AQUATIC INFORMATICS™

FASTER ANALYSIS.BETTER DECISIONS.

AQUARIUS 3.10 Publish Web Service API

April 8, 2016

Copyright © 2016, Aquatic Informatics Inc.

Table of Contents

1 Introduction **3**

2 Programming Environment: **3**

3 General Description of API **3**

4 Data Types and Technical Specs **4**

Method: GetAuthToken *5*
Method: GetParameterList *6*
Method: GetGradeList *7*
Method: GetApprovalList *8*
Method: GetPublishViewList *9*
Method: GetLocations *10*
Method: GetLocationsByFolderId *13*
Method: GetDataSetsList *14*
Method: GetTimeSeriesData *15*
Method: GetTimeSeriesRawData *19*
Method: GetTimeSeriesDataResampled *22*
Method: GetRatingTable *24*
Method: GetRatingCurveList *26*
Method: GetRatingTableAsCsvByIdentifier *28*
Method: GetRatingTableExtension *29*
Method: GetTemplateList *31*
Method: GetReportData *32*
Method: IsConnectionValid *33*
Method: KeepConnectionAlive *34*

Appendix 1: Sample Code **35**

REST (HTTP) Client Example (C#) 36

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

1 Introduction

This document describes software interfaces for retrieving time series and related data from AQUARIUS Server. The interface uses industry standard SOAP or REST web services whose outputs are simple CSV formatted strings.

2 Programming Environment:

For AQUARIUS software and hardware requirements, please refer to the AQUARIUS System Requirements document.

External tools can connect to the AQUARIUS Publish web service using either SOAP or REST protocols. If Microsoft Visual Studio is used, then the web service can be referenced directly, otherwise interfaces can be generated from the WSDL description or coded by hand.

Example code is shown in Appendix 1.

3 General Description of API

The API is based on a simple token-based authentication mechanism. Users acquire a token at the beginning of the session and use it for all subsequent access during the session. For the REST version of the interface, the token can be passed explicitly in the query string of embedded in the HTTP header. Examples of both are show in Appendix 1.

To acquire a token you use the GetAuthToken method. Passwords passed to this method can be either clear-text or RSA encrypted. If using RSA encryption, the RSA public key is:


```
<RSAKeyValue>
<Modulus>
3GmH/grhUyuiWlBDhNliZN3PYy2JPxJ5nG4t1CmQW8ZGPSsWY1AYIEU3b0A7eKOL3BbAiQzn0GpaSU9
HT452zKhqXE+G2nxw7Y0tJYfBzsTbBSPcHaUTHd/YyqDwDhuuJ+RrCqQfgnkq+YXnlCU1CJwVlF5HUy
LWWHaUgaNlEqs=
</Modulus>
<Exponent>AQAB</Exponent>
</RSAKeyValue>
```


Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

The API includes a set of methods for retrieving various lists of metadata from AQUARIUS (such as Approval codes, Quality codes, Parameter definitions, etc.) as well as methods for retrieving other data. The best practice is to get the lists of metadata types and use those values in the data retrieval methods, but for relatively static systems it is also fine to ignore the metadata definitions and simply hard code the known values into the retrieval methods.

AQUARIUS uses the concept of *Publish Views* to control access to time series data. The server administrator defines these views and specifies the data approval levels that each view is allowed to see. For example, it might make sense to set up a "Public" view that hides data at an "unapproved" approval level, or a "Trusted" view that hides nothing.

4 Data Types and Technical Specs

- Unless otherwise specified, all string data is UTF8 Unicode and all byte arrays are UTF8 encoded strings.
- Any string that contains a comma must be wrapped in double quotes.
- Dates are represented as ISO 8601 formatted strings with fractional seconds and time zone. The format is:
  `yyyy-MM-ddTHH:mm:ss.fffzzz  (e.g., 1997-07-16T19:20:30.451+01:00)`
- The SOAP interface is found at the following URL:
  `http://<yourserver>/AQUARIUS/Publish/AquariusPublishService.svc`
  And the WSDL is here:
  `http://<yourserver>/AQUARIUS/Publish/AquariusPublishService.svc?WSDL`
- The REST interface is at the following URL:
  `http://<yourserver>/AQUARIUS/Publish/AquariusPublishRestService.svc`

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

Method: GetAuthToken

This method is used to log in to the AQUARIUS Publish web service and obtain a token for accessing the API.

Interface

SOAP method: `string GetAuthToken (string user, string encodedPassword)`

REST method: `/GetAuthToken?user={user}&encPwd={encodedPassword}`

Input	
String user	AQUARIUS publish view access user (as configured in AQUARIUS Manager)
string encodedPassword	Hashed password. May be passed as clear text, or as RSA-encrypted hash (see above for RSA public key).

