

## Contents

1	Introduction .....	2
2	Requirements.....	2
3	Installation .....	2
3.1	Niagara Software Module Installation .....	2
3.2	Niagara Software Licensing.....	2
4	Overview of API Operation .....	3
5	Setting Up ossRestApiServer .....	4
5.1	Install API Service .....	4
5.2	Resource Selection.....	4
5.3	Install API Client Application .....	5
5.4	Generate Client Credentials.....	5
5.5	Resource Access Permissions.....	7
5.6	Oss Rest Api Service And Application Example .....	8
6	ossRestApiServer Reference .....	9
6.1	OssRestApiService Enabled .....	9
6.2	Resources .....	9
6.3	Token Expiry Time.....	9
6.4	Block Repeated Failed Auth Attempts .....	9
6.5	Max Failed Auth Attempts .....	9
6.6	Block Credential Reuse .....	9
7	OssApiClientApplication Reference .....	10
7.1	Application Name.....	10
7.2	Enabled .....	10
7.3	Authorized.....	10
7.4	Debug Enabled .....	10
7.5	Failed Auth Attempts .....	10
7.6	Read Access User .....	10
7.7	Write Access User .....	10
7.8	Client Credentials .....	10
7.9	Auth Grant.....	11
8	Usage Notes .....	12
8.1	Token Storage .....	12
9	Appendix .....	12
9.1	<b>End User License Agreement .....</b>	<b>12</b>

## 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 RESTful 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 install and configure the software in Niagara.

There is a separate document that describes how to access the API from another computer.

## 2 Requirements

It is assumed that the user is trained in the use of Niagara 4 and is familiar with the use of web APIs.

The ossRestApiServer software is only compatible with Niagara versions 4.8 or later.

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

## 3 Installation

### 3.1 Niagara Software Module Installation

Before the ossRestApiServer software can be used, the Niagara software modules must be installed on the user's PC and the target host.

The following files supplied by One Sight Solutions must be installed on the host using the Niagara platform software manager.

- ossRestApiServer-rt.jar
- ossRestApiServer-wb.jar

These software files must also be installed in the appropriate Niagara modules directory of any PC on which you intend to access the ossRestApiServer software.

### 3.2 Niagara Software Licensing

The host running the ossRestApiServer software must be licensed by One Sight Solutions before it will operate correctly.

The following information is required to generate the ossRestApiServer license:

- Niagara host identifier (e.g. Qnx-TITAN-0000-XXXX-XXXX).
- Niagara host serial number.

Please send this information to [One Sight Solutions: info@onesight.solutions](mailto:info@onesight.solutions)

It is recommended that the license is installed on the Niagara host using the platform license manager import capability to download the license directly from the Niagara license server over the internet.

If it is not possible for the host to download the license from the internet then One Sight Solutions can supply the required license files which can then be installed manually.

Ensure that the One Sight Solutions certificate is also installed, so the Niagara platform license manager should show both of the following files:

- OneSightSolutions.license
- OneSightSolutions.certificate

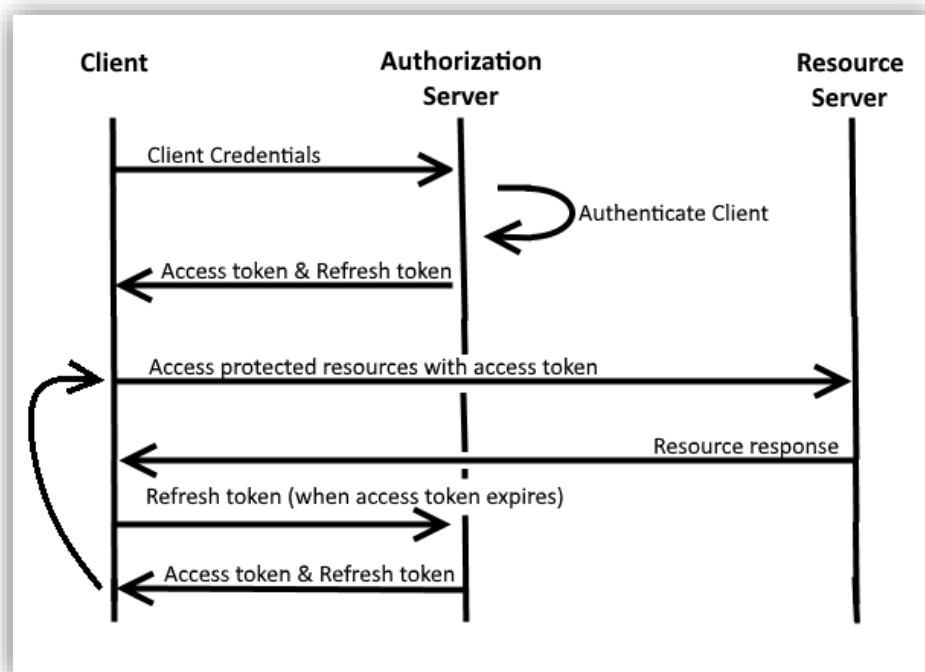
The host should reboot during the license installation process.

