



GE VERNOVA

PROFICY® SOFTWARE & SERVICES

PROFICY HISTORIAN

REST APIs Reference
Manual

Proprietary Notice

The information contained in this publication is believed to be accurate and reliable. However, GE Vernova assumes no responsibilities for any errors, omissions or inaccuracies. Information contained in the publication is subject to change without notice.

No part of this publication may be reproduced in any form, or stored in a database or retrieval system, or transmitted or distributed in any form by any means, electronic, mechanical photocopying, recording or otherwise, without the prior written permission of GE Vernova. Information contained herein is subject to change without notice.

© 2024 GE Vernova and/or its affiliates. All rights reserved.

Trademark Notices

“GE VERNOVA” is a registered trademark of GE Vernova. The terms “GE” and the GE Monogram are trademarks of the General Electric Company, and are used with permission.

Microsoft® is a registered trademark of Microsoft Corporation, in the United States and/or other countries.

All other trademarks are the property of their respective owners.

We want to hear from you. If you have any comments, questions, or suggestions about our documentation, send them to the following email address:
doc@ge.com

Contents

- Chapter 1. Historian REST APIs..... 4**
- Introduction to Historian REST APIs.....4
- Historian APIs.....4
- About Security and Authentication.....4
- Standards..... 6
- API Methods..... 7
- API Status Messages..... 7
- Common API Parameters..... 8
- Overview of Commonly Used API Parameters..... 8
- TagNames Parameter..... 8
- Start and End Timestamps Parameter..... 9
- TagSamples Parameter..... 9
- DataSample Parameter..... 11
- SamplingModeType Parameter..... 12
- Direction Parameter..... 14
- CalculationModeType Parameter..... 14
- FilterModeType Parameter..... 20
- ReturnDataFields Parameter..... 21
- Payload Parameter..... 22
- Error Code Definitions..... 28
- Historian REST APIs.....31
- Overview of the Historian REST APIs.....31
- Managing Systems..... 31
- Managing Historian Model.....64
- Managing Collector Instances..... 106
- Collector Type and Subtype..... 116
- Managing Collectors..... 117

Managing Data Stores.....	142
Managing Tags.....	158
Managing Alarms and Events.....	194
Swagger Documentation.....	204

Chapter 1. Historian REST APIs

Introduction to Historian REST APIs

Historian APIs

Historian is a high performance data archiving system designed to collect, store, and retrieve time-based information at extremely high speed efficiently. The Historian environment provides a set of REST APIs to query data from the archives.

This document provides links for setting up your development environment, as well as information for getting started with the Historian services and their associated APIs.

Starting Historian 8.0, the default https port is 443. If you use the default port, you need not include it in the Rest API calls.

Also, the default admin client name is changed from admin to hostname.admin, and it is case-sensitive.

Example:

```
curl -u admin:adminsecret https://<nodename>:443/uaa/oauth/token -d  
'grant_type=client_credentials'
```

should be replaced with

```
curl -u hostname.admin:adminsecret https://<nodename>/uaa/oauth/token -d  
'grant_type=client_credentials'
```

See the following topics for more information:

- [About Security and Authentication \(on page 4\)](#)
- [Standards \(on page 6\)](#)
- [API Methods \(on page 7\)](#)
- [API Status Messages \(on page 7\)](#)

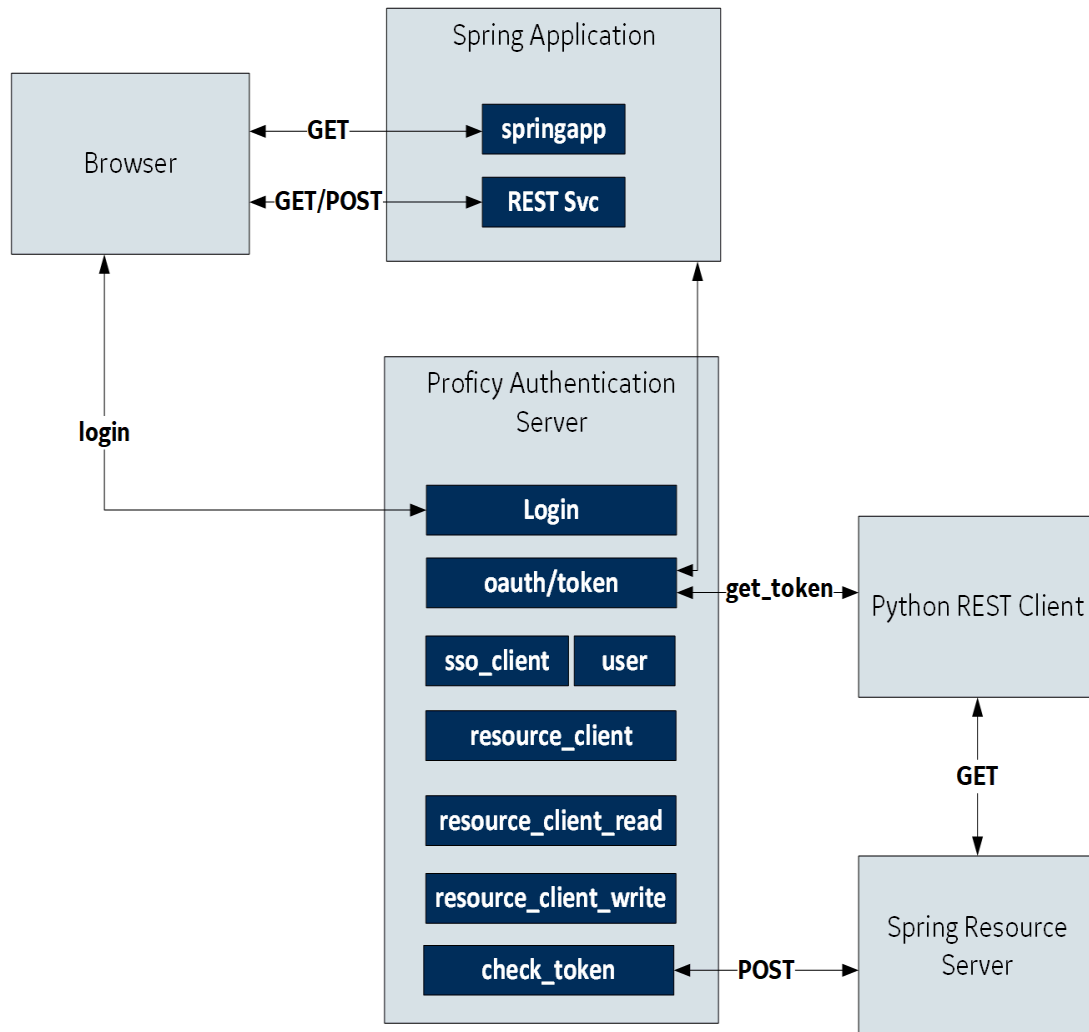
About Security and Authentication

For security purposes, Historian uses the Proficy Authentication service as a trusted source of tokens issued for authentication. It is a multi-tenant identity management service, used in Cloud Foundry, but also available as a standalone OAuth2 server. Its primary role is as an OAuth2 provider, issuing tokens for client applications to use when they act on behalf of Cloud Foundry users. It can also authenticate users with Cloud Foundry credentials, and can act as an SSO service using those credentials, or others.

It contains endpoints for managing user accounts, registering OAuth2 clients, and other management functions.

The following diagram shows how the Proficy Authentication server functions with a Python REST client:

Figure 1. Proficy Authentication Server and Python REST Client



Authorization

For exchanging data between the client-server system, user authentication is required. Once you have provided your client credentials, an access or bearer token is generated. This token is used for the REST APIs.

cURL command format for generating an oauth token for clients: `curl -u <client ID>:<client secret>
https://<node name>:443/uaa/oauth/token -d 'grant_type=client_credentials'`

Example: `curl -u server1.admin:adminsecret https://server1:443/uaa/oauth/token -d 'grant_type=client_credentials'`

cURL command format for generating an oauth token for Proficy Authentication users: `curl -d`

```
"client_id=<value>&client_secret=<value>&grant_type=password&username=<value>&password=<value>&token_format=opaque"
https://<node name>:8080/uaa/oauth/token
```

Example: `curl -d`

```
"client_id=historian_public_rest_api&client_secret=publicapisecret&grant_type=password&username=<value>&password=<value>"
https://<nodename>/uaa/oauth/token
```

In the following image, the actual token text is blurred for security concerns.

Figure 2. OAuth Access Token Sample



Client applications can access data using service REST API endpoints. Your application makes an HTTP request and parses the response. You can use any web-development language to access the APIs.

Standards

Historian APIs use a REST application architecture constrained by Hypermedia as the Engine of Application State (HATEOAS) that distinguishes it from most other network application architectures. Therefore, a client interacts with a network application entirely through hypermedia provided dynamically by application servers. The REST client doesn't need prior knowledge about how to interact with a particular application or server beyond a basic understanding of hypermedia.

As defined by the query parameters, the Historian APIs use "search" functions to access raw data using cURL and HTTP, while responses are in JSON format.

cURL is a command-line utility used to transfer data from or to a server, using one of the supported protocols, such as DICT, FILE, FTP, FTPS, GOPHER, HTTP, HTTPS, IMAP, IMAPS, LDAP, LDAPS, POP3, POP3S, RTMP, RTSP, SCP, SFTP, SMTP, SMTPS, TELNET and TFTP. The command is designed to work without user interaction.

cURL offers many useful functions such as proxy support, user authentication, FTP upload, HTTP post, SSL connections, cookies, file transfer resume, user and password authentication, and more.

You can run the sample commands provided in this document from Bash on Windows in the Windows operating system, and also in Linux Shell in the Linux operating system.

As a prerequisite, make sure you install cURL on your system, if it is not already installed. Run the `curl --version` command on Windows Bash or Linux shell to check if cURL is installed on your system.



Important:

Do not create your own URIs. Instead, use the links in this document and in the responses to navigate between resources.

API Methods

The Historian APIs use GET, POST, PUT, and DELETE methods.

Method	Usage
GET	Retrieves a resource.
POST	Creates (or adds) a resource.
PUT	Updates a resource.
DELETE	Removes a resource.

API Status Messages

In its use of the following HTTP status codes, the Historian API services adhere as closely as possible to standard HTTP and REST conventions.

Status Code	Usage
200 OK	Success message. The request has completed.
201 Created	Success message. A new resource has been created. The resource URI is available from the location header in the response.
204 No Content	Success message. An update to an existing resource has been applied.
400 Bad Request	Error message. The request was malformed. The response body provides additional information.

Status Code	Usage
401 Unauthorized	Error message. Either you are not authenticated, or the authentication is incorrect. You must re-authenticate and try again.
403 Forbidden	Error message. You do not have permission to access this resource.
404 Not Found	Error message. The requested resource does not exist.
500 Internal Error	Error message. The server encountered an unexpected condition that prevented it from fulfilling the request.

Common API Parameters

Overview of Commonly Used API Parameters

The Historian REST service provides various REST API calls to retrieve the current tags list and query data with different sampling modes. Most of these API calls use the following common parameters:

- [tagNames \(on page 8\)](#)
- [Start and End timestamps \(on page 9\)](#)
- [TagSamples \(on page 9\)](#)
- [DataSamples \(on page 11\)](#)
- [SamplingModeType \(on page 12\)](#)
- [Direction \(on page 14\)](#)
- [CalculationModeType \(on page 14\)](#)
- [FilterModeType \(on page 20\)](#)
- [ReturnDataFields \(on page 21\)](#)
- [Payload \(on page 22\)](#)
- [Error Code Definitions \(on page 28\)](#)

TagNames Parameter

By default, the Historian REST service provides support to read samples for multiple tags. Multiple tag names are separated by semicolons (;). For example, "tagname1;tagname2;tagname3".

```
https://<historianservername>:443/historian-rest-api
/v1/datapoints/currentvalue?tagNames=tagName1;tagName2;tagName3

https://RestTestsNode/historian-rest-api/v1/datapoints/currentvalue?tagNames=TAG01;TAG02
```

Encode the semicolon as %3B if using the URI format, as the semicolon is also a valid character for a Historian name, and the web service parses the tag names incorrectly if a tag name contains a semicolon.

Start and End Timestamps Parameter

For the Start and End Timestamps parameter, the Timestamp format in the URI must be in ISO data format, such as `YYYY-MM-DDTHH:mm:ss.SSSZ`.

EPOCH time (standard base time) is only valid in the JSON-format request body or response body, such as `\Date(928167600000-0500)\`. If you use the same timestamp for start and end timestamps, the request returns a single result.

All timestamps passed to the REST service must be formatted as UTC timestamps.

Object Name	Description
StartTime	Start time of the query. This represents the earliest timestamp for any tag contained in the query. If no StartTime is specified, the start time is two hours prior to running the query.
EndTime	End time of the query. This represents the latest timestamp for any tag contained in the query. If no EndTime is specified, the end time is the time that the query runs.

TagSamples Parameter

The TagSamples parameter is the output from the REST API calls.

Property Name	Property Type	Description
TagName	String	Name of the tag.
DataType	String	Tag Data Type Value:

Property Name	Property Type	Description
		<ul style="list-style-type: none"> • Blob – Stores tags as binary large objects. The Blob datatype generally refers to undetermined binary data types, such as an Excel spreadsheet, a PDF file, or a Word file. • Boolean (one byte) – Stores boolean values. Valid values for the boolean data type are 0=FALSE and 1=TRUE. If the user sends zero, the value is taken as zero. Anything other than zero, the value is treated as one. • Byte (one byte) – Stores integer values. Valid values for the byte data type are -128 to +127. • SingleFloat (four bytes) – Stores decimal values up to six places. Valid ranges for the single float data type are 1.175494351e-38F to 3.402823466e+38F • DoubleFloat (eight bytes) – Stores decimal values up to 15 places. Valid values for the double float data type are 2.2250738585072014e-308 to 1.7976931348623158e+308. • SingleInteger (two bytes) – Stores whole numbers, without decimal places. Valid values for the single integer data type are -32767 to +32767. • DoubleInteger (four bytes) – Stores whole numbers, without decimal places. Valid values for the double integer data type are -2147483648 to +2147483648. • FixedString (Configured by user) – Stores string data of a fixed size. Valid values are between 0 and 255 bytes. • Float – Single float. • Integer – Single integer. • MultiField – Stores string data that has multiple words. • QuadInteger (eight bytes) – Stores whole numbers without decimal places. Valid values for the quad integer data type are -9,223,372,036,854,775,808 (negative nine quintillion) to +9,223,372,036,854,775,807 (positive nine quintillion). • Scaled (two bytes) – Lets you store a four-byte float as a twobyte integer in the Historian archive. The scaled data type saves disk space but sacrifices data precision as a result. • Time – Returns or sets the type of time stamping applied to data at collection time.

Property Name	Property Type	Description
		<ul style="list-style-type: none"> • UDoubleInteger (Unsigned Double Integer) (four bytes) – Stores whole numbers without decimal places. Valid values for the unsigned double integer data type are 0 to 4,294,967, 295 (4.2 billion). • Undefined – Data type is not defined. • UQuadInteger (Unsigned Quad Integer) (eight bytes) – Stores whole numbers without decimal places. Valid values for the unsigned quad integer data type are 0 to 18,446,744,073,709,551,615 (19 quintillion). • USingleInteger (Unsigned Single Integer) (two bytes) – Stores whole numbers without decimal places. Valid values for the unsigned single integer data type are 0 to 65535. • VariableString (No fixed size) – Stores string values of undetermined size. This data type is useful if you cannot rely on a constant string length from your data source. • Array – Returns an array of tags from your data source. You can specify orientation, size, and number of rows returned in the array.
ErrorCode	Error Code	<p>Error Code Definition</p> <p>See Error Code Definition (on page 28) for more information.</p>
Samples	Data Sample	See DataSample Parameter (on page 11) for more information.

DataSample Parameter

The DataSample Parameter specifies the number of data samples to retrieve from the archive. Samples are evenly spaced within the time range defined by start time and end time for most sampling modes.

Property Name	Property Type	Description
Value	String	<p>Format for a multi-field tag like</p> <pre>{ "field1": "1", "field2": "1000.0" }</pre> <p>(user-defined type tag).</p>

Property Name	Property Type	Description
		JavaScript code can parse the value string as a JSON object. All field values are string.
TimeStamp	DateTime	Start and end times of the query. If no start time is specified, the start time is two hours prior to running the query. If no EndTime is specified, the end time is the time the query runs.
Quality	Integer (Enumerated value of DataQuality.StatusType)	Data type consisting of a set of named values called elements, members or enumerators of the type. Property values reflect quality as "quality is good" or " quality is bad". Value and Status <ul style="list-style-type: none"> • 0 – Bad • 1 – Uncertain • 2 – NA • 3 – Good

SamplingModeType Parameter

The SamplingModeType parameter is the mode of sampling data from the archive. The default setting for the Sampling Mode is `Calculated`.

