

Contents

1	Introduction	2
2	Requirements.....	2
3	Overview of API Operation	2
4	Client Access Credentials	3
5	Request Overview	3
5.1	General Request Information	3
5.2	Request Responses	4
6	Authorization Requests.....	4
6.1	Token Request	4
6.2	Token Refresh Request	5
7	Data Requests	6
7.1	Data Request Methods	6
7.2	Data Request URL	6
7.3	Authorisation Header.....	6
7.4	Data Request Responses.....	6
7.5	Access Rights.....	6
8	Point Data Read Requests.....	7
8.1	List Points	7
8.2	Reading Point Data.....	7
9	Point Data Write Requests.....	8
9.1	Write Point Method	8
9.2	Write Point Slots	8
9.3	Write Point URL.....	8
9.4	Write Point Type	8
10	History Data Read Requests.....	9
10.1	List Histories	9
10.2	Reading History Data	9
10.3	History Time Range	10
11	Appendix	12
11.1	End User License Agreement	12

1 Introduction

The One Sight Solutions Niagara ossRestApiServer software provides a way for third party applications to read and write point data, and read history data via HTTP API requests to a Niagara station.

It provides an implementation of OAuth 2.0 token authentication to make this data transfer secure.

This guide describes how to access the API from an external computer.

There is a separate document that describes how to install and configure the software in Niagara.

2 Requirements

It is assumed that the reader is familiar with the use of web APIs.

The ossRestApiServer software must have been installed on a Niagara host and a client application must have been created.

You must read the [End User License Agreement](#) at the end of this document before using this software.

3 Overview of API Operation

A simple overview of the API interaction is shown in Figure 1.

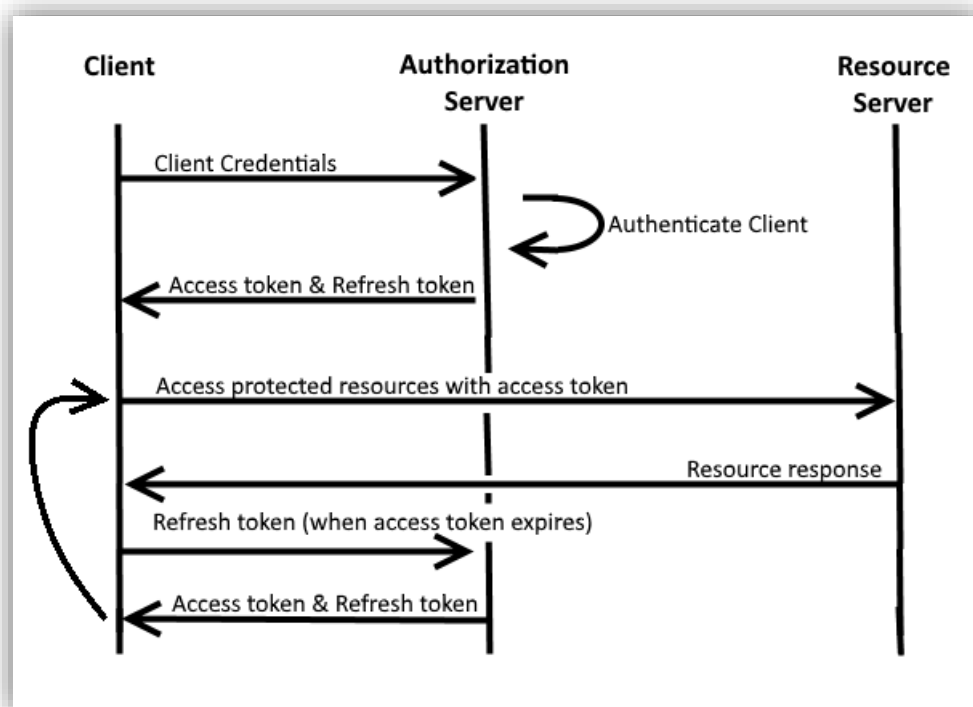


Figure 1: API Interaction Overview

For each API client the ossRestApiServer will generate a set of client credentials.

These pieces of information are transferred to the client application manually (e.g. cut and paste).

The client application will then use those credentials when making a token request to the API, the ossRestApiServer software will then generate an access token and a refresh token and send them back to the client.

The client may then request data from the API using the access token for authorisation.

The access token is only valid for a limited period of time, when this time expires the client may use the refresh token to obtain a new access token and then continue making data requests using the new access token.

