

1. IUCLID6 Public REST API	2
1.1 Overview	2
1.2 Definitions	6
1.3 Search	18
1.4 Document handling	37
1.5 Sharing (data access across security groups)	54
1.6 Import / Export	57

IUCLID6 Public REST API

- [Overview](#)
- [Definitions](#)
- [Search](#)
- [Document handling](#)
- [Sharing \(data access across security groups\)](#)
- [Import / Export](#)
- [Appendix I: IUCLID5 WS API](#)

Overview

- [1 Media Types](#)
 - [1.1 IUCLID Document media types](#)
- [2 HTTP Methods and Status Codes](#)
 - [2.1 Semantics of HTTP methods](#)
 - [2.2 Semantics of HTTP status codes](#)
 - [2.2.1 Success Codes](#)
 - [2.2.2 Error codes](#)
 - [2.2.3 Polling](#)
- [3 Authorization](#)
- [4 Base URL](#)
- [5 About](#)

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.

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.

IUCLID Document media types

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

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

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.

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.

Semantics of HTTP status codes

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 <i>import</i> 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>

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.

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.

1. 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.
2. The client should call the monitoring URI with a GET to retrieve progress information.
3. When servicing the monitoring URI,
 - a. 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.
 - b. 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)
 - c. 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.

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.

Base URL

Every REST resource of the public API is under:

Base url	<code>/iuclid6-ext/api/ext/v1/</code>
-----------------	---------------------------------------

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	<code>/iuclid6-ext/api/ext/v1/</code>
------------	---------------------------------------

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.

Definitions

- [Introduction](#)
- [Legislation providers](#)
- [Submission types](#)
- [Trees](#)
- [Documents](#)
- [Phrase-groups](#)
- [Phrases](#)
- [Text templates](#)

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/`

Legislation providers

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

GET `/definition/providers`

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.

Submission types

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
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

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"
  },
  {
    "provider": "bpr",
    "identifier": "BIOC_ACTIVE_SUBSTANCE_FOR_MIXTURES",
    "title": "BPR Active substance application (representative product)",
    "applicableFor": "MIXTURE"
  },
  {
    "provider": "bpr",
    "identifier": "BIOC_BIOCIDAL_PRODUCT",
    "title": "BPR Biocidal product authorisation",
    "applicableFor": "MIXTURE"
  },
  {
    "provider": "oecd",
    "identifier": "OECD_HT",
    "title": "OECD harmonised templates",
    "applicableFor": "MIXTURE"
  },
  {
    "provider": "bpr",
    "identifier": "BIOC_BASIC_INFORMATION_MIXTURE",
    "title": "BPR Basic information (mixture)",
    "applicableFor": "MIXTURE"
  },
  {
    "provider": "core",
    "identifier": "COMPLETE",
    "title": "Complete table of contents",
    "applicableFor": "SUBSTANCE"
  },
  {
    "provider": "reach",
    "identifier": "R_A15_REST",
    "title": "REACH Annex XV - Restriction",
    "applicableFor": "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 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.
---------------	--------	---

Trees

You can retrieve the section tree of a submission type, applicable to a specific entity type at the following resource:

```
GET /definition/tree/{submission_id}?for={entity_type}
```

