/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055/SUBSTANCE/IUC5-cea09f98-9007-410d-9c2f-fe51f113297a/FLEXIBLE_RECORD.Sites/ce5b8eba-295a-3307-a11c-1ed89cf0ce44",
    "params": [
      ""
    ]
  },
  {
    "code": null,
    "level": "INFO",
    "value": "Imported: FLEXIBLE_RECORD / RITMII-926_TC235_Reg1-10_ReqUpdate_nonTCC_OCCOK_EoP_Automatic / 038de362-29b4-347a-9d1c-e4b2eb6e3441/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055",
    "documentUri": "iuclid6:/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055/SUBSTANCE/IUC5-cea09f98-9007-410d-9c2f-fe51f113297a/FLEXIBLE_RECORD.SubstanceComposition/038de362-29b4-347a-9d1c-e4b2eb6e3441",
    "params": [
      ""
    ]
  },
  {
    "code": null,
    "level": "INFO",
```

```
"value": "Imported: FLEXIBLE_SUMMARY / PBT assessment / IUC5-41089113-deae-42c1-bbb6-de6f0fdbefbc/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055",
  "documentUri": "iuclid6:/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055/SUBSTANCE/IUC5-cea09f98-9007-410d-9c2f-fe51f113297a/FLEXIBLE_SUMMARY.PbtAssessment/IUC5-41089113-deae-42c1-bbb6-de6f0fdbefbc",
  "params": [
    "",
    "",
    "",
    "",
    ""
  ]
},
{
  "code": null,
  "level": "INFO",
  "value": "Imported: ENDPOINT_STUDY_RECORD / Acute toxicity: dermal.001 / IUC5-449d6591-6a32-4f52-a004-f86a955dd1ca/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055",
  "documentUri": "iuclid6:/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055/SUBSTANCE/IUC5-cea09f98-9007-410d-9c2f-fe51f113297a/ENDPOINT_STUDY_RECORD.AcuteToxicityDermal/IUC5-449d6591-6a32-4f52-a004-f86a955dd1ca",
  "params": [
    "",
    "",
    "",
    "",
    ""
  ]
},
{
  "code": null,
  "level": "INFO",
  "value": "Imported: ENDPOINT_STUDY_RECORD / Acute toxicity: inhalation.001 / IUC5-0e2146bf-5c3e-4b64-9bf5-6600ded1c98f/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055",
  "documentUri": "iuclid6:/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055/SUBSTANCE/IUC5-cea09f98-9007-410d-9c2f-fe51f113297a/ENDPOINT_STUDY_RECORD.AcuteToxicityInhalation/IUC5-0e2146bf-5c3e-4b64-9bf5-6600ded1c98f",
  "params": [
    "",
    "",
    "",
    "",
    ""
  ]
},
{
  "code": null,
  "level": "INFO",
```

```
"value": "Imported: ENDPOINT_STUDY_RECORD / Acute toxicity: oral.001 / IUC5-1b15a65c-1566-49f7-b169-0bf1dc033b69/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055",
  "documentUri": "iuclid6:/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055/SUBSTANCE/IUC5-cea09f98-9007-410d-9c2f-fe51f113297a/ENDPOINT_STUDY_RECORD.AcuteToxicityOral/IUC5-1b15a65c-1566-49f7-b169-0bf1dc033b69",
  "params": [
    "",
    "",
    "",
    "",
    ""
  ]
},
{
  "code": null,
  "level": "INFO",
  "value": "Imported: ENDPOINT_STUDY_RECORD / Acute toxicity: other routes.001 / IUC5-55b2ba5f-1b7f-44c8-a000-c73c6eb552b4/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055",
  "documentUri": "iuclid6:/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055/SUBSTANCE/IUC5-cea09f98-9007-410d-9c2f-fe51f113297a/ENDPOINT_STUDY_RECORD.AcuteToxicityOtherRoutes/IUC5-55b2ba5f-1b7f-44c8-a000-c73c6eb552b4",
  "params": [
    "",
    "",
    "",
    "",
    ""
  ]
},
{
  "code": null,
  "level": "INFO",
  "value": "Imported: ENDPOINT_STUDY_RECORD / Adsorption / desorption.001 / IUC5-f41e8d70-3bfc-4da9-bc72-6015d547298c/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055",
  "documentUri": "iuclid6:/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055/SUBSTANCE/IUC5-cea09f98-9007-410d-9c2f-fe51f113297a/ENDPOINT_STUDY_RECORD.AdsorptionDesorption/IUC5-f41e8d70-3bfc-4da9-bc72-6015d547298c",
  "params": [
    "",
    "",
    "",
    "",
    ""
  ]
},
{
  "code": null,
```