Properties	Description	Value
Undefined	Sampling mode is not defined.	0
CurrentValue	Retrieves the current value. The time- interval criteria are ignored.	1
Interpolated	Retrieves evenly-spaced, interpolated values based on interval or NumberOf-Samples and the time-frame criteria.	2
Trend	Returns the raw minimum and raw maximum value for each specified interval.	3

Properties	Description	Value
	Use the Trend sampling mode to maximize performance when retrieving data points for plotting. For the Trend sampling mode, if the sampling interval does not evenly divide by the interval length, Historian ignores any leftover values at the end, rather than putting them into a smaller interval.	
RawByTime	Retrieves raw archive values based on time-frame criteria.	4
RawByNumber	Retrieves raw archive values based on the StartTime criteria, the NumberOfSamples, and Direction criteria. The EndTime criteria is ignored for this sampling mode.	5
Calculated	Retrieves evenly spaced calculated values based on NumberOfSamples, interval, the time frame criteria, and the CalculationMode criteria.	6
Lab	Returns actual collected values without interpolation.	7
InterpolatedtoRaw	Provides raw data in place of interpolated data when the number of samples are fewer than the available samples.	8
TrendtoRaw	The TrendtoRaw sampling mode almost always produces the same results as the Trend sampling mode. However, when more samples are requested than there are raw data points, the TrendtoRaw sampling mode returns all available raw data points with no further processing. Use TrendtoRaw in place of Trend when this condition exists.	9

Properties	Description	Value
LabtoRaw	Provides raw data for the selected calculated data, when NumberOfSamples is less than the available samples.	10
RawByFilterToggle	<p>Returns filtered time ranges using the following values:</p> <ul style="list-style-type: none"> • 1 – true • 0 – false <p>This sampling mode is used with the time range and filter tag conditions. The response string starts with a starting time stamp and ends with an ending timestamp.</p>	11

Direction Parameter

The Direction Parameter specifies the direction (Forward or Backward from the starting time) of data sampling from the archive. The default value is Forward.

Direction	Value
Forward	1
Backward	0

CalculationModeType Parameter

The CalculationModeType parameter is only applied if the Sampling Mode is set to Calculated. It represents the type of calculation to use on the archive data. The default Calculation Mode, if none is specified, is Average.

Calculation Mode Type	Description	Value
Undefined	Calculation mode is not defined.	0
Average	Retrieves the time-weighted average for each calculation interval.	1

Calculation Mode Type	Description	Value
StandardDeviation	Retrieves the time-weighted standard deviation for each calculation interval.	2
Total	<p>Retrieves the time-weighted rate total for each calculation interval.</p> <p>Use rate totals when working with a continuous measurement. Time weighting takes into account that compressed data is not evenly spaced in time. A factor must be applied to the total value to convert into appropriate engineering units. As a rate total, the default is Units/Day. If the actual units of the continuous measurement are Units/Minute, you would multiply the results by 1440 (minutes per day) to convert the total into appropriate engineering units.</p>	3
Minimum	Retrieves the minimum value for each calculation interval.	4
Maximum	Retrieves the maximum value for each calculation interval.	5
Count	<p>Counts the number of raw samples found with good quality in the interval.</p> <p>Value is the count of raw samples with good quality in the interval. The values of each sample are ignored. The Count does not include any samples of bad quality, including the start and end of collection markers.</p> <p>For Quality, the percentage of good samples is always 100, even if the interval does not contain any raw samples, or contains only bad quality samples.</p> <p>Count is useful for analyzing the distribution of the raw data samples to determine the ef-</p>	6

Calculation Mode Type	Description	Value
	<p>fect of compression deadbands. It is also useful to determine which tags are consuming the most archive space.</p>	
RawAverage	<p>Retrieves the arithmetic average of all good quality raw samples for each calculation interval.</p> <p>Value is the sum of all good quality samples in the interval, divided by the number of good quality samples in the interval. All bad quality samples are ignored. That is RawAverage is equivalent to RawTotal divided by the Count.</p> <p>For Quality, if there are no raw samples in the interval or if they all are bad quality, then the percentage of good is 0. Otherwise, the percentage of good is always 100, even if the interval contains bad quality samples.</p> <p>RawAverage is useful for calculating an accurate average when a sufficient number of raw samples are collected.</p>	7
RawStandardDeviation	<p>Retrieves the arithmetic standard deviation of raw values for each calculation interval.</p> <p>For Value, any raw point of bad data quality is ignored.</p> <p>For Quality, if there are no raw samples in the interval or they all have bad quality, then the percentage of good is 0. Otherwise, the percentage of good is always 100, even if the interval contains bad quality samples.</p> <p>RawStandardDeviation is useful for calculating an accurate standard deviation when</p>	8

Calculation Mode Type	Description	Value
	a sufficient number of raw samples are collected.	
RawTotal	<p>Retrieves the arithmetic total (sum) of sampled values for each interval.</p> <p>Value is the sum of the good quality values of all raw samples in the interval. All bad quality samples are ignored.</p> <p>For Quality, the percentage of good samples is always 100, even if the interval does not contain any raw samples or it contains only bad quality samples.</p> <p>If the same start and end times are used, and the time span is treated as a single interval, then all values are added together.</p> <p>RawTotal is useful for calculating an accurate total when a sufficient number of raw samples are collected. Note that unlike ihTotal, this is a simple sum with no assumption that the values are rate values.</p>	9
MinimumTime	Retrieves the timestamp of the minimum value found within each calculation interval. It can be a raw or an interpolated value. The minimum must be a good data quality sample.	10
MaximumTime	Retrieves the timestamp of the maximum value found within each calculation interval. It can be a raw or an interpolated value. The maximum must be a good data quality sample.	11
TimeGood	Retrieves the amount of time (milliseconds) during the interval when the data is of good quality and the filter condition is met.	12

Calculation Mode Type	Description	Value
StateCount	Retrieves the amount of time a tag uses to transition to another state from a previous state during a time interval.	13
StateTime	Retrieves the duration that a tag stayed in a given state within an interval.	14
OPCQAnd	<p>Retrieves the OPCQAND, bit-wise AND operation of all the 16-bit OPC qualities of the raw samples stored in the specified interval.</p> <p>Note that OPC Quality is a subfield for Quality-Value-Timestamp (QVT), so when this calculation mode is used, OPC Quality is considered for calculation.</p>	15
OPCQOr	<p>Retrieves the OPCQOR, bit-wise OR operation of all the 16-bit OPC qualities of the raw samples stored in the specified interval.</p> <p>Note that OPC Quality is a subfield for Quality-Value-Timestamp (QVT), so when this calculation mode is used, OPC Quality is considered for calculation.</p>	16
FirstRawValue	<p>Retrieves the first good raw sample value for a given interval.</p> <p>Value is the value of the raw sample, or zero if there are no good raw samples in the interval.</p> <p>For Quality, if there are not good raw samples in the interval, then the percentage of good is 0. Otherwise, the percentage of good is always 100, even if the interval contains bad quality samples. Note that Quality is the same for FirstRawValue and FirstRawTime.</p>	17

Calculation Mode Type	Description	Value
	<p>The Raw sample has a quality of Good, Bad, or Uncertain, and that is converted to a 0 or 100 percent.</p>	
FirstRawTime	<p>Retrieves the first good raw timestamp for a given interval.</p> <p>Value is the timestamp of the sample or the year 1969 if there are no good raw samples in the interval.</p> <p>For Quality, if there are not good raw samples in the interval, then the percentage of good is 0. Otherwise, the percentage of good is always 100, even if the interval contains bad quality samples. Note that Quality is the same for FirstRawValue and FirstRawTime.</p> <p>The Raw sample has a quality of Good, Bad, or Uncertain, and that is converted to a 0 or 100 percent.</p>	18
LastRawValue	<p>Retrieves the last good raw sample value for a given time interval.</p> <p>Value is the value of the raw sample or zero if there are no good raw samples in the interval.</p> <p>For Quality, if there are no good raw samples in the interval, the percentage of good samples is 0. Otherwise, the percentage of good is always 100, even if the interval contains bad samples. Note that Quality is the same for LastRawValue and LastRawTime.</p> <p>The Raw sample has a quality of Good, Bad, or Uncertain, and that is converted to a 0 or 100 percent.</p>	19

Calculation Mode Type	Description	Value
LastRawTime	<p>Retrieves the last good timestamp of the last value for a given time interval.</p> <p>Value is the timestamp of the sample or the year 1969 if there are no good raw samples in the interval.</p> <p>For Quality, if there are no good raw samples in the interval, the percentage of good samples is 0. Otherwise, the percentage of good is always 100, even if the interval contains bad samples. Note that Quality is the same for LastRawValue and LastRawTime.</p> <p>The Raw sample has a quality of Good, Bad, or Uncertain, and that is converted to a 0 or 100 percent.</p>	20
TagStats	Retrieves the statistics for a tag from the archive stored in the specified interval.	21

FilterModeType Parameter

The FilterModeType parameter defines how time periods before and after transitions in the filter condition should be handled.

When the FilterModeType parameter is applied, then the Start time and End time are specified as:

- ExactTime
- BeforeTime
- AfterTime
- BeforeAndAfterTime

For example, AfterTime indicates that the filter condition should be True starting at the timestamp of the archive value that triggered the True condition, and leading up to the timestamp of the archive value that triggered the False condition.

Properties	Description	Value
ExactTime	Retrieves data for the exact times that the filter condition is True.	1
BeforeTime	Retrieves data from the timestamp of the last False filter condition to the timestamp of the next True condition.	2
AfterTime	Retrieves data from the timestamp of the True filter condition to the timestamp of the next False condition.	3
BeforeAndAfterTime	Retrieves data from the timestamp of the last False filter condition to the timestamp of the next False condition.	4

ReturnDataFields Parameter

The ReturnDataFields bitwise parameter specifies which data fields are returned in a query. Using it in a query returns data such as TimeStamp, and each field returns a Boolean value.

Each time-series data sample contains QVT (quality, value, and timestamp) values. If ReturnDataFields is not provided, then the default value of 0 is considered, and all QVT values are returned for each data sample. To return one of the data field properties, such as TimeStamp, use the TimeStamp option as shown in the table.

Properties	Description	Field value (Boolean)
All Fields	Specifies that all data fields are returned in the query.	0 (0000)
TimeStamp	The time stamp of the collected sample or an interval time stamp. When specified in the query, returns the TimeStamp property.	1 (0001)
Value	The collected value or sampled value; the data type of the value will be the same	2 (0010)

Properties	Description	Field value (Boolean)
	data type as the tag's raw data.	
Quality	<p>When specified in the query, returns the Quality property. Each sample in Current Value and Raw query retrieval has a quality of:</p> <ul style="list-style-type: none"> • Good (3) • Not Available (2) • Uncertain (1) • Bad (0) <p>Interpolated and Lab Retrieval express quality as "percent good".</p>	4 (0100)

Payload Parameter

This parameter queries for the tag properties requested from the server.

Use the Payload parameter to query for all the tag properties to return from the server. In the Update Tag Configuration API, you must provide the actual tag property value. However, in the Get Tag Properties API, you must provide the property name and value of 1 (true), so the property can be read from the server and returned.

The properties listed in the following table are valid in APIs that use the Payload parameter, unless otherwise specified. For Property Names used in the Get Tag Properties API, the property name is always a Boolean (true/false) value, while it can be a string or integer for other APIs.

Property Name	Property Type	Description
AllFields	Boolean	Used for Get Tag Properties API.
Name	Boolean, String	Used for the Get Tag Properties API, Add Single Tag API, and Add Bulk Tags API.
Description	String	
EngineeringUnits	String	

Property Name	Property Type	Description
Comment	String	
DataType : ihDataType	SignedInteger	Type definition is an enumerated type "ihDataType". <pre> { ihDataTypeUndefined = 0, ihScaled, ihFloat, ihDoubleFloat, ihInteger, ihDoubleInteger, ihFixedString, ihVariableString, ihBlob, ihTime, ihInt64, ihUInt64, ihUInt32, ihUInt16, ihByte, ihBool, ihMultiField, ihArray, ihMaxDataType } ihDataType;</pre>
FixedStringLength	UnsignedChar	
CollectorName	String	
SourceAddress	String	
CollectionType : ihCollectionType	SignedInteger	Type definition is an enumerated type "ihCollectionType". <pre> { ihUnsolicited = 1, ihPolled } ihCollectionType;</pre>

Property Name	Property Type	Description
CollectionInterval	SignedIntegral	
CollectionOffset	UnsignedLong	
LoadBalancing	Boolean	
TimeStampType : ihTimeStampType	SignedIntegral	Type definition is an enumerated type "ihTimeStampType". <pre>{ ihSource = 1, ihInterface, } ihTimeStampType;</pre>
HiEngineeringUnits	Double	
LoEngineeringUnits	Double	
InputScaling	Boolean	
HiScale	Double	
LoScale	Double	
CollectorCompression	Boolean	
CollectorDeadbandPercentRange	Float	
ArchiveCompression	Boolean	
ArchiveDeadbandPercentRange	Float	
General1	String	
General2	String	
General3	String	
General4	String	
General5	String	
ReadSecurityGroup	String	
WriteSecurityGroup	String	

Property Name	Property Type	Description
AdministratorSecurityGroup	String	
LastModified	Boolean	Used for Get Tag Properties API.
LastModifiedUser	Boolean	Used for Get Tag Properties API.
InterfaceType	Boolean	Used for Get Tag Properties API.
CollectorType : ihInterfaceType	SignedIntegral	<p>Type definition is an enumerated type "ihInterfaceType".</p> <pre> { ihInterfaceUndefined = 0, ihIFix, ihRandom, ihOPC, ihFile, ihIFixLabData, ihManualEntry, ihOther, ihCalcEngine, ihServerToServer, ihPI, ihOPCAE, ihCIMPE, ihPIDistributor, ihCIMME, ihPerfTag, ihCustom, ihServerToServerDistributor, ihWindowsPerfMon, } ihInterfaceType; </pre>
UTCBias	SignedIntegral	
AverageCollectionTime	Boolean	Used for Get Tag Properties API.
CalculationDependencies	StringArray	
CollectionDisabled	Boolean	

Property Name	Property Type	Description
ArchiveCompressionTimeout	Unsigned-Long	
CollectorCompressionTimeout	Unsigned-Long	
SpikeLogic	Boolean	
SpikeLogicOverride	Boolean	
CollectorAbsoluteDeadbanding	Boolean	
CollectorAbsoluteDeadband	Double	
ArchiveAbsoluteDeadbanding	Boolean	
ArchiveAbsoluteDeadband	Double	
StepValue	Boolean	
TimeResolution : ihTimeResolution	SignedIntegral	<p>Type definition is an enumerated type "ihTimeResolution".</p> <pre> { ihSeconds = 0, ihMilliseconds, ihMicroseconds, ihNanoseconds } ihTimeResolution; </pre>
ConditionCollectionEnabled	Boolean	
ConditionCollectionTriggerTag	String	
ConditionCollectionComparison : ihConditionCollectionComparison	SignedIntegral	<p>Type definition is an enumerated type "ihConditionCollectionComparison".</p> <pre> { ihConditionComparisonUndefined = 0, ihConditionComparisonEqual, ihConditionComparisonLessThan, ihConditionComparisonLessThanEqual, ihConditionComparisonGreaterThan, ihConditionComparisonGreaterThanEqual, </pre>

Property Name	Property Type	Description
		<pre>ihConditionComparisonNotEqual } ihConditionCollectionComparison;</pre>
ConditionCollectionCompareValue	String	
ConditionCollectionMarkers	Boolean	
Calculation	String	When the Calculation field is used, then two more conditions are required. Calculation is not a specific field for a tag property. If the tag's collector or interface type is Server-to-server and the Calculation field is set (not Null), then the field value is set to the source address.
TagId	Boolean	Used for Get Tag Properties API.
EnumeratedSetName	String	
DataStoreName	String	
DefaultQueryModifiers	Long Long	
UserDefinedTypeName	String	
NumberOfElements	SignedIntegral	
DataDensity : ihTagDataDensity	SignedIntegral	<p>Type definition is an enumerated type "ihTagDataDensity".</p> <pre>{ ihDataDensityUndefined = 0, ihDataDensityMinimum = 1, ihDataDensityMedium = 4, ihDataDensityMaximum = 7 } ihTagDataDensity;</pre>
CalcType : ihTagCalcType	SignedIntegral	<p>Type definition is an enumerated type "ihCalcType".</p> <pre>{ ihRawTag = 0, ihAnalyticTag = 1,</pre>

Property Name	Property Type	Description
		<pre>ihPythonExprTag = 2 } ihTagCalcType;</pre>
HasAlias	Boolean	Used for Get Tag Properties API.
IsStale	Boolean	Used for Get Tag Properties API.

Error Code Definitions

The following table provides the values and definitions for the ErrorCode parameter.

Table 1. Error Code Definitions

Error Code Value:	Error Code Definition
Success = 0	Operation successful.
Failed = -1	Operation failed.
Timeout = -2	Operation failed due to timeout.
NotConnected = -3	Not connected to Historian server.
CollectorNotFound = -4	The given collector does not exist on the server.
NotSupported = -5	Operation not supported.
DuplicateData = -6	Attempt to overwrite an existing data sample.
InvalidUsername = -7	Bad user name or password.
AccessDenied = -8	Insufficient permissions for operation.
WriteInFuture = -9	Attempted data write too far in the future.
WriteArchiveOffline = -10	Attempted data write to an offline archive.
WriteArchiveReadOnly = -11	Attempted data write to a read-only archive.
WriteOutsideActiveRange = -12	Attempted data write beyond the configured active range.
WriteNoArchiveAvailable = -13	Attempted data write with no available archives.
InvalidTagname = -14	The requested tag was not found.
LicensedTagCountExceeded = -15	Number of licensed tags exceeded.