Output	
string authToken	The token is a SHA signature. The server caches this token for use during the session. Tokens expire after 30 minutes of no use.

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

Method: GetParameterList

This method returns the complete list of parameters defined in AQUARIUS. Common examples are HG (Gauge Height) and QR (Discharge) but there are often hundreds of others.

Interface

SOAP method: `string GetParameterList()`

REST method: `/GetParameterList?token={string}`

Input

<none>

Output

CSV string, format:

```
Parameter,DisplayId,Name,Interpolation,Unit
HG,Stage,Gauge Height,1,m
QR,Flow,Discharge,1,m^3/s
```

Line 1: Header
Line 2: Data

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

Method: GetGradeList

This method returns the complete list of grades defined in AQUARIUS.

Interface

SOAP method: `string GetGradeList ()`

REST method: `/GetGradeList?token=s{tring}`

Input

<none>

Output

CSV string, format:

```
Code,TypeCode,Description,Color
-2,BAD,Bad data,#ff00ff
0,,Undefined Quality,
10,A,Partial Day,#34c203
15,B, Backwater,
...
```

Line 1: Header
Line 2: Data

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

Method: GetApprovalList

This method returns the complete list of approval levels defined in AQUARIUS.

Interface

SOAP method: `string GetApprovalList()`

REST method: `/GetApprovalList?token={string}`

Input

<none>

Output

CSV string, format:

```
Code,Description,Color
0,Unprocessed,#ffffff
1,Preliminary,#ff0000
2,Reviewed,#00ff00
3,Checked,#0000ff
4,Approved,#c0c0c0
...
```

Line 1: Header
Line 2: Data

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

Method: GetPublishViewList

This method returns the complete list of publishing views defined in AQUARIUS. Publishing views are used to control who sees what data based on approval levels.

The output is a list of publish views where each view has a code number, a list of visible approval levels, and a view name.

Interface

SOAP method: `string GetPublishViewList()`

REST method: `/GetPublishViewList?token={string}`

Input

<none>

Output

CSV string, format:

```
Code,Approvals,PublishView
0,4,Public
1,0 3 4,Partner
2,0 2 3 4,InternalAccess
...
```

Line 1: Header
Line 2: Data

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

Method: GetLocations

This method is used to get a list of locations in the AQUARIUS database. The returned list can be filtered by supplying an optional collection of matching criteria via the filter parameter.

The output is a list of locations with data columns for name, identifier and other location attributes. There is a basic fixed set of data columns for the built-in attributes but columns for extended attributes are also returned.

The output is a list of locations which are contained within the given location folder. The operation is analogous to Springboard (i.e., any location within the given folder, *or any subfolder*, will be returned).

The filter parameter supports a flexible pattern of queries. It takes the form of a semicolon-separated list of filter criteria of the form:

<attribute><comparator><pattern>

<attribute> can be from the following list (these are based on the column names of the LOCATION table in the AQUARIUS database):

- LocationName (string, e.g., 'Wind River near Dubois')
- FolderName (string, e.g., 'Ottawa', based on Location Path)
- LocationTypeName (string, e.g., 'Hydrology')
- Identifier (string, e.g., '02GA010')
- It may also be the database column name of an extended attribute.

<comparator> can be from the following list:

- >
- >=
- <
- <=
- <>
- !=
- =

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

<pattern> is a wildcard string, such as *Foo*

For example: Identifier=H25*;Type=Hydrometic;Basin=Fraser

Criteria are processed with 'AND' semantics.

The back-slash character (\) is required to escape the following characters in the <pattern> string:

.(period)

*(asterisk)

\ (backslash)

For example:

- LocationName=AA\.AA*AA
- LocationName=AA\\AA;

Interface

SOAP method:
`string GetLocations(long folderId, string filter)`

REST method:
`/GetLocations?token={token}&filter={filter}`

Input	
string filter	Allows further filtering of the returned locations. See above for detailed description.

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

Output

CSV string, format: (assuming “NUMBEROFSENSORS” and “LOCSIZE” are configured as extended attributes for all location types and “BASIN” and “PHONENUMBER” are configured as extended attributes for Hydrology Station location types.)

```
LOCATIONID,LOCATIONFOLDERID,LASTMODIFIED,LOCATIONNAME,DESCRIPTIO
N,IDENTIFIER,LOCATIONTYPEID,LATITUDE,LONGITUDE,SRID,ELEVATIONUNI
TS,ELEVATION,UTCOFFSET,TIMEZONE,DEFAULTROLEID,AQUSERDATA_,NUMBER
OFSENSORS,LOCSIZE,LOCATIONTYPENAME,ATTRIBUTETABLENAME,
LOCATIONPATH, BASIN, PHONENUMBER
245,244,2012-08-24T14:29:03.210, Foo River at the Bar,Foo River
Description,02GA010,1,,,,,,-8.000,,,,45,Medium,Hydrology
Station,HYDSTNATTRIBTABLE,All Locations.Level1.Level2, Fraser
Basin, 604-123-9999
```

Line 1: Header

Line 2: Data

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

Method: GetLocationsByFolderId

Returns a list of locations for a specific location folder, and which meet an optional set of filter criteria.

