soda4LCA release 2.0.0 Service API

Table of Contents

1. Namespace URIs

If not otherwise specified, response elements (except in full dataset mode) belong to the http://www.ilcd-network.org/ILCD/ServiceAPI namespace. The following table lists the keys that are used to reference the namespaces for response elements.

Table 1. Namespace URIs

Key	Namespace URI
serviceapi	http://www.ilcd-network.org/ILCD/ServiceAPI
xlink	http://www.w3.org/1999/xlink
process	http://www.ilcd-network.org/ILCD/ServiceAPI/Process
flow	http://www.ilcd-network.org/ILCD/ServiceAPI/Flow
flowProperty	http://www.ilcd-network.org/ILCD/ServiceAPI/FlowProperty
unitGroup	http://www.ilcd-network.org/ILCD/ServiceAPI/UnitGroup
lciamethod	http://www.ilcd-network.org/ILCD/ServiceAPI/LCIAMethod
source	http://www.ilcd-network.org/ILCD/ServiceAPI/Source
contact	http://www.ilcd-network.org/ILCD/ServiceAPI/Contact
datastock	http://www.ilcd-network.org/ILCD/ServiceAPI/DataStock
nodeinfo	http://www.ilcd-network.org/ILCD/ServiceAPI/NodeInfo

2. GET Datasets

Basic GET operations that return a list of datasets. Applies to all dataset types.

Requests

Syntax

```
GET /processes
GET /flows
GET /flowproperties
GET /unitgroups
GET /sources
GET /contacts
GET /lciamethods
```

Request Parameters

Table 2.

Name	Description
startIndex	As all result sets are paged, this specifies the index of the first item of the entire result set of the operation that shall be included in the response. Type: Integer Default: 0
pageSize	The page size (number of items) for the response. Type: Integer Default: 500
search	Perform a search query that will return results matching the given query parameters. Type: Boolean Default: false
distributed	Perform a distributed search across all registered network nodes. Type: Boolean Default: false
name	search parameter Type: String Default: None
description	search parameter Type: String Default: None

Responses

Response Elements

The response returned is a list of dataset overview objects, wrapped in a dataSetList object. See the section "DataSetList Response Elements" for a detailed description.

Examples

Sample Request

```
GET /processes
```

Sample Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<serviceapi:dataSetList xmlns:serviceapi="http://www.ilcd-
network.org/ILCD/ServiceAPI" xmlns:xlink="http://www.w3.org/1999/
xlink" xmlns:process="http://www.ilcd-network.org/ILCD/ServiceAPI/
Process" serviceapi:totalSize="4" serviceapi:startIndex="0" serviceapi:pageSize="500">
    <process:process serviceapi:sourceId="ACME" xlink:href="http://lci.acme.com/DB/
processes/0a1b40db-5645-4db8-a887-eb09300b7b74">
        <serviceapi:uuid>0a1b40db-5645-4db8-a887-eb09300b7b74</serviceapi:uuid>
        <serviceapi:permanentUri>http://lca.jrc.ec.europa.eu/lcainfohub/datasets/elcd/
processes/0a1b40db-5645-4db8-a887-eb09300b7b74.xml</serviceapi:permanentUri>
        <serviceapi:dataSetVersion>03.00.000</serviceapi:dataSetVersion>
        <serviceapi:name xml:lang="en">Electricity Mix;AC;consumption mix, at consumer;1kV -
 60kV</serviceapi:name>
        <serviceapi:classification name="ilcd">
            <serviceapi:class level="0">Energy carriers and technologies</serviceapi:class>
            <serviceapi:class level="1">Electricity</serviceapi:class>
        </serviceapi:classification>
        <process:type>LCI result</process:type>
        <process:location>EU-27</process:location>
        <process:time>
            <process:referenceYear>2002</process:referenceYear>
            <process:validUntil>2010</process:validUntil>
        </process:time>
        <process:parameterized>false</process:parameterized>
        <process:hasResults>false</process:hasResults>
        <process:lciMethodInformation>
            <process:methodPrinciple>Attributional</process:methodPrinciple>
            <process:approach>Allocation - mass</process:approach>
            <process:approach>Allocation - market value</process:approach>
            <process:approach>Allocation - exergetic content</process:approach>
            <process:approach>Allocation - net calorific value</process:approach>
        </process:lciMethodInformation>
        <process:complianceSystem name="ILCD Data Network - Entry-level">
            <serviceapi:reference type="source data set" version="00.00.000" uri="../sources/
d92a1a12-2545-49e2-a585-55c259997756.xml"/>
            <process:overallCompliance>Not compliant</process:overallCompliance>
            <process:nomenclatureCompliance>Fully compliant</process:nomenclatureCompliance>
            <process:methodologicalCompliance>Fully compliant</
process:methodologicalCompliance>
            <process:reviewCompliance>Not compliant</process:reviewCompliance>
            <process:documentationCompliance>Not compliant</process:documentationCompliance>
            <process:qualityCompliance>Not defined</process:qualityCompliance>
        </process:complianceSystem>
        <process:accessInformation/>
    </process:process>
</serviceapi:dataSetList>
```

3. GET Dataset

Basic GET operations for all dataset types to retrieve a single dataset

Requests

Syntax