Table 1. Error Code Definitions (continued)

Error Code Value:	Error Code Definition
LicensedConnectionCountExceeded = -16	Number of licensed server connections exceeded.
InternalLicenseError = -17	Internal license error.
NoValue = -18	No available tag data.
DuplicateCollector = -19	The given collector name already exists on the server.
NotLicensed = -20	Server or feature is not licensed.
CircularReference = -21	Circular reference detected in calculation.
BackupInsufficientSpace = -22	Insufficient disk space to perform backup.
InvalidServerVersion = -23	Operation unsupported due to server version.
QueryResultSizeExceeded = -24	Upper limit on query results exceeded.
DeleteOutsideActiveRange = -25	Attempted data delete outside allowed modification interval.
AlarmArchiverUnavailable = -26	Alarms and Events subsystem unreachable.
ArgumentException = -27	A supplied argument is invalid.
ArgumentNullException = -28	A supplied argument is NULL.
ArgumentOutOfRangeException = -29	A supplied argument is outside the valid range.
InvalidEnumeratedSet = -30	The requested Enumerated Set was not found.
InvalidDataStore = -31	The requested data store was not found.
NotPermitted = -32	Operation not permitted.
InvalidCustomDataType = -33	The Custom data type is not supported.
ihSTATUS_EXISTING_USERDEF_REFERENCES = -34	N/A
ihSTATUS_INVALID_TAGNAME_DELETEDTAG = -35	N/A
ihSTATUS_INVALID_DHS_NODENAME = -36	N/A
ihSTATUS_DHS_SERVICE_IN_USE = -37	N/A

Table 1. Error Code Definitions (continued)

Error Code Value:	Error Code Definition
ihSTATUS_DHS_STORAGE_IN_USE = -38	N/A
ihSTATUS_DHS_TOO_MANY_NODES_IN_MIRROR = -39	N/A
ihSTATUS_ARCHIVE_IN_SYNC = -40	N/A
InvalidArchiveName= -41	N/A
InvalidSession = 1	Session id is invalid.
SessionExpired = 2	Session has expired.
UnknownError = 3	Unknown error, please check server log.
NoValidClientBufferManager= 4	No valid client buffer manager.
NoValueInDataSet = 5	No value in returned data set.
TagNotExisting = 6	Tag doesn't exist.
ClientBufferManagerCommunicationError = 7	Service call to central buffer server fail.
TagTypeNotSupported=8	Tag type is not supported.
ValueTypeNotMatchTagDataType = 9	Value type doesn't match tag data type.
InvalidParameter=10	Invalid query parameter.
TagSearchResultIsHuge = 11	Tag Search Criteria result was more than 5000.
InvalidHistorianServer=12	No valid server or historian server name isn't in the server list.
ihSTATUS_INVALID_INTERFACETYPE = -49	The collector type is not valid. For a list of collector types, refer to Collector Type and Subtype (on page 116) .
ihSTATUS_INTERFACE_START_FAIL = -50	Starting the collector has failed.
ihSTATUS_INTERFACE_STOP_FAIL = -51	Stopping the collector has failed.

Historian REST APIs

Overview of the Historian REST APIs

Historian provides REST APIs to manage Historian systems, collectors, data stores, and tags. In addition, it provides APIs to install and manage collector instances.



Important:

Port 443 is used in examples and sample code. If you copy and paste the sample code from Help, you must change this port to your installed port.

Managing Systems

The Get DHS Machines API

Using the Get DHS Machines API, you can view the list of DHS machines in a location.

METHOD	GET
URI	<code>https://<historianservername>/historian-rest-api/v1/dhsmachines?storageName=</code>
SAMPLE QUERY PARAM GET URL	<code>https://<historianservername>/historian-rest-api/v1/dhsmachines?storageName=xx</code>
SAMPLE RESPONSE	<pre>{ "ErrorCode": 0, "ErrorMessage": null, "Data": [{ "NodeName": "xyz", "IsAlreadyAdded": true }] }</pre>

	<pre>] } } </pre>
SAMPLE cURL COMMAND	<pre> curl -i -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://<historianservername>/historian-rest-a pi/v1/dhsmachines?storageName=xxx </pre>

Table 2. Query Parameters

Parameter	Description	Required?	Values
storageName	The value of the location whose DHS machines you want to view.	Yes	String

Table 3. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Get DHS Services API

Using the Get DHS Services API, you can view the list of DHS services in a data store.

METHOD	GET
URI	<pre> https://<historianservername>/historian-rest-a pi/v1/dhsservices?dhSServiceMask=&withReason=fa lse </pre>
SAMPLE QUERY PARAM GET URL	<pre> https://<historianservername>/historian-rest-a pi/v1/dhsservices?dhSServiceMask =*&withReason=false </pre>

SAMPLE RESPONSE

```
{

  "ErrorCode": 0,

  "ErrorMessage": null,

  "Data": [

    {

      "LogicalName":

      "ConfigManager_NPI212611749M1",

      "NodeName": "NPI212611749M1",

      "ServiceType": 4,

      "Status": 1,

      "TCPPort": 14002

    },

    {

      "LogicalName":

      "DataArchiver_NPI212611749M1",

      "NodeName": "NPI212611749M1",

      "ServiceType": 2,

      "Status": 1,

      "TCPPort": 14001

    },

  ],

}
```

```
{  
  
  "LogicalName":  
  "ClientManager_NPI212611749M1",  
  
  "NodeName": "NPI212611749M1",  
  
  "ServiceType": 3,  
  
  "Status": 1,  
  
  "TCPPort": 14000  
  
},  
  
{  
  
  "LogicalName":  
  "DiagnosticsManager_NPI212611749M1",  
  
  "NodeName": "NPI212611749M1",  
  
  "ServiceType": 5,  
  
  "Status": 1,  
  
  "TCPPort": 14003  
  
},  
  
{  
  
  "LogicalName":  
  "DataArchiver_distmachine2",  
  
  "NodeName": "distmachine2",
```

```
        "ServiceType": 2,

        "Status": 0,

        "TCPPort": 14001

    },

    {

        "LogicalName":

        "DataArchiver_distmachine1",

        "NodeName": "distmachine1",

        "ServiceType": 2,

        "Status": 1,

        "TCPPort": 14001

    },

    {

        "LogicalName":

        "ClientManager_distmachine1",

        "NodeName": "distmachine1",

        "ServiceType": 3,

        "Status": 1,

        "TCPPort": 14000

    }

}
```

	<pre> }, { "LogicalName": "DiagnosticsManager_distmachine1", "NodeName": "distmachine1", "ServiceType": 5, "Status": 0, "TCPPort": 14003 }] } } </pre>
<p>SAMPLE cURL COMMAND</p>	<pre> curl -i -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://<historianservername>/historian-rest-a pi/v1/dhsservices?dhSServiceMask=*&withReason=fa lse </pre>

Table 4. Query Parameters

Parameter	Description	Required?	Values
withReason	Indicates whether the reason must be retrieved in the API response.	Yes	Boolean
dhSServiceMask	The value of the DHS service mask.	Yes	String

Table 5. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Get Server Properties API

Using the Get Server Properties API, you can view the list of properties of a server.

METHOD	GET
URI	<code>https://<historianservername>/historian-rest-api/v1/serverproperties</code>
SAMPLE QUERY PARAM GET URL	<code>https://<historianservername>/historian-rest-api/v1/serverproperties</code>
SAMPLE RESPONSE	<pre>{ "ErrorCode": 0, "ErrorMessage": null, "Data": { "Storages": [{ "StorageName": "System Storage", "StorageType": 2, "NumberOfDataStores": 1, "NumberOfArchivers": 0, "DataStores": [</pre>

```
        "System"

    ],

    "Id":
"861C2743-72E0-46FC-9B31-90E28CC39B8D",

    "IsDefault": false,

    "LastModifiedUser": null,

    "LastModifiedTime":
"1970-01-01T00:00:00.000Z",

    "ArchiverServices": []

},

{

    "StorageName": "xyz",

    "StorageType": 0,

    "NumberOfDataStores": 3,

    "NumberOfArchivers": 1,

    "DataStores": [

        "ScadaBuffer",

        "DHSSystem",

        "User"
```

```
    ],  
  
    "Id":  
    "5F267DF3-879A-4222-8A0E-D31EDEA83C14",  
  
    "IsDefault": true,  
  
    "LastModifiedUser": null,  
  
    "LastModifiedTime":  
    "1970-01-01T00:00:00.000Z",  
  
    "ArchiverServices": [  
  
        {  
  
            "LogicalName":  
            "DataArchiver_xyz",  
  
            "NodeName": "xyz",  
  
            "ServiceType": 2,  
  
            "IsAlreadyAdded": true,  
  
            "TCPPort": 14001  
  
        }  
  
    ]  
  
    }  
  
    ],  
  
    "Servers": [  

```



```
{  
  
  "LogicalName":  
"DataArchiver_xyz0",  
  
  "NodeName": "xyz",  
  
  "ServiceType": 2,  
  
  "Status": 1,  
  
  "TCPPort": 14001,  
  
  "MemoryVMSize": "4778",  
  
  "TotalFailedWrites": "0",  
  
  "WriteCacheHitRatio": "0.748",  
  
  "TotalOutOfOrder": "3",  
  
  "CompressionRatio": "0.321",  
  
  "ReadQueueSize": "0",  
  
  "WriteQueueSize": "0",  
  
  "MsgQueueSize": "0",  
  
  "ReadQueueProcessingRate": "1",  
  
  "WriteQueueProcessingRate":  
"31",  
  
  "MsgQueueProcessingRate": "0"  
  
}
```

	<pre>] } } </pre>
SAMPLE cURL COMMAND	<pre> curl -i -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://<historianservername>/historian-rest-a pi/v1/serverproperties </pre>

Table 6. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Get System Statistics API

Using the Get System Statistics API, you can view the statistics of a system.

METHOD	GET
URI	<pre> https://<historianservername>/historian-rest-a pi/v1/systemstats </pre>
SAMPLE QUERY PARAM GET URL	<pre> https://<historianservername>/historian-rest-a pi/v1/systemstats </pre>
SAMPLE RESPONSE	<pre> { "ErrorCode": 0, "ErrorMessage": null, "Data": { "Utilization": { </pre>

```
"WriteCacheHitRatio": "0.499",  
  
"SpaceConsumptionRate": "",  
  
"CompressionRatio": "0.199",  
  
"ReadQueueSize": "0",  
  
"WriteQueueSize": "0",  
  
"MsgQueueSize": "0",  
  
"ReadQueueProcessRate": "3",  
  
"WriteQueueProcessRate": "0",  
  
"MsgQueueProcessRate": "0",  
  
"MemoryVMUsage": "62",  
  
"OutOfOrderRate": "0",  
  
"ReadThreadUsage": "0",  
  
"WriteThreadUsage": "0",  
  
"FailedWriteRate": "0",  
  
"DiskFreeSpace": "59828"  
  
},  
  
"AlarmEvents": {  
  
"AverageAlarmRate": ""
```

```
    },  
  
    "TotalCollectors": {  
  
        "TotalCollectors": 1,  
  
        "RunningCollectors": 1,  
  
        "StoppedCollectors": 0,  
  
        "UnknownCollectors": 0  
  
    },  
  
    "Licence": {  
  
        "ActualDataStores": 3,  
  
        "MaxDataStores": 200,  
  
        "ActualTags": 0,  
  
        "MaxTags": 2147483647,  
  
        "ActualUsers": 0,  
  
        "MaxUsers": 1000  
  
    }  
  
    }  
  
}
```

SAMPLE cURL COMMAND

```
curl -i -H "Accept: application/json" -H  
"Authorization: Bearer <TOKEN>"
```

	<pre>https://<historianservername>/historian-rest-api/systemstats</pre>
--	---

Table 7. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Get Read Sample and Receive Rate API

Using the Get Read Sample and Receive Rate API, you can view the read rate and receive rate of a system.

METHOD	GET
URI	<p>Read Sample Rate</p> <pre>https://<historianservername>/historian-rest-api/v1/performancecounter/perftagdata/PerfTag_AverageEventRate/--/starttime/endtime/interval</pre> <p>Receive Rate</p> <pre>https://<historianservername>/historian-rest-api/v1/performancecounter/perftagdata/PerfTag_AverageReadRawRate/--/starttime/endtime/interval</pre>
SAMPLE GET URI	<pre>https://<historianservername>/historian-rest-api/v1/performancecounter/perftagdata/PerfTag_AverageEventRate/--/2020-12-15T11:19:01.719Z/2020-12-15T12:19:01.719Z/360000</pre>
SAMPLE RESPONSE	<pre>{ "ErrorCode": 0,</pre>

```
"ErrorMessage": null,  
  
"Data": [  
  
  {  
  
    "TagName":  
"PerfTag_AverageEventRate",  
  
    "ErrorCode": 0,  
  
    "DataType": "DoubleFloat",  
  
    "Samples": [  
  
      {  
  
        "TimeStamp":  
"2020-11-18T05:35:22.612Z",  
  
        "Value": "0",  
  
        "Quality": 0  
  
      },  
  
      {  
  
        "TimeStamp":  
"2020-11-18T05:47:22.612Z",  
  
        "Value": "0",  
  
        "Quality": 0  
  
      },  
  
    ]  
  
  }  
  
]
```

```
{  
  
    "TimeStamp":  
    "2020-11-18T05:53:22.612Z",  
  
    "Value": "0",  
  
    "Quality": 0  
  
},  
  
{  
  
    "TimeStamp":  
    "2020-11-18T06:11:22.612Z",  
  
    "Value": "0",  
  
    "Quality": 0  
  
},  
  
{  
  
    "TimeStamp":  
    "2020-11-18T06:29:22.612Z",  
  
    "Value": "0",  
  
    "Quality": 0  
  
}  
  
]  
  
}
```

	<pre> } } </pre>
SAMPLE cURL COMMAND	<pre> curl -i -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://<historianservername>/historian-rest-a pi/v1/performancecounter/perftagdata/ PerfTag_AverageEventRate/--/2020-12-15T11:19:01 .719Z/2020-12-15T12:19:01.719Z/360000 </pre>

Table 8. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Get Storages API

Using the Get Storages API, you can view the list of locations in a system.

METHOD	GET
URI	<pre> https://<historianservername>/historian-rest-a pi/v1/storages?storageMask= </pre>
SAMPLE QUERY PARAM GET URL	<pre> https://<historianservername>/historian-rest-a pi/v1/storages?storageMask=* </pre>
SAMPLE RESPONSE	<pre> { "ErrorCode": 0, "ErrorMessage": null, "Data": [{ </pre>


```
    "StorageName": "System Storage",

    "StorageType": 2,

    "NumberOfDataStores": 1,

    "NumberOfArchivers": 0,

    "DataStores": [

        "System"

    ],

    "Id":

"861C2743-72E0-46FC-9B31-90E28CC39B8D",

    "IsDefault": false,

    "LastModifiedUser": null,

    "LastModifiedTime":

"1970-01-01T00:00:00.000Z",

    "ArchiverServices": []

},

{

    "StorageName": "srinivaswin10",

    "StorageType": 0,

    "NumberOfDataStores": 3,
```

```
"NumberOfArchivers": 1,

"DataStores": [

  "ScadaBuffer",

  "DHSystem",

  "User"

],

"Id":

"5F267DF3-879A-4222-8A0E-D31EDEA83C14",

"IsDefault": true,

"LastModifiedUser": null,

"LastModifiedTime":

"1970-01-01T00:00:00.000Z",

"ArchiverServices": [

  {

    "LogicalName":

    "DataArchiver_xyz",

    "NodeName": "xyz",

    "ServiceType": 2,

    "TCPPort": 14001

  }

]
```

	<pre>] }] } </pre>
SAMPLE cURL COMMAND	<pre> curl -i -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://<historianservername>/historian-rest-a pi/v1/storages?storageMask=* </pre>

Table 9. Query Parameters

Parameter	Description	Required?	Values
storageMask	The value of the location mask.	No	String

Table 10. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Add Machine API

Using the Add Machine API, you can add a server in a Historian system.

METHOD	POST
URI	<pre> https://<historianservername>/historian-rest-a pi/v1/machine </pre>
SAMPLE URI	<pre> https://<historianservername>/historian-rest-a pi/v1/machine Payload { </pre>

	<pre>"nodeName": "node1" } }</pre>
SAMPLE RESPONSE	<pre>{ "ErrorCode": 0, "ErrorMessage": null }</pre>
SAMPLE cURL COMMAND	<pre>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -d "{ \"nodeName\": \"node1\"}" -X POST https://<historianservername>/historian-rest-a pi/v1/machine</pre>

Table 11. Query Parameters

Parameter	Description	Required?	Values
Payload	Contains the machine name of the server that you want to add.	Yes	Multiple

Table 12. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Delete Machine API

Using the Delete Machine API, you can remove a server from a Historian system.