The output is a list of locations which are contained within the given location folder. The operation is analogous to Springboard (i.e., any location within the given folder, *or any subfolder*, will be returned.)

An optional filter argument may also be provided. This is described above (see GetLocations).

Interface

SOAP method:
`string GetLocationsByFolderId(long folderId, string filter)`

REST method:
`/GetLocationsByFolderId?token={token}&folderId={folderId}&filter={filter}`

Input	
long folderId	Location folder identifier. This is a required field.
string filter	Allows further filtering of the returned locations. See GetLocations for more information.

Output	
CSV string, format:	
`LOCATIONID,LOCATIONFOLDERID,LASTMODIFIED,LOCATIONNAME,DESCRIPTION,IDENTIFIER,LOCATIONTYPEID,LATITUDE,LONGITUDE,SRID,ELEVATIONUNITS,ELEVATION,UTCOFFSET,TIMEZONE,DEFAULTROLEID,AQUSERDATA_,LOCATIONTYPENAME,ATTRIBUTETABLENAME,LOCATIONPATH`	Line 1: Header
`245,244,2012-08-24T14:29:03.210,Loc005,,Loc005,1,,,,,,-8.000,,,,Hydrology Station,,All Locations.Level1.Level2`	Line 2: Data

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

Method: GetDataSetsList

This method is used to get a list of all the time series that are in the AQUARIUS database for a given location or for all locations. The output is a list of datasets, where each line has a dataset name, the parameter it is measuring, the unit of measurement, and a set of simple statistics such as max, min, mean, total number of samples, etc. There is an option to return a list of only those time series that have been modified since a given time for a given location or for all locations.

Interface

SOAP method:
`string GetDataSetsList(string locationId, string changesSinceTime)`

REST method:
`/GetDataSetsList?token={string}&locId={string}&changesSince={datetime}`

Input	
string locationId	AQUARIUS location identifier (e.g., 02GA010). This is currently a required value. However, in the future, if this is not specified, then time series for all locations are returned.
string changesSinceTime	By specifying this time you limit the returned time series to those that have been modified since this time. If not specified then all time series are returned.

Output

CSV string, format:

```
LocationId,DataId,Parameter,Unit,Min,Max,Mean,StartTime,StartVal
ue,EndTime,EndValue,TotalGaps,TotalSamples,LastModified
02GA010,HG.Working@02GA010,HG,m,2.131,7.304,3.455,1996-01-
01T00:15:00:00.000-06:00,6.212,2010-05-05T20:00:01.000-
06:00,6.707,60,153771,2010-07-01T00:00:01.000-06:00
02GA010,QR.Working@02GA010,QR,m^3/s,0,422.525,17.152,1996-01-
01T00:15:00:00.000-06:00,4.321,2010-05-05T20:00:01.000-
06:00,5.775,60,153771,2010-07-01T00:00:01.000-06:00
...
```

Line 1: Header

Line 2: Data

Line 3

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

Method: GetTimeSeriesData

This method is used to retrieve the contents of a given time series. It can:

- retrieve the entire signal, or it can look at a particular time range.
- retrieve only data that has changed since a specified time.
- "roll back the clock" to look at the signal as it existed at some time in the past.

It is important to note that the data returned by this method is governed by the publishView that is specified, which in turn depends on the approval levels of data in the time series. This means that data can come into "view" at any time (based on approval changes) but can also go out of view.

The method returns two sets of data:

1- The time ranges that are relevant (one line per time range)
2- The data for each time range (many lines, one per data point)

This method always returns the corrected signal.

If changesSinceTime is specified then the method will return data that has changed (or come into view) since that time. Note that the returned data may include some points that were returned in a previous poll.

If the caller is maintaining a copy of the time series then the returned data must be processed as follows for each returned time range:

1- First, the overall time range should be deleted
2- Then the returned data should be copied in

> **IMPORTANT:** This may result in data being deleted as well as added. In the extreme case, a given time range may not have any corresponding points, in which case it is a full delete.

If changesSinceTime is not specified then the time series will be returned in a single time range (according to the specified publishView).

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

In the case where no data is available, the number of ranges will be 0 (i.e., the file will have only 2 lines).

Interface

SOAP method:

```
string GetTimeSeriesData(string dataId, string publishView,
                         string queryFromTime, string queryToTime,
                         string changesSinceTime, string asAtTime,
                         bool returnFullCoverage, bool includeGapMarkers)
```

REST method:

```
/GetTimeSeriesData?token={string}&dataId={string}&view={string}
                  &queryFrom={datetime}&queryTo={datetime}
                  &changesSince={datetime}&asAtTime={datetime}
                  &returnFullCoverage={bool}&includeGapMarkers{bool}
```

Input	
String dataId	AQUARIUS data identifier. This should be taken from the DataId parameter as returned from the call to GetDataSetsList. e.g. HG.Working@02GA010 QR.Working@02GA010 TA.Field Visits@02GA010
string publishView	See method GetPublishViewList. If the current user has a default publishing view assigned then the method will filter based on user's default publishing view. If the current user does not have a default publishing view specified a publishing view can be specified via this parameter. If the user has a publishing view specified and the publishView parameter is not null or empty GetTimeSeriesData will return an error message instead of data.
string queryFromTime	The start time of the range to be returned. For example if you are interested in only 2009 then this might be 2009-01-01T00:00:00.000-05:00. If not specified then the start of signal is used.
string queryToTime	The end time of the range to be returned. For example if you are interested in only 2009 then this might be 2009-12-31T23:59:59.999-05:00. If not specified then the end of signal is used.
string changesSinceTime	By specifying this time you limit the returned data to only changes that have

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

	occurred since this time. The data is returned in multiple time ranges according to what data has changed. If not specified then all data is returned and there will be only one time range.
String asAtTime	By specifying this time you can roll back the clock to retrieve the signal as it was at this time. Note that the "roll back" does not include changes made to correction history, such as "un-do" of a correction, or editing a correction. If this parameter is not specified then the current signal is used.
bool returnFullCoverage	When set to "true", the behavior of this flag is as follows: • if there is no point at the requested *queryFromTime*, then the first point prior to *queryFromTime* will be added to the returned time series points list. • if there is no point at the requested *queryToTime*, then the first point after *queryToTime* will be added to the returned time series points list. • if there is a point at the requested *queryFromTime*/*queryToTime*, or the requested *queryFromTime*/*queryToTime* time is outside of the time range of the time series, then no additional points are returned. When set to "false", the data returned will be within the time period between *queryFromTime* and *queryToTime*.
bool includeGapMarkers	When set to "true", 2 empty points (no-value in CSV) will be returned at 0.003 sec after/before the valid points on either side of each identified gap. In the following example, there is a gap between 03:30 – 05:45: 1,2011-04-02T03:15:00.000-08:00,27.216,,1, 1,2011-04-02T03:30:00.000-08:00,27.216,,1, 1,2011-04-02T03:30:00.003-08:00,,,1, 1,2011-04-02T05:44:59.997-08:00,,,1, 1,2011-04-02T05:45:00.000-08:00,27.213,,1, 1,2011-04-02T06:00:00.000-08:00,27.213,,1, When set to "false", no new empty points will be added to the output.

Output

CSV string, format:

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

Empty values will be represented as points with no Value in the CSV string.

```
Time,Parameter,LocationId,DataId,NumRanges,NumPoints,
PublishView, QueryFrom,QueryTo,ChangesSince, AsAtTime
2010-07-01T00:00:01.000-
06:00,HG,02GA010,HG.Telemetry@02GA010,2,35,Public,,,,,
RangeNumber,StartTime,EndTime,NumPoints
1,2010-02-01T00:00:01.000-06:00,2010-02-06T23:59:59.000-
06:00,20
2,2010-06-01T00:00:01.000-06:00,2010-06-06T23:59:59.000-
06:00,15
RangeNumber,Time,Value,Quality,Interpolation,Approval
1,2010-02-01T00:00:01.000-06:00,10.1,10,1,3
1,2010-02-01T00:15:01.000-06:00,10.2,10,1,3
1,2010-02-01T00:30:01.000-06:00,,10,1,3
1,2010-02-01T00:45:01.000-06:00,,10,1,3
1,2010-02-01T01:00:01.000-06:00,10.5,10,1,3
...
2,2010-06-01T00:00:01.000-06:00,10.1,10,1,3
2,2010-06-01T00:15:01.000-06:00,10.2,10,1,3
2,2010-06-01T00:30:01.000-06:00,10.3,10,1,3
2,2010-06-01T00:45:01.000-06:00,10.4,10,1,3
2,2010-06-01T01:00:01.000-06:00,10.5,10,1,3
...
```

Line 1: Meta header
Line 2: Meta data

Table1: Data Ranges

Table 2: Data

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

Method: GetTimeSeriesRawData

This method is used to retrieve the contents of the *raw* signal for a given time series.

The behavior is identical to that of GetTimeSeriesData, with the exception that this API works on the resampled data.