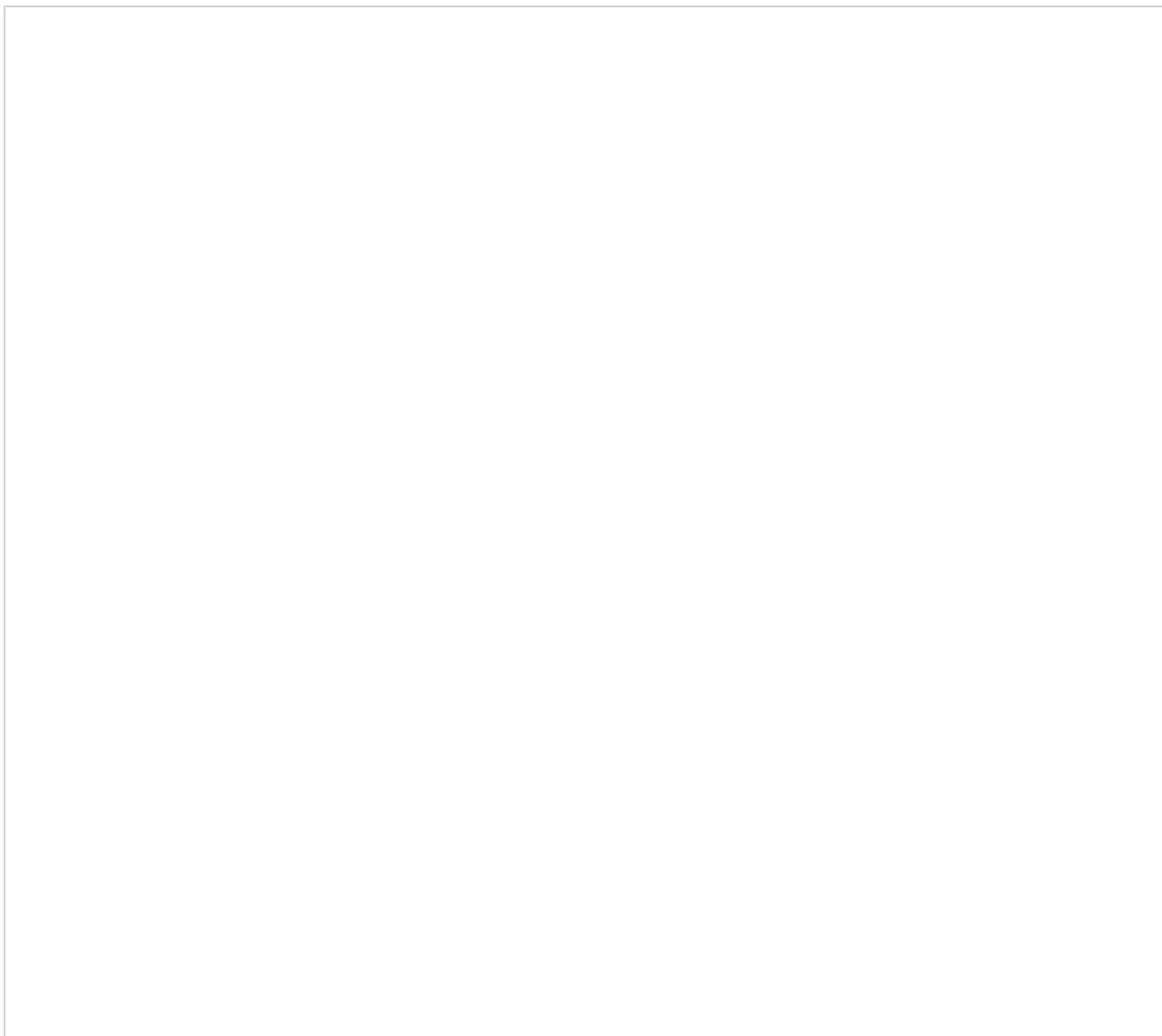
The response type is not `application/json` as it should be, but `application/vnd.iuclid6.ext+json`

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
```

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
            }
          ]
        }
      ],
    },
    {
      "code": "1.2",
      "title": "Composition",
      "sections": [],
      "documents": [
        {
          "code": "FLEXIBLE_RECORD.SubstanceComposition",
          "title": "Composition",
          "access": "NO_ACCESS",
          "required": true,
          "single": false
        }
      ]
    },
    {
      "code": "1.3",
      "title": "Identifiers",
      "sections": [],
      "documents": [
        {
          "code": "FIXED_RECORD.Identifiers",
          "title": "Identifiers",
          "access": "NO_ACCESS",
          "required": false,
          "single": true
        }
      ]
    }
  ]
}
...

```

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
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.
access	string	If the request is authenticated, it contains the privileges of the authenticated user on the specific document type. Can be one of: <ul style="list-style-type: none">• NO_ACCESS• READ_ONLY• READ_WRITE• FULL_ACCESS When the request is unauthenticated it will always contain 'NO_ACCESS'
required	boolean	True if the specific document type is required for the submission.
single	boolean	True if only one document of the specific type is allowed. e.g. The "Identification" of the substance

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
type	string	The specific type of the element: Can be one of: <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	The localized name of the element, 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.
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
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
	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	<p>The allowed boundary qualifiers. Depending on 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. • < • <= • > • >=
	lowerBoundQualifiers		
upperBoundQualifiers			
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`

Phrase-groups

Phrase-groups organize several `phrases` together in a set identified by a code. Several iucld 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")
------	---------	---

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
  },
  ...
]
```

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}

Search

- 1 Results
 - 1.1 Document Secured Representation
 - 1.2 Ordering
- 2 Available queries

- 2.1 byDate
- 2.2 byName
- 2.3 byContact
- 2.4 bySubstance
- 2.5 byMixture
- 2.6 byLegalEntity
- 2.7 byReferenceSubstance
- 2.8 byTemplate
- 2.9 byLiterature
- 2.10 byAnnotation
- 2.11 bySite
- 2.12 byAdditive
- 2.13 byImpurity
- 2.14 byConstituent
- 2.15 byGhs
- 2.16 byDsd
- 2.17 byJointSubmission

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

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
```

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-aald-387b1f8fefb5/SUBSTANCE/db63a1af-6304-427f-aald-387b1f8fefb5",
      "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
    }
  ]
}
```

```
8-87c9-d220c866997a",
  "representation": null
},
{
  "uri":
"iucld6:/0/SUBSTANCE/IUC4-f696abe6-513a-3771-b2ee-76e4110e8103/SUBSTANCE/IUC4-f696abe
6-513a-3771-b2ee-76e4110e8103",
  "representation": null
},
{
  "uri":
"iucld6:/0/SUBSTANCE/326ab136-d74c-4088-9425-8d9d5642b96e/SUBSTANCE/326ab136-d74c-408
8-9425-8d9d5642b96e",
  "representation": null
},
{
  "uri":
"iucld6:/0/SUBSTANCE/1ef71d52-ddc7-4b55-9c2f-ea72efa55ddd/SUBSTANCE/1ef71d52-ddc7-4b5
5-9c2f-ea72efa55ddd",
  "representation": null
},
{
  "uri":
"iucld6:/0/SUBSTANCE/7cec4254-65f7-4d9a-9a6a-2a74dd1c9b1c/SUBSTANCE/7cec4254-65f7-4d9
a-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.

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

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
```


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	<p>The name of group that the document should be shared to.</p>
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:

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"

byName

byName

Find entities of specific name.

Parameter name	Description
doc.name	The name of the document

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

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.

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.

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.

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.

byTemplate

byTemplate

Find 'Templates' satisfying certain criteria.

Note: the `doc.type` parameter should be omitted.

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

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"

byAnnotation

byAnnotation

Find annotations by various criteria.

Note: the `doc.type` parameter should be omitted.

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

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
<code>site.name</code>	The name of the site.
<code>site.country</code>	Picklist. The country of the site. Should be one of the phase codes in Phasegroup N03
<code>site.town</code>	The town of the site.
<code>site.owner</code>	The name of the legal entity owner of the site.

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.

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.

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.

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:</i> <i>FLEXIBLE_RECORD.Ghs.Labelling.HazardStatementsBlock.HazardSt</i> Should be one of the phase codes in Phasegroup GHS65
ghs.hazard_pictogram	Picklist. The hazard pictogram (<i>path:</i> <i>FLEXIBLE_RECORD.Ghs.Labelling.HazardPictogramBlock.HazardPic</i> Should be one of the phase codes in Phasegroup DM02
ghs.precaution_statement	Picklist. The precautionary statement (<i>path:</i> <i>FLEXIBLE_RECORD.Ghs.Labelling.PrecautionaryStatementsBlock.Pre</i> Should be one of the phase codes in Phasegroup GHS66

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:</i> <i>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:</i> <i>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:</i> <i>FLEXIBLE_RECORD.DsdDpd.Labelling.SafetyPhrases.Safeties.Code</i>) Should be one of the phase codes in Phasegroup N31

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.

Document handling

- 1 Datatypes
 - 1.1 JsonDocumentEnvelope
 - 1.1.1 JsonDocumentHeader
 - 1.1.2 JsonDocumentContent
 - 1.1.2.1 Container types
 - 1.1.2.2 Field types
 - 1.2 Response types
 - 1.2.1 Links
 - 1.2.2 Error
- 2 Semantics of HTTP status codes
 - 2.1 Success Codes
 - 2.2 Error codes
- 3 Load
 - 3.1 Raw data
 - 3.1.1 Entities
 - 3.1.2 Sections
 - 3.1.3 Attachments
 - 3.2 Dossier
 - 3.2.1 Header document
 - 3.2.2 Subject
 - 3.2.3 Sections of subject
 - 3.2.4 Submitter
 - 3.2.5 Dossier Components
 - 3.2.6 Dossier attachments
- 4 Write raw data
 - 4.1 Entities
 - 4.1.1 Create
 - 4.1.1.1 Error responses
 - 4.1.2 Update
 - 4.1.2.1 Error responses
 - 4.1.3 Delete
 - 4.1.3.1 Error responses
 - 4.2 Sections
 - 4.2.1 Create
 - 4.2.2 Update
 - 4.2.3 Delete
 - 4.3 Attachments
- 5 Create Dossier
 - 5.1 Error responses
- 6 Delete Dossier

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

Datatypes

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-c2cadef74a0e/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.

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 <code>{uuid}/{snapshot}</code>
definition	String	The document definition identifier. e.g. <code>FLEXIBLE_RECORD.SubstanceComposition</code>

parentKey	String	Only if the document is a section. The document key of the parent, formatted as <code>{uuid}/{snapshot}</code>
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.
attachments	Array	An array of attachment keys formatted as <code>{uuid}/{snapshot}</code>
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>

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`.

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> <div style="border: 1px solid gray; padding: 10px; margin: 10px 0;"> <p>e.g. The "GeneralInformation" header of the "SubstanceComposition" document</p> <pre>"GeneralInformation": { "Name": "substance-composition-1", "TypeOfComposition": { "code": "61423" } }</pre> </div>
Repeatable Set	Array	<p>It defines a grouping of fields and other container elements, that can be repeated.</p> <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> <div style="border: 1px solid gray; padding: 10px; margin: 10px 0;"> <p>e.g. The "Constituents" repeatable set of the "SubstanceComposition"</p> </div>

Repeatable List

document

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

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

PickListWithRemarks32000		<pre> "other": "other", "remarks": "remarks" } </pre>																		
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>																		
RangeDecimalWithPickList	object	<p>An object containing the following:</p> <table border="1"> <thead> <tr> <th>field</th> <th>type</th> <th>description</th> </tr> </thead> <tbody> <tr> <td>lowerQualifier</td> <td>string</td> <td> one of: <ul style="list-style-type: none"> • ca. • > • >= </td> </tr> <tr> <td>lowerValue</td> <td>number</td> <td>the lower numeric boundary</td> </tr> <tr> <td>upperQualifier</td> <td>string</td> <td> one of: <ul style="list-style-type: none"> • ca. • < • <= </td> </tr> <tr> <td>upperValue</td> <td>number</td> <td>the upper numeric boundary</td> </tr> <tr> <td>unit</td> <td>object</td> <td>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.																		
RangeDecimal	object	same as 'RangeDecimalWithPickList' without the 'unit' field.																		
HalfBounded	object	<p>An object containing the following:</p> <table border="1"> <thead> <tr> <th>field</th> <th>type</th> <th>description</th> </tr> </thead> <tbody> <tr> <td>lowerQualifier</td> <td>string</td> <td> one of: <ul style="list-style-type: none"> • ca. • < • <= • > • >= </td> </tr> <tr> <td>lowerValue</td> <td>number</td> <td>the lower numeric boundary</td> </tr> <tr> <td>unit</td> <td>object</td> <td>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	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																		
unit	object	a picklist object.																		

Confidentiality	object	<p>An object containing the following:</p> <table border="1"> <thead> <tr> <th>field</th> <th>type</th> <th>description</th> </tr> </thead> <tbody> <tr> <td>confidentiality</td> <td>object</td> <td> <p>a picklist object with codes from phase-group N64:</p> <ul style="list-style-type: none"> • 2732 (<i>confidential business information</i>) • 2859 (<i>intellectual property</i>) • 3441 (<i>not publicly available</i>) </td> </tr> <tr> <td>legislations</td> <td>array</td> <td>an array of picklist objects. The phrasegroup of each picklist is N78</td> </tr> <tr> <td>justification</td> <td>string</td> <td>text up to 32786 characters. Can be populated with text template TT_501</td> </tr> </tbody> </table>	field	type	description	confidentiality	object	<p>a picklist object with codes from phase-group N64:</p> <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
field	type	description												
confidentiality	object	<p>a picklist object with codes from phase-group N64:</p> <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	<p>An array of picklist objects with/without remarks.</p> <p>e.g.</p> <pre>[{ "code": "60224", "remarks": "remarks-1" }, { "code": "60225", "remarks": "remarks-2" }, { "code": "60230" }]</pre>												
MultiSelectWithRemarks														
MultiSelectWithRemarks2000														
MultiSelectWithRemarks32000														
EntityReferenceField	string	<p>a document key, formatted as {uuid}/{snapshot}</p> <p>e.g. "4f88bc7f-395c-4d0b-997b-14e8c9aef605/0"</p>												
EndpointReferenceField														
SingleFileAttachment	string	<p>an attachment key, formatted as {uuid}/{snapshot}</p> <p>e.g. "4f88bc7f-395c-4d0b-997b-14e8c9aef605/0"</p>												
Image														
EntityReferenceList	array	<p>an array of strings, that represent keys, either of documents or attachments.</p> <p>The keys are formatted as {uuid}/{snapshot}</p> <p>e.g.</p> <pre>["77748886-6911-4a41-af93-7d5a86fc075f/0", "f9d571cd-d0ed-403d-b944-21e8d26c7c26/0"]</pre>												
EndpointReferenceList														
AttachmentsList														
LiteratureReferenceList														

Response types

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.

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

Semantics of HTTP status codes

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>

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.
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.

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
```

Raw data

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

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: *****'
```

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.

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-c06  
296b43820/document/FLEXIBLE_RECORD.SubstanceComposition/laca8956-fbfe-4f2e-80f7-54182b  
c72454 \  
  --header 'accept: application/vnd.iuclid6.ext+json; type=iuclid6.Document' \  
  --header 'iuclid6-user: SuperUser' \  
  --header 'iuclid6-pass: *****'
```

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

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.

Header document

You access the contents of the header document by:

```
GET /dossier/{dossier_uuid}/header
```

Subject

The dossier subject is available on:

```
GET /dossier/{dossier_uuid}/subject
```

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}
```

Submitter

The submitting legal entity of the dossier is accessible through:

```
GET /dossier/{dossier_uuid}/submitter
```

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](#).

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

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
```

Entities

Create

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

```
POST /raw/{entity_type}
```

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 code	Payload	Reason
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	<p>e.g.</p> <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>	<p>The document content is invalid.</p> <p>This can be either due to incorrect structure, or value content.</p>
-----	---	---

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-9
06c7b864443"

```

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

Error responses

Same as in create scenario.

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:/0/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.

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:

Create	POST	/raw/{entity_type}/{entity_uuid}/document/{doc_def}
Update	PUT	/raw/{entity_type}/{entity_uuid}/document/{doc_def}/{doc_uuid}
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.

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

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-afd4-4da9-9c01-4a
c12b1bc9c0/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	<p>e.g.</p> <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>	<p>The document content is invalid.</p> <p>This can be either due to incorrect structure, or value content.</p>
-----	--	---

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.

Sharing (data access across security groups)

- [1 Introduction](#)
- [2 Datatypes](#)
- [3 Semantics of HTTP status codes](#)
 - [3.1 Success Codes](#)
 - [3.2 Error codes](#)
- [4 Retrieve data access](#)
- [5 Modify data access](#)

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

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.

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

Semantics of HTTP status codes

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>

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.

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: *****'
```

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"}]'
```


Import / Export

- 1 Supported Types
 - 1.1 Entities
 - 1.2 Dossiers
- 2 Export
 - 2.1 Export Entity
 - 2.2
 - 2.3 Export Dossier
- 3 Import

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

Supported Types

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.

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).

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
```

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-4e8458  
79cc99/result \  
  --header 'iuclid6-user: SuperUser' \  
  --header 'iuclid6-pass: *****'
```

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-e73182e08  
764/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-e6f8-4823-8725-1750df  
a82e5e/status \  
  --header 'accept: text/plain' \  
  --header 'iuclid6-user: SuperUser' \  
  --header 'iuclid6-pass: *****'
```

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

```
SUCCEEDED
```

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-e6f8-4823-8725-1750df \  
  a82e5e/result \  
  --header 'iuclid6-user: SuperUser' \  
  --header 'iuclid6-pass: *****'
```

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-52160
e442055;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-b4d2-7a
8da62fbeat",
  "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-59923ee8-3913-46
b3-89c6-5cc696f567a6",
  "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",
```



```
    "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-e4b2eb6
e3441",
    "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-de6f0fdb
efbc",
    "params": [
        "",
        "",
        "",
        "",
        ""
    ]
},
{
    "code": null,
    "level": "INFO",
    "value": "Imported: ENDPOINT_STUDY_RECORD / Acute toxicity: dermal.001 /
IUC5-449d6591-6a32-4f52-a004-f86a955ddlca/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-a0
04-f86a955ddlca",
    "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-4b6
```

```
4-9bf5-6600ded1c98f",
  "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-44
c8-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-b
c72-6015d547298c",
  "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-bfacdf4cf
136",
      "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-5c14-45
a3-8518-042d743f1ecf",
      "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-3790-49e9-8de1-30de33ceclb2/IUC5-e44ee572-3f11-4ad9-bdeb-52160e442055",
      "documentUri": null,
      "params": [
        "",
        "",
        ""
      ]
    }
  ],
  {
    "code": null,
    "level": "INFO",
    "value": "Imported: ATTACHMENT / 152-16-9-V2.jpeg /
ECB5-4653ee4e-3790-49e9-8de1-30de33ceclb2/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"  
}
```