METHOD	DELETE
URI	<pre>https://<historianservername>/historian-rest-a pi/v1/machine</pre>

<p>SAMPLE URI</p>	<pre>https://<historianservername>/historian-rest-api/v1/machine Payload { "nodeName": "", }</pre>
<p>SAMPLE RESPONSE</p>	<pre>{ "ErrorCode": 0, "ErrorMessage": null }</pre>
<p>SAMPLE cURL COMMAND</p>	<pre>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -d "{ \"nodeName \": \"name\"}" -X DELETE https://<historianservername>/historian-rest-api/v1/machine</pre>

Table 13. Query Parameters

Parameter	Description	Required?	Values
Payload	Contains the name of the machine that you want to remove.	Yes	Multiple

Table 14. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Create Mirror Group API

Using the Create Mirror Group API, you can create a mirror group.

METHOD	POST
URI	<code>https://<historianservername>/historian-rest-api/v1/mirrorsgroup</code>
SAMPLE URI	<code>https://<historianservername>/historian-rest-api/v1/mirrorsgroup</code>
	Payload
	<pre>{ "mirrorStorageName": "storagename", "nodes": "node1;node2" }</pre>
SAMPLE RESPONSE	<pre>{ "ErrorCode": 0, "ErrorMessage": null }</pre>
SAMPLE cURL COMMAND	<pre>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -d "{ \" mirrorStorageName \": \"name\", \" nodes \": \"xx;yy\"}" -X POST https://<historianservername>/historian-rest-api/v1/mirrorsgroup</pre>

Table 15. Query Parameters

Parameter	Description	Required?	Values
Payload	Contains the mirror group name and the servers you want to add to the group.	Yes	Multiple

Table 16. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Add Mirror Machine API

Using the Add Mirror Machine API, you can add a server to a mirror group.

METHOD	POST
URI	<code>https://<historianservername>/historian-rest-api/v1/mirrormachine</code>
SAMPLE URI	<pre> https://<historianservername>/historian-rest-api/v1/mirrormachine Payload { "mirrorStorageName": "Mirror2", "mirrorMachineName": "distmachine1" } </pre>
SAMPLE RESPONSE	<pre> { "ErrorCode": 0, "ErrorMessage": null </pre>

	<pre>} </pre>
SAMPLE cURL COMMAND	<pre>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -d "{ \" mirrorStorageName \": \"name\", \" nodes \": \"xx;yy\"}" -X POST https://<historianservername>/historian-rest-a pi/v1/mirrormachine</pre>

Table 17. Query Parameters

Parameter	Description	Required?	Values
Payload	Contains the machine name of the server that you want to add.	Yes	Multiple

Table 18. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Mirror Group Update API

Using the Mirror Group Update API, you can update the name of a mirror group.

METHOD	PUT
URI	<pre>https://<historianservername>/historian-rest-a pi/v1/mirrorsgroup</pre>
SAMPLE URI	<pre>https://<historianservername>/historian-rest-a pi/v1/mirrorsgroup Payload { "mirrorStorageName": "Mirror2",</pre>

	<pre>"mirrorStorageNewName": "Mirror3" }</pre>
SAMPLE RESPONSE	<pre>{ "ErrorCode": 0, "ErrorMessage": null }</pre>
SAMPLE cURL COMMAND	<pre>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -d "{ \"mirrorStorageName \": \"name\", \" mirrorStorageNewName \": \"sname\"}" -X PUT https://<historianservername>/historian-rest-a pi/v1/mirrorsgroup</pre>

Table 19. Query Parameters

Parameter	Description	Required?	Values
Payload	Contains the existing and new names of the mirror group that you want to rename.	Yes	Multiple

Table 20. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Delete Mirror Machine API

Using the Delete Mirror Machine API, you can remove a server from a mirror group.

METHOD	DELETE
URI	<pre>https://<historianservername>/historian-rest-api/v1/mirrormachine</pre>
SAMPLE URI	<pre>https://<historianservername>/historian-rest-api/v1/mirrormachine Payload { "mirrorStorageName": "Mirror 2", "mirrorMachineName": "dismachine1" }</pre>
SAMPLE RESPONSE	<pre>{ "ErrorCode": 0, "ErrorMessage": null }</pre>
SAMPLE cURL COMMAND	<pre>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -d "{ \"mirrorStorageName \": \"name\", \" mirrorMachineName\": \"name\"}" -X DELETE https://<historianservername>/historian-rest-api/v1/mirrormachine</pre>

Table 21. Query Parameters

Parameter	Description	Required?	Values
Payload	Contains the name of the machine that you want to remove	Yes	Multiple

Table 21. Query Parameters (continued)

Parameter	Description	Required?	Values
	and the name of the mirror group.		

Table 22. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Delete Mirror Group API

Using the Delete Mirror Group API, you can delete a mirror group.

METHOD	DELETE
URI	<code>https://<historianservername>/historian-rest-api/v1/mirrorsgroup</code>
SAMPLE URI	<code>https://<historianservername>/historian-rest-api/v1/mirrorsgroup</code> Payload <pre>{ "mirrorStorageName": "Mirror3", }</pre>
SAMPLE RESPONSE	<pre>{ "ErrorCode": 0, "ErrorMessage": null }</pre>

<p>SAMPLE cURL COMMAND</p>	<pre>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -d "{ \"mirrorStorageName \": \"name\"}" -X DELETE https://<historianservername>/historian-rest-a pi/v1/mirrorgroup</pre>
----------------------------	--

Table 23. Query Parameters

Parameter	Description	Required?	Values
Payload	Contains the name of the machine that you want to remove.	Yes	Multiple

Table 24. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Local OPC Servers API

Using the Local OPC Servers API, you can view the list of OPC servers installed on a specified machine.

METHOD	GET
URI	<pre>http://<historianservername>/v1/localopcserve rs/<machine name></pre>
SAMPLE QUERY PARAM GET URL	<pre>http://<historianservername>/v1/localopcserve rs/<machine name></pre>
SAMPLE RESPONSE	<pre>{ "ErrorCode": 0, "ErrorMessage": null, "ServerIDs": ["ID1", "ID2"] }</pre>

	} }
SAMPLE cURL COMMAND	<pre>curl -i -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://<historianservername>/historian-rest-a pi/v1/localopcserver/xyz</pre>

Table 25. Query Parameters

Parameter	Description	Required?	Values
machine name	The machine name of the OPC server.	Yes	String

Table 26. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Local OPC AE Servers API

Using the Local OPC AE Servers API, you can view the list of OPC Alarms and Events servers installed on a specified machine.

METHOD	GET
URI	<pre>http://<historianservername>/v1/localopcaeserve rs/<machine name></pre>
SAMPLE QUERY PARAM GET URL	<pre>http://<historianservername>/v1/localopcaeserve rs/<machine name></pre>
SAMPLE RESPONSE	<pre>{ "ErrorCode": 0, "ErrorMessage": null, "ServerIDs": ["ID1", "ID2"] }</pre>

	} }
SAMPLE cURL COMMAND	<pre>curl -i -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://<historianservername>/historian-rest-a pi/v1/localopcaeservers/abc</pre>

Table 27. Query Parameters

Parameter	Description	Required?	Values
machine name	The machine name of the OPC Alarms and Events server.	Yes	String

Table 28. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Local OPC HDA Servers API

Using the Local OPC HDA Servers API, you can view the list of OPC HDA servers installed on a specified machine.

METHOD	GET
URI	<pre>http://<historianservername>/v1/localopchdaserve rs/<machine name></pre>
SAMPLE QUERY PARAM GET URL	<pre>http://<historianservername>/v1/localopchdaserve rs/<machine name></pre>
SAMPLE RESPONSE	<pre>{ "ErrorCode": 0, "ErrorMessage": null, "ServerIDs": ["ID1", "ID2"] }</pre>

	} }
SAMPLE cURL COMMAND	<pre>curl -i -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://<historianservername>/historian-rest-a pi/v1/localopchdaservers/xyz</pre>

Table 29. Query Parameters

Parameter	Description	Required?	Values
machine name	The machine name of the OPC HDA server.	Yes	String

Table 30. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The DHS Service Port Update API

Using the DHS Service Port Update API, you can change the port and other details of a DHS service.

METHOD	PUT
URI	<pre>https://<historianservername>/historian-rest-a pi/v1/dhsservice/<DHS service name></pre>
SAMPLE URI	<pre>https://<historianservername>/historian-rest-a pi/v1/dhsservice/ DataArchiver_xxx { "LogicalName": "DataArchiver_xxx", "NodeName": "xxx",</pre>

	<pre>"ServiceType": 2, "Status": 1, "TCPPort": 14005 }</pre>
<p>SAMPLE RESPONSE</p>	<pre>{ "ErrorCode": 0, "ErrorMessage": null, "Data": { "LogicalName": "DataArchiver_xxx", "NodeName": "xxx", "ServiceType": 2, "Status": 1, "TCPPort": 14005 } }</pre>
<p>SAMPLE cURL COMMAND</p>	<pre>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -d "{ \" LogicalName \":\": \"DataArchiver_xxx \", \" NodeName \": \"xxx\", \" ServiceType \": 2, \" Status\": 1, \" TCPPort \": 14005}" -X PUT https://<historianservername>/historian-rest-a pi/v1/dhsservice/DataArchiver_xxx</pre>

Table 31. Query Parameters

Parameter	Description	Required?	Values
Payload	Contains the values of the attributes of the data store that you want to change.	Yes	Multiple

Table 32. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

Managing Historian Model

The Add Object Type API

Using this API, you can create an object type.

METHOD	POST
URI	<code>https://<historianservername>/historian-model/v2/objectTypes</code>
SAMPLE QUERY PARAM POST URL	<code>https://localhost:443/historian-model/v2/objectTypes</code>
SAMPLE PAYLOAD	<pre>[{ "name": "Vehicle", "description": "Vehicle description", "dataVariables": [], "containedObjectTypes": [], "objectTypeTemplates": [], "lastModifiedUser": "", "lastmodifiedTime": "", "versionNumber": 0 }]</pre>

Table 33. Payload Parameters

Parameter	Description	Required?	Data Type
<code>name</code>	The object type name (must be unique).	Yes	String
<code>description</code>	The description of the object type.	No	String
<code>containedObjectTypes</code>	The list of contained types with details	No	Array
<code>objectTypeTemplates</code>	The list of object type templates, along with their details.	No	Array
<code>lastModifiedUser</code>	The username of the user who created the object type.	Yes	String
<code>lastModifiedTime</code>	The current time.	Yes	String
<code>versionNumber</code>	The version number of the object type. Start with 1 and increment by 1 each time you want to modify the object type.	Yes	Number

Table 34. Response Parameters

Parameter	Data Type	Description
<code>detail</code>	String	Description of the error.
<code>error_code</code>	Integer	The error code. A value of 0 indicates that the operation was successful.
<code>status</code>	String	The http request status.
<code>title</code>	String	The title of the error.
<code>type</code>	String	The type of the error.

The View Object Type API

Using this API, you can access an object type.

METHOD	POST
--------	------

URI	<code>https://<historianservername>/historian-model/v2/objectTypes/info</code>
SAMPLE QUERY PARAM POST URL	<code>https://localhost:443/historian-model/v2/objectTypes/info</code>
SAMPLE RESPONSE	<pre>{ "containedObjectTypes": [], "dataVariables": [], "description": "Vehicle description", "lastModifiedUser": "XYZ.admin", "lastmodifiedTime": "2022-01-21 08:58:20", "name": "Vehicle", "objectTypeTemplates": [{"containedObjectTypes": [], "dataVariables": [], "defaultTemplate": true, "description": "Default_Template_Vehicle", "lastModifiedUser": "XYZ.admin", "lastmodifiedTime": "2022-01-21 08:58:20", "name": "Default_Template_Vehicle", "objectType": "Vehicle", "substituteParameters": [], "versionNumber": 3}], "versionNumber": 3 }</pre>
SAMPLE PAYLOAD	<code>{"databaseName": "XYZ", "name": "Vehicle"}</code>

Table 35. Payload Parameters

Parameter	Description	Required?	Data Type
<code>name</code>	The object type name (must be unique).	Yes	String
<code>databaseName</code>	The name of the Historian server.	Yes	String

Table 36. Response Parameters

Parameter	Data Type	Description
name	String	The object type name (must be unique).
description	String	The description of the object type.
containedObject- Types	Array	The list of contained types with details
dataVariables	Array	The list of variables (with details) in the object type.
objectTypeTem- plates	Array	The list of object type templates, along with their details.
lastModifiedUser	String	The username of the user who created the object type.
lastModifiedTime	String	The current time.
versionNumber	Number	The version number of the object type. Start with 1 and increment by 1 each time you want to modify the object type.

Table 37. objectTypeTemplates Array Parameters

Parameter	Data Type	Description
name	String	The object type name (must be unique).
description	String	The description of the object type.
containedObject- Types	Array	The list of contained types with details
dataVariables	Array	The list of variables (with details) in the object type.
lastModifiedUser	String	The username of the user who created the object type.
lastModifiedTime	String	The current time.
versionNumber	Number	The version number of the object type. Start with 1 and increment by 1 each

Table 37. objectTypeTemplates Array Parameters (continued)

Parameter	Data Type	Description
		time you want to modify the object type.
ObjectType	String	The object type name.
substituteParameters	Array	Blank (not applicable)

The Update Object Type API

Using this API, you can modify an object type.

METHOD	PUT
URI	<code>https://<historianservername>/historian-model/v2/objectTypes</code>
SAMPLE QUERY PARAM POST URL	<code>https://localhost:443/historian-model/v2/objectTypes</code>
SAMPLE RESPONSE	Error code and message, specifying whether the object type has been modified (200 for success and other codes for failure).
HEADER	<ul style="list-style-type: none"> • Content type: multipart/form-data • Boundary: ---WebKitFormBoundary-KeyName
SAMPLE PAYLOAD	<pre> -----WebKitFormBoundaryViBtYYvOfWhTYyhH Content-Disposition: form-data; name="objectTypeInfo"; filename="blob" Content-Type: application/json { "containedObjectTypes":[], "dataVariables": [{ "baseType":"NUMBER", "description":""," </pre>

```
"name": "Speed",
"variableType": "Direct"
},
{
"baseType": "STRING",
"description": "",
"name": "Type",
"variableType": "Indirect"
}
],
"description": "Vehicle description",
"lastModifiedUser": "XYZ.admin",
"lastmodifiedTime": "2022-01-21 11:37:56",
"name": "Vehicle",
"objectTypeTemplates":
[
{
"containedObjectTypes": [],
"dataVariables":
[
{
"baseType": "NUMBER",
"description": "",
"name": "Speed",
"properties":
{"blockType": "AI",
"propertyDefinition": []
},
"required": true,
"templateTagName": "",
"variableType": "DIRECT"
},
{
"baseType": "STRING",
"description": "",
"name": "Type",
"properties":
```

```
{
  "blockType": "AI",
  "propertyDefinition": []
},
"required": true,
"templateTagName": "",
"variableType": "INDIRECT"
}
],
"defaultTemplate": true,
"description": "Default_Template_Vehicle",
"lastModifiedUser": "XYZ.admin",
"lastmodifiedTime": "2022-01-21 11:37:56",
"name": "Default_Template_Vehicle",
"objectType": "Vehicle",
"substituteParameters": [],
"versionNumber": 5
}
],
"versionNumber": 5,
"databaseName": "XYZ",
"containedObjectTypesChanges":
[
{
  "newName": "",
  "oldName": "Engine"
}
],
"dataVariablesChanges":
[
{
  "newName": "",
  "oldName": "Color"
}
],
"objectTypeTemplatesChanges":
[
```

```
{
  "newName": "",
  "oldName": "Car"
}
]
}
```

-----WebKitFormBoundaryViBtYYvOfWhTYyhH--

Table 38. Payload Parameters

Parameter	Description	Required?	Data Type
name	The object type name (must be unique).	Yes	String
description	The description of the object type.	No	String
dataVariables	The list of variables in the object type.	No	Array
containedObjectTypes	The list of contained types with details	No	Array
objectTypeTemplates	The list of object type templates, along with their details.	No	Array
lastModifiedUser	The username of the user who created the object type.	Yes	String
lastModifiedTime	The current time.	Yes	String
versionNumber	The version number of the object type. Start with 1 and increment by 1 each time you want to modify the object type.	Yes	Number

Table 39. Response Parameters

Parameter	Data Type	Description
name	String	The object type name (must be unique).
description	String	The description of the object type.