Most installations will only allow the client application credentials to be used once to generate the access tokens. Then only the tokens can be used and must be refreshed when they expire.

4 Client Access Credentials

The ossRestApiServer software implements the OAuth Client Credentials grant.

To access the API you will need the client credentials for the application you will be using.

These values will have been created by the person who set up the ossRestApiServer application on the Niagara host you will be connecting to.

The client credentials consist of a client identifier and a client secret which are both alphanumeric values which can be cut and pasted into emails and documents.

Client credentials should be kept secret and transferred from the ossRestApiServer application to the client accessing the API in a secure manner.

Note that the client identifier and client secret are each single line values. Any line breaks in these values should be removed. Care should be taken if these values are transferred between non-windows systems as line break characters can differ between these systems and values can get corrupted. Contact the person who set up the ossRestApiServer application if you receive client credential values that do not consist of single line values and are unsure how to reformat them.

5 Request Overview

5.1 General Request Information

Information is read and written to the ossRestApiServer API using HTTP requests.

All HTTP requests to the API must use the secure extension HTTPS (i.e. the URL should start with 'https://' scheme), so any client must be capable of supporting HTTPS.

The person who set up the ossRestApiServer application should be able to tell you the hostname or IP address of the host and the port number to use for the HTTPS requests. Note that the port number will often not be specified if the default HTTPS port number of 443 is being used.

All request types use a URL of the format:

`https://<hostname>[:<port-number>]/ossRestApiServer/<requestPath>`

The HTTP authorisation headers of each request must be assigned correctly for the access to succeed, as described in the following sections for each type of request.

5.2 Request Responses

Most API requests will result in one of three responses as described in Table 1.

HTTP Status Code	Response	Description
200	OK	The request succeeded.
400	Bad Request	The request was invalid.
401	Unauthorized	The request authorisation was invalid, the client may need to refresh the access token.

Table 1: Request Responses

By default request failure responses will only return basic fault information, for security reasons.

The ossRestApiServer software can be configured in Niagara to return more informative debug information in API failure responses. This can be useful during initial development of systems integrating with the API. It is recommended that this debug information is disabled when it is no longer required.

6 Authorization Requests

Authorisation requests are used to obtain and update the access token required for API data requests.

6.1 Token Request

6.1.1 Token Request URL

Requests to obtain access tokens from the API use the HTTP **POST** request method.

The token request URL has the format:

`https://<hostname>/ossRestApiServer/token`

Specific HTTP headers must also be included in the token request as well as body data, which are described in the following sections.

6.1.2 Authorization Header

The authorisation header for the token request must contain the client identifier (as user name) and client secret (as password) [base64](#) encoded according to the Basic Access Authentication (BA) mechanism (as defined in [RFC 7617](#)).

This involves creating a raw credentials text value in the following format:

`<client-identifier>:<client-secret>`

This is the client identifier text followed by a colon character (:) followed by the client secret text.

This raw credentials value is then [base64](#) encoded to create the encoded credentials.

The encoded credentials are then included in the HTTP header in the basic authorisation field as follows:

`Authorization: Basic <encoded-credentials>`

6.1.3 Content Type Header

The content type header for the token request must be as follows:

Content-Type: application/x-www-form-urlencoded

6.1.4 Request Body

As the token request uses the HTTP POST method it must contain data in the body in the following format:

grant_type=client_credentials&scope=<access-scope>

Where access-scope is one of the following according to what type of access the client requires:

- read
- read write

Note the space between the words read and write in the second option.

6.1.5 Token Request Response

The response returned by the token request is JSON data containing the three fields described in Table 2.

<u>JSON Response Field</u>	<u>Description</u>
access_token	The token value to use for API access requests.
refresh_token	The token to use to refresh the access token when it expires.
expires_in	The number of milliseconds the access token is valid for.

Table 2: Token Request Response Fields

The following is an example response:

```
{
  "access_token": "780bd378e051391e8728819826ac3f7d",
  "refresh_token": "5c221b522cd970230b08c515bccd3391",
  "expires_in": 3600000
}
```

6.2 Token Refresh Request

6.2.1 Token Refresh Request URL

Requests to refresh the access token use the HTTP **POST** request method.

The token request URL has the format:

https://<hostname>/ossRestApiServer/token/renew

Specific HTTP headers must also be included in the token request as well as body data, which are described in the following sections.