Interface

SOAP method:

```
string GetTimeSeriesRawData(string dataId, string publishView,
                            string queryFromTime, string queryToTime,
                            string changesSinceTime, string asAtTime,
                            bool returnFullCoverage, bool includeGapMarkers)
```

REST method:

```
/GetTimeSeriesRAwData?token={string}&dataId={string}&view={string}
                  &queryFrom={datetime}&queryTo={datetime}
                  &changesSince={datetime}&asAtTime={datetime}
                  &returnFullCoverage={bool},&includeGapMarkers={bool}
```

Input	
String dataId	AQUARIUS data identifier. This should be taken from the DataId parameter as returned from the call to GetDataSetsList. e.g. HG.Working@02GA010 QR.Working@02GA010 TA.Field Visits@02GA010
string publishView	As per description in GetTimeSeriesData and GetPublishViewList.
string queryFromTime	The start time of the range to be returned. For example if you are interested in only 2009 then this might be 2009-01-01T00:00:00.000-05:00. If not specified then the start of signal is used.
string queryToTime	The end time of the range to be returned. For example if you are interested in only 2009 then this might be 2009-12-31T23:59:59.999-05:00. If not specified then the end of signal is used.
string changesSinceTime	As per description in GetTimeSeriesData.
string asAtTime	As per description in GetTimeSeriesData.

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

bool returnFullCoverage	When set to “true”, the behavior of this flag is as follows: • if there is no point at the requested *queryFromTime*, then the first point prior to *queryFromTime* will be added to the returned time series points list. • if there is no point at the requested *queryToTime*, then the first point after *queryToTime* will be added to the returned time series points list. • if there is a point at the requested *queryFromTime*/*queryToTime*, or the requested *queryFromTime*/*queryToTime* time is outside of the time range of the time series, then no additional points are returned. When set to “false”, the data returned will be within the time period between *queryFromTime* and *queryToTime*.
bool includeGapMarkers	When set to "true", 2 empty points (no-value in CSV) will be returned at 0.003 sec after/before the valid points on either side of each identified gap. In the following example, there is a gap between 03:30 – 05:45: 1,2011-04-02T03:15:00.000-08:00,27.216,,1, 1,2011-04-02T03:30:00.000-08:00,27.216,,1, 1,2011-04-02T03:30:00.003-08:00,,,1, 1,2011-04-02T05:44:59.997-08:00,,,1, 1,2011-04-02T05:45:00.000-08:00,27.213,,1, 1,2011-04-02T06:00:00.000-08:00,27.213,,1, When set to “false”, no new empty points will be added to the output.

Output

CSV string, format:

Empty values will be represented as points with no Value in the CSV string.

`Time,Parameter,LocationId,DataId,NumRanges,NumPoints,PublishView, QueryFrom,QueryTo,ChangesSince`	Line 1: Meta header
`2010-07-01T00:00:01.000-06:00,HG,02GA010,HG.Telemetry@02GA010,2,35,Public,,,,,`	Line 2: Meta data
`RangeNumber,StartTime,EndTime,NumPoints`	Table1: Data Ranges
`1,2010-02-01T00:00:01.000-06:00,2010-02-06T23:59:59.000-06:00,20`	
`2,2010-06-01T00:00:01.000-06:00,2010-06-06T23:59:59.000-`	

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

```
06:00,15
RangeNumber,Time,Value,Quality,Interpolation,Approval
1,2010-02-01T00:00:01.000-06:00,10.1,10,1,3
1,2010-02-01T00:15:01.000-06:00,10.2,10,1,3
1,2010-02-01T00:30:01.000-06:00,,10,1,3
1,2010-02-01T00:45:01.000-06:00,,10,1,3
1,2010-02-01T01:00:01.000-06:00,10.5,10,1,3
...
2,2010-06-01T00:00:01.000-06:00,10.1,10,1,3
2,2010-06-01T00:15:01.000-06:00,10.2,10,1,3
2,2010-06-01T00:30:01.000-06:00,10.3,10,1,3
2,2010-06-01T00:45:01.000-06:00,10.4,10,1,3
2,2010-06-01T01:00:01.000-06:00,10.5,10,1,3
...
```

Table 2: Data

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

Method: GetTimeSeriesDataResampled

This method is used to retrieve the contents of a given time series, resampled based on a given interval. The behavior is generally similar to that of GetTimeSeriesData.

The entire corrected signal will be resampled. All points which fall within the endpoints of the original, and after the specified "anchor time," will be returned. The specified sampling interval may be used to "up-sample" or "down-sample" the signal.