Table 39. Response Parameters (continued)

Parameter	Data Type	Description
containedObject-Types	Array	The list of contained types with details.
dataVariables	Array	The list of variables (with details) in the object type.
objectTypeTemplates	Array	The list of object type templates, along with their details.
lastModifiedUser	String	The username of the user who created the object type.
lastModifiedTime	String	The current time.
versionNumber	Number	The version number of the object type. Start with 1 and increment by 1 each time you want to modify the object type.
databaseName	The name of the Historian server.	Yes
containedObject-TypeChanges	The list of changes in the contained type.	No
dataVariables-Changes	The list of changes in the object type variables.	No
objectTypeTemplatesChanges	The list of changes in the object type templates.	No

Table 40. objectTypeTemplates Array Parameters

Parameter	Description	Data Type
name	The object type name (must be unique).	String
description	The description of the object type.	String
containedObject-Types	The list of contained types with details	Array

Table 40. objectTypeTemplates Array Parameters (continued)

Parameter	Description	Data Type
<code>dataVariables</code>	The list of variables (with details) in the object type.	Array
<code>lastModifiedUser</code>	The username of the user who created the object type.	String
<code>lastModifiedTime</code>	The current time.	String
<code>versionNumber</code>	The version number of the object type. Start with 1 and increment by 1 each time you want to modify the object type.	Number
<code>ObjectType</code>	The object type name.	String
<code>substituteParameters</code>	Blank (not applicable)	Array

Table 41. dataVariables Array Parameters

Parameter	Description	Re-quired?	Data Type
<code>name</code>	The name of the object type name (must be unique).	Yes	String
<code>description</code>	The description of the object type.	No	String
<code>variableType</code>	The type of the variable (direct, indirect, or static).	No	String
<code>baseType</code>	Not applicable	No	String
<code>dbTagName</code>	Not applicable	No	String

Table 42. containedObjectTypes Array Parameters

Parameter	Description	Re-quired?	Data Type
<code>name</code>	The name of the object type (must be unique).	Yes	String
<code>description</code>	The description of the object type).	No	String
<code>containedObjectType</code>	The name of the object type that you want to use as a contained type.	No	String

Table 42. containedObjectTypes Array Parameters (continued)

Parameter	Description	Re-quired?	Data Type
<code>baseType</code>	The type of the contained type (ThingName indicates a contained type).	No	String

Table 43. Response Parameters

Parameter	Data Type	Description
<code>detail</code>	String	The description of the error.
<code>error_code</code>	Integer	The error code. A value of 0 indicates that the operation was successful.
<code>status</code>	String	The http request status.
<code>title</code>	String	The title of the error.
<code>type</code>	String	The type of the error.

The Contained Type List API

Using this API, you can get a list of contained types in each object type.

METHOD	GET
URI	<code>https://<historianservername> /historian-model/v2/ containedObjectTypesMap</code>
SAMPLE QUERY PARAM GET URL	<code>https://localhost:443/historian-model/v2/containedObjectTypesMap</code>

Table 44. Response Parameters

Parameter	Description	Data Type
<code>containedObjectTypes</code>	The list of contained types.	Array
<code>name</code>	The object type in which the contained type is used.	String

The Create Object Instance API

Using this API, you can create an object instance.

METHOD	POST
URI	https://<historianservername> /historian-model/v2/objects
SAMPLE QUERY PARAM POST URL	https://localhost:443/historian-model/v2/objects
SAMPLE PAYLOAD	<pre>[{ "containedObjects":[], "containedType":false, "dataVariables":[], "databaseName":"XYZ", "description":"", "fullName":"", "name":"Audi", "objectDataValueChanges":[], "objectDataValues":[], "objectType":"Vehicle", "objectTypeTemplate":"Car", "override":false, "parentName":"", "substituteParameters":[], "v1Type":false, "versionNumber":0 }]</pre>

Table 45. Payload Parameters

Parameter	Description	Required?	Data Type
containedObjects	The list of contained object types.	No	Array
containedType	Indicates whether the object type is used as a contained type (by default, False).	Yes	Boolean

Table 45. Payload Parameters (continued)

Parameter	Description	Required?	Data Type
data-Variables	The list of variables with details.	No	Array
description	The description of the object instance.	No	String
fullName	The full name of the object instance, which contains the name of the original object type, followed by the contained type name (as applicable).	No	String
name	The name of the object instance.	Yes	String
object-Data-Value-Changes	The list of changes in the object instance.	No	Array
Object-Type	The object type from which this instance must be created.	Yes	String
object-TypeTemplate	The name of the template you want to use in the object instance.	Yes	String
Override	True or false (required, but not used in Historian).	No	Boolean
parent-Name	The name of the object type.	No	String
substituteParameters	Not applicable	No	String
v1Type	Indicates whether it is an Operations Hub model or a Historian model (by default, False, indicating a Historian model).	Yes	Boolean
version-Number	The version number of the object instance.	Yes	Number

Table 45. Payload Parameters (continued)

Parameter	Description	Required?	Data Type
<code>data-baseName</code>	The name of the Historian system.	Yes	String

Table 46. Response Parameters

Parameter	Data Type	Description
<code>detail</code>	String	Description of the error.
<code>error_code</code>	Integer	The error code. A value of 0 indicates that the operation was successful.
<code>status</code>	String	The http request status.
<code>title</code>	String	The title of the error.
<code>type</code>	String	The type of the error.

The Object Instance Info API

Using this API, you can view an object instance.

METHOD	POST
URI	<code>https://<historianservername>/historian-model/v2/objects/info</code>
SAMPLE QUERY PARAM POST URL	<code>https://localhost:443/historian-model/v2/objects/info</code>
SAMPLE RESPONSE	<pre>{ "containedObjects": [{ "alias": "Engine", "baseType": "THINGNAME", "containedObjectType": "Engine", "containedObjectTypeTemplate": "Default_Template_Engine", "description": "Engine", "fullName": "Audi1>Engine",</pre>

```
"name": "Engine",
"variableType": "ContainedAsset"
}
],
"containedType": false,
"dataVariables":
[
{
"baseType": "NUMBER",
"blockType": "AI",
"dbTagName": "Speed",
"description": "",
"name": "Speed",
"properties": [],
"required": true,
"variableType": "Direct"
},
{
"baseType": "STRING",
"blockType": "AI",
"dbTagName": "Type",
"description": "",
"name": "Type",
"properties": [],
"required": true,
"variableType": "InDirect"
},
{
"baseType": "NUMBER",
"blockType": "AI",
"dbTagName": "Color",
"description": "",
"name": "Color",
"properties": [],
"required": true,
"variableType": "Static"
}
}
```

	<pre>], "databaseName": "XYZ", "description": "", "lastModifiedUser": "XYZ.admin", "lastmodifiedTime": "2022-01-21 11:38:59", "name": "Audil", "objectType": "Vehicle", "objectTypeTemplate": "Car", "override": false, "parentName": ">", "substituteParameters": [], "vlType": false, "versionNumber": 3 } </pre>
SAMPLE PAYLOAD	<pre> {"name": "Audil", "databaseName": "XYZ"} </pre>

Table 47. Payload Parameters

Parameter	Description	Re-quired?	Data Type
name	The name of the object instance.	Yes	String
databaseName	The name of the Historian server.	No	String

Table 48. Response Parameters

Parameter	Description	Data Type
contained-Objects	The list of contained object types.	Array
contained-Type	Indicates whether the object type is used as a contained type.	Boolean
dataVariables	The list of variables with details.	Array
description	The description of the object instance.	String
fullName	The full name of the object instance.	String

Table 48. Response Parameters (continued)

Parameter	Description	Data Type
name	The name of the object instance.	String
object-DataValue-Changes	The list of changes in the object instance.	Array
ObjectType	The name of the object instance from which this instance is created.	String
objectType-Template	The name of the object type template used in the object instance.	String
Override	True or false.	Boolean
parentName	The name of the parent.	String
substitute-Parameters	Not applicable	String
v1Type	Indicates whether it is an Operations Hub model or a Historian model (by default, False, indicating a Historian model).	Boolean
versionNumber	The version number of the object instance.	Number
database-Name	The name of the Historian server.	String

Table 49. containedObjectTypes Array Parameters

Parameter	Description	Re-quired?	Data Type
alias	The name of the variable.	No	String
variableType	The type of the variable (direct, indirect, or static).	No	String
name	The name of the object type (must be unique).	Yes	String
description	The description of the object type).	No	String

Table 49. containedObjectTypes Array Parameters (continued)

Parameter	Description	Re-quired?	Data Type
<code>containedObject- Type</code>	The name of the object type that you want to use as a contained type.	No	String
<code>containedObject- TypeTemplate</code>	The template in the contained type that you want to use.	No	String
<code>baseType</code>	The type of the contained type (ThingName indicates a contained type).	No	String

Table 50. Data Variables Parameters

Parameter	Description	Re-quired?	Data Type
<code>baseType</code>	The type of the contained type (ThingName indicates a contained type).	No	String
<code>blockType</code>	Not applicable	No	String
<code>dbTagName</code>	The tag mapped with the variable (in the case of a direct or indirect variable).	No	String
<code>descrip- tion</code>	The description of the object instance.	No	String
<code>name</code>	The name of the variable.	No	String
<code>proper- ties</code>	Not applicable	No	Array
<code>required</code>	Indicates whether this variable is included in the object instance.	No	String
<code>variable- Type</code>	The data type of the variable (direct, indirect, or static).	No	String

The Object Instance with Contained Type Variables API

Using this API, you can view the contained type variables in an object instance.

METHOD	POST
--------	------

URI	<a href="https://<historianservername>/historian-model/v2/types-InstancesList/objects">https://<historianservername>/historian-model/v2/types-InstancesList/objects
SAMPLE QUERY PARAM POST URL	https://localhost:443/historian-model/v2/typesInstancesList/objects
SAMPLE RESPONSE	<pre>[{ "hasItems":true, "id":"Instances_Audil_V_", "loaded":false, "parentId":"Instances_Audil", "text":"Variables", "type":"objectInstanceVariableRootNode" }, { "hasItems":false, "id":"Instances_Audil_V_0", "loaded":false, "objectFullName":"Audil", "parentId":"Instances_Audil_V_", "parentObjectName":"Audil", "text":"Speed", "type":"objectInstanceVariableNode" }, { "hasItems":false, "id":"Instances_Audil_V_1", "loaded":false, "objectFullName":"Audil", "parentId":"Instances_Audil_V_", "parentObjectName":"Audil", "text":"Type", "type":"objectInstanceVariableNode" }, { "hasItems":false, "id":"Instances_Audil_V_2",</pre>

	<pre> "loaded":false, "objectFullName":"Audil", "parentId":"Instances_Audil_V_", "parentObjectName":"Audil", "text":"Color", "type":"objectInstanceVariableNode" }, { "hasItems":true, "id":"Instances_Audil0", "loaded":false, "objectFullName":"Audil>Engine", "objectType":"Engine", "objectTypeTemplate":"Default_Template_Engine", "parentId":"Instances_Audil", "parentObjectName":"Audil", "status":"state_new", "text":"Engine", "type":"objectInstanceContainedAssetNode" }, { "hasItems":false, "id":"Instances_Audil0_V_", "loaded":false, "parentId":"Instances_Audil0", "text":"Variables", "type":"objectInstanceVariableRootNode" } }] </pre>
SAMPLE PAYLOAD	<pre> {"name":"Audil","databaseName":"XYZ"} </pre>

Table 51. Payload Parameters

Parameter	Description	Re-quired?	Data Type
name	The name of the object instance.	Yes	String

Table 51. Payload Parameters (continued)

Parameter	Description	Re-quired?	Data Type
databaseName	The name of the Historian system.	Yes	String

Table 52. Response Parameters

Parameter	Description	Data Type
hasItems	Indicates whether the variable is a static text or a variable (be default, false, indicating that it is static text).	Boolean
id	The sequence ID to render the model tree.	String
loaded	Not applicable	Boolean
parentId	The ID of the parent.	String
text	Contains the static text value or, in the case of variables, contains the text <code>Variables</code> .	String
type	The type of the variable: <ul style="list-style-type: none"> • objectInstanceVariableRootNode: Indicates static text. • objectInstanceVariableNode: Indicates direct, indirect, or a static variable. • objectInstanceContainedAssetNode: Indicates a variable in a contained type. 	String

The Object Instance - Variable Information API

Using this API, you can view the variable details of an object instance.

METHOD	POST
URI	<code>https://<historianservername> /historian-model/v2/ objects/variables/info</code>
SAMPLE QUERY PARAM POST URL	<code>https://localhost:443/historian-model/v2/objects/variables/info</code>

SAMPLE RESPONSE	<pre>{ "blockType": "AI", "variableName": "Speed", "variableTagName": "Audil>Speed", "variableType": "Direct", "variableValue": "Audil>Speed" }</pre>
SAMPLE PAYLOAD	<pre>{"databaseName": "XYZ", "name": "Audil>Speed" }</pre>

Table 53. Payload Parameters

Parameter	Description	Re-quired?	Data Type
name	The name of the object instance.	Yes	String
databaseName	The name of the Historian system.	Yes	String

Table 54. Response Parameters

Parameter	Description	Data Type
block- Type	Not applicable	String
vari- ableName	The name of the variable.	String
vari- ableTag- Name	The name of the tag associated with the variable (not applicable for a static variable).	String
vari- ableType	The type of the variable (static, direct, or indirect).	String
variabl- eValue	The value of the variable (in case of a static variable) or the associated tag (in case of a direct or indirect variable).	String

The Historian Model API

Using this API, you can view the Historian model of a Historian system.

METHOD	POST
URI	<code>https://<historianservername>/historian-model/v2/typesObjectsList</code>
SAMPLE QUERY PARAM POST URL	<code>https://localhost:443/historian-model/v2/typesObjectsList</code>
SAMPLE PAYLOAD	<code>{"databaseName": "XYZ" }</code>

Table 55. Payload Parameters

Parameter	Description	Re-quired?	Data Type
<code>databaseName</code>	The name of the Historian system.	Yes	String

Table 56. Response Parameters

Parameter	Description	Data Type
<code>assetTemplates</code>	List of all the templates in the Historian model.	Array
<code>assetTemplatesInfo</code>	List of the details of all the templates in the Historian model.	Array
<code>assetTypes</code>	List of all the object types in the Historian model.	Array
<code>assetTypesInfo</code>	List of the details of all the object types in the Historian model.	Array
<code>assets</code>	List of all the object instances in the Historian model.	Array
<code>assetsInfo</code>	List of the details of all the object types in the Historian model.	Array

Table 57. assetTemplates Parameters

Parameter	Description	Data Type
<code>defaultTemplate</code>	Indicates whether it is the default template.	Boolean
<code>description</code>	The description of the template.	String

Table 57. assetTemplates Parameters (continued)

Parameter	Description	Data Type
lastModified-User	The username of the user who last modified the template.	String
lastmodified-Time	The date and time when the template was last modified.	Date and time
name	The name of the template.	String
objectType	The name of the object type associated with the template.	String
versionNumber	The version number of the template.	Integer

Table 58. assets Parameters

Parameter	Description	Data Type
assetFullName	The full name of the object instance.	Array
assetName	The name of the object instance.	String
lastModified-User	The username of the user who last modified the object instance.	String
lastmodified-Time	The date and time when the object instance was last modified.	Date and time
assetParent-Name	The parent name of the asset.	String
containedAssets	List of all the contained types in the object type.	Array
versionNumber	The version number of the object instance.	Integer

Table 59. assetsInfo Parameters

Parameter	Description	Data Type
containedType	Indicates whether the object type is a contained type.	Boolean
databaseName	The name of the Historian system.	String

Table 59. assetsInfo Parameters (continued)

Parameter	Description	Data Type
lastModified-User	The username of the user who last modified the object instance.	String
lastmodified-Time	The date and time when the object instance was last modified.	Date and time
description	The description of the object instance.	String
name	The name of the object instance.	String
objectType	The name of the object type associated with the object instance.	String
objectTypeTemplate	The name of the template used in the object instance.	String
Override	Updated or not.	Boolean
ParentName	The parent name of the asset.	String
status	Not applicable	String
v1Type	Not applicable	Boolean
versionNumber	The version number of the object instance.	number

The Export Historian Model API

When you create an object type or an object instance, you can use it only in the Historian system in which you have created it. If, however, you want to use the object type/instance in a different Historian system, you can export it and then import it into the other Historian system.

Using this API, you can export a Historian model, along with the object types, object instances, variables, templates, and contained types, into another Historian system.

METHOD	GET
URI	<code>http://<historianservername>/historian-model/v2/export?type-s=<true/false>&templates=<true/false>&instances=<true/false>&databaseName=<encrypted Historian system name></code>

SAMPLE QUERY PARAM GET URL	<code>http://localhost:443/historian-model/v2/export?types=true&templates=true&instances=true&databaseName=KNQWYA</code>
SAMPLE RESPONSE	<pre>{ "admissionTime": 1642600209, "jobId": 0 }</pre>

Table 60. Query Parameters

Parameter	Description	Required?	Data Type
<code>types</code>	Indicates whether you want to include object types while exporting the Historian model.	Yes	Boolean
<code>templates</code>	Indicates whether you want to include templates while exporting the Historian model.	No	Boolean
<code>instances</code>	Indicates whether you want to include object instances while exporting the Historian model.	No	Boolean
<code>databaseName</code>	The Historian system from which you want to export the Historian model.	Yes	String (Encrypted)

Table 61. Response Parameters

Parameter	Description	Data Type
<code>admissionTime</code>	The timestamp in Epoch time format when the model is exported.	Unix Time Stamp
<code>jobId</code>	The job id with which the operation started.	Integer

The Export Historian Model to Historian - Job Status API

Using this API, you can view the status of exporting a Historian model to another Historian system.