6.2.2 Authorization Header

The authorisation header for the token refresh request must contain the refresh token (from the previous token request) as a bearer token.

Therefore, the authorisation field should be as follows:

Authorization: Bearer <refresh-token>

6.2.3 Request Body

The token refresh request uses the same body data and content type header as the original token request, as described in sections 6.1.4 and 6.1.3.

6.2.4 Token Request Response

The response returned by the token refresh request is the same as that returned for the initial token request, as described in section 6.1.5.

7 Data Requests

7.1 Data Request Methods

All data read requests use the HTTP GET method and all data write requests use the HTTP PUT method.

7.2 Data Request URL

All data requests use a URL of the following format:

https://<hostname>/ossRestApiServer/query/<resource-path>

The resource path required for each request type is described in the following sections.

7.3 Authorisation Header

All data requests require a HTTP authorisation header field that contains a valid access token in the following format:

Authorization: Bearer <access-token>

7.4 Data Request Responses

All valid request responses return JSON data.

The JSON data may contain an array or a single JSON object depending on the type of request.

7.5 Access Rights

Note that access to both point and history resources are controlled by licensing and a user access rights system in Niagara.

Therefore, the request may fail if the Niagara station or the ossRestApiServer software is not configured correctly to access the specified resource.

8 Point Data Read Requests

8.1 List Points

This request will return a list of all accessible points on the Niagara host.

8.1.1 List Points Request URL

`https://<hostname>/ossRestApiServer/query/listPoints`

8.1.2 List Points Response Data Format

The request returns a JSON array of strings which contain the point slot path values.

For example:

```
[
  "/Folder1/BooleanTestPoint",
  "/Folder1/NumericTestPoint",
  "/Folder2/enum_point"
]
```

8.2 Reading Point Data

This request will return the value of a single point.

8.2.1 Point Read Request URL

`https://<hostname>/ossRestApiServer/query/readPoint/<slotPath>`

8.2.2 Point Read Response Data Format

The request returns a single JSON object which contains the point value and status using the keys and values described in Table 3.

JSON Key	Value Description
value	A string containing the point value.
status	A string containing a comma separated list of status flag identifiers that apply to the point.

Table 3: Read Point Response Keys

Refer to the Niagara documentation for a list of the status flag identifier values and their meaning.

For example:

```
{
  "value": "70.00",
  "status": "ok"
}
```

9 Point Data Write Requests

9.1 Write Point Method

Requests to write data to the API use the HTTP PUT method.

9.2 Write Point Slots

Points in Niagara have a list of priority slots with values 1 to 16.

When writing to a point the client must specify which point slot is to be written to.

Please read the Niagara documentation or contact the ossRestApiServer administrator for more information regarding what slot to select for writing.

9.3 Write Point URL

The URL to use to write to a slot has the following format:

`https://<hostname>/ossRestApiServer/query/writePoint/<write-spec>`

Where the **<write-spec>** field has the following structure:

`<slotPath>?value=<value>&priority=<priority>`

The write-spec parameters are described in Table 4.

<u>Field Name</u>	<u>Description</u>
slotPath	The point to write to.
value	The value to write.
priority	The slot to write to (1 to 16).

Table 4: Write Point Fields

For example:

```
.../query/writePoint/Heating/setpoint?value=22.3&priority=5
```

9.4 Write Point Type

The API write request supports writing to Boolean, Numeric, Enum or String point types.

You will receive a bad request (400) error response if you attempt to write an invalid type value to a point.

Boolean point values should be specified using the values **'true'** or **'false'**.

Enum point values should be specified using a numeric integer ordinal value.

10 History Data Read Requests

10.1 List Histories

This request will return a list of all accessible histories on the Niagara host.

10.1.1 List Histories Request URL

`https://<hostname>/ossRestApiServer/query/listHistories`

10.1.2 List Histories Response Data Format

The request returns a JSON array of strings which contain the history identifier values.

Note that the history identifier value will always start with the name of the Niagara station.

For example:

```
[  
  "/TestStation/officeTemp",  
  "/TestStation/officeCo2",  
  "/TestStation/elecMeter"  
]
```

10.2 Reading History Data

This request will return an array of history record values.

10.2.1 History Request URL

`https://<hostname>/ossRestApiServer/query/history/<history-id>`

Where the 'history-id' field is of the form shown in the list of histories (i.e. starting with the station name).

10.2.2 History Response Data Format

The request returns a JSON array of arrays, where each secondary array represent a history record.

Each history record array contains three strings as described in Table 5.

Array Index	Value Description
time	The time of the history record. This time is specified as the number of milliseconds since the epoch (midnight, January 1, 1970 UTC).
value	The history record data value.
status	A string containing a comma separated list of status flag identifiers for the history record.

Table 5: Standard History Response Keys

Refer to the Niagara documentation for a list of the status flag identifier values and their meaning.

For example:

```
[
  ["1564389180012", "40.04", "ok"],
  ["1564389240015", "0.0", "fault"],
  ["1564389300019", "155.82", "ok"],
  ["1564389360026", "128.38", "ok"]
]
```

10.3 History Time Range

History requests support specifying a time period, either using a keyword or by specifying the absolute begin and/or end times.

A time period is specified by adding parameters to the end of the URL

10.3.1 History Time Range Keyword

A history time period may be specified by using the '**range**' parameter to define the time range keyword.

The valid time range keywords are shown in the following list:

- TODAY
- YESTERDAY
- WEKTODATE
- LASTWEEK
- LAST7DAYS
- MONTHTODATE
- LASTMONTH
- YEARTODATE
- LASTYEAR

For example:

```
.../ossRestApiServer/query/history/Station/officeTemp?range=YESTERDAY
```

10.3.2 History Time Range Absolute Begin/End

A history time period may be specified by defining the absolute begin and/or end times for the data required.

The begin time can be specified using the '**begin**' parameter.

The end time can be specified using the '**end**' parameter.

Both time parameters must use the following ([ISO 8601](#)) format:

yyyy-MM-ddThh:mm:ss.fff

Note that the 'T' character appears literally as shown, but the other fields should be given numeric values as described in Table 6.

<u>Time Field</u>	<u>Description</u>
yyyy	Four digit year
MM	Two digit month (01-12)
dd	Two digit day of the month (01-31)
hh	Two digit hour, 24 hour clock (00-23)
mm	Two digit minute (00-59)
ss	Two digit second (00-59)
fff	Up to three digit fraction of a second

Table 6: History Time Field Descriptions

For example:

```
.../query/history/Floor2/elecMeter?begin=2019-03-21T15:34:51.713Z&end=2019-03-23T20:45:11.371Z
```

11 Appendix

11.1 End User License Agreement

11.1.1 License

By receiving, opening the file package, and/or using ossRestApiServer ("Software") containing this software, you agree that this End User License Agreement (EULA) is a legally binding and valid contract and agree to be bound by it. You agree to abide by the intellectual property laws and all of the terms and conditions of this Agreement.

Unless you have a different license agreement signed by One SightSolutions Ltd your use of ossRestApiServer indicates your acceptance of this license agreement and warranty.

Subject to the terms of this Agreement, One SightSolutions Ltd grants to you a limited, non-exclusive, non-transferable license, without right to sub-license, to use ossRestApiServer in accordance with this Agreement and any other written agreement with One SightSolutions Ltd. One SightSolutions Ltd does not transfer the title of ossRestApiServer to you; the license granted to you is not a sale. This agreement is a binding legal agreement between One SightSolutions Ltd and the purchasers or users of ossRestApiServer.

If you do not agree to be bound by this agreement, remove ossRestApiServer from your computer now and, if applicable, promptly return to One SightSolutions Ltd by mail any copies of ossRestApiServer and related documentation and packaging in your possession.

11.1.2 Distribution

ossEasyAPI and the license herein granted shall not be copied, shared, distributed, re-sold, offered for re-sale, transferred or sub-licensed in whole or in part except that you may make one copy for archive purposes only. For information about redistribution of ossRestApiServer contact One SightSolutions Ltd.

11.1.3 User Agreement

11.1.3.1 Use

Your license to use ossRestApiServer is limited to the number of licenses purchased by you. You shall not allow others to use, copy or evaluate copies of ossRestApiServer.

11.1.3.2 Use Restrictions

You shall use ossRestApiServer in compliance with all applicable laws and not for any unlawful purpose. Without limiting the foregoing, use, display or distribution of ossRestApiServer together with material that is pornographic, racist, vulgar, obscene, defamatory, libellous, abusive, promoting hatred, discriminating or displaying prejudice based on religion, ethnic heritage, race, sexual orientation or age is strictly prohibited.

Each licensed copy of ossRestApiServer may be used on one single computer location by one user. Use of ossRestApiServer means that you have loaded, installed, or run ossRestApiServer on a computer or similar device. If you install ossRestApiServer onto a multi-user platform, server or network, each and every individual user of ossRestApiServer must be licensed separately.

You may make one copy of ossRestApiServer for backup purposes, providing you only have one copy installed on one computer being used by one person. Other users may not use your copy of ossRestApiServer. The assignment, sublicense, networking, sale, or distribution of copies of ossRestApiServer are strictly forbidden without the prior written consent of One SightSolutions Ltd. It is a violation of this agreement to assign, sell, share, loan, rent, lease, borrow, network or transfer the use of ossRestApiServer. If any person other than yourself uses ossRestApiServer registered in your name, regardless of whether it is at the same time or different times, then this agreement is being violated and you are responsible for that violation!

11.1.3.3 Copyright Restriction

This Software contains copyrighted material, trade secrets and other proprietary material. You shall not, and shall not attempt to, modify, reverse engineer, disassemble or decompile ossRestApiServer. Nor can you create any derivative works or other works that are based upon or derived from ossRestApiServer in whole or in part.

One SightSolutions Ltd's name, logo and graphics file that represents ossRestApiServer shall not be used in any way to promote products developed with ossRestApiServer. One SightSolutions Ltd retains sole and exclusive ownership of all right, title and interest in and to ossRestApiServer and all Intellectual Property rights relating thereto.

Copyright law and international copyright treaty provisions protect all parts of ossRestApiServer, products and services. No program, code, part, image, audio sample, or text may be copied or used in any way by the user except as intended within the bounds of the single user program. All rights not expressly granted hereunder are reserved for One SightSolutions Ltd.

11.1.3.4 Limitation of Responsibility

You will indemnify, hold harmless, and defend One SightSolutions Ltd, its employees, agents and distributors against any and all claims, proceedings, demand and costs resulting from or in any way connected with your use of One SightSolutions Ltd's Software.

In no event (including, without limitation, in the event of negligence) will One SightSolutions Ltd, its employees, agents or distributors be liable for any consequential, incidental, indirect, special or punitive damages whatsoever (including, without limitation, damages for loss of profits, loss of use, business interruption, loss of information or data, or pecuniary loss), in connection with or arising out of or related to this Agreement, ossRestApiServer or the use or inability to use ossRestApiServer or the furnishing, performance or use of any other matters hereunder whether based upon contract, tort or any other theory including negligence.

One SightSolutions Ltd's entire liability, without exception, is limited to the customers' reimbursement of the purchase price of the Software (maximum being the lesser of the amount paid by you and the suggested retail price as listed by One SightSolutions Ltd) in exchange for the return of the product, all copies, registration papers and manuals, and all materials that constitute a transfer of license from the customer back to One SightSolutions Ltd.

11.1.3.5 Warranties

Except as expressly stated in writing, One SightSolutions Ltd makes no representation or warranties in respect of this Software and expressly excludes all other warranties, expressed or implied, oral or written, including, without limitation, any implied warranties of merchantable quality or fitness for a particular purpose.

11.1.3.6 Governing Law

This Agreement shall be governed by the law of the United Kingdom applicable therein. You hereby irrevocably attorn and submit to the non-exclusive jurisdiction of the courts of United Kingdom therefrom. If any provision shall be considered unlawful, void or otherwise unenforceable, then that provision shall be deemed severable from this License and not affect the validity and enforceability of any other provisions.

11.1.3.7 Termination

Any failure to comply with the terms and conditions of this Agreement will result in automatic and immediate termination of this license. Upon termination of this license granted herein for any reason, you agree to immediately cease use of ossRestApiServer and destroy all copies of ossRestApiServer supplied under this Agreement. The financial obligations incurred by you shall survive the expiration or termination of this license.

11.1.4 Disclaimer Of Warranty

This software and the accompanying files are sold "as is" and without warranties as to performance or merchantability or any other warranties whether expressed or implied. this disclaimer concerns all files generated and edited by ossRestApiServer as well.

11.1.5 Consent Of Use Of Data

You agree that One SightSolutions Ltd may collect and use information gathered in any manner as part of the product support services provided to you, if any, related to ossRestApiServer. One SightSolutions Ltd may also use this information to provide notices to you which may be of use or interest to you.

End of license agreement