Interface

SOAP method:

```
string GetTimeSeriesDataResampled(string dataId, string publishView,
                        string queryFromTime, string queryToTime,
                        int sampleIntervalInMinutes, string anchorTime)
```

REST method:

```
/GetTimeSeriesDataResampled?token={string}&dataId={string}&view={string}
                  &queryFrom={datetime}&queryTo={datetime}
                  & sampleIntervalInMinutes={int}&anchorTime={datetime}
```

Input	
String dataId	AQUARIUS data identifier. This should be taken from the DataId parameter as returned from the call to GetDataSetsList. e.g., HG.Working@02GA010 QR.Working@02GA010 TA.Field Visits@02GA010
string publishView	As per description in GetTimeSeriesData and GetPublishViewList.
string queryFromTime	The start time of the range to be returned. For example, if you are interested in only 2009 then this might be 2009-01-01T00:00:00.000-05:00. If not specified then the start of signal is used.
string queryToTime	The end time of the range to be returned. For example, if you are interested in only 2009 then this might be 2009-12-31T23:59:59.999-05:00. If not specified then the end of signal is used.
int sampleIntervalInMinutes	Interval between resampled points, in minutes.

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

string anchorTime	Specifies the time of the first point of the resampled signal. This could be used, for example, to create an hourly signal which has points on the half-hour. If the anchor time falls before the start of the signal, then the resampled signal will start at the first point of the signal. If the anchor time falls after the end of the signal, then no points will be returned.

Output

CSV string, format:

Empty values will be represented as points with no Value in the CSV string.

Gap markers will not be included in the CSV string.

`Time,Parameter,LocationId,DataId,NumRanges,NumPoints, PublishView, QueryFrom,QueryTo,AnchorTime`	Line 1: Meta header
`2010-07-01T00:00:01.000-06:00,HG,02GA010,HG.Telemetry@02GA010,2,35,Public,,, 2010-11-15T09:22:00.000-08:00`	Line 2: Meta data
`RangeNumber,StartTime,EndTime,NumPoints` `1,2010-02-01T00:00:01.000-06:00,2010-02-06T23:59:59.000-06:00,20`	Table1: Data Ranges

`RangeNumber,Time,Value,Quality,Interpolation,Approval`
`1,2010-11-15T09:22:00.000-08:00,0.662,,1,`
`1,2010-11-15T09:27:00.000-08:00,0.663,,1,`
`1,2010-11-15T09:32:00.000-08:00,0.663,,1,`
`1,2010-11-15T09:37:00.000-08:00,0.663,,1,`
`1,2010-11-15T09:42:00.000-08:00,0.664,,1,`
	Table 2: Data

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

Method: GetRatingTable

This method is used to get the rating curve(s) for a given location. It returns a set of interpolated curves in the form of regularly sampled rating tables, each with its own period of applicability.

The output is composed of two logical tables. The first table lists the rating curves and their periods of applicability. The second table gives the points for each curve.

Interface

SOAP method:

```
string GetRatingTable(string locationId, string inputParameter,
                      string outputParameter)
```

REST method:

```
/GetRatingTable?token={string}&locId={string}&input={string}&output={string}
```

Input	
string locationId	AQUARIUS location identifier. (e.g., 02GA010)
string inputParameter	Independent parameter (e.g., HG)
string outputParameter	Dependent parameter (e.g., QR)

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

Output

CSV string, format:

```
LocationId,InputParameter,OutputParameter,NumTables
02GA010,HG,QR,2
TableNumber,StartDate,EndDate,NumPoints
1,1990-06-01T00:00:01.000-06:00,2001-06-01T00:00:01.000-06:00,56
2,2001-06-01T00:00:01.000-06:00,2002-06-01T00:00:01.000-06:00,87
TableNumber,Input,Output
1,12.1,20.3
1,12.2,20.4
1,12.2,20.4
1,12.2,20.4
1,12.2,20.4
...
2,12.2,20.4
2,12.2,20.4
2,12.2,20.4
2,12.2,20.4
...
```

Line 1: Meta header
Line 2: Meta data
Table 1: Rating table info

Table 2: Rating points

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

Method: GetRatingCurveList

This method is used to retrieve the list of rating curve(s) which match a given set of filter parameters. It returns a list of rating curve set of identifiers which may then be used to query the details, e.g. by calling GetRatingTableAsCsvByIdentifier.

Interface

SOAP method:
```
string[] GetRatingCurveList(RatingCurveFilter ratingCurveFilter)
```

REST method:
```
/GetRatingCurveList?token={string}& locationIdentifier
={string}&changedSince={string}
```

Input	
RatingCurveFilter ratingCurveFilter (SOAP)	Information about the set of rating curves to be retrieved. A rating curve must match all the specified filters in order to be retrieved. • string LocationIdentifier: Optional. If specified, only the rating curves under that location will be retrieved. Otherwise, all rating curves in all locations will be fetched for further filtering. • string ChangedSince : Last modified time string in ISO format. Optional. If specified, only the rating curves modified on or after the time will be retrieved. Otherwise, the last modified time of a rating curve is ignored. The format is of ISO time format ("yyyy-MM-ddTHH:mm:ss.fffzzz"). For example, “1994-11-05T08:15:30.000-05:00” corresponds to November 5, 1994, 8:15:30 AM, US Eastern Standard Time). An exception is thrown if the input is of a bad format or has invalid values.
string locationIdentifier (REST)	See note on “LocationIdentifier” above.
string changedSince (REST)	See note on “ChangedSince” above.

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

Output

SOAP method: Array of rating curve identifiers. Returns an empty array if no rating curves found.

REST method: A string separated with newline characters. Each line is a rating curve identifier. Returns an empty string if no rating curves found.

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

Method: GetRatingTableAsCsvByIdentifier

This method is used to get the rating curve(s) for a given rating curve identifier. See the notes on GetRatingTable for details.

Interface

SOAP method:
```
string GetRatingTableAsCsvByIdentifier(string ratingCurveIdentifier)
```

REST method:
```
/ GetRatingTableAsCsv?token={string}& ratingCurveIdentifier ={string}
```

Input	
string ratingCurveIdentifier	A single rating curve identifier in “text identifier” format, e.g. “HG-QR.Rating Curve@02GA006”.

Output
Rating tables in CSV format. See the Output notes on GetRatingTable API. If the specified identifier is valid but such a rating curve does not exist, an empty string is returned.

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

Method: GetRatingTableExtension

This method is used to get the expanded rating table for a given location and specified time. It returns a sampled rating table with applied shifts. The points are interpolated with precision of 0.01.

Interface

SOAP method:

```
string GetRatingTableExtension(string locationId, string inputParameter,
                      string outputParameter, string asAtTime)
```

REST method:

```
/GetRatingTable?token={string}&locId={string}&input={string}&output={string}&time={st
ring}
```

Input	
string locationId	AQUARIUS location identifier. (e.g., 02GA010)
string inputParameter	Independent parameter (e.g., HG)
string outputParameter	Dependent parameter (e.g., QR)
string asAtTime	This time is used to determine which rating and which shift is in effect to create rating table. If the time is not specified the current date and time is used.

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

Output

CSV string, format:

```
LocationId,InParameter,OutParameter,StartDate,EndDate,ExpandedPo
ints,OriginalPoints,Remark
02GA010,HG,QR, 2010-01-01T00:00:00.000-08:00,2382-12-
30T12:00:00.000-08:00,898,8,shifted curve
Stage,Discharge,ShiftStage,ShiftDischarge
8.000,127.741,8.500,204.419
8.010,128.990,8.509,206.097
8.020,130.249,8.518,207.786
```

Line 1: Meta header

Line 2: Meta data

Table 1: Rating points

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

Method: GetTemplateList

This method returns the complete list of report templates defined in AQUARIUS. Report templates are used to generate reports for a given data set.

The output is a list of report templates where each template has an id, a label and a type.

Interface

SOAP method:
`string GetTemplateList()`

REST method:
`/GetTemplateList?token={string}`

Input

<none>

Output

CSV string, format:

```
Id,Label,Type (0=Template; 1=Description)                    Line 1: Meta header
798fa260-ab19-435c-9997-850373d8da71, Daily Mean by Year,0    Line 2: Meta data
```

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

Method: GetReportData

This method generates and retrieves a report for a given data set using the provided report template. The report templates are configured using the AQUARIUS Report Tool User Interface. The output is the report as generated by the AQUARIUS reporting engine. The contents of the report are specific according to the provided report template and data set.

Interface

SOAP method:

```
string GetReportData(string dataId, string reportId, int reportType,
                     string outputPath, int reportName)
```

REST method:

```
/GetReportData?token={string}&dataId={string}&reportId={string}&reportType={int}&
               outputPath={string}&reportName={string}
```

Input	
string dataId	AQUARIUS data identifier, e.g., HG.Working@02GA010
string reportId	See method GetTemplateList. The report template is set up in the AQUARIUS Reporting Tool.
Int reportType	See method GetTemplateList: 0 = Report Template 1 = Report Description
string outputPath	Full path to the location on disk where the report will be saved. e.g., C:\ProgramData\Aquatic Informatics\AQUARIUS\Reports
string reportName	Report name (file extension will be appended to the report name once the report is generated according to how the specified report template is set up). e.g., DailyMeansReport_01012011.

Output
(Report output in format as defined by report template or report description in Report Tool)

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

Method: IsConnectionValid

This method checks if authentication token is still valid for the created connection.

Interface

SOAP method:
`bool IsConnectionValid()`

REST method:
`/IsConnectionValid?token={string}`

Input
<none>

Output	
bool	Returns true if the token is still valid

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

Method: KeepConnectionAlive

This method keeps authentication token valid for the created connection.

Interface

SOAP method:
`bool KeepConnectionAlive()`

REST method:
`/KeepConnectionAlive?token={string}`

Input

<none>

Output

Bool	Returns true if the token is still valid

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

Appendix 1: Sample Code

SOAP Client Example (C#)

This example shows sample code for using the SOAP interface. If you are using Visual Studio you can create the client proxy class by using the “Add Service Reference” feature. Otherwise you can generate a proxy class from the WSDL some other way (not shown). In AQUARIUS 3.3, and newer versions, the WSDL indicates that the authentication is required in the SOAP header, so you no longer have to manually add it to the header. In .NET the authentication token must be passed as the first parameter of each method. Other frameworks/languages may use an alternate means of setting the token, depending on the implementation

```
//Add web references "http://aiapp1/AQUARIUS/Publish/AquariusPublishService.svc"
AquariusPublishServiceClient client = new AquariusPublishServiceClient(
     "BasicHttpBinding_IAquariusPublishService",
     @"http://aiapp1/AQUARIUS/Publish/AquariusPublishService.svc");

//get token and set it into the SOAP header
string authToken = client.GetAuthToken("dummy_user", "dummy_pw");

//basic methods
string paramList = client.GetParameterList(authToken);
string qualityList = client.GetGradeList(authToken);
string approvalList = client.GetApprovalList(authToken);
string ratingTable = client.GetRatingTable(authToken, "02GA010", "HG", "QR");
string dataList = client.GetDatasetsList(authToken, "02GA010");

//Time series changes in the last 30 hours
string changeSince = (DateTime.Now-TimeSpan.FromHours(30)).ToString(
                      "yyyy-MM-ddTHH:mm:ss.fffzzz");
byte[] datas =
client.GetTimeSeriesData(authToken,"HG.Telemetry@02GA010","Public",
                                         null, null, changeSince, null);
string csv = System.Text.Encoding.UTF8.GetString(datas);
```

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

REST (HTTP) Client Example (C#)

This example shows a mechanism for using the REST interface. To simplify things it includes a custom HttpGet function (included below) that deals with making HTTP requests, setting the authToken, and marshalling the response data.

```
string address =
     "http://aiapp1/AQUARIUS/Publish/AquariusPublishRestService.svc";

//get auth token
string authToken = HttpGet(address + "/GetAuthToken” +
                          “?user=dummy.supe&encPwd=dummy1", null);

//basic methods
string paramList = HttpGet(address + "/GetParameterList", authToken);
string qualityList = HttpGet(address + "/GetGradeList", authToken);
string approvalList = HttpGet(address + "/GetApprovalList", authToken);
string ratingTable = HttpGet(address + "/GetRatingTable” +
                             “?locId=02GA010” +
                             “&input=HG” +
                             ”&output=QR",
                             authToken);
string dataList = HttpGet(address + "/GetDataSetsList” +
                         “?locId=02GA010",
                         authToken);

//Time series changes in the last 30 hours
string changeSince = (DateTime.Now – TimeSpan.FromHours(30)).ToString(
                      "yyyy-MM-ddTHH:mm:ss.fffzzz");
string csv = HttpGet(address + "/GetTimeSeriesData” +
                    “?dataId=HG.Telemetry@02GA010” +
                    “&view=Public” +
                    “&changesSince=" + changeSince,
                    authToken);
```

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.

```
//-------------------------------------------------------
// Here is the custom HttpGet function, in C#
//-------------------------------------------------------
public static string HttpGet(string url, string authToken)
{
    return HttpGet(url, authToken, System.Text.Encoding.UTF8);
}
public static string HttpGet(string url, string authToken,
                             System.Text.Encoding encoding)
{
    System.Net.HttpWebRequest request = null;
    System.Net.WebResponse response = null;
    System.IO.Stream responses = null;
    System.IO.StreamReader sr = null;
    try
    {
        request = (System.Net.HttpWebRequest)System.Net.WebRequest.Create(url);
        request.Method = "GET";

        if (!string.IsNullOrEmpty(authToken))
        {
              url = url + "?AQAuthToken=" + authToken;
        }

        // It is also possible to include the authToken in the header
        //if (!string.IsNullOrEmpty(authToken))
        //{
        //    request.Headers.Add("AQAuthToken", authToken);
        //}

        response = request.GetResponse();
        responses = response.GetResponseStream();
        sr = new System.IO.StreamReader(responses, encoding);
        string ret = sr.ReadToEnd();
        return ret;
    }
    catch (Exception e)
    {
        throw e;
    }
    finally
    {
        if (sr != null) { try { sr.Close(); } catch { } }
        if (responses != null) { try { responses.Close(); } catch { } }
    }
}
```

Copyright © 2014, Aquatic Informatics Inc. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Do not distribute.