METHOD	GET
URI	<code>http://<historianservername>/historian-model/import-export/status/?jobId=<jobId></code>

SAMPLE QUERY PARAM GET URL	<code>http://localhost:443/historian-model/import-export/status/?jobId=1</code>
SAMPLE RESPONSE	<pre>{ "admissionTime": 1642601025, "completionTime": 1642601025, "isRunning": 0, "processing": "Completed", "processingElement": 14, "startTime": 1642601025, "statusPercentage": 100 }</pre>

Table 62. Query Parameters

Parameter	Description	Required?	Data Type
<code>jobId</code>	The job created when the export command is run.	Yes	Integer

Table 63. Response Parameters

Parameter	Description	Data Type
<code>admissionTime</code>	The timestamp in Epoch time format when the model is exported.	Unix Time Stamp
<code>completionTime</code>	The completing time of the job.	Unix Time Stamp
<code>isRunning</code>	The status of the job.	String
<code>processing</code>	Indicates whether the elements are being processed.	String
<code>processingElement</code>	The element that is being processed.	String
<code>startTime</code>	The start time of the job.	Unix Time Stamp
<code>statusPercentage</code>	The percentage of job completion.	Integer

The Historian Model to Historian - Job Log API

Using this API, you can view the job log of exporting a Historian model to another Historian system.

METHOD	GET
URI	http: //<historianservername>/historian-model/im- port-export/log/?jobId=<jobId>
SAMPLE QUERY PARAM GET URL	http://localhost:443/historian-model/import-ex- port/log/?jobId=1
SAMPLE RESPONSE	<pre>[{ "Elements": ["2022-01-19T08-08-43\t"], "LogStringCode": 12304 }, { "Elements": ["2022-01-19T08-08-43\t","3"], "LogStringCode": 12309 }, { "Elements": ["2022-01-19T08-08-43\t","2.1"], "LogStringCode": 12310 }]</pre>

Table 64. Query Parameters

Parameter	Description	Re-quired?	Data Type
<i>jobId</i>	The job created when you run the export command.	Yes	Integer

Table 65. Response Parameters

Parameter	Description	Data Type
Array of Elements	List of elements that are processed.	Array
LogStringCode	The processing code.	Integer

The Export Historian Model to Historian - Job Result API

Using this API, you can export and retrieve a Historian model in a CSV file. You can then import it into another Historian model.

METHOD	GET
URI	<code>http: //<historianservername>/historian-model/im- port-export/result/?jobId=<jobId></code>
SAMPLE QUERY PARAM GET URL	<code>http://localhost:443/historian-model/import-export/re- sult/?jobId=1</code>

Table 66. Query Parameters

Para- meter	Description	Re- quired?	Data Type
jobId	The job created when you run the export command.	Yes	Integer

Table 67. Response Parameters

Para- meter	Description	Data Type
CSV file	Contains the exported model	file

The Export Historian Model to Operations Hub API

When you create an object type or an object instance, you can use it only in the Historian system in which you have created it. If, however, you want to use the object type/instance in Operations Hub, you can export it and then import it into Operations Hub.

Using this API, you can export a Historian model to Operations Hub.

METHOD	GET
URI	<code>http://<historianservername>/historian-model/v2/export?type- s=<true/false>&instances=<true/false>&forOpsHub=<true/ false>&databaseName=<encrypted Historian system name></code>
SAMPLE QUERY PARAM GET URL	<code>http://localhost:443/historian-model/v2/export?types=true&in- stances=true&forOpsHub=true&databaseName=KNOGEYA</code>
SAMPLE RESPONSE	<pre>{ "admissionTime": 1642600209,</pre>

```

{
  "jobId": 0
}
    
```

Table 68. Query Parameters

Parameter	Description	Required?	Data Type
types	Indicates whether you want to include object types while exporting the Historian model.	Yes	Boolean
in- stances	Indicates whether you want to include object instances while exporting the Historian model.	No	Boolean
data- base- Name	The Historian system from which you want to export the Historian model.	Yes	String (Encrypted)

Table 69. Response Parameters

Parameter	Description	Data Type
admission- Time	The timestamp in Epoch time format when the model is exported.	Unix Time Stamp
jobId	The job id with which the operation started.	Integer

The Export Historian Model to Operations Hub Job Status API

Using this API, you can get the job status of exporting a Historian model to Operations Hub.

METHOD	GET
URI	<code>http://<historianservername>/historian-model/import-export/status/?jobId=<jobId></code>
SAMPLE QUERY PARAM GET URL	<code>http://localhost:443/historian-model/import-export/status/?jobId=1</code>
SAMPLE RESPONSE	<pre> { "admissionTime": 1642601025, "completionTime": 1642601025, "isRunning": 0, "processing": "Completed", "processingElement": 14, </pre>

	<pre>"startTime": 1642601025, "statusPercentage": 100 }</pre>
--	---

Table 70. Query Parameters

Parameter	Description	Required?	Data Type
jobId	The job created when you run the export command.	Yes	Integer

Table 71. Response Parameters

Parameter	Description	Data Type
admissionTime	The timestamp in Epoch time format when the model is exported.	Unix Time Stamp
completionTime	The completion time of the job.	Unix Time Stamp
isRunning	The status of the job.	String
processing	Indicates whether the elements are being processed.	String
processingElement	The element which is being processed.	String
startTime	The start time of the job.	Unix Time Stamp
statusPercentage	The percentage of job completion.	Integer

The Export Historian Model to Operations Hub Job Log API

Using this API, you can get the log of the job when exporting a Historian model to Operations Hub.

METHOD	GET
URI	http: //<historianservername>/historian-model/import-export/log/?jobId=<jobId>
SAMPLE QUERY PARAM GET URL	http://localhost:443/historian-model/import-export/log/?jobId=1

<p>SAMPLE RESPONSE</p>	<pre>[{ "Elements": ["2022-01-19T08-08-43\t", "LogStringCode": 12304 }, { "Elements": ["2022-01-19T08-08-43\t", "3"], "LogStringCode": 12309 }, { "Elements": ["2022-01-19T08-08-43\t","2.1"], "LogStringCode": 12310 }]</pre>
------------------------	---

Table 72. Query Parameters

Parameter	Description	Required?	Data Type
jobId	The job created when you run the export command.	Yes	Integer

Table 73. Response Parameters

Parameter	Description	Data Type
Array of Elements	List of elements that are processed.	Array
LogStringCode	The processing code.	Integer

The Export Historian Model to Operations Hub Job Result API

Using this API, you can export and retrieve a Historian model in a CSV file. You can then import it into Operations Hub.

METHOD	GET
URI	<code>http://<historianservername>/historian-model/import-export/result/?jobId=<jobId></code>

SAMPLE QUERY PARAM GET URL	<code>http://localhost:443/historian-model/import-export/re- sult/?jobId=1</code>
-------------------------------	---

Table 74. Query Parameters

Para- meter	Description	Re- quired?	Data Type
<code>jobId</code>	The job created when you run the export command.	Yes	Integer

Table 75. Response Parameters

Para- meter	Description	Data Type
<code>CSV file</code>	Contains the exported model.	file

The Import Model to Historian API

Using this API, you can import a Historian model from one system to another.



Important:

If the name of a tag associated with a variable in a model contains a period (.), you cannot import the tag while importing the model into a Historian system.

METHOD	POST
URI	<code>http://<historianservername>/historian-model/v2/import?type- s=<true/false>&templates=<true/false>&instances=<true/ false>&databaseName=<encrypted Historian system name></code>
SAMPLE QUERY PARAM GET URL	<code>http://localhost:443/historian-model/v2/import?types=true&tem- plates=true&instances=true&databaseName=KGEYA</code>
SAMPLE RESPONSE	<pre>{ "admissionTime": 1642600209, "jobId": 0 }</pre>

Table 76. Payload Parameters

Pay-load	Description	Re-quired?	Data Type
body	CSV file that you want to import	Yes	file

Table 77. Query Parameters

Para-meter	Description	Re-quired?	Data Type
types	Indicates whether you want to import object types.	Yes	Boolean
templates	Indicates whether you want to import templates.	No	Boolean
instances	Indicates whether you want to import object instances.	No	Boolean
database-Name	The Historian system name in to which you want to import the model.	Yes	String (Encrypted)

Table 78. Response Parameters

Parameter	Description	Data Type
admission-Time	The timestamp in Epoch time format when the model is imported.	Unix Time Stamp
jobId	The job id with which the operation started.	Integer

The Import Historian Model to Historian - Job Status API

Using this API, you can view the job status of importing a Historian model to another Historian system.

METHOD	GET
URI	<code>http://<historianservername>/historian-model/import-export/status/?jobId=<jobId></code>
SAMPLE QUERY PARAM GET URL	<code>http://localhost:443/historian-model/import-export/status/?jobId=1</code>
SAMPLE RESPONSE	<pre>{ "admissionTime": 1642601025,</pre>

```

"completionTime": 1642601025,
"isRunning": 0,
"processing": "Completed",
"processingElement": 14,
"startTime": 1642601025,
"statusPercentage": 100
}
    
```

Table 79. Query Parameters

Parameter	Description	Required?	Data Type
jobId	The job created when the import command is run.	Yes	Integer

Parameter	Description	Data Type
admissionTime	The error in text format.	Unix Time Stamp
completionTime	The completed time of the job.	Unix Time Stamp
isRunning	The status of the job.	String
processing	Indicates whether the elements are processed.	String
processingElement	The element that is being processed.	String
startTime	The start time of the job.	Unix Time Stamp
statusPercentage	The percentage of job completion.	Integer

The Import Historian Model to Historian - Job Log API

Using this API, you can view the job log of importing a Historian model into another Historian model.

METHOD	GET
URI	http://<historianservername>/historian-model/import-export/log/?jobId=<jobId>
SAMPLE QUERY PARAM GET URL	http://localhost:443/historian-model/import-export/log/?jobId=1
SAMPLE RESPONSE	{ {

```

"Elements": ["2022-01-19T08-08-43\t"],
"LogStringCode": 12304
},
{
"Elements": ["2022-01-19T08-08-43\t","3"],
"LogStringCode": 12309
},
{
"Elements": ["2022-01-19T08-08-43\t","2.1"],
"LogStringCode": 12310
}
]
    
```

Table 80. Query Parameters

Parameter	Description	Required?	Data Type
jobId	The job created when the import command is run.	Yes	Integer

Table 81. Response Parameters

Parameter	Description	Data Type
Array of Elements	The list of elements that are processed.	Array
LogStringCode	The processing code.	Integer

The Import Historian Model to Historian - Result API

Using this API, you can view the result of importing a Historian model into another Historian system.

METHOD	GET
URI	<code>http://<historianservername>/historian-model/import-export/result/?jobId=<jobId></code>
SAMPLE QUERY PARAM GET URL	<code>http://localhost:443/historian-model/import-export/result/?jobId=1</code>

<p>SAMPLE RESPONSE</p>	<pre>[{ "Elements":["2022-01-19T08-06-59\t","72","Contained Asset","Engine1"], "LogStringCode":12214 }, { "Elements":["2022-01-19T08-06-59\t","73","Contained Asset","cylinder1"], "LogStringCode":12214 }, { "Elements":["2022-01-19T08-06-59\t","74","Contained Asset","piston1"], "LogStringCode":12214 }]</pre>
------------------------	---

Table 82. Payload Parameters

Parameter	Description	Required?	Data Type
jobId	The job created when the import command is run.	Yes	Integer

Table 83. Response Parameters

Parameter	Description	Data Type
Array of Elements	The list of elements that are processed.	Array
LogStringCode	The processing code.	Integer

The Duplicate Object Instance API

Using this API, you can copy an object instance.

METHOD	POST
URI	http: //<historianservername>/historian-mod- el/v2/objects/clone

SAMPLE QUERY PARAM POST URL	<code>http://localhost:443/historian-model/v2/objects/clone</code>
SAMPLE PAYLOAD	<pre>{ "databaseName": "XYZ", "source": "Vehicle", "destination": "Audi", "description": "the new object" }</pre>

Table 84. Payload Parameters

Parameter	Description	Re-quired?	Data Type
database-Name	The name of the Historian system.	Yes	String
source	The name of the object instance that you want to copy.	Yes	String
destina-tion	The name that you want to provide for the copied object instance.	Yes	String
descrip-tion	The description of the new object instance.	No	String

The Duplicate Object Type API

Using this API, you can copy an object type. When you copy an object type, all the templates and variables are copied too.

METHOD	POST
URI	<code>http://<historianservername>/historian-model/v2/objectTypes/clone</code>
SAMPLE QUERY PARAM POST URL	<code>http://localhost:443/historian-model/v2/objectTypes/clone</code>
SAMPLE PAYLOAD	<pre>{ "source": "Vehicle", "destination": "Audi", }</pre>

	<pre>"description": "the new object" }</pre>
--	--

Table 85. Payload Parameters

Parameter	Description	Re-quired?	Data Type
source	The name of the object type that you want to copy.	Yes	String
destina- tion	The name that you want to provide for the copied object type.	Yes	String
descrip- tion	The description of the new object type.	No	String

The Export Object Instance API

Using this API, you can export an object instance into another Historian system.

METHOD	POST
URI	<code>http://<historianservername>/historian-model/v2/export/objects</code>
SAMPLE QUERY PARAM POST URL	<code>http://localhost:443/historian-model/v2/export/objects</code>
SAMPLE PAYLOAD	<pre>{ "objects": ["object_Instance_1", "Object_Instance_2"], "objectType": "Audi", "objectTypeTemplate": "Vehicle", "databaseName": "XYZ" }</pre>

Table 86. Payload Parameters

Parameter	Description	Re-quired?	Data Type
<code>objects</code>	The list of the object instances that you want to export.	Yes	Array
<code>objectType</code>	The object type associated with the object instances that you want to export.	Yes	String
<code>objectType-Template</code>	The template type of the object instances that you want to export.	No	String
<code>database-Name</code>	The name of the Historian system.	Yes	String

The Export Object Type API

Using this API, you can export an object type into another Historian system.

METHOD	POST
URI	<code>http://<historianservername>/historian-model/export/objectTypes</code>
SAMPLE QUERY PARAM POST URL	<code>http://localhost:443/historian-model/export/objectTypes?databaseName= KOGEYA</code>
SAMPLE PAYLOAD	<pre>{ "objectType": ["object_type_1", "object_type_2"] }</pre>

Table 87. Payload Parameters

Parameter	Description	Re-quired?	Data Type
<code>objectType</code>	The list of object types that you want to export.	Yes	Array

Table 88. Query Parameters

Parameter	Description	Required?	Data Type
<code>data-baseName</code>	The name of the Historian system from which you want to export the object type.	Yes	String (encrypted)

The Delete Object Instance API

Using this API, you can delete an object instance. If there are direct variables in the object type, you can also choose to delete the tags associated with these variables (along with their data).

METHOD	DELETE
URI	<code>http://<historianservername>/historian-model/v2/objects</code>
SAMPLE QUERY PARAM DELETE URL	<code>http://localhost:443/historian-model/v2/objects</code>
SAMPLE PAYLOAD	<pre>{ "databaseName": "XYZ", "name": "Vehicle1", "permanentDelete": true }</pre>

Table 89. Payload Parameters

Parameter	Description	Required?	Data Type
<code>data-baseName</code>	The name of the Historian system.	Yes	String
<code>name</code>	The name of the object instance that you want to delete.	Yes	String
<code>permanentDelete</code>	Indicates whether you want to permanently delete the object instance. If you do so, the tags are deleted as well.	Yes	Boolean

The Incoming Dependencies API

Using this API, you can view a list of object instances associated with an object type. This is used to check if an object type contains object instances before you delete the object type.

METHOD	POST
URI	<code>http://<historianservername>/historian-model/v2/object-Types/IncomingDependencies</code>
SAMPLE QUERY PARAM POST URL	<code>http://localhost:443/historian-model/v2/objectTypes/IncomingDependencies</code>
SAMPLE PAYLOAD	<pre>{ "name": "Vehicle", "databaseName": "XYZ" }</pre>

Table 90. Payload Parameters

Parameter	Description	Re-quired?	Data Type
<code>name</code>	The name of the object type that you want to delete.	Yes	String
<code>database-Name</code>	The name of the Historian system associated with the object type.	Yes	String

The Delete Object Type API

Using this API, you can delete an object type. You can delete multiple object types together; however, all the object types must belong to the same Historian system. You cannot delete an object type if it is used in an object instance; you must first delete the object instance.

METHOD	DELETE
URI	<code>http://<historianservername>/historian-model/v2/objectTypes</code>
SAMPLE QUERY PARAM DELETE URL	<code>http://localhost:443/historian-model/v2/objectTypes</code>
SAMPLE PAYLOAD	<pre>{ "databaseName": "XYZ", "objectTypes": ["Vehicle", "Engine"] }</pre>

Table 91. Payload Parameters


Parameter	Description	Re-quired?	Data Type
object-Types	The list of object types that you want to delete.	Yes	Array
database-Name	The name of the Historian system associated with the object types.	Yes	String

Managing Collector Instances

The Create Collector Instance API

Using the Create Collector Instance API, you can create a collector instance.