## 4 Overview of API Operation

It is useful for the person configuring the ossRestApiServer software to have a high-level understanding of how the API functions so they can make the correct choice of configuration values.

While it is not necessary to know the full technical details of the requests and responses involved in the use of the API, understanding the fundamental elements and interactions will help.

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.

## 5 Setting Up ossRestApiServer

### 5.1 Install API Service

First the OssRestApiService must be installed in the station.

This component can be found in the ossRestApiServer palette, as shown in Figure 2.

Drag the OssRestApiService component from the palette into the station services folder.

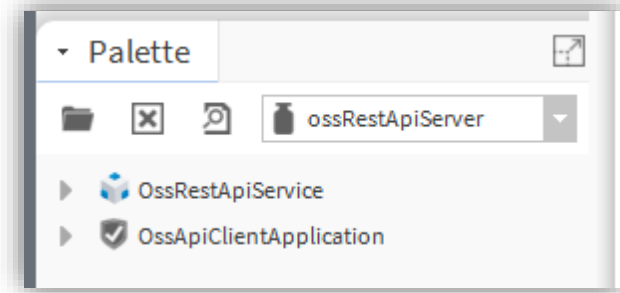


Figure 2: ossRestApiServer Palette

### 5.2 Resource Selection

There are license restrictions on the number of points and histories that can be accessed using the API service.

The point and history resources that will be available must be selected in advance using the Api Resource Manager. Double click on the 'Resources' section of the OssRestApiService to access the Api Resource Manager.

At the bottom of the resource discovery window there are a set of buttons, as shown in Figure 3.



Figure 3: Resource Discovery Buttons

To add points to the list of allocated available resources click the 'Find Points' button.

You will then be prompted to select an item from the station hierarchy which will define the base location from where the points will be searched. If you just click OK then all points on the station will be listed.

To add histories to the list of allocated available resources click the 'Find Histories' button.

Once a list of discovered points or histories has appeared you can add them to the allocated resource list using the usual Niagara methods (double click, drag or using the add button).

Note that duplicate entries will not be created, so if you attempt to add a resource that already exists in the allocated resource list then it will be ignored.

If the resource licence limit is exceeded any additional resources added will show a fault and will not be available for use by the API. All resources in the allocation list that are not in fault will continue to work with the API.

Note that if you exceed the licence limit and then delete some previously allocated resources you can use the 'Refresh' action on the 'Resources' components to reallocate the remaining resources.

The point paths and history identifiers will not update if the source resources are renamed or moved in the station, so you must manually re-discover the resources if they are modified.

### 5.3 Install API Client Application

Many different clients may access the API, each with different credentials and access rights.

Each client must have its own client application installed and configured.

For each client, drag an OssApiClientApplication component from the ossRestApiServer palette and drop it in the OssRestApiService component, as shown in Figure 4.

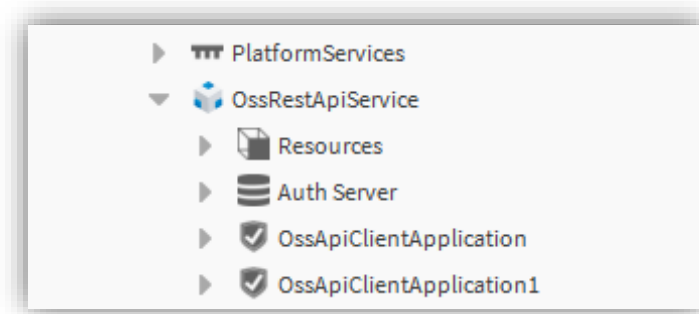


Figure 4: Installed OssRestApiService And Applications

### 5.4 Generate Client Credentials

Once the OssApiClientApplication is installed in the OssRestApiService the access credentials can be generated.

Enter a client name in the 'Application Name' property of the OssApiClientApplication component. This name can contain spaces and is only used to identify the client application.

Then view the 'Client Credentials' property (which has a special 'Generate Client Credentials View' by default).

Click on the 'Generate Credentials' button at the bottom of the view, as shown in Figure 5.

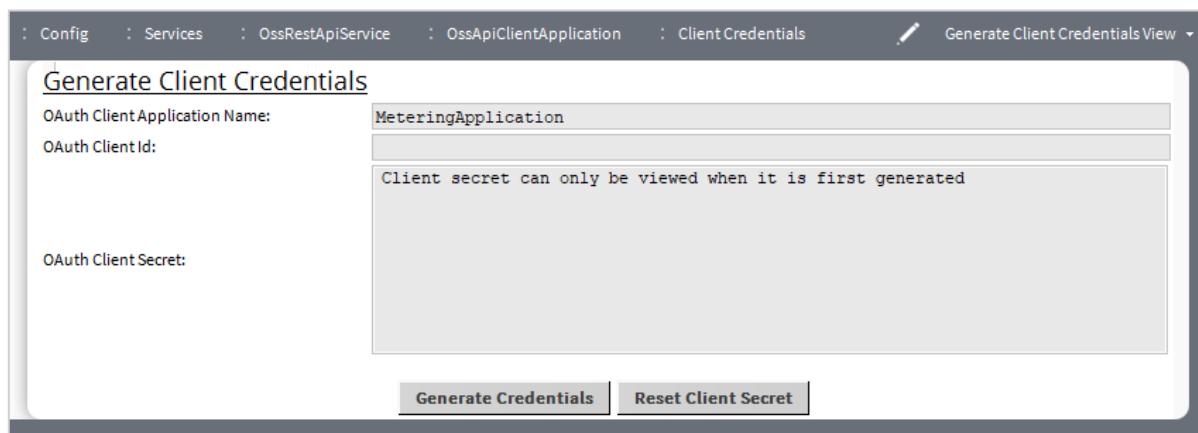


Figure 5: Client Credentials Generation

You will see a warning window indicating that the credentials that are about to be generated will only be temporarily visible and should be copied before continuing (shown in Figure 6).

Click OK on this window.

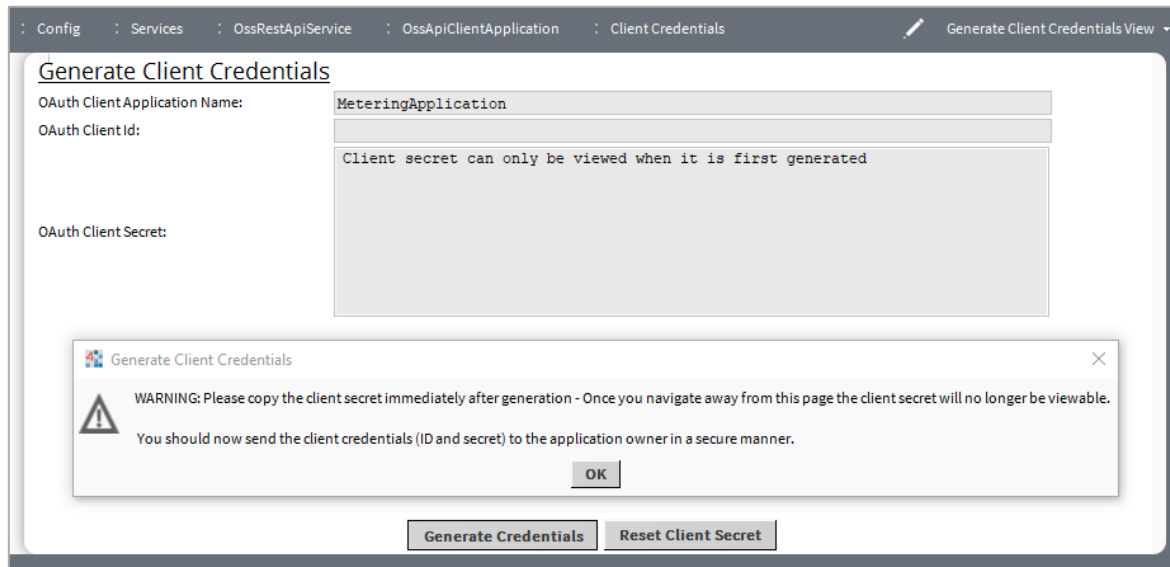


Figure 6: Client Credentials Warning

Text values will now be available in the 'OAuth Client Id' and 'OAuth Client Secret' properties, as shown in Figure 7.