```
    "level": "INFO",
    "value": "Imported: ENDPOINT_STUDY_RECORD / Melting point/freezing point.001 / IUC5-693459ee-92b5-43a0-9741-bfacdf4cf136/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055",
    "documentUri": "iuclid6:/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055/SUBSTANCE/IUC5-cea09f98-9007-410d-9c2f-fe51f113297a/ENDPOINT_STUDY_RECORD.Melting/IUC5-693459ee-92b5-43a0-9741-bfacdf4cf136",
    "params": [
      "",
      "",
      "",
      "",
      ""
    ]
  },
  {
    "code": null,
    "level": "INFO",
    "value": "Imported: ENDPOINT_STUDY_RECORD / Phototransformation in air.001 / IUC5-767c00cf-5c14-45a3-8518-042d743f1ecf/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055",
    "documentUri": "iuclid6:/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055/SUBSTANCE/IUC5-cea09f98-9007-410d-9c2f-fe51f113297a/ENDPOINT_STUDY_RECORD.PhototransformationInAir/IUC5-767c00cf-5c14-45a3-8518-042d743f1ecf",
    "params": [
      "",
      "",
      "",
      "",
      ""
    ]
  },
  {
    "code": null,
    "level": "INFO",
    "value": "Imported: DOSSIER / RITMII-926_TC235_Reg1-10_Initial / IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055",
    "documentUri": "iuclid6:/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055",
    "params": [
      "",
      "",
      "",
      "",
      ""
    ]
  },
  {
    "code": null,
    "level": "INFO",
```

```
"value": "Imported: ATTACHMENT / 152-16-9-V2.jpeg / ECB5-4653ee4e-3790-49e9-8de1-30de33cec1b2/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055",
"documentUri": null,
"params": [
  "",
  "",
  "",
  "",
  ""
]
},
{
  "code": null,
  "level": "INFO",
  "value": "Imported: ATTACHMENT / 119-04-0-V2.jpeg / ECB5-bc147df9-10b6-4d5d-9f24-10ab2e751784/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055",
  "documentUri": null,
  "params": [
    "",
    "",
    "",
    "",
    ""
  ]
},
{
  "code": null,
  "level": "INFO",
  "value": "Import of 20 document(s) completed",
  "documentUri": null,
  "params": [
    "",
    "",
    "",
    "",
    ""
  ]
},
{
  "code": null,
  "level": "INFO",
  "value": "20 document(s) were imported / 0 document(s) were skipped / 0 document(s) failed",
  "documentUri": null,
  "params": [
    "",
    "",
    "",
    "",
    ""
  ]
}
```

```
    },
    {
      "code": null,
      "level": "INFO",
      "value": "Import of 2 attachment(s) completed",
      "documentUri": null,
      "params": [
        "",
        "",
        "",
        "",
        ""
      ]
    },
    {
      "code": null,
      "level": "INFO",
      "value": "2 attachment(s) were imported / 0 attachment(s) were skipped / 0 attachment(s)
failed\n",
      "documentUri": null,
      "params": [
        "",
        "",
        "",
        "",
        ""
      ]
    },
    {
      "code": null,
      "level": "INFO",
      "value": "Security pool: Common (R/W/D)",
      "documentUri": null,
      "params": [
        "",
        "",
        "",
        "",
        ""
      ]
    },
    {
      "code": null,
      "level": "INFO",
      "value": "Owner: SuperUser",
      "documentUri": null,
      "params": [
        "",
        "",
        "",
        ""
      ]
    }
  ]
}
```



```

    ""
  ]
},
{
  "code": null,
  "level": "INFO",
  "value": "Import process succeeded",
  "documentUri": null,
  "params": []
}
]
}
}
}

```

To fetch information on the imported entity only, use the 'application/vnd.iuclid6.ext+json;type=standard.URI' accept header

```

curl --request GET
      --url 'http://localhost:8080/iuclid6-ext/api/ext/v1/system/job/eb685af2-b9bc-45a0-a2b1-
b470a0283bd8'
      --header 'accept: application/vnd.iuclid6.ext+json;type=standard.URI'
      --header 'IUCLID6-USER: SuperUser'
      --header 'IUCLID6-PASS: root'

```

JobInfo

```

{
  "id": "eb685af2-b9bc-45a0-a2b1-b470a0283bd8",
  "status": "SUCCEEDED",
  "uri": "/system/jobs/eb685af2-b9bc-45a0-a2b1-b470a0283bd8",
  "representation": "iuclid6:/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055"
}

```

6.4. Validation assistant

The validation assistant checks if a IUCLID6 dossier contains errors, that would reject its submission, and it produces a report containing information about these errors, if any.

The system supports the execution of the validation assistant, both on created dossiers, and on raw datasets.

However, since the validation assistant operates on dossiers, to validate raw datasets, you need to specify the additional data that would create a dossier from this dataset, ie the submission type and the dossier header.

6.4.1. Raw entities

To execute the validation assistant on a raw dataset you can **POST** a **DossierCreationDTO** object to the following resource:

POST	/raw/{entity_type}/{entity_uuid}/dossiers/{submission_type}/validation
-------------	--

For example, to validate a **REACH Inquiry** dossier of a specific substance:

```
curl -X POST -H "IUCLID6-USER: SuperUser" -H "IUCLID6-PASS: *****" -H "Content-Type: application/vnd.iuclid6.ext+json; type=iuclid6.DossierCreationDTO" -H "Accept: application/vnd.iuclid6.ext+json; type=iuclid6.ValidationReport" -d '{
  "excludeLegalEntity": true,
  "excludeNotRequiredDocuments": true,
  "excludeAnnotations": false,
  "excludeConfidentialFields": false,
  "excludeDetailFields": false,
  "filterCategoryMemberChildren": false,
  "excludeConfidentialityRestrictions": [],
  "excludeRegulationRestrictions": [],
  "excludeDocuments": [],
  "header": [
    {
      "definition": "DOSSIER.R_INQUIRY",
      "name": "demo dossier"
    },
    {
      "DossierSpecificInformation": {
        "Remarks": "demo remarks"
      }
    }
  ]
}' "http://localhost:8080/iuclid6-ext/api/ext/v1/raw/SUBSTANCE/5307a86a-afd4-4da9-9c01-4ac12b1bc9c0/dossiers/R_INQUIRY/validation"
```

The above request will respond with the validation report:

ValidationReport

```
{
  "subjectUuid": "9f7c364c-8e29-4d3c-8c4a-56baea903414",
  "snapshotUuid": "3bf15315-f76f-4d61-84d0-0ac0335ad338",
  "rulesetId": "SC0005",
  "rulesetName": "SC0005 - Inquiry",
  "ruleResults": [
```

```

{
  "ruleId": "BR018",
  "ruleType": "BR",
  "ruleStatus": "FAIL",
  "checks": [
    {
      "ruleCheckId": "BR018",
      "level": "FAILURE",
      "element": {
        "uri": "iuclid6:/3bf15315-f76f-4d61-84d0-0ac0335ad338/SUBSTANCE/9f7c364c-8e29-4d3c-8c4a-56baea903414;section=R_INQUIRY:1.1/SUBSTANCE/9f7c364c-8e29-4d3c-8c4a-56baea903414#ReferenceSubstance.ReferenceSubstance",
        "description": "",
        "documentName": "test1",
        "sectionName": "Identification"
      },
      "message": "A reference substance must be linked in IUCLID section 1.1."
    }
  ]
},
...
]
}

```

The **ValidationReport** contains the following properties

Name	Type	Description
subjectUuid	string	The uuid of the dossier subject
snapshotUuid	string	The uuid of the dossier
rulesetId	string	The identification of the ruleset
rulesetName	string	The title of the ruleset
ruleResults	array	An array of RuleResult objects

The **RuleResult** contains the following properties

Name	Type	Description
ruleId	string	The identification of the rule
ruleType	string	The type of the rule. Can be one of: TCC BR QLT SID

Name	Type	Description
ruleStatus	string	The status of the rule. Can be on of: PASS FAIL EXCEPTION
checks	array	An array of RuleCheckResult objects

The **RuleCheckResult** contains the following properties

Name	Type	Description
ruleCheckId	string	The uuid of the dossier subject
level	string	The severity of the check. Can be one of: REMINDER WARNING FAILURE
message	string	A localized description of the check.
element	object	The RuleCheckElement the check applies to.
relatedElements	array	An array of related RuleCheckElement objects, that the check might apply to
conditions	array	An array of ReportCondition objects. <i>A check might be relevant only if some additional conditions are met.</i>

The **RuleCheckElement** contains the following properties

Name	Type	Description
uri	string	The I6Uri of the element
description	string	An optional description for the element
documentName	string	The name of the document the element is contained in.
sectionName	string	The name of the section in the tree of the document containing the element.

The **ReportCondition** contains the following properties

Name	Type	Description
conditionKey	string	The key of the condition
conditionValue	string	The value of the condition
localizedCondition	string	A localized title of the condition

6.4.2. Dossiers

To execute the validation assistant on created dossier, you can make a **POST** request without any payload, to the following resource:

POST	dossier/{dossier_uuid}/validation
-------------	-----------------------------------

For example, the following request executes the validation assistant for a dossier with uuid: 3bf15315-f76f-4d61-84d0-0ac0335ad338

```
curl -X POST -H "IUCLID6-USER: SuperUser" -H "IUCLID6-PASS: *****" -H "Accept: application/vnd.iuclid6.ext+json; type=iuclid6.ValidationReport" "http://localhost:8080/iuclid6-ext/api/ext/v1/dossier/3bf15315-f76f-4d61-84d0-0ac0335ad338/validation"
```

6.5. Dossier comparison (since v3.0.0)

You can **compare** two dossiers and receive a comparison report in HTML format by performing the following call:

POST	/dossier/{uuid1}/compare
-------------	--------------------------

where uuid1 corresponds to the identifier of the first dossier.

The uuid of the second dossier is passed through the body of the HTTP request:

Body	{uuid2}
-------------	---------

The **Content-Type** header needs to be set as follows:

Content-Type	application/json
---------------------	------------------

The **Accept** header needs to be set as follows:

Accept	text/html
---------------	-----------

For example, in order to compare a dossier with uuid "uuid-1" to a dossier with uuid "uuid-2" we may perform the following call:

```
curl --request POST \
  --url http://localhost:8080/iuclid6-ext/api/ext/v1/dossier/uuid-1/compare \
  --header 'content-type: application/json' \
  --header 'accept: text/html' \
  --header 'iuclid6-user: username' \
  --header 'iuclid6-pass: *****' \
  --data 'uuid-2'
```

6.6. Report generation (since v3.0.0)

6.6.1. Document Printing

In order to **print** a dataset or dossier to PDF format, one can perform the following call:

POST	/system/script
-------------	----------------

To specify the resource to be printed, a valid I6Uri needs to be provided in the HTTP request body

Body	"{i6Uri}"
-------------	-----------

The **Content-Type** header needs to be set as follows:

Content-Type	application/vnd.iuclid6.ext+json;type=iuclid6.print
---------------------	---

The **Accept** header needs to be set as follows:

Accept	application/vnd.iuclid6.ext+pdf;type=iuclid6.pdf
---------------	--

The following optional query parameters can also be used to control the printing operation:

Query parameter name	Default value	Description
----------------------	---------------	-------------

Query parameter name	Default value	Description
skipEmpty	true	Whether to ignore sections and fields that have no data
skipConfidential	true	Whether to skip fields that are declared as confidential (as per their definition, not any related data protection)
showCover	true	Whether to include a cover page
showAnnotations	false	Whether to include annotations of the documents that are printed

For example, in order to print a dossier with uuid "dossier-uuid" we may perform the following call:

```
curl --request POST \
  --url http://localhost:8080/iuclid6-
  ext/api/ext/v1/system/script \
  --header 'content-type:
  application/vnd.iuclid6.ext+json;type=iuclid6.print' \
  --header 'accept:
  application/vnd.iuclid6.ext+pdf;type=iuclid6.pdf' \
  --header 'iuclid6-user: username' \
  --header 'iuclid6-pass: *****' \
  --data "iuclid6:/dossier-uuid"
```

Valid resources for printing are entities, dossiers or single documents within a composite entity. Some example I6URIs can be found below:

Dossier: "iuclid6:/<dossier-uuid>"
 Substance: "iuclid6:/0/SUBSTANCE/<substance-uuid>"
 Substance Composition document: "iuclid6:/0/SUBSTANCE/<substance-
 uuid>/FLEXIBLE_RECORD.SubstanceComposition/<composition-doc-uuid>"

6.6.2. CSR Report

In order to generate the **CSR report** for a dataset or dossier to RTF format, one can perform the following call:

POST	/system/script
-------------	----------------

To specify the resource to be printed, a valid I6Uri needs to be provided in the HTTP request body

Body	"{i6Uri}"
-------------	-----------

The **Content-Type** header needs to be set as follows:

Content-Type	application/vnd.iuclid6.ext+json;type=web.csr
---------------------	---

The **Accept** header needs to be set as follows:

Accept	application/vnd.iuclid6.ext+json;type=web.rtf
---------------	---

For example, in order to produce the CSR a dossier with uuid "dossier-uuid" we may perform the following call:

```
curl --request POST \
  --url http://localhost:8080/iuclid6-ext/api/ext/v1/system/script \
  --header 'content-type: application/vnd.iuclid6.ext+json;type=web.csr' \
  --header 'accept: application/vnd.iuclid6.ext+json;type=web.rtf' \
  --header 'iuclid6-user: username' \
  --header 'iuclid6-pass: *****' \
  --data "iuclid6:/dossier-uuid"
```


6.7. Dossier filtering (since v3.0.0)

One may **filter a dossier** by performing the following call:

POST	/dossier/{dossier-uuid}/filter
-------------	--------------------------------

The **Content-Type** header needs to be set as follows:

Content-Type	application/vnd.iuclid6.ext+json;type=web.FilterRequest
---------------------	---

The **Accept** header needs to be set as follows:

Accept	application/vnd.iuclid6.ext+json;type=web.FilterReport
---------------	--

The body of the HTTP request should be an empty object:

Body	{ }
-------------	-----

The filtering ruleset configuration used is the built-in configuration with id 'reachV2.all-dossier-except-core'.

The result of this call is a filtering report comprised of a list of rows. Each row represents an event emitted by the filtering engine and has the following structure:

Name	Type	Description
------	------	-------------

Name	Type	Description
entity	string	<p>The entity related to the filtering event.</p> <p><i>Possible values:</i></p> <p><i>"Document",</i> <i>"Dossier",</i> <i>"Substance",</i> <i>"Mixture/Product",</i> <i>"Category",</i> <i>"Template",</i> <i>"Reference substance",</i> <i>"Legal entity",</i> <i>"Legal entity site",</i> <i>"Contact",</i> <i>"Literature Reference",</i> <i>"Endpoint Study Record",</i> <i>"Endpoint Summary",</i> <i>"Flexible Summary",</i> <i>"Flexible Record",</i> <i>"Fixed Record",</i> <i>"Assessment entity",</i> <i>"Annotation",</i> <i>"Test materials"</i></p>
sectionName	string	The section name where the relevant entity belongs to (e.g., "1.2 Composition")
documentName	string	The name of the affected document (e.g., "Boundary composition.001")
field	string []	The localized path of the affected field as an array of localized path segments (e.g., ["Composition", "General Information", "Name"])

Name	Type	Description
outcome	string	<p>The filtering outcome.</p> <p><i>Possible values:</i></p> <p><i>"Published",</i> <i>"Not published",</i> <i>"Flagged confidential – Not published",</i> <i>"Not published (ineligible for dissemination)",</i> <i>"Filtered dossier contained confidentiality claim on tonnage band – Not published",</i> <i>"Filtered dossier contained confidentiality claim on the registration number – Not published",</i> <i>"Removed open picklist 'Other' field",</i> <i>"Removed picklist remark",</i> <i>"Filtered section 0.4 Assessment entities",</i> <i>"Removed constituent",</i> <i>"Filtered category member dataset",</i> <i>"Filtered category member dataset - Substance identity claimed confidential"</i></p>
sourceDocumentKey	string	<p>The document key of the affected document (e.g., "164eb2f1-6b40-493b-95a6-851cc072d47b/59749183-d14c-40f1-975b-39d1024dcc4f")</p>
path	string	<p>The path of the affected entity (e.g., FLEXIBLE_RECORD/SubstanceComposition.GeneralInformation.Name)</p>

The JSON structure of the response looks like below:

```

{
"rows": [
...
{
"entity": "Flexible Record",
"sectionName": "1.2 Composition",
"documentName": "Boundary composition.001",
"field": [
"Composition",
"General Information",
"Justification for deviations"
],
"outcome": "Not published",
"sourceDocumentKey": "164eb2f1-6b40-493b-95a6-851cc072d47b/59749183-d14c-40f1-975b-39d1024dcc4f",
"path":
"FLEXIBLE_RECORD/SubstanceComposition.GeneralInformation.JustificationForDeviations"
},
{
"entity": "Flexible Record",
"sectionName": "1.2 Composition",
"documentName": "Boundary composition.001",
"field": [
"Composition",
"General Information",
"Related composition(s)",
"Related composition"
],
"outcome": "Published",
"sourceDocumentKey": "164eb2f1-6b40-493b-95a6-851cc072d47b/59749183-d14c-40f1-975b-39d1024dcc4f",
"path":
"FLEXIBLE_RECORD/SubstanceComposition.GeneralInformation.RelatedCompositions.RelatedComposition"
},
...
]
}

```

For example, in order to filter the dossier with uuid "dossier-uuid" we may perform the following call:

```
curl --request POST \  
--url http://localhost:8080/iuclid6-ext/api/ext/v1/dossier/dossier-uuid/filter \  
--header 'content-type: application/vnd.iuclid6.ext+json;type=web.FilterRequest' \  
--header 'accept: application/vnd.iuclid6.ext+json;type=web.FilterReport' \  
--header 'iuclid6-user: username' \  
--header 'iuclid6-pass: *****' \  
--data '{}'
```

7. Inventories

Every REST resource in this document is under `/iuclid6-ext/api/ext/v1/`

The IUCLID6 Rest API supports paged inventory entry listing and searching. This is an unsecured resource. In other words **no authentication** is needed to access this resource

GET	<code>/inventory</code>
------------	-------------------------

The above resource supports the following query parameters:

Param	Type	Description	Default
count	boolean	include the total count of the results. <i>If the total count is not needed, omitting it improves performance.</i>	true
l	integer	the number of results (<i>limit</i>) per page.	10
o	integer	the offset of the first result in page. e.g. to fetch the 3rd page of 25 ($l=25$) results specify $o=50$ ($o = l * \text{zero_indexed_page_number}$)	0
code	string	matches the code of the inventory. E.g. 'EC'	
number	string	matches the number of the inventory entry. E.g. '500-001*'	
name	string	matches the name of the inventory entry. E.g. '*Methyl*'	
cas	string	matches the cas number of the inventory entry. E.g. '50-*	
formula	string	matches the molecular formula of the inventory entry. E.g. 'C11H15Cl2NO.ClH'	
q	string	matches the specified value in either of the following inventory entry fields: cas number molecular formula number name	

7.1. Results

The inventory entry searching results are wrapped inside a standard payload that contains:

- the results of the search
- plus some additional information to support paging, such as

- the limit
- the offset
- and the total count of the results.

Each result comprises of two values, the URI and the representation:

```
{
  "limit": 15,
  "totalCount": 407,
  "results": [
    {
      "uri": "iuclid6:/EC/200-057-3",
      "representation": {
        "inventoryCode": "EC",
        "inventoryTitle": "EC Inventory",
        "inventoryVersion": "1.3",
        "number": "200-057-3",
        "status": "ACTIVE",
        "casNumber": "50-66-8",
        "name": "6-(methylthio)purine",
        "molecularFormula": "C6H6N4S",
        "createdOn": "1990-06-15T00:00:00Z",
        "modifiedOn": "2015-02-19T22:00:00Z"
      }
    },
    ...
  ]
}
```