METHOD	POST
URI	<code>https://<historianservername>/historian-rest-api/v1/collector/createnewinstance</code>
SAMPLE URI	<p><code>https://<historianservername>/historian-rest-api/v1/collector/createnewinstance</code></p> <p>Payload</p> <pre>{ "mode": 1, "CollectorSystemName": "xyz", "InterfaceDescription": "xyz_Simulation_<IP address>_2", "DataPathDirectory": "C:\\Proficiency Historian Data", "CollectorDestination": "Historian", "winUserName": "", "winPassword": "", "InterfaceSubType": "", "DestinationHistorianUserName": "abc", "DestinationHistorianPassword": "password", "DestinationHistorian": "<IP address>", "General1": "", "General2": "", "General3": "123",</pre>

	<pre> "General4":""," "General5":""," "Type":2, "InterfaceName":"<source server>_<type of the collector>_<destination server>" } </pre> <div data-bbox="820 499 1339 1060" style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note:</p> <ul style="list-style-type: none"> • The DestinationHistorian parameter will not have a value for offline collector configuration. • To connect to MQTT destinations such as AWS IoT and Google Cloud Platform (GCP), you must provide an encrypted password. </div>
<p>SAMPLE RESPONSE</p>	<pre> { "ErrorCode": 0, "ErrorMessage": null } </pre>
<p>SAMPLE cURL COMMAND</p>	<pre> curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -d "{ \"mode\":1,\"CollectorSystemName\": \"xyz\", \"InterfaceDescription\": \"xyz_Simulation_<IP address>_2\", \"DataPathDirectory\": \"C:\ \Proficy Historian Data\", \"CollectorDestination\": \"Historian\", \"winUserName\": \"\", \"winPassword\": \"\", </pre>

```

\"InterfaceSubType\":
\\\", \"DestinationHistorianUserName\": \"abc\",
\"DestinationHistorianPassword\": \"password\",
\"DestinationHistorian\": \"<IP
address>\", \"General1\": \"\",
\"General2\": \"\", \"General3\": \"xyz\",
\"General4\": \"\", \"General5\": \"\",
\"Type\": 2, \"InterfaceName\": \"<source
server>_<type of the collector>_<destination
server>\"} -X POST
https://<historianservername>/historian-rest-a
pi/v1/collector/createnewinstance
    
```

Table 92. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Get Collector Instance Details API

Using the Get Collector Instance Details API, you can view the details of a collector instance.

METHOD	GET
URI	https://<historianservername>/historian-rest-api/v1/collector/instancedetails/<interface name>
SAMPLE QUERY PARAM GET URL	https://<historianservername>/historian-rest-api/v1/collector/instancedetails/<interface name>
SAMPLE RESPONSE	<pre> { "ErrorCode": 0, "ErrorMessage": null, "Data": { "CloudDestination": "", "InterfaceSubType": "", "CollectorSystemName": "xyz", </pre>

	<pre> "Type":2, "DefaultCompression":false, "CloudInformationLogLevel":0, "InterfaceDataDir":"C:\\Proficy Historian Data", "SourceServer":""," "Username":""," "Password":""," "DestinationType":"Historian", "DestinationServer":"abc", "DebugLogLevel":0, "InterfaceInstallDrive":"C", "ConnectionString":"xyz" } } </pre>
<p>SAMPLE cURL COMMAND</p>	<pre> curl -i -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://<historianservername>/historian-rest-a pi/v1/collector/instancedetails/xyz_Simulation_ <IP address>_2 </pre>

Table 93. Query Parameters


Parameter	Description	Required?	Values
<p><code>interface</code> <code>name</code></p>	<p>The interface name of the collector whose details you want to view.</p>	<p>Yes</p>	<p>String</p>

Table 94. Response Parameters

Parameter	Data Type	Required?	Description
<p><code>ErrorCode</code></p>	<p>Integer</p>	<p>Yes</p>	<p>For example, ErrorCode = 0 implies the operation was successful.</p>
<p><code>ErrorMessage</code></p>	<p>String</p>	<p>Yes</p>	<p>For example, NULL.</p>

The Edit Collector Instance API

Using the Edit Collector Instance API, you can modify the cloud parameters of a collector instance. The collector instance will be restarted after you make changes.

METHOD	PUT
URI	<pre>https://<historianservername>/historian-rest-api/v1/collector/editInstance</pre>
SAMPLE URI	<pre>https://<historianservername>/historian-rest-api/v1/collector/editInstance</pre> <p>Payload</p> <pre>{ "InterfaceName": "<source server>_<type of the collector>_<destination server>", "messageCompression": 0, "azureLogLevel": 1, "debugMode": 0, "CollectorDestination": "Predix", "DestinationHistorian": "abc", "mode": 1, "CloudDestinationAddress": "wss://def.run.abc.ice.predix.io/v1/stream/messages", "IdentityIssuer": "https://1234.predix-uaa.run.abc.ice.predix.io/oauth/token", "ClientID": "xyz", "ClientSecret": "123", "ZoneID": "1234", "Proxy": "http://1.2.3.4:80", "ProxyUserName": "", "ProxyPassword": "", "DatapointAttribute1": "", "DatapointAttribute2": "", "DatapointAttribute3": "", "DatapointAttribute4": "" }</pre> <div style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note:</p> <ul style="list-style-type: none"> • The DestinationHistorian parameter will not have a value for offline collector configuration. </div>



- To connect to MQTT destinations such as AWS IoT and Google Cloud Platform (GCP), you must provide an encrypted password.

SAMPLE RESPONSE

```
{
  "ErrorCode": 0,
  "ErrorMessage": null
}
```

SAMPLE cURL COMMAND

```
curl -i -H "Accept: application/json" -i -H
  "Content-Type: application/json"
-H "Authorization: Bearer <TOKEN>" -d
  "{ \"InterfaceName\": \"<source server>_<type of
  the collector>_<destination server>\",
  \"InterfaceName\": \"<source server>_<type
  of the collector>_<destination
  server>\",
  \"messageCompression\": 0, \"azureLogLevel\": 1,
  \"debugMode\": 0, \"CollectorDestination\":
  \"Predix\", \"DestinationHistorian\": \"abc\",
  \"mode\": 1,
  \"CloudDestinationAddress\":
  \"wss://wss://def.run.abc.ice.predix.io/v1/stre
  am/messages\",
  \"IdentityIssuer\":
  \"https://1234.predix-uaa.run.abc.ice.predix.
  io/oauth/token\",
  \"ClientID\": \"HistQA\", \"ClientSecret\":
  \"Gei32litc\", \"ZoneID\": \"1234\",
  \"Proxy\":
  \"http://1.2.3.4:80\", \"ProxyUserName\":
  \"\", \"ProxyPassword\": \"\"\",
```



```

\DatapointAttribute1\":"\","DatapointAttribut
e2\":"\","DatapointAttribute3\":"\","
\DatapointAttribute4\":"\"}" -X PUT
https://<historianservername>/historian-rest-a
pi/v1/collector/editInstance
    
```

Table 95. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Azure Log Level API

Using the Azure Log Level API, you can set the debug information log level for destination - Azure IoT Hub. You can set a value ranging from 0 to 4.

METHOD	PUT
URI	https://<historianservername>/historian-rest-api/v1/collector/azureloglevel
SAMPLE URI	<p>https://<historianservername>/historian-rest-api/v1/collector/azureloglevel</p> <p>Payload</p> <pre> {"InterfaceName": "<source server>_<type of the collector>_<destination server>", "azureLogLevel": 1,} </pre>
SAMPLE RESPONSE	<pre> { "ErrorCode": 0, "ErrorMessage": null } </pre>

<p>SAMPLE cURL COMMAND</p>	<pre>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -d {"\InterfaceName\":"<source server>_<type of the collector>_<destination server>","\azureLogLevel\":"1}" -X PUT https://<historianservername>/historian-rest-a pi/v1/collector/azureloglevel</pre>
-----------------------------------	--

Table 96. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Install Component Details API

Using the Install Component Details API, you can view the install component details from the collector machine.

METHOD	GET
URI	<pre>https://<historianservername>/historian-rest-a pi/v1/ installcomponentdetails/collectorType/collectorS ubType/machine</pre>
SAMPLE URI	<pre>https://<historianservername>/historian-rest-a pi/v1/installcomponentdetails/2/-/abc</pre>
SAMPLE RESPONSE	<pre>{ "ErrorCode":0, "ErrorMessage":null, "Data": { "InterfaceInstallDrive":"C", "InterfaceDataDir":"C:\\Proficiency Historian Data", "CertPathDir":"NONE"</pre>

	<pre>} }</pre>
SAMPLE cURL COMMAND	<pre>curl -i -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://<historianservername>/historian-rest-a pi/v1/installcomponentdetails/2/-/abc</pre>

Table 97. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Message Compression API

Using the Message Compression API, you can enable or disable message compression.

METHOD	PUT
URI	<pre>https://<historianservername>/historian-rest-a pi/v1/collector/messagecompression</pre>
SAMPLE REQUEST	<pre>{ "InterfaceName": "<source server>_<type of the collector>_<destination server>", "messageCompression": 1 }</pre>
SAMPLE RESPONSE	<pre>{ "ErrorCode": 0, "ErrorMessage": null, }</pre>
SAMPLE cURL COMMAND	<pre>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -d {"\InterfaceName\":"<source server>_<type of the collector>_<destination server>\", \messageCompression\":1}" -X PUT</pre>

	<pre>https://<historianservername>/historian-rest-api/v1/collector/messagecompression</pre>
--	---

Table 98. Query Parameters

Parameter	Description	Required?	Values
<code>interface name</code>	The interface name of the collector whose message compression you want to enable or disable.	Yes	String
<code>messagecompression</code>	Identifies whether you want to enable or disable message compression. The valid values are 0 and 1.	Yes	

Table 99. Response Parameters

Parameter	Data Type	Required?	Description
<code>ErrorCode</code>	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
<code>ErrorMessage</code>	String	Yes	For example, NULL.

The Delete Instance API

Using the Delete Instance API, you can delete a collector instance.

METHOD	PUT
URI	<pre>https://<historianservername>/historian-rest-api/v1/collector/deleteinstance</pre>
SAMPLE REQUEST	<pre>https://<historianservername>/historian-rest-api/v1/collector/deleteinstance Payload { "InterfaceName": "<source server>_<type of the collector>_<destination server>", "deleteTags": true }</pre>

<p>SAMPLE RESPONSE</p>	<pre>{ "ErrorCode":0, "ErrorMessage":null, }</pre>
<p>SAMPLE cURL COMMAND</p>	<pre>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -d "{\"InterfaceName\":\"<source server>_<type of the collector>_<destination server>\", \"deleteTags\":true}" -X PUT https://<historianservername>/historian-rest-a pi/v1/collector/deleteinstance</pre>

Table 100. Query Parameters

Parameter	Description	Required?	Values
<code>interface name</code>	The interface name of the collector whose details you want to delete.	Yes	String
<code>deleteTags</code>	Identifies whether you want to delete the tags. The valid values are true and false.	Yes	Boolean

Table 101. Response Parameters

Parameter	Data Type	Required?	Description
<code>ErrorCode</code>	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
<code>ErrorMessage</code>	String	Yes	For example, NULL.

Collector Type and Subtype

The following table provides a list of collector type and subtype for each collector, which you will provide in APIs.

Collector	Collector Type	Collector Subtype
The Calculation collector	8	
The CygNet collector	16	Cygnnet
The File collector	4	
The HAB collector	16	HAB
The iFIX Alarms and Events collector	11	iFixAE
The iFIX collector	1	
The MQTT collector	16	MQTT
The ODBC collector	16	ODBC
The OPC Classic Alarms and Events collector	11	
The OPC Classic DA collector	3	
The OPC Classic HDA collector	16	OPCHDA
The OPC UA DA collector	16	OPCUA
The OSI PI collector	10	
The OSI PI distributor	13	
The Server-to-Server collector	9	
The Server-to-Server distributor	17	
The Simulation collector	2	
The Windows Performance collector	18	
The Wonderware collector	16	Wonderware

Managing Collectors

The Start Collector API

Using the Start Collector API, you can start a collector.

METHOD	PUT
URI	<pre>https://<historianservername>/historian-rest-api/v1/collector/start</pre>
SAMPLE URI	<p>Sample URI for the service mode:</p> <pre>https://<historianservername>/historian-rest-api/v1/collector/start</pre> <p>Payload</p> <pre>{ "interfaceName": "<source server>_<type of the collector>_<destination server>", "mode": 1 }</pre> <p>Sample URI for the command line mode:</p> <pre>https://<historianservername>/historian-rest-api/v1/collector/start</pre> <p>Payload</p> <pre>{ "interfaceName": "<source server>_<type of the collector>_<destination server>", "mode": 2, "winUserName": "", "winPassword": "" }</pre>
SAMPLE RESPONSE	<pre>{ "ErrorCode": 0, "ErrorMessage": "", "Data": "Collector Start Initiated" }</pre>
SAMPLE cURL COMMAND	<pre>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json"</pre>

```
-H "Authorization: Bearer <TOKEN>" -d
"{ \"interfaceName\": \"<source server>_<type of
the collector>_<destination server>\",
\"mode\": 1}" -X PUT
https://<historianservername>/historian-rest-api/v1/collector/start
```

Query parameters include the Payload parameter, which is a JSON file, which contains the following properties.

Table 102. Query Parameters

Parameter	Description	Required?	Values
<code>interfaceName</code>	The interface name of the collector.	Yes	String
<code>mode</code>	The mode to use to manage the collector.	Yes	<ul style="list-style-type: none"> • 1: service mode • 2: command-line mode
<code>winUserName</code>	The Windows username.	Yes (only if you want to use the command-line mode)	String
<code>winPassword</code>	The Windows password	Yes (only if you want to use the command-line mode)	String

Table 103. Response Parameters

Parameter	Data Type	Required?	Description
<code>ErrorCode</code>	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
<code>ErrorMessage</code>	String	Yes	For example, NULL.
<code>Data</code>	String	No	Indicates if the task has been initiated.

The Stop Collector API

Using the Stop Collector API, you can stop a collector.

METHOD	PUT
URI	<code>https://<historianservername>/historian-rest-api/v1/collector/stop</code>
SAMPLE URI	<pre>https://<historianservername>/historian-rest-api/v1/collector/stop Payload { "interfaceName": "<source server>_<type of the collector>_<destination server>" "winUserName": "TestAdmin", "winPassword": "TestAdminPassword" }</pre>
SAMPLE RESPONSE	<pre>{ "ErrorCode": 0, "ErrorMessage": null, "Data": "Collector Stop Initiated" }</pre>
SAMPLE cURL COMMAND	<pre>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -d "{ \"interfaceName\": \"<source server>_<type of the collector>_<destination server>\"}" -X PUT https://<historianservername>/historian-rest-api/v1/collector/stop</pre>

Query parameters include the Payload parameter, which is a JSON file, which contains the following properties.

Table 104. Query Parameters

Parameter	Description	Required?	Values
<code>interfaceName</code>	The interface name of the collector.	Yes	String
<code>winUserName</code>	The Windows username.	Yes (only if you want to use the command-line mode)	String
<code>winPassword</code>	The Windows password	Yes (only if you want to use the command-line mode)	String

Table 105. Response Parameters

Parameter	Data Type	Required?	Description
<code>ErrorCode</code>	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
<code>ErrorMessage</code>	String	Yes	For example, NULL.
<code>Data</code>	String	No	Indicates if the task has been initiated.

The Restart Collector API

Using the Restart Collector API, you can restart a collector.

METHOD	PUT
URI	<code>https://<historianservername>/historian-rest-api/v1/collector/restart</code>
SAMPLE URI	<code>https://<historianservername>/historian-rest-api/v1/collector/restart</code> Payload <pre>{</pre>

	<pre>"interfaceName": "<source server>_<type of the collector>_<destination server>" "winUserName": "", "winPassword": "" }</pre>
SAMPLE RESPONSE	<pre>{ "ErrorCode": 0, "ErrorMessage": null, "Data": "Collector Restart Initiated" }</pre>
SAMPLE cURL COMMAND	<pre>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -d "{ \"interfaceName\": \"<source server>_<type of the collector>_<destination server>\"}" -X PUT https://<historianservername>/historian-rest-a pi/v1/collector/restart</pre>

Query parameters include the Payload parameter, which is a JSON file, which contains the following properties.

Table 106. Query Parameters

Parameter	Description	Required?	Values
interfaceName	The interface name of the collector.	Yes	String
winUserName	The Windows username.	Yes	String
winPassword	The Windows password	Yes	String

Table 107. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

Table 107. Response Parameters (continued)

Parameter	Data Type	Required?	Description
Data	String	No	Indicates if the task has been initiated.

The Pause Data Collection API

Using the Pause Data Collection API, you can pause the data collection of a collector.