Figure 7: Client Credentials Values

These two values should be copied for use by the external client application that will access the API.

As indicated by the warning window, once the user navigates away from this credentials view the client secret value will no longer be available and will have to be regenerated.

It is important to make sure that the client ID and client secret values are kept private to the ossRestApiServer administrator and the client application that will be accessing the API. Any transmission or storage of these values should be secure.

For additional security the client credentials have a validity period and if they are not used within that period they become invalid and have to be re-generated. The default period is 1 week (168 hours) but can be changed with the ['Valid Time' property](#).

Once the client credentials have been created a new ['AuthGrant' component](#) will be created inside the client application with additional status and configuration properties, (see section 7.9).

## 5.5 Resource Access Permissions

Niagara station points and histories will only be accessible if the application is configured correctly with the correct permissions.

API resource permissions are managed using the Niagara user, role and category mechanisms to assign permissions to a user to be able to read or write to resources. Please read the Niagara documentation for instructions on how to set up these permissions.

These users are then assigned to each application using the 'Read Access User' and 'Write Access User' properties. Note that different users can be created for read and write permissions, or these can both be the same.

The Niagara users created are only used for setting permissions. If not used for other purposes these users can be disabled to increase security. The users passwords are not used by ossRestApiServer.

If write access is required then the 'Write Access' Boolean must also be set to true in the 'Auth Grant' subsection of the application properties.

## 5.6 Oss Rest Api Service And Application Example

An example configuration is shown in Figure 8.

Property Sheet	
OssApiService (Oss Api Service)	
Status	{ok}
Fault Cause	
Enabled	<input checked="" type="radio"/> true
Auth Server	Auth Server
OssApiClientApplication	Oss Api Client Application
Application Name	MeteringApplication
Enabled	<input checked="" type="radio"/> true
Authorized	<input type="radio"/> false
Debug Enabled	<input type="radio"/> false
Failed Auth Attempts	0
Read Access User	MeterUser
Write Access User	ConfigUser
Client Credentials	Oss Api Client Credentials
Client Id	wdK1TfjzWoxkPAJPjiFwpGSIpGYZGjUM
Client Secret	.....
AuthGrant	Oss Api Auth Grant
Authorized	<input type="radio"/> false
Authorization Time	null
Time Generated	27-Aug-2019 03:46 PM BST
Valid Time	+00168h 00m 00s
Write Access	<input checked="" type="radio"/> true
Fault Cause	
Fault Time	27-Aug-2019 03:46 PM BST

Figure 8: Oss Rest Api Service And Application Example



## 6 ossRestApiServer Reference

This section describes all the properties of the OssRestApiService.

### 6.1 OssRestApiService Enabled

If the OssRestApiService Enabled property is set to false then all API requests will be refused with a 400 HTTP status code.

Credentials or access tokens do not have to be regenerated once the service is re-enabled as long as they remain valid.

### 6.2 Resources

This section contains the allocated point and history resources that are allowed to be accessed via the API as defined by the license that has been obtained for the host.

These resources and the resource manager view are described in section 5.2.

### 6.3 Token Expiry Time

The Auth Server Token Expiry Time determines how long the access token will be valid from the time it is generated. Once the access token has expired the refresh token must be used to obtain a new access token.

### 6.4 Block Repeated Failed Auth Attempts

This Auth Server property is used in conjunction with the [‘Max Failed Auth Attempts’](#) property to prevent brute force attacks.

If set to true then the API will refuse all authentication attempts after the maximum number of failures has been reached, even if the correct credentials are used.

If set to false then an unlimited number of failures are allowed and the API will still successfully authenticate if the correct credentials are used.

### 6.5 Max Failed Auth Attempts

This Auth Server property is used in conjunction with the [‘Block Repeated Failed Auth Attempts’](#) property to prevent brute force attacks.

The property value determines the number of failed authentication attempts that are allowed before the API refuses all further authentication attempts, (even if the correct credentials are used).

### 6.6 Block Credential Reuse

By default, the API will only authorise the client credentials once and after that only the access token and refresh tokens can be used to access the API.

If the Auth Server ‘Block Credential Reuse’ property is set to false then the credentials can be re-used.

If the ‘Block Credential Reuse’ property is set to true then any attempts to reuse the client credentials (after the initial authorisation) will result in an unauthorized error response (401).

This feature is mainly used when first developing an API integration solution and for security reasons should be set to true as soon as the API access is working correctly.

## 7 OssApiClientApplication Reference

This section describes all the properties of the OssApiClientApplication.

### 7.1 Application Name

The application name property is required, but is only used to identify the client application.

### 7.2 Enabled

If the OssApiClientApplication Enabled property is set to false then all API requests using the credentials or tokens for this application will be refused with a 400 HTTP status code.

Credentials or access tokens do not have to be regenerated once the service is re-enabled as long as they remain valid.

### 7.3 Authorized

This is a read-only property that indicates if an application in the OssRestApiService has been successfully authorised by a client and therefore has issued tokens.

### 7.4 Debug Enabled

This property controls whether debug information is added to API request failure responses.

This debug information is disabled by default for security reasons.

### 7.5 Failed Auth Attempts

This is a read-only property that indicates the number of failed authentication attempts this API application has received. This can be reset with the 'Reset Failed Auths' action.

This is related to the ['Max Failed Auth Attempts'](#) OssRestApiService property.

### 7.6 Read Access User

All resources accessed by this API application must be readable by the Niagara user specified by the read access user property.

See section 5.5 ['Resource Access Permissions'](#).

### 7.7 Write Access User

All resources written to by this API application must be writable by the Niagara user specified by the write access user property.

See section 5.5 ['Resource Access Permissions'](#).

Note that to enable clients to write to points the Auth Grant 'Write Access' property must be set to true as well as the write user being configured correctly.

### 7.8 Client Credentials

The client credentials component contains the 'Client Id' and 'Client Secret' properties.

These client credentials are generated by the ossRestApiServer software using the buttons available on the default 'Generate Client Credentials View'.

Note that once generated, if you close the view the 'Client Secret' property is no longer able to be viewed and cannot be recovered from the system. If the secret is lost then the only solution is to reset the client secret using the button available on the credentials view. If the secret is reset then any clients still using the old secret value will no longer be able to access the API application.

### 7.8.1 Copying Client Credentials

Note that although the client secret is displayed as a multi-line value in the credentials view, both the client identifier and client secret should be transferred as 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.

### 7.8.2 Resetting Client Credentials

If the client credentials for an application are lost or compromised and new ones are required then they can be regenerated.

From the 'Generate Client Credentials View' of the Client Credentials component in the Oss API Client Application there is a 'Reset Client Secret' button. This button will cause the client secret to be regenerated.

Note that as before, this secret will only be available while the generation view is visible. If you close this view you will not be able to go back and access the secret again.

If the client credentials for an application are reset then any clients using the previous credentials and tokens will no longer be able to access the API.

## 7.9 Auth Grant

Note that the AuthGrant component is only created after the client credentials are generated.

### 7.9.1 Authorized

This is a read-only property that indicates if the API application has been successfully authorised by a client and therefore has issued tokens.

### 7.9.2 Authorization Time

This is a read-only property that indicates the last time a client was successfully authorised with the application.

### 7.9.3 Time Generated

This is a read-only property that shows the time the client credentials were last generated.

### 7.9.4 Valid Time

This property is used to set how long the client credentials will remain valid for after they are generated.

The client Id and client secret credentials must be used to generate a token within this time or they will become invalid.

### 7.9.5 Write Access

This property is used to control whether this application will allow write access.

### 7.9.6 Fault Cause

This is a read-only property that shows the current application fault cause, or blank if there is no fault.

### 7.9.7 Fault Time

This is a read-only property that shows the last application fault time.

## 8 Usage Notes

### 8.1 Token Storage

Whenever API tokens are generated they must be stored in the Niagara station so they persist if a station is rebooted.

Tokens are generated when authorisation requests are processed and when tokens are refreshed.

To ensure that these important security values are not lost the ossRestApiServer software will force the Niagara station to be saved whenever these tokens change. This writes the values to non-volatile storage.

It is possible that an engineer may kill the station at the same time the access token is being saved and therefore some of the station data may not be saved and the new access token may be lost. In this situation the client credentials will have to be regenerated. Note this will not happen if the station is shutdown or rebooted normally as the station will be saved correctly in these circumstances.

## 9 Appendix

### 9.1 End User License Agreement

#### 9.1.1 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.

#### 9.1.2 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.

#### 9.1.3 User Agreement

##### 9.1.3.1 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.

#### *9.1.3.2 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!

#### *9.1.3.3 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.

#### *9.1.3.4 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.

#### *9.1.3.5 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.

#### *9.1.3.6 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.

#### *9.1.3.7 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.

### **9.1.4 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.**

#### *9.1.5 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