```
GET /processes/{uuid}?version={version}
GET /flows/{uuid}?version={version}
GET /flowproperties/{uuid}?version={version}
GET /unitgroups/{uuid}?version={version}
GET /sources/{uuid}?version={version}
GET /contacts/{uuid}?version={version}
GET /lciamethods/{uuid}?version={version}
```

Versioning

By default, the most recent version of a dataset is retrieved. When specifying the version parameter, that specific version will be returned.

Request Parameters

Table 3.

Name	Description
version	The version number of the dataset to retrieve. If omitted, always the most recent version is retrieved. Type: Version number of the form 00.00.000 Default: None
format	Specifies the format of the response. Values: XML, HTML Default: HTML
view	Specifies whether the response should be summary of the dataset, the full dataset or just the metadata section. Values: overview, full, metadata Default: full

Responses

Response Elements

By default (and with proper access permissions), the full ILCD-formatted dataset is returned as HTML representation (if not otherwise specified). See the section "Response Elements" for response elements in overview.

Examples

Sample Request

```
GET /processes/00000000-0000-0000-0000-000000000000?
format=xml&view=overview
```

Sample Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<process xmlns:serviceapi="http://www.ilcd-network.org/ILCD/ServiceAPI" xmlns:xlink="http://
www.w3.org/1999/xlink" xmlns="http://www.ilcd-network.org/ILCD/ServiceAPI/
Process" xmlns:flow="http://www.ilcd-network.org/ILCD/ServiceAPI/
Flow" xmlns:flowProperty="http://www.ilcd-network.org/ILCD/ServiceAPI/
FlowProperty" xmlns:unitGroup="http://www.ilcd-network.org/ILCD/ServiceAPI/
UnitGroup" xmlns:lciamethod="http://www.ilcd-network.org/ILCD/ServiceAPI/
LCIAMethod" xmlns:source="http://www.ilcd-network.org/ILCD/ServiceAPI/
Source" xmlns:contact="http://www.ilcd-network.org/ILCD/ServiceAPI/
Contact" serviceapi:accessRestricted="true">
    <serviceapi:uuid>00000000-0000-0000-0000-000000000000</serviceapi:uuid>
    <serviceapi:permanentUri>http://db.ilcd-network.org/data/processes/processtest</
serviceapi:permanentUri>
    <serviceapi:dataSetVersion>01.00.000</serviceapi:dataSetVersion>
    <serviceapi:name xml:lang="en">Foo unit process</serviceapi:name>
    <serviceapi:classification>
        <serviceapi:class level="0">Energy systems</serviceapi:class>
        <serviceapi:class level="1">Foo energy systems</serviceapi:class>
    </serviceapi:classification>
    <serviceapi:generalComment xml:lang="en">foo bar</serviceapi:generalComment>
    <serviceapi:synonyms xml:lang="en">Foobar</serviceapi:synonyms>
    <serviceapi:synonyms xml:lang="de">Fubar</serviceapi:synonyms>
    <type>Unit process, single operation</type>
    <quantitativeReference>
        <referenceFlow>
            <name xml:lang="en">electricity mix</name>
            <meanValue>0.0</meanValue>
            <serviceapi:reference type="flow data
 set" refObjectId="00000000-0000-0000-0000-000000000000">
                <serviceapi:shortDescription xml:lang="en">foo flow</
serviceapi:shortDescription>
            </serviceapi:reference>
        </referenceFlow>
        <functionalUnit xml:lang="en">Foonctional Unit</functionalUnit>
    </quantitativeReference>
    <location>RER</location>
    <time>
        <referenceYear>2009</referenceYear>
        <validUntil>2012</validUntil>
    </time>
    <parameterized>true</parameterized>
    <hasResults>true</hasResults>
    <containsProductModel>false</containsProductModel>
    <lciMethodInformation>
        <methodPrinciple>Attributional</methodPrinciple>
        <approach>Allocation - gross calorific value</approach>
        <approach>Allocation - element content</approach>
    </lciMethodInformation>
    <completenessProductModel>All relevant flows quantified</completenessProductModel>
    <complianceSystem name="ILCD Compliance - entry level">
        <overallCompliance>Fully compliant</overallCompliance>
    </complianceSystem>
    <review type="Independent external review">
        <scope name="LCI results or Partly terminated system">
            <method name="Element balance"/>
```

```
        </scope>
        <dataQualityIndicators>
            <dataQualityIndicator name="Completeness" value="Good"/>
        </dataQualityIndicators>
        <reviewDetails xml:lang="en">details here</reviewDetails>
    </review>
    <overallQuality>FOOOO</overallQuality>
    <useAdvice xml:lang="en">hear my advice</useAdvice>
    <accessInformation>
        <copyright>true</copyright>
        <licenseType>Free of charge for all users and uses</licenseType>
        <useRestrictions xml:lang="en">Rated R</useRestrictions>
    </accessInformation>
    <format>ILCD 1.0</format>
    <ownership type="contact data set" refObjectId="00000000-0000-0000-0000-000000000000">
        <serviceapi:shortDescription xml:lang="en">JRC</serviceapi:shortDescription>
    </ownership>
</process>
```

Sample Request

```
GET /processes/00000000-0000-0000-0000-000000000000?format=xml
```

Sample Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet version="1.0" href="../../stylesheets/process2html.xsl" type="text/xsl"?>
<processDataSet xmlns="http://lca.jrc.it/ILCD/Process" xmlns:common="http://lca.jrc.it/ILCD/
Common" locations="../ILCDLocations.xml" version="1.1">
<processInformation>
<dataSetInformation>
<common:UUID>00000000-0000-0000-0000-000000000000</common:UUID>
<name>
<baseName xml:lang="en">Electricity Mix, Foo</baseName>
<treatmentStandardsRoutes xml:lang="en">AC</treatmentStandardsRoutes>
<mixAndLocationTypes xml:lang="en">consumption mix, at consumer</mixAndLocationTypes>
<functionalUnitFlowProperties xml:lang="en">1kV - 60kV</functionalUnitFlowProperties>
</name>
<common:synonyms xml:lang="en">power grid mix</common:synonyms>
<classificationInformation>
<common:classification>
<common:class level="0">Energy carriers and technologies</common:class>
<common:class level="1">Electricity</common:class>
</common:classification>
</classificationInformation>
<common:generalComment xml:lang="en">Good overall data quality. Energy carrier mix
 information based on official statistical information including import/export. Detailed
 power plant models were used, which combine measured emissions plus calculated values for
 not measured emissions of e.g. organics or heavy metals. Energy carrier extraction and
 processing data is of sufficient to good (e.g. refinery) data quality. Inventory is partly
 based on primary industry data, partly on secondary literature data.</common:generalComment>
</dataSetInformation>
<quantitativeReference type="Reference flow(s)">
<referenceToReferenceFlow>63</referenceToReferenceFlow>
</quantitativeReference>
<time>
<common:referenceYear>2002</common:referenceYear>
<common:dataSetValidUntil>2010</common:dataSetValidUntil>
<common:timeRepresentativenessDescription xml:lang="en">Annual average</
common:timeRepresentativenessDescription>
</time>
<geography>
...
</processDataSet>
```

4. GET Exchanges of a Process Dataset

GET operation to retrieve the list of exchanges for a specific process dataset.

Requests

Syntax

```
GET /processes/{uuid}/exchanges
```

Request Parameters

Table 4.

Name	Description
direction	The direction of the exchanges to be retrieved. Optional Values: in, out Default: None
type	The type of the exchanges to be retrieved. Optional Values: Elementary flow, Product flow, Waste flow, Other flow Default: None

Responses

Response Elements

A datasetList object containing flow objects is returned as response. See the sections "DatasetList Response Elements" and "Flow Response Elements" for a detailed description.

Examples

Sample Request

```
GET /processes/0cbf76cc-0192-4617-acd3-0fdb3cecf6c7/exchanges?
direction=in
```

Sample Response

```
HTTP/1.1 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<serviceapi:dataSetList xmlns:serviceapi="http://www.ilcd-network.org/ILCD/
ServiceAPI" xmlns:xlink="http://www.w3.org/1999/xlink"
```

```
    xmlns:flow="http://www.ilcd-network.org/ILCD/ServiceAPI/
Flow" serviceapi:totalSize="60" serviceapi:startIndex="0" serviceapi:pageSize="60">
    <flow:flow serviceapi:sourceId="ACME" xlink:href="http://localhost:8091/Node/flows/
fe0acd60-3ddc-11dd-ae5d-0050c2490048">
        <serviceapi:uuid>fe0acd60-3ddc-11dd-ae5d-0050c2490048</serviceapi:uuid>
        <serviceapi:permanentUri>http://lca.jrc.ec.europa.eu/lcainfohub/datasets/ilcd/flows/
fe0acd60-3ddc-11dd-ae5d-0050c2490048_02.00.000.xml</serviceapi:permanentUri>
        <serviceapi:dataSetVersion>02.00.000</serviceapi:dataSetVersion>
        <serviceapi:name xml:lang="en">lead</serviceapi:name>
        <flow:flowCategorization name="ilcd">
            <serviceapi:category level="0">Resources</serviceapi:category>
            <serviceapi:category level="1">Resources from ground</serviceapi:category>
            <serviceapi:category level="2">Non-renewable element resources from ground</
serviceapi:category>
        </flow:flowCategorization>
        <flow:type>Elementary flow</flow:type>
        <flow:referenceFlowProperty xlink:href="">
            <flow:name xml:lang="de">Formaldehyd</flow:name>
            <flow:defaultUnit>kg</flow:defaultUnit>
            <serviceapi:reference type="flow property data set" version="02.00.000" uri="../
flowproperties/93a60a56-a3c8-11da-a746-0800200b9a66_02.00.000.xml"/>
        </flow:referenceFlowProperty>
    </flow:flow>
    <flow:flow serviceapi:sourceId="ACME" xlink:href="http://localhost:8091/Node/
flows/1729ef88-6556-11dd-ad8b-0800200c9a66">
        <serviceapi:uuid>1729ef88-6556-11dd-ad8b-0800200c9a66</serviceapi:uuid>
        <serviceapi:permanentUri>http://lca.jrc.ec.europa.eu/lcainfohub/datasets/ilcd/
flows/1729ef88-6556-11dd-ad8b-0800200c9a66_02.00.000.xml</serviceapi:permanentUri>
        <serviceapi:dataSetVersion>02.00.000</serviceapi:dataSetVersion>
        ...
    </flow:flow>
</serviceapi:dataSetList>
```

5. GET Producers or Consumers of a Flow

GET operation to identify processes that produce or consume a specific flow.

Requests

Syntax

```
GET /flows/{uuid}?version={version}/producers/
GET /flows/{uuid}?version={version}/consumers/
```

Request Parameters

Table 5.

Name	Description
	Type:
	Default:

Responses

Response Elements

By default (and with proper access permissions), the full ILCD-formatted dataset is returned as HTML representation (if not otherwise specified). See the section "Response Elements" for response elements in overview.

Examples

Sample Request

Sample Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

6. GET Digital File of a Source Dataset

GET operation to retrieve the digital file attachment for a specific source dataset.

Requests

Syntax

```
GET /sources/{uuid}/{filename}
```

The filename must be URL encoded.

```
GET /sources/{uuid}/digitalfile
```

The latter variant will retrieve the first digital file entry for the specified dataset.

Request Parameters

Responses

The digital file.

The returned MIME type will be `image/*` for images and `application/pdf` for files that carry a `.pdf` extension.

Examples

Sample Request

```
GET /sources/0a34866e-ce75-48c8-82e6-0080739e7154/100512%20System
%20boundaries%20diagram%20-%20with%20reuse.jpg
```

Sample Response

```
HTTP/1.1 200 OK
Content-Type: image/jpg
```

(the image)

Sample Request

```
GET /sources/cb1c5d4a-50ed-4d7b-828b-6fcd560ee17b/digitalfile
```

Sample Response

```
HTTP/1.1 200 OK
Content-Type: application/pdf
```

(the PDF document)

7. POST Datasets

Requests

Syntax

```
POST /
```

Request Parameters

Table 6.

Name	Description
	Type: String
	Default: None

Responses

Response Elements

Table 7.

Name	Description
	Type:
	Ancestors:

Examples

Sample Request

```
POST /
```

Sample Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

8. GET Process Datasets (Query)

Requests

Syntax

```
GET /
```

Request Parameters

Table 8.

Name	Description
	Type: String
	Default: None

Responses

Response Elements

Table 9.

Name	Description
	Type:
	Ancestors:

Examples

Sample Request

```
GET /
```

Sample Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

9. GET Datastocks

GET operation that returns a list of all datastocks.

Requests

Syntax

```
GET /datastocks
```

Request Parameters

None.

Responses

Response Elements

Table 10.

Name	Description
dataStockList (datastock)	The container element for the list of data stock objects. Type: Container Ancestors: none.
dataStock (datastock)	The container element for the data stock object. Type: Container may occur multiple times Ancestors: dataStockList
@root (datastock)	Indicates whether the data stock is a root data stock. Type: Boolean Ancestors: dataStock
shortName	The short name (handle) of the data stock. Type: String Ancestors: dataStock
name	The name of the data stock. Type: String Multilang may occur multiple times Ancestors: dataStock
description	A description for the data stock.

Name	Description
(datastock)	Type: String Multilang
	may occur multiple times
	Ancestors: dataStock

Examples

Sample Request

```
GET /datastocks
```

Sample Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dataStockList xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.ilcd-network.org/ILCD/
ServiceAPI/DataStock ../schemas/ILCD_Service_API_DataStocks.xsd"
    xmlns="http://www.ilcd-network.org/ILCD/ServiceAPI/
DataStock" xmlns:ds="http://www.ilcd-network.org/ILCD/ServiceAPI/
DataStock" xmlns:serviceapi="http://www.ilcd-network.org/ILCD/
ServiceAPI">
    <dataStock ds:root="true">
        <serviceapi:shortName>default</serviceapi:shortName>
        <serviceapi:name xml:lang="en">Default Root Data Stock</
serviceapi:name>
            <serviceapi:name xml:lang="de">deutscher Name</
serviceapi:name>
        <description xml:lang="en">description</description>
        <description xml:lang="de">deutsche Beschreibung</description>
    </dataStock>
    <dataStock>
        <serviceapi:shortName>other</serviceapi:shortName>
        <serviceapi:name xml:lang="en">Other, Non-Root Data Stock</
serviceapi:name>
        <description xml:lang="en">other data</description>
    </dataStock>
</dataStockList>
```

10. GET Datasets from a Data Stock

Basic GET operations that return a list of datasets from a specific data stock. Applies to all dataset types.

Requests

Syntax

```
GET /datastocks/{datastock-uuid}/processes
    GET /datastocks/{datastock-uuid}/flows
    GET /datastocks/{datastock-uuid}/flowproperties
    GET /datastocks/{datastock-uuid}/unitgroups
    GET /datastocks/{datastock-uuid}/sources
    GET /datastocks/{datastock-uuid}/contacts
    GET /datastocks/{datastock-uuid}/lciamethods
```

Request Parameters

All request parameters as for the basic GET Datasets operation are supported. See section Request Parameters for GET Datasets for the full list.

Responses

Response Elements

The response returned is a list of dataset overview objects, wrapped in a dataSetList object. See the section "DataSetList Response Elements" for a detailed description.

Examples

Sample Request

```
GET
    /datastocks/aca74e60-146e-11e2-892e-0800200c9a66/processes
```

Sample Response

```
HTTP/1.1 200 OK
    Content-Type: application/xml
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<serviceapi:dataSetList xmlns:serviceapi="http://www.ilcd-
network.org/ILCD/ServiceAPI" xmlns:xlink="http://www.w3.org/1999/
xlink" xmlns:process="http://www.ilcd-network.org/ILCD/ServiceAPI/
Process" serviceapi:totalSize="4" serviceapi:startIndex="0" serviceapi:pageSize="500">
    <process:process serviceapi:sourceId="ACME" xlink:href="http://lci.acme.com/DB/
processes/0a1b40db-5645-4db8-a887-eb09300b7b74">
        <serviceapi:uuid>0a1b40db-5645-4db8-a887-eb09300b7b74</serviceapi:uuid>
        <serviceapi:permanentUri>http://lca.jrc.ec.europa.eu/lcainfohub/datasets/elcd/
processes/0a1b40db-5645-4db8-a887-eb09300b7b74.xml</serviceapi:permanentUri>
```

```
        <serviceapi:dataSetVersion>03.00.000</serviceapi:dataSetVersion>
        <serviceapi:name xml:lang="en">Electricity Mix;AC;consumption mix, at consumer;1kV -
60kV</serviceapi:name>
        <serviceapi:classification name="ilcd">
            <serviceapi:class level="0">Energy carriers and technologies</serviceapi:class>
            <serviceapi:class level="1">Electricity</serviceapi:class>
        </serviceapi:classification>
        <process:type>LCI result</process:type>
        <process:location>EU-27</process:location>
        <process:time>
            <process:referenceYear>2002</process:referenceYear>
            <process:validUntil>2010</process:validUntil>
        </process:time>
        <process:parameterized>false</process:parameterized>
        <process:hasResults>false</process:hasResults>
        <process:lciMethodInformation>
            <process:methodPrinciple>Attributional</process:methodPrinciple>
            <process:approach>Allocation - mass</process:approach>
            <process:approach>Allocation - market value</process:approach>
            <process:approach>Allocation - exergetic content</process:approach>
            <process:approach>Allocation - net calorific value</process:approach>
        </process:lciMethodInformation>
        <process:complianceSystem name="ILCD Data Network - Entry-level">
            <serviceapi:reference type="source data set" version="00.00.000" uri="../sources/
d92a1a12-2545-49e2-a585-55c259997756.xml"/>
            <process:overallCompliance>Not compliant</process:overallCompliance>
            <process:nomenclatureCompliance>Fully compliant</process:nomenclatureCompliance>
            <process:methodologicalCompliance>Fully compliant</
process:methodologicalCompliance>
            <process:reviewCompliance>Not compliant</process:reviewCompliance>
            <process:documentationCompliance>Not compliant</process:documentationCompliance>
            <process:qualityCompliance>Not defined</process:qualityCompliance>
        </process:complianceSystem>
        <process:accessInformation/>
    </process:process>
</serviceapi:dataSetList>
```

11. Authentication

Login

In order to perform certain operations, authentication may be required. This GET operation performs authentication against the application, setting a session cookie if successful.

Requests

Syntax

```
GET /authenticate/login
```

Request Parameters

Table 11.

Name	Description
userName	The username to authenticate with Type: String Default: None
password	The password to authenticate with Type: String Default: None

Responses

Response Elements

None

Examples

Sample Request

```
GET /authenticate/login?username=foo&password=bar
```

Sample Response if Login Successful

```
HTTP/1.1 200 OK
Content-Type: text/plain
```

```
Login successful
```

Sample Response if Already Logged In

```
HTTP/1.1 200 OK
Content-Type: text/plain
```

```
You are already logged in as a user
```

Sample Response if Wrong User Name or Password

```
HTTP/1.1 200 OK
Content-Type: text/plain
```

```
incorrect password or user name
```

Sample Response if User Name or Password Missing in Request

```
HTTP/1.1 200 OK
Content-Type: text/plain
```

```
user name and password must have a value
```

Logout

With this GET operation, the session for a currently authenticated user can be closed.

Requests

Syntax

```
GET /authentication/logout
```

Request Parameters

Responses

Response Elements

None.

Examples

Sample Request

```
GET /authentication/logout
```

Sample Response if authenticated

```
HTTP/1.1 200 OK
Content-Type: text/plain
```

```
successfully logged out
```

Sample Response if not authenticated

```
HTTP/1.1 200 OK
Content-Type: text/plain
```

```
currently not authenticated
```

Status

With this GET operation, the current authentication status (authenticated or not) can be retrieved, as well the user name and any associated roles if authenticated.

Requests

Syntax

```
GET /authentication/status
```

Request Parameters

Responses

Response Elements

Table 12.

Name	Description
authInfo	Contains the elements that describe the authentication status. Type: Container Ancestors: None
authenticated	Indicates whether the session is currently authenticated or not. Type: Boolean Ancestors: authInfo
userName	The username for the current session Type: String Ancestors: authInfo
role	One entry for each role associated for the current session Type: String may occur multiple times Ancestors: authInfo

Examples

Sample Request

```
GET /authentication/status
```

Sample Response if authenticated

```
HTTP/1.1 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<authInfo xmlns="http://www.ilcd-network.org/ILCD/ServiceAPI">
    <authenticated>true</authenticated>
    <userName>admin</userName>
    <role>READ</role>
    <role>EXPORT</role>
    <role>CHECKOUT</role>
    <role>CHECKIN</role>
    <role>RELEASE</role>
    <role>DELETE</role>
    <role>MANAGE_USER</role>
    <role>ADMIN</role>
    <role>SUPER_ADMIN</role>
</authInfo>
```

Sample Response if not authenticated

```
HTTP/1.1 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<authInfo xmlns="http://www.ilcd-network.org/ILCD/ServiceAPI">
    <authenticated>false</authenticated>
</authInfo>
```

12. Node Information

This GET operation retrieves information about the node.

Requests

Syntax

```
GET /nodeinfo
```

Request Parameters

None.

Responses

Response Elements

Table 13.

Name	Description
nodeInfo	Contains the elements with the node information. Type: Container Ancestors: None.
nodeID	The ID of the node. Type: String (no spaces allowed) Ancestors: nodeInfo
name	The full name of the node Type: String Ancestors: nodeInfo
operator	The person or entity operating this node Type: String Ancestors: nodeInfo
description	A description of the node Type: String Ancestors: nodeInfo
baseURL	The base URL of the node's service interface. Type: String Ancestors: nodeInfo

Name	Description
administrativeContact	The element carrying information about the administrative contact for the node. Type: String Ancestors: nodeInfo
centralContactPoint	The central contact point Type: String Ancestors: nodeInfo.centralContactPoint
email	The email address Type: String Ancestors: nodeInfo.centralContactPoint
phone	The phone number Type: String Ancestors: nodeInfo.centralContactPoint
www	The www address Type: String Ancestors: nodeInfo.centralContactPoint

Examples

Sample Request

```
GET /nodeinfo
```

Sample Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
```

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<nodeInfo xmlns="http://www.ilcd-network.org/ILCD/ServiceAPI/
NodeInfo" xmlns:contact="http://www.ilcd-network.org/ILCD/ServiceAPI/
Contact">
    <nodeID>ACME</nodeID>
    <name>ACME Public LCI Database</name>
    <operator>ACME Inc.</operator>
    <description xml:lang="en">Free Text Description</description>
    <baseURL>http://lci.acme.com/DB</baseURL>
    <administrativeContact>
        <contact:centralContactPoint>ACME Inc. Worldwide
 Headquarters</contact:centralContactPoint>
        <contact:email>info@acme.com</contact:email>
        <contact:phone>+49 721 555 4242</contact:phone>
        <contact:www>www.acme.com</contact:www>
```

```
        </administrativeContact>
</nodeInfo>
```

13. Response Elements

DatasetList Response Elements

Table 14.

Name	Description
dataSetList	The container element for the list of dataset objects. Type: Container Ancestors: None
@sourceId	Node ID of the originating node. Type: String Ancestors: dataSetList
@totalSize	Total size of the result set, i.e. number of contained dataset objects. Type: Integer Ancestors: dataSetList
@startIndex	The index of the first item in this list in relation to the entire result set. Type: Integer Ancestors: dataSetList
@pageSize	The page size (number of items) for this list. Type: Integer Ancestors: dataSetList
process (process)	A process dataset object. See section "Process Response Elements" for a detailed description. Type: Container optional, may occur multiple times Ancestors: dataSetList
flow (flow)	A flow dataset object. See section "Flow Response Elements" for a detailed description. Type: Container optional, may occur multiple times Ancestors: dataSetList
flowProperty (flowproperty)	A flow property dataset object. See section "FlowProperty Response Elements" for a detailed description. Type: Container optional, may occur multiple times Ancestors: dataSetList

Name	Description
unitGroup (unitgroup)	A unit group dataset object. See section "UnitGroup Response Elements" for a detailed description. Type: Container optional, may occur multiple times Ancestors: dataSetList
contact (contact)	A source dataset object. See section "Source Response Elements" for a detailed description. Type: Container optional, may occur multiple times Ancestors: dataSetList
contact (contact)	A contact dataset object. See section "Contact Response Elements" for a detailed description. Type: Container optional, may occur multiple times Ancestors: dataSetList
lciamethod (lciamethod)	An LCIA method dataset object. See section "LCIAMethod Response Elements" for a detailed description. Type: Container optional, may occur multiple times Ancestors: dataSetList

Process Response Elements

Table 15.

Name	Description
process (process)	<description here> Type: String Ancestors: None
@accessRestricted	<description here> Type: String Ancestors: process
uuid	<description here> Type: String Ancestors: process
permanentUri	<description here> Type: String Ancestors: process
dataSetVersion	<description here> Type: String Ancestors: process
name	<description here> Type: String Multilang Ancestors: process
classification	<description here> Type: String Ancestors: process
class	<description here> Type: String may occur multiple times Ancestors: process.classification
@level	<description here> Type: String Ancestors: process.classification.class
generalComment	<description here>

Name	Description
	Type: String Multilang Ancestors: process
synonyms	<description here> Type: String Multilang may occur multiple times Ancestors: process
type (process)	<description here> Type: String Ancestors: process
quantitativeReference (process)	<description here> Type: String Ancestors: process
referenceFlow (process)	<description here> Type: String Ancestors: process.quantitativeReference
name (process)	<description here> Type: String Multilang Ancestors: process.quantitativeReference.referenceFlow
meanValue (process)	<description here> Type: String Ancestors: process.quantitativeReference.referenceFlow
reference	<description here> Type: String Ancestors: process.quantitativeReference.referenceFlow
@type	<description here> Type: String Ancestors: process.quantitativeReference.referenceFlow.reference
@refObjectId	<description here> Type: String Ancestors: process.quantitativeReference.referenceFlow.reference
shortDescription	<description here>

Name	Description
	Type: String Multilang Ancestors: process.quantitativeReference.referenceFlow.reference
functionalUnit (process)	<description here> Type: String Multilang Ancestors: process.quantitativeReference
location (process)	<description here> Type: String Ancestors: process
time (process)	<description here> Type: String Ancestors: process
referenceYear (process)	<description here> Type: String Ancestors: process.time
validUntil (process)	<description here> Type: String Ancestors: process.time
parameterized (process)	<description here> Type: String Ancestors: process
hasResults (process)	<description here> Type: String Ancestors: process
containsProductModel (process)	<description here> Type: String Ancestors: process
lciMethodInformation (process)	<description here> Type: String Ancestors: process
methodPrinciple (process)	<description here> Type: String

Name	Description
	Ancestors: process.lciMethodInformation
approach (process)	<description here> Type: String may occur multiple times Ancestors: process.lciMethodInformation
completenessProduct-Model (process)	<description here> Type: String Ancestors: process
complianceSystem (process)	<description here> Type: String Ancestors: process
@name	<description here> Type: String Ancestors: process.complianceSystem
overallCompliance (process)	<description here> Type: String Ancestors: process.complianceSystem
review (process)	<description here> Type: String Ancestors: process
@type	<description here> Type: String Ancestors: process.review
scope (process)	<description here> Type: String Ancestors: process.review
@name	<description here> Type: String Ancestors: process.review.scope
method (process)	<description here> Type: String

Name	Description
	Ancestors: process.review.scope
@name	\<description here\> Type: String Ancestors: process.review.scope.method
dataQualityIndicators (process)	\<description here\> Type: String Ancestors: process.review
dataQualityIndicator (process)	\<description here\> Type: String Ancestors: process.review.dataQualityIndicators
@name	\<description here\> Type: String Ancestors: process.review.dataQualityIndicators.dataQualityIndicator
@value	\<description here\> Type: String Ancestors: process.review.dataQualityIndicators.dataQualityIndicator
reviewDetails (process)	\<description here\> Type: String Multilang Ancestors: process.review
overallQuality (process)	\<description here\> Type: String Ancestors: process
useAdvice (process)	\<description here\> Type: String Multilang Ancestors: process
accessInformation (process)	\<description here\> Type: String Ancestors: process
copyright (process)	\<description here\> Type: String Ancestors: process.accessInformation

Name	Description
licenseType (process)	<description here> Type: String Ancestors: process.accessInformation
useRestrictions (process)	<description here> Type: String Multilang Ancestors: process.accessInformation
format (process)	<description here> Type: String Ancestors: process
ownership (process)	<description here> Type: String Ancestors: process
@type	<description here> Type: String Ancestors: process.ownership
@refObjectId	<description here> Type: String Ancestors: process.ownership
shortDescription	<description here> Type: String Multilang Ancestors: process.ownership

Flow Response Elements

Table 16.

Name	Description
flow (flow)	\<description here\> Type: String Ancestors: None
uuid	\<description here\> Type: String Ancestors: flow
permanentUri	\<description here\> Type: String Ancestors: flow
dataSetVersion	\<description here\> Type: String Ancestors: flow
name	\<description here\> Type: String Multilang Ancestors: flow
generalComment	\<description here\> Type: String Multilang Ancestors: flow
synonyms	\<description here\> Type: String Multilang may occur multiple times Ancestors: flow
flowCategorization (flow)	\<description here\> Type: String Ancestors: flow
category	\<description here\> Type: String may occur multiple times

Name	Description
	Ancestors: flow.flowCategorization
@level	<description here>
	Type: String
	Ancestors: flow.flowCategorization.category
type (flow)	<description here> Type: String Ancestors: flow
casNumber (flow)	<description here> Type: String Ancestors: flow
sumFormula (flow)	<description here> Type: String Ancestors: flow
referenceFlowProperty (flow)	<description here> Type: String Ancestors: flow
@href ()	<description here> Type: String Ancestors: flow.referenceFlowProperty
name (flow)	<description here> Type: String Multilang Ancestors: flow.referenceFlowProperty
defaultUnit (flow)	<description here> Type: String Ancestors: flow.referenceFlowProperty
reference	<description here> Type: String Ancestors: flow.referenceFlowProperty
@type	<description here> Type: String Ancestors: flow.referenceFlowProperty.reference

Name	Description
@refObjectId	<description here> Type: String Ancestors: flow.referenceFlowProperty.reference
shortDescription	<description here> Type: String Multilang Ancestors: flow.referenceFlowProperty.reference

FlowProperty Response Elements

Table 17.

Name	Description
flowProperty (flow)	\<description here\> Type: String Ancestors: None
uuid	\<description here\> Type: String Ancestors: flowProperty
permanentUri	\<description here\> Type: String Ancestors: flowProperty
dataSetVersion	\<description here\> Type: String Ancestors: flowProperty
name	\<description here\> Type: String Multilang Ancestors: flowProperty
generalComment	\<description here\> Type: String Multilang Ancestors: flowProperty
synonyms	\<description here\> Type: String Multilang may occur multiple times Ancestors: flowProperty
unitGroup (flow)	\<description here\> Type: String Ancestors: flowProperty
name (flow)	\<description here\> Type: String Multilang Ancestors: flowProperty.unitGroup
defaultUnit	\<description here\>

Name	Description
(flow)	Type: String Ancestors: flowProperty.unitGroup
reference	<description here> Type: String Ancestors: flowProperty.unitGroup
@type	<description here> Type: String Ancestors: flowProperty.unitGroup.reference
@refObjectId	<description here> Type: String Ancestors: flowProperty.unitGroup.reference
shortDescription	<description here> Type: String Multilang Ancestors: flowProperty.unitGroup.reference

UnitGroup Response Elements

Table 18.

Name	Description
unitGroup (unitgroup)	<description here> Type: String Ancestors: None
uuid	<description here> Type: String Ancestors: unitGroup
permanentUri	<description here> Type: String Ancestors: unitGroup
dataSetVersion	<description here> Type: String Ancestors: unitGroup
name	<description here> Type: String Multilang Ancestors: unitGroup
generalComment	<description here> Type: String Multilang Ancestors: unitGroup
referenceUnit (unitgroup)	<description here> Type: String Ancestors: unitGroup

Source Response Elements

Table 19.

Name	Description
source (source)	<description here> Type: String Ancestors: None
uuid	<description here> Type: String Ancestors: source
permanentUri	<description here> Type: String Ancestors: source
dataSetVersion	<description here> Type: String Ancestors: source
name	<description here> Type: String Multilang may occur multiple times Ancestors: source
generalComment	<description here> Type: String Multilang Ancestors: source
citation (source)	<description here> Type: String Multilang Ancestors: source
publicationType (source)	<description here> Type: String Ancestors: source
file (source)	<description here> Type: String Ancestors: source
@type	<description here>

Name	Description
	Type: String Ancestors: source.file
@href ()	\<description here\> Type: String Ancestors: source.file
shortDescription	\<description here\> Type: String Multilang Ancestors: source.file
belongsTo (source)	\<description here\> Type: String Ancestors: source
@type	\<description here\> Type: String Ancestors: source.belongsTo
@refObjectId	\<description here\> Type: String Ancestors: source.belongsTo
shortDescription	\<description here\> Type: String Multilang Ancestors: source.belongsTo

Contact Response Elements

Table 20.

Name	Description
contact (contact)	<description here> Type: String Ancestors: None
@sourceId	<description here> Type: String Ancestors: contact
uuid	<description here> Type: String Ancestors: contact
permanentUri	<description here> Type: String Ancestors: contact
dataSetVersion	<description here> Type: String Ancestors: contact
name	<description here> Type: String Multilang Ancestors: contact
classification	<description here> Type: String Ancestors: contact
class	<description here> Type: String may occur multiple times Ancestors: contact.classification
@level	<description here> Type: String Ancestors: contact.classification.class
generalComment	<description here>

Name	Description
	Type: String Multilang Ancestors: contact
centralContactPoint (contact)	<description here> Type: String Ancestors: contact
phone (contact)	<description here> Type: String Ancestors: contact
fax (contact)	<description here> Type: String Ancestors: contact
email (contact)	<description here> Type: String Ancestors: contact
www (contact)	<description here> Type: String Ancestors: contact

LCIAMethod Response Elements

Table 21.

Name	Description
LCIAMethod (lciamethod)	<description here> Type: String Ancestors: None
uuid	<description here> Type: String Ancestors: LCIAMethod
permanentUri	<description here> Type: String Ancestors: LCIAMethod
dataSetVersion	<description here> Type: String Ancestors: LCIAMethod
name	<description here> Type: String Multilang Ancestors: LCIAMethod
classification	<description here> Type: String Ancestors: LCIAMethod
class	<description here> Type: String may occur multiple times Ancestors: LCIAMethod.classification
@level	<description here> Type: String Ancestors: LCIAMethod.classification.class
generalComment	<description here> Type: String Multilang Ancestors: LCIAMethod
type	<description here>

Name	Description
(lciamethod)	Type: String Ancestors: LCIAMethod
methodology (lciamethod)	<description here> Type: String may occur multiple times Ancestors: LCIAMethod
impactCategory (lciamethod)	<description here> Type: String Ancestors: LCIAMethod
areaOfProtection (lciamethod)	<description here> Type: String Ancestors: LCIAMethod
impactIndicator (lciamethod)	<description here> Type: String Ancestors: LCIAMethod
time (lciamethod)	<description here> Type: String Ancestors: LCIAMethod
referenceYear (lciamethod)	<description here> Type: String Multilang Ancestors: LCIAMethod.time
duration (lciamethod)	<description here> Type: String Multilang Ancestors: LCIAMethod.time