METHOD	PUT
URI	<code>https://<historianservername>/historian-rest-api/v1/collector/pausecollection/{interfaceName}</code>
SAMPLE URI	<code>https://<historianservername>/historian-rest-api/v1/collector/pausecollection/RSSERVER2012-02_Simulation</code>
SAMPLE RESPONSE	<pre>{ "ErrorCode": 0, "ErrorMessage": "" }</pre>
SAMPLE cURL COMMAND	<pre>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -X PUT https://<historianservername>/historian-rest-api/v1/collector/pausecollection/RSSERVER2012-02_Simulation</pre>

Table 108. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.
Data	String	No	Indicates if the task has been initiated.

The Resume Data Collection API

Using the Resume Data Collection API, you can resume the data collection of a collector.

METHOD	PUT
URI	<code>https://<historianservername>/historian-rest-api/v1/collector/resumecollection/{interfaceName}</code>
SAMPLE URI	<code>https://<historianservername>/historian-rest-api/v1/collector/resumecollection/RSSERVER2012-02_Simulation</code>
SAMPLE RESPONSE	<pre>{ "ErrorCode": 0, "ErrorMessage": "" }</pre>
SAMPLE cURL COMMAND	<pre>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -X PUT https://<historianservername>/historian-rest-api/v1/collector/resumecollection/RSSERVER2012-02_Simulation</pre>

Table 109. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.
Data	String	No	Indicates if the task has been initiated.

The Add Tag Comment API

Using the Add Tag Comment API, you can add a comment to a tag.

METHOD	POST
URI	<code>https://<historianservername>/historian-rest-api/v1/tags/addcomment</code>

<p>SAMPLE URI</p>	<pre>https://<historianservername>/historian-rest-api/v1/tags/addcomment Payload { "tagName": "rsserver2012-02.Simulation00003", "comment": "Retest", "timeStamp": "2020-04-22T00:00:00.000Z" }</pre>
<p>SAMPLE RESPONSE</p>	<pre>{ "ErrorCode": 0, "ErrorMessage": null, "Data": "" }</pre>
<p>SAMPLE cURL COMMAND</p>	<pre>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -d "{ \"tagName\": \" rsserver2012-02.Simulation00003\", \"comment\": \\\"Test10\\\", \"timeStamp\": \\\"2020-04-22T00:00:00.000Z \\\"}" -X POST https://<historianservername>:443/historian-rest-api/v1/tags/addcomment</pre>

Query parameters include the Payload parameter, which is a JSON file, which contains the following properties.

Table 110. Query Parameters


Parameter	Description	Required?	Values
tagName	The name of the tag.	Yes	String
timestamp	The timestamp of the comment.	Yes	String
comment	The comment.	Yes	String

Table 111. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Get Tag Comment API

Using the Get Tag Comment API, you can view the comments added to a tag.

METHOD	GET
URI	<pre>https://<historianservername>/historian-rest-api/v1/tags/comments/{tagNames}/{start}/{end}</pre>
SAMPLE QUERY PARAM GET URI	<pre>https://<historianservername>/historian-rest-api/v1/tags/comments/?tagNames=rserver2012-02.Simulation00003;rserver2012-02.Simulation00004&start=2020-04-19T00:00:00.000Z&end=2020-04-24T00:00:00.000Z</pre> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note: The query parameter supports multiple tags.</p> </div>
SAMPLE RESPONSE	<pre>{ "ErrorCode": 0, "ErrorMessage": null, "Data": [{ "TagName": "Motor Temperature", "ErrorCode": 0, "Comments": [{</pre>

```

        "TimeStamp":
"2020-04-22T00:00:00.000Z",
        "Comment": "Heat run test:
Starting temperature"
    },
    {
        "TimeStamp":
"2020-04-22T00:00:00.000Z",
        "Comment": "Heat run test:
Temperature of the stator"
    },
    {
        "TimeStamp":
"2020-04-22T00:00:00.000Z",
        "Comment": "Heat run test:
Temperature of the rotor"
    },
    {
        "TimeStamp":
"2020-04-22T00:00:00.000Z",
        "Comment": "Heat run test:
Temperature of the shaft"
    },
    {
        "TimeStamp":
"2020-04-22T00:00:00.000Z",
        "Comment": "Heat run test:
Temperature of the endshield"
    }
]
}
]
}

```

SAMPLE cURL COMMAND

```

curl -i -H "Accept: application/json" -H
"Authorization: Bearer <TOKEN>"
http://<historianservername>

```



```
/historian-rest-api/v1/tags/comments/<tagName>
s>/<start>/<end>
```

Query parameters include the Payload parameter, which is a JSON file, which contains the following properties.

Table 112. Query Parameters

Parameter	Description	Required?	Values
tagNames	The names of the tag as a semi-colon-separated list. For example: HISTWIN20161.ctag1; HISTWIN20161.ctag2	Yes	String
start	The start time of the query, in ISO data format (YYYY-MM-DDTHH:mm:ss.SSSZ).	Yes	DateTime
end	The end time of the query, in ISO format (YYYY-MM-DDTHH:mm:ss.SSSZ).	Yes	DateTime

Table 113. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Set Debug Mode API

Using the Set Debug Mode API, you can set the debug mode of a collector.

METHOD	PUT
URI	<code>https://<historianservername>/historian-rest-api/v1/collector/setdebugmode</code>
SAMPLE URI	<code>https://<historianservername>/historian-rest-api/v1/collector/setdebugmode</code>

	<pre>Payload { "interfaceName": "<source server>_<type of the collector>_<destination server>", "debugMode": 255 }</pre>
SAMPLE RESPONSE	<pre>{ "ErrorCode": 0, "ErrorMessage": "" }</pre>
SAMPLE cURL COMMAND	<pre>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -d "{ \"interfaceName\": \"<source server>_<type of the collector>_<destination server>\", \"debugMode\": \"}\"" -X PUT https://<historianservername>/historian-rest-a pi/v1/collector/setdebugmode</pre>

Query parameters include the Payload parameter, which is a JSON file, which contains the following properties.

Table 114. Query Parameters

Parameter	Description	Required?	Values
interfaceName	The interface name of the collector.	Yes	String
debugMode	The debug log level for the collector.	Yes	<ul style="list-style-type: none"> • 0: Normal log level • 255: Debug log level

Table 115. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.

Table 115. Response Parameters (continued)

Parameter	Data Type	Required?	Description
ErrorMessage	String	Yes	For example, NULL.

The Buffer File Control API

Using the Buffer File Control API, you can delete or move the buffer files. It is recommended to move the buffer files to a new folder within the same drive.



Note:

Moving files to a network shared drive is not supported.

METHOD	PUT
URI	<pre>https://<historianservername>/historian-rest-api/v1/collector/buffercontrol</pre>
SAMPLE URI	<p>Sample URI for moving buffer files:</p> <pre>https://<historianservername>/historian-rest-api/v1/collector/buffercontrol</pre> <p>Payload</p> <pre>{ "interfaceName": "<source server>_<type of the collector>_<destination server>", "bufferMode": 2, "winUserName": "Administrator", "winPassword": "xxxxxxx", "bufferPath": "C:\\Users\\bufffiles" }</pre> <p>Sample URI for deleting buffer files:</p> <pre>https://<historianservername>/historian-rest-api/v1/collector/buffercontrol</pre> <p>Payload</p> <pre>{</pre>

	<pre> "interfaceName": "<source server>_<type of the collector>_<destination server>", "bufferMode": 1 "winUserName": "Administrator", "winPassword": "xxxxxxx", } </pre>
<p>SAMPLE RESPONSE</p>	<p>Sample response for moving buffer files:</p> <pre> { "ErrorCode": 0, "ErrorMessage": null, "Data": "BufferFiles Move Initiated. Collector is in the Stopped state." } </pre> <p>Sample response for deleting buffer files:</p> <pre> { "ErrorCode": 0, "ErrorMessage": null, "Data": "BufferFiles Delete Initiated. Collector is in the Stopped state." } </pre>
<p>SAMPLE cURL COMMAND</p>	<pre> curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -d "{ \"interfaceName\": \"<source server>_<type of the collector>_<destination server>\", \"bufferMode \": 1, \"bufferPath \": \" C:\\Users\\bufffiles\"}" -X PUT https://<historianservername>/historian-rest-a pi/v1/collector/buffercontrol </pre>

Query parameters include the Payload parameter, which is a JSON file, which contains the following properties.

Table 116. Query Parameters


Parameter	Description	Required?	Values
<code>interfaceName</code>	The interface name of the collector.	Yes	String
<code>bufferMode</code>	Indicates whether you want to move or delete the files.	Yes	<ul style="list-style-type: none"> • 1: Indicates that the buffer files will be deleted. • 2: Indicates that the buffer files will be moved to the location specified in the <code>bufferPath</code> parameter.
<code>bufferPath</code>	<p>The directory to which you want to move the buffer files. For example: <code>C:\\Data\\New-BufferFilesLocation</code> or <code>C:/Data/NewBufferPathLocation</code></p> <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: The double slash (\\) is required in the JSON format. </div>	Yes (only if you want to move the buffer files)	String
<code>winUserName</code>	The Windows username.	Yes (only if you want to use the command-line mode)	String
<code>winPassword</code>	The Windows password	Yes (only if you want to use the com-	String

Table 116. Query Parameters (continued)

Parameter	Description	Required?	Values
		mand-line mode)	

Table 117. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.
Data	String	No	Indicates if the task has been initiated (and if the collector is in the stopped state).

The Server Node Change API

Using the Server Node Change API, you can change the server node of a collector to a machine that has Historian 8.1 installed on it.

METHOD	PUT
URI	<code>https://<historianservername>/historian-rest-api/v1/collector/historiannodechange</code>
SAMPLE URI	<pre> https://<historianservername>/historian-rest-api/v1/collector/historiannodechange Payload { "interfaceName":"<source server>_<type of the collector>_<destination server>", "mode":2, "winUserName":"TestAdministrator", "winPassword":"TestPassword", "historianNode":"VMHISTWEBAUTO", "historianUserName":" TestAdministrator2 ", "historianpassword":" TestPassword2" </pre>

	<pre>} </pre>
SAMPLE RESPONSE	<pre>{ "ErrorCode": 0, "ErrorMessage": "" }</pre>
SAMPLE cURL COMMAND	<pre>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -d "{ \"interfaceName\": \"<source server>_<type of the collector>_<destination server>\", \"userName winUserName\": \"tesrt\", \"winpPassword \": \"password\", \"historianNode \": \"nodename\", \" historianUserName \": \"husername\", \" historianpassword \": \"hpassword\"}" -X PUT https://<historianservername>/historian-rest-a pi/v1/collector/ historiannodechange</pre>

Query parameters include the Payload parameter, which is a JSON file, which contains the following properties.

Table 118. Query Parameters

Parameter	Description	Required?	Values
<code>interfaceName</code>	The interface name of the collector.	Yes	String
<code>mode</code>	The mode to use to manage the collector.	Yes	<ul style="list-style-type: none"> • 1: service mode • 2: command-line mode
<code>winUserName</code>	The Windows username.	Yes (only if you want to use the command-line mode)	String

Table 118. Query Parameters (continued)

Parameter	Description	Required?	Values
<code>winPassword</code>	The Windows password.	Yes (only if you want to use the command-line mode)	String
<code>historianNode</code>	The host name of the new Historian destination machine. The destination machine must have Historian 8.1.	Yes	String
<code>historian-UserName</code>	The Windows username of the new Historian destination machine.	Yes (only if you want to use the command-line mode)	String
<code>historian-Password</code>	The Windows password of the new Historian destination machine.	Yes (only if you want to use the command-line mode)	String

Table 119. Response Parameters

Parameter	Data Type	Required?	Description
<code>ErrorCode</code>	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
<code>ErrorMessage</code>	String	Yes	For example, NULL.

The Get Collector Version API

Using the Get Collector Version API, you can view the version number of a collector.

METHOD	GET
--------	-----

URI	<code>https://<historianservername>/historian-rest-api/v1/collector/version/{interfaceName}</code>
SAMPLE QUERY PARAM GET URL	<code>https://<historianservername>/historian-rest-api/v1/collector/version/?interfaceName=<source server>_<type of the collector>_<destination server></code>
SAMPLE RESPONSE	<pre>{ "ErrorCode": 0, "ErrorMessage": null, "Data": { "Version": "8.1.2068.0" } }</pre>
SAMPLE cURL COMMAND	<pre>curl -i -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" http://<historianservername>/historian-rest-api/v1/collector/version/RSSERVER2012-02_Simulation</pre>

Table 120. Query Parameters

Parameter	Description	Required?	Values
<code>interfaceName</code>	The interface name of the collector.	Yes	String

Table 121. Response Parameters

Parameter	Data Type	Required?	Description
<code>ErrorCode</code>	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
<code>ErrorMessage</code>	String	Yes	For example, NULL.
<code>Data</code>	String	Yes	Returns the version of the collector.

The Get Collector Status API

Using the Get Collector Status API, you can view the status of a collector.

METHOD	GET
URI	<code>https://<historianservername>/historian-rest-api/v1/collector/status/{interfaceName}</code>
SAMPLE GET URI	<code>https://<historianservername>/historian-rest-api/v1/collector/status/RSSERVER2012-02_Simulation</code>
SAMPLE RESPONSE	<pre>{ "ErrorCode": 0, "ErrorMessage": "null" "Data":{ "Status": "Running" } }</pre>
SAMPLE cURL COMMAND	<pre>curl -i -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://<historianservername>/historian-rest-api/v1/collector/status/RSSERVER2012-02_Simulation</pre>

Table 122. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.
Data	String	Yes	Returns the status of the collector.

The Collector Manager List API

Using the Collector Manager List API, you can view the list of collector agents machines associated with the Historian server.

METHOD	GET
--------	-----

<p>URI</p>	<p>https://<historianservername>/historian-rest-api/v1/collectormanagerlist</p>
<p>SAMPLE QUERY PARAM GET URL</p>	<p>https://<historianservername>/historian-rest-api/v1/collectormanagerlist</p>
<p>SAMPLE RESPONSE</p>	<pre> { "ErrorCode": 0, "ErrorMessage": null, "Data": [{ "Name": "CollectorManager::abc", "IPAddress": "[::ffff:<IP address>]", "Status": 1, "ComputerName": "abc " }, { "Name": "CollectorManager::xyz", "IPAddress": "[::ffff:<IP address>]", "Status": 1, "ComputerName": "xyz" }, { "Name": "CollectorManager::abc", "IPAddress": "[::ffff:<IP address>]", "Status": 1, "ComputerName": "abc" }, { "Name": "CollectorManager::123", "IPAddress": "[::ffff:<IP address>]", "Status": 1, "ComputerName": "123" }] } </pre>

	<pre>] } } } } </pre>
SAMPLE cURL COMMAND	<pre> curl -i -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://<historianservername>/historian-rest-a pi/v1/collectormanagerlist </pre>

Table 123. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Collector Mode API

Using the Collector Mode API, you can view the running mode of a collector.

METHOD	GET
URI	<pre> https://<historianservername>/historian-rest-a pi/v1/collector/mode/<collector interface name> </pre>
SAMPLE QUERY PARAM GET URL	<pre> https://<historianservername>/historian-rest-a pi/v1/collector/mode/<collector interface name> </pre>
SAMPLE RESPONSE	<pre> { "ErrorCode": 0, "ErrorMessage": null, "Data": { "RunningMode": "Service Mode" } } </pre>
SAMPLE cURL COMMAND	<pre> curl -i -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://<historianservername>/historian-rest-a pi/v1/collector/mode/<host name>_Simulation_<IP address>_2 </pre>

Table 124. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Collector Details API

Using the Collector Details API, you can view the details of a collector.

METHOD	GET
URI	<code>https://<historianservername>/historian-rest-api/v1/collector/details</code>
SAMPLE QUERY PARAM GET URL	<code>https://<historianservername>/historian-rest-api/v1/collector/details</code>
SAMPLE RESPONSE	<pre>{ "ErrorCode":0, "ErrorMessage":null, "Data": [{ "Name":"<value>", "ComputerName":"<value>", "Status":"Running", "ReportRate":0, "MaximumEventRate":0, "MinimumEventRate":0, "OutOfOrderEvents":0, "Overruns":0, "OverrunsPercent":0, "TotalEventsCollected":0, "TotalEventsReported":0, "LastDataValue":"\\\\"1970-01-01T00:00:00.000Z\\", "Redundency":"" }] }</pre>

	<pre>"Comments": "<username>--test2--\\ \"2020-12-15T07:19:42.000Z\\\"";", "Version": "9.0.4326.0", "CollectorCompression": 0, "TagsCount": 0 }] }</pre>
SAMPLE cURL COMMAND	<pre>curl -i -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://<historianservername>/historian-rest-a pi/v1/collector/details</pre>

Table 125. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Offline Collectors API

Using the Offline Collectors API, you can view a list of offline collectors.

METHOD	GET
URI	<pre>https://<historianservername>/historian-rest-a pi/v1/offlinecollectors</pre>
SAMPLE QUERY PARAM GET URL	<pre>https://<historianservername>/historian-rest-a pi/v1/offlinecollectors</pre>
SAMPLE RESPONSE	<pre>{ "ErrorCode": 0, "ErrorMessage": null, "Data": [{ "Name": "DISTMACHINE1_Simulation", "ComputerName": "DISTMACHINE1",</pