

# Nokis CSW 2.0.2 Client API



smile consult GmbH  
Vahrenwalder Straße 4  
30165 Hannover



16.11.2011

smile consult GmbH  
 Vahrenwalder Straße 4  
 30165 Hannover

Tel.: 0511 / 54 36 17 - 40  
 Fax: 0511 / 54 36 17 - 66

[info@smileconsult.de](mailto:info@smileconsult.de)  
[www.smileconsult.de](http://www.smileconsult.de)

## Table of Contents

<b>1.</b>	<b>Introduction</b>	<b>3</b>
<b>2.</b>	<b>Hits request and Results request</b>	<b>5</b>
2.1.	Hits request .....	5
2.2.	Results request .....	5
<b>3.</b>	<b>GetRecordById request</b>	<b>7</b>
<b>4.</b>	<b>GetRecords request</b>	<b>8</b>
4.1.	Request with literal condition .....	8
4.2.	Request with literal and temporal conditions .....	8
4.3.	Request with literal and spatial conditions .....	9
<b>5.</b>	<b>Transaction requests</b>	<b>11</b>
5.1.	Insert .....	11
5.2.	Update .....	11
5.3.	Delete .....	12

## 1. Introduction

**Note:** This client has been tested with Nokis server software version 2.1.523.1. Higher versions of the server may contain subtle changes that make it incompatible with this API.

The central component of the API is the **NokisCSWClient** class, which can be instantiated using the following constructor:

```
NokisCSWClient (      NokisCSWClient.RequestMethod method,
                     NokisCSWClient.Version version,
                     String url,
                     String username,
                     String password);
```

With:

- **method:** enum parameter of type **NokisCSWClient.RequestMethod** that denotes the request method. Available values are **NokisCSWClient.RequestMethod.SOAP** and **NokisCSWClient.RequestMethod.POST**
- **version:** enum parameter of type **NokisCSWClient.Version** that denotes the CSW version. For CSW 2.0.2 it should always be **NokisCSWClient.Version.CSW\_2\_0\_2**
- **url:** the URL string for the **GetCapabilities** request. Depending on the **method** parameter, a suitable URL must be provided (there are separate URLs for different types of methods, i.e. POST, SOAP)
- **username & password:** the username and password strings for server authentication. When null values or invalid username and/or password are used, the server will respond with only valid records that are externally shared

Example:

```
NokisCSWClient hzgcswclient = new NokisCSWClient (
    NokisCSWClient.RequestMethod.SOAP,
    NokisCSWClient.Version.CSW_2_0_2,
    "http://.../csw/v_2_0_2/GetCapabilitiesServant$Post$SOAP",
    "username",
    "password");
```

When the **NOKISCSWClient** object (hereafter referred to as **hzgcswclient** in the sample code) is instantiated, a **GetCapabilities** request will be sent and the response, if successfully received, will be stored for later reference. Several methods can then be called on this client object to perform different requests, with the needed URL for each operation being fetched from the stored **GetCapabilities** response. Important methods include the following:

- **QueryMetadataById(String[] uuids):** takes an array of UUIDs and performs a **GetRecordById** request to fetch the records
- **QueryMetadataByCondition(CSWQueryCondition c):** takes a **CSWQueryCondition** object in which query conditions are specified and performs a **GetRecords** request

- ***insertMetadata(de.disy.preludio2.base.MDMetaData metadata)***: performs a transactional ***Insert*** operation
- ***updateRecordStatus(String uuid, RecordStatus v)***: update the status of the record whose UUID is provided, using the transactional ***Update*** request
- ***deleteRecord(String uuid)***: delete the record whose UUID is specified, using the transactional ***Delete*** request

The first 2 methods return a ***CSWQueryResult*** object holding the response that can be accessed via several methods, e.g. *getMetadata()*, *getProjects()*... The last 3 methods return a ***CSWInsertOrUpdateOrDeleteResult*** object containing methods such as *getInserted()*, *getUpdated()*, *getDeleted()* that give information about how many records have been affected in the transaction.

## 2. Hits request and Results request

### 2.1. Hits request

Generated request document:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns7:GetRecords xmlns="http://www.opengis.net/gml"
xmlns:ns2="http://www.w3.org/1999/xlink" xmlns:ns3="http://www.w3.org/2001/SMIL20/"
xmlns:ns4="http://purl.org/dc/elements/1.1/" xmlns:ns5="http://purl.org/dc/terms/"
xmlns:ns6="http://www.opengis.net/ogc"
xmlns:ns7="http://www.opengis.net/cat/csw/2.0.2"
xmlns:ns8="http://www.opengis.net/ows"
xmlns:ns9="http://www.disy.de/preludio2/base"
xmlns:preludio="http://www.disy.de/preludio2/nokis"
xmlns:ns11="http://www.w3.org/2001/SMIL20/Language" resultType="hits"
outputSchema="http://www.opengis.net/cat/csw/2.0.2" startPosition="1"
maxRecords="10" service="CSW" version="2.0.2">
  <ns7:Query xmlns:ns12="http://www.opengis.net/gml" xmlns=""
typeName="preludio:metadata">
    <ns7:ElementSetName>full</ns7:ElementSetName>
    <ns7:Constraint version="1.1.0">
      <ns6:Filter>
        <ns6:And>
          <ns6:PropertyIsLike wildCard="*" singleChar="?" escapeChar="\ ">
            <ns6:PropertyName>AnyText</ns6:PropertyName>
            <ns6:Literal>*</ns6:Literal>
          </ns6:PropertyIsLike>
        </ns6:And>
      </ns6:Filter>
    </ns7:Constraint>
    <ns6:SortBy>
      <ns6:SortProperty>
        <ns6:PropertyName>base:dateStamp</ns6:PropertyName>
        <ns6:SortOrder>DESC</ns6:SortOrder>
      </ns6:SortProperty>
    </ns6:SortBy>
  </ns7:Query>
</ns7:GetRecords>
```

Code:

```
// instantiate the condition object and specify the conditions
CSWQueryCondition qc = new CSWQueryCondition();
qc.addLiteralCondition("*", MetadataField.AnyText);
qc.setResultType(ResultType.Hits);
qc.addTypeName(TypeName.Metadata_202);

// send the GetRecords request with the specified conditions
CSWQueryResult queryresult = hzgcswclient.QueryMetadatabyCondition(qc);
```

### 2.2. Results request

Generated request document:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns7:GetRecords xmlns="http://www.opengis.net/gml"
xmlns:ns2="http://www.w3.org/1999/xlink" xmlns:ns3="http://www.w3.org/2001/SMIL20/"
xmlns:ns4="http://purl.org/dc/elements/1.1/" xmlns:ns5="http://purl.org/dc/terms/"
xmlns:ns6="http://www.opengis.net/ogc"
xmlns:ns7="http://www.opengis.net/cat/csw/2.0.2"
xmlns:ns8="http://www.opengis.net/ows"
```

```

xmlns:ns9="http://www.disy.de/preludio2/base"
xmlns:preludio="http://www.disy.de/preludio2/nokis"
xmlns:ns11="http://www.w3.org/2001/SMIL20/Language" resultType="results"
outputSchema="http://www.opengis.net/cat/csw/2.0.2" startPosition="1"
maxRecords="10" service="CSW" version="2.0.2">
  <ns7:Query xmlns:ns12="http://www.opengis.net/gml" xmlns=""
typeNames="preludio:metadata">
    <ns7:ElementSetName>full</ns7:ElementSetName>
    <ns7:Constraint version="1.1.0">
      <ns6:Filter>
        <ns6:And>
          <ns6:PropertyIsLike wildCard="*" singleChar="?" escapeChar="\ ">
            <ns6:PropertyName>AnyText</ns6:PropertyName>
            <ns6:Literal>*</ns6:Literal>
          </ns6:PropertyIsLike>
        </ns6:And>
      </ns6:Filter>
    </ns7:Constraint>
    <ns6:SortBy>
      <ns6:SortProperty>
        <ns6:PropertyName>base:dateStamp</ns6:PropertyName>
        <ns6:SortOrder>DESC</ns6:SortOrder>
      </ns6:SortProperty>
    </ns6:SortBy>
  </ns7:Query>
</ns7:GetRecords>

```

**Code:**

```

// instantiate the condition object and specify the conditions
CSWQueryCondition qc = new CSWQueryCondition();
qc.addLiteralCondition("*", MetadataField.AnyText);
qc.setElementset(ElementSet.Full);
qc.setResulttype(ResultType.Results);
qc.addType_name(TypeName.Metadata_202);
qc.setMaxresult(10);

// send the GetRecords request with the specified conditions
CSWQueryResult queryresult = hzgcswclient.QueryMetadatabyCondition(qc);

```

### 3. GetRecordById request

#### Generated request document:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns7:GetRecordById xmlns="http://www.opengis.net/gml"
xmlns:ns2="http://www.w3.org/1999/xlink" xmlns:ns3="http://www.w3.org/2001/SMIL20/"
xmlns:ns4="http://purl.org/dc/elements/1.1/" xmlns:ns5="http://purl.org/dc/terms/"
xmlns:ns6="http://www.opengis.net/ogc"
xmlns:ns7="http://www.opengis.net/cat/csw/2.0.2"
xmlns:ns8="http://www.opengis.net/ows"
xmlns:ns9="http://www.disy.de/preludio2/base"
xmlns:preludio="http://www.disy.de/preludio2/nokis"
xmlns:ns11="http://www.w3.org/2001/SMIL20/Language"
outputSchema="http://www.disy.de/preludio2/nokis" service="CSW" version="2.0.2">
  <ns7:Id>ff8081812a4df559012a6a594a6f00de</ns7:Id>
  <ns7:Id>4028478a211aa41201212e805a8e00cd</ns7:Id>
  <ns7:ElementSetName>full</ns7:ElementSetName>
</ns7:GetRecordById>
```

#### Code:

```
// instantiate the array of UUIDs
String[] uuids = new String[]{"ff8081812a4df559012a6a594a6f00de",
"4028478a211aa41201212e805a8e00cd"};

// send the GetRecordById request
CSWQueryResult queryresult = hzgcswclient.QueryMetadatabyId(uuids);
```

## 4. GetRecords request

### 4.1. Request with literal condition

Generated request document:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns7:GetRecords xmlns="http://www.opengis.net/gml"
xmlns:ns2="http://www.w3.org/1999/xlink" xmlns:ns3="http://www.w3.org/2001/SMIL20/"
xmlns:ns4="http://purl.org/dc/elements/1.1/" xmlns:ns5="http://purl.org/dc/terms/"
xmlns:ns6="http://www.opengis.net/ogc"
xmlns:ns7="http://www.opengis.net/cat/csw/2.0.2"
xmlns:ns8="http://www.opengis.net/ows"
xmlns:ns9="http://www.disy.de/preludio2/base"
xmlns:preludio="http://www.disy.de/preludio2/nokis"
xmlns:ns11="http://www.w3.org/2001/SMIL20/Language" resultType="results"
outputSchema="http://www.opengis.net/cat/csw/2.0.2" startPosition="1"
maxRecords="10" service="CSW" version="2.0.2">
  <ns7:Query xmlns:ns12="http://www.opengis.net/gml" xmlns=""
typeName="preludio:metadata">
    <ns7:ElementSetName>full</ns7:ElementSetName>
    <ns7:Constraint version="1.1.0">
      <ns6:Filter>
        <ns6:And>
          <ns6:PropertyIsLike wildCard="*" singleChar="?" escapeChar="\ ">
            <ns6:PropertyName>AnyText</ns6:PropertyName>
            <ns6:Literal>elbe</ns6:Literal>
          </ns6:PropertyIsLike>
        </ns6:And>
      </ns6:Filter>
    </ns7:Constraint>
    <ns6:SortBy>
      <ns6:SortProperty>
        <ns6:PropertyName>base:dateStamp</ns6:PropertyName>
        <ns6:SortOrder>DESC</ns6:SortOrder>
      </ns6:SortProperty>
    </ns6:SortBy>
  </ns7:Query>
</ns7:GetRecords>
```

Code:

```
// instantiate the condition object and specify the conditions
CSWQueryCondition qc = new CSWQueryCondition();
qc.addLiteralCondition("elbe", MetadataField.AnyText);
qc.setElementset(ElementSet.Full);
qc.setResultType(ResultType.Results);
qc.addTypename(TypeName.Metadata_202);
qc.setMaxresult(10);

// send the GetRecords request with the specified conditions
CSWQueryResult queryresult = hzgcswclient.QueryMetadataByCondition(qc);
```

### 4.2. Request with literal and temporal conditions

Generated request document:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns7:GetRecords xmlns="http://www.opengis.net/gml"
xmlns:ns2="http://www.w3.org/1999/xlink" xmlns:ns3="http://www.w3.org/2001/SMIL20/"
xmlns:ns4="http://purl.org/dc/elements/1.1/" xmlns:ns5="http://purl.org/dc/terms/"
xmlns:ns6="http://www.opengis.net/ogc"
xmlns:ns7="http://www.opengis.net/cat/csw/2.0.2"
```



```

xmlns:ns8="http://www.opengis.net/ows"
xmlns:ns9="http://www.disy.de/preludio2/base"
xmlns:preludio="http://www.disy.de/preludio2/nokis"
xmlns:ns11="http://www.w3.org/2001/SMIL20/Language" resultType="results"
outputSchema="http://www.opengis.net/cat/csw/2.0.2" startPosition="1"
maxRecords="10" service="CSW" version="2.0.2">
  <ns7:Query xmlns:ns12="http://www.opengis.net/gml" xmlns=""
typeNames="preludio:metadata">
    <ns7:ElementSetName>full</ns7:ElementSetName>
    <ns7:Constraint version="1.1.0">
      <ns6:Filter>
        <ns6:And>
          <ns6:PropertyIsBetween>
            <ns6:PropertyName>TemporalExtent</ns6:PropertyName>
            <ns6:LowerBoundary>
              <ns6:Literal>2001-01-01</ns6:Literal>
            </ns6:LowerBoundary>
            <ns6:UpperBoundary>
              <ns6:Literal>2010-12-01</ns6:Literal>
            </ns6:UpperBoundary>
          </ns6:PropertyIsBetween>
          <ns6:PropertyIsLike wildCard="*" singleChar="?" escapeChar="\ ">
            <ns6:PropertyName>AnyText</ns6:PropertyName>
            <ns6:Literal>elbe</ns6:Literal>
          </ns6:PropertyIsLike>
        </ns6:And>
      </ns6:Filter>
    </ns7:Constraint>
    <ns6:SortBy>
      <ns6:SortProperty>
        <ns6:PropertyName>base:dateStamp</ns6:PropertyName>
        <ns6:SortOrder>DESC</ns6:SortOrder>
      </ns6:SortProperty>
    </ns6:SortBy>
  </ns7:Query>
</ns7:GetRecords>

```

#### Code:

```

// instantiate the condition object and specify the conditions
CSWQueryCondition qc = new CSWQueryCondition();
qc.addLiteralCondition("elbe", MetadataField.AnyText);
qc.setElementset (ElementSet.Full);
qc.setResulttype (ResultType.Results);
qc.setLogicalOperator (CSWQueryCondition.AND);
qc.addTypename (TypeName.Metadata_202);
qc.setMaxresult (10);

// add temporal condition to the condition object
SimpleDateFormat sdf = new SimpleDateFormat("dd.MM.yyyy");
qc.addTemporalCondition(sdf.parse("01.01.2001"), sdf.parse("01.12.2010"));

// send the GetRecords request with the specified conditions
CSWQueryResult queryresult = hzgcswclient.QueryMetadatabyCondition(qc);

```

### 4.3. Request with literal and spatial conditions

#### Generated request document:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns7:GetRecords xmlns="http://www.opengis.net/gml"
xmlns:ns2="http://www.w3.org/1999/xlink" xmlns:ns3="http://www.w3.org/2001/SMIL20/"
xmlns:ns4="http://purl.org/dc/elements/1.1/" xmlns:ns5="http://purl.org/dc/terms/"
xmlns:ns6="http://www.opengis.net/ogc"

```

```

xmlns:ns7="http://www.opengis.net/cat/csw/2.0.2"
xmlns:ns8="http://www.opengis.net/ows"
xmlns:ns9="http://www.disy.de/preludio2/base"
xmlns:preludio="http://www.disy.de/preludio2/nokis"
xmlns:ns11="http://www.w3.org/2001/SMIL20/Language" resultType="results"
outputSchema="http://www.opengis.net/cat/csw/2.0.2" startPosition="1"
maxRecords="10" service="CSW" version="2.0.2">
  <ns7:Query xmlns:ns12="http://www.opengis.net/gml" xmlns=""
typeNames="preludio:metadata">
    <ns7:ElementSetName>full</ns7:ElementSetName>
    <ns7:Constraint version="1.1.0">
      <ns6:Filter>
        <ns6:And>
          <ns6:BBOX>
            <ns6:PropertyName>Envelope</ns6:PropertyName>
            <ns12:Envelope>
              <ns12:lowerCorner>7.63 51.93</ns12:lowerCorner>
              <ns12:upperCorner>9.63 55.93</ns12:upperCorner>
            </ns12:Envelope>
          </ns6:BBOX>
          <ns6:PropertyIsLike wildCard="*" singleChar="?" escapeChar="\ ">
            <ns6:PropertyName>AnyText</ns6:PropertyName>
            <ns6:Literal>elbe</ns6:Literal>
          </ns6:PropertyIsLike>
        </ns6:And>
      </ns6:Filter>
    </ns7:Constraint>
    <ns6:SortBy>
      <ns6:SortProperty>
        <ns6:PropertyName>base:dateStamp</ns6:PropertyName>
        <ns6:SortOrder>DESC</ns6:SortOrder>
      </ns6:SortProperty>
    </ns6:SortBy>
  </ns7:Query>
</ns7:GetRecords>

```

### Code:

```

// instantiate the condition object and specify the conditions
CSWQueryCondition qc = new CSWQueryCondition();
qc.addLiteralCondition("elbe", MetadataField.AnyText);
qc.setElementset(ElementSet.Full);
qc.setResulttype(ResultType.Results);
qc.setLogicalOperator(CSWQueryCondition.AND);
qc.addTypename(TypeName.Metadata_202);
qc.setMaxresult(10);

// add spatial condition to the condition object
double north = 55.93;
double south = 51.93;
double east = 9.63;
double west = 7.63;
qc.addSpatialCondition(north, south, east, west);

// send the GetRecords request with the specified conditions
CSWQueryResult queryresult = hzgcswclient.QueryMetadatabyCondition(qc);

```

## 5. Transaction requests

### 5.1. Insert

Generated request document:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns7:Transaction xmlns="http://www.opengis.net/gml"
xmlns:ns2="http://www.w3.org/1999/xlink" xmlns:ns3="http://www.w3.org/2001/SMIL20/"
xmlns:ns4="http://purl.org/dc/elements/1.1/" xmlns:ns5="http://purl.org/dc/terms/"
xmlns:ns6="http://www.opengis.net/ogc"
xmlns:ns7="http://www.opengis.net/cat/csw/2.0.2"
xmlns:ns8="http://www.opengis.net/ows"
xmlns:ns9="http://www.disy.de/preludio2/base"
xmlns:preludio="http://www.disy.de/preludio2/nokis"
xmlns:ns11="http://www.w3.org/2001/SMIL20/Language" service="CSW" version="2.0.2">
  <ns7:Insert>
    <ns10:metadata xmlns:ns10="http://www.disy.de/preludio2/nokis"
xmlns:ns1="http://www.opengis.net/gml" uuid="4028478a20f2f1b101210f7d84c5085e">
      ...
    </ns10:metadata>
  </ns7:Insert>
</ns7:Transaction>
```

Code:

```
// construct the metadata record
MDMeta data md = ...;

// send the Insert request
CSWInsertOrUpdateOrDeleteResult result = hzgcswclient.insertMetadata(md);
```

### 5.2. Update

Generated request document:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns7:Transaction xmlns="http://www.opengis.net/gml"
xmlns:ns2="http://www.w3.org/1999/xlink" xmlns:ns3="http://www.w3.org/2001/SMIL20/"
xmlns:ns4="http://purl.org/dc/elements/1.1/" xmlns:ns5="http://purl.org/dc/terms/"
xmlns:ns6="http://www.opengis.net/ogc"
xmlns:ns7="http://www.opengis.net/cat/csw/2.0.2"
xmlns:ns8="http://www.opengis.net/ows"
xmlns:ns9="http://www.disy.de/preludio2/base"
xmlns:preludio="http://www.disy.de/preludio2/nokis"
xmlns:ns11="http://www.w3.org/2001/SMIL20/Language" service="CSW" version="2.0.2">
  <ns7:Update>
    <ns7:RecordProperty>
      <ns7:Name>ns9:status</ns7:Name>
      <ns7:Value xmlns:ns12="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ns13="http://www.w3.org/2001/XMLSchema"
ns12:type="ns13:string">sharedExternally</ns7:Value>
    </ns7:RecordProperty>
    <ns7:Constraint version="1.1.0">
      <ns6:Filter>
        <ns6:GmlObjectId id="4028478a211aa41201212e745ce60089"/>
      </ns6:Filter>
    </ns7:Constraint>
  </ns7:Update>
</ns7:Transaction>
```

Code:

```
// instantiate the UUID string
String uuid = "4028478a200e518f0120422a0b800cff";

// send the Update request
CSWInsertOrUpdateOrDeleteResult result = hzgcswclient.updateRecordStatus(uuid,
RecordStatus.external);
```

### 5.3. Delete

#### Generated request document:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns7:Transaction xmlns="http://www.opengis.net/gml"
xmlns:ns2="http://www.w3.org/1999/xlink" xmlns:ns3="http://www.w3.org/2001/SMIL20/"
xmlns:ns4="http://purl.org/dc/elements/1.1/" xmlns:ns5="http://purl.org/dc/terms/"
xmlns:ns6="http://www.opengis.net/ogc"
xmlns:ns7="http://www.opengis.net/cat/csw/2.0.2"
xmlns:ns8="http://www.opengis.net/ows"
xmlns:ns9="http://www.disy.de/preludio2/base"
xmlns:preludio="http://www.disy.de/preludio2/nokis"
xmlns:ns11="http://www.w3.org/2001/SMIL20/Language" service="CSW" version="2.0.2">
  <ns7:Delete>
    <ns7:Constraint version="1.1.0">
      <ns6:Filter>
        <ns6:GmlObjectId id="4028478a200e518f01204200592d0c55"/>
      </ns6:Filter>
    </ns7:Constraint>
  </ns7:Delete>
</ns7:Transaction>
```

#### Code:

```
// instantiate the UUID string
String uuid = "4028478a20f2f1b101210f7d84c5085e";
// send the Delete request
CSWInsertOrUpdateOrDeleteResult result = hzgcswclient.deleteRecord(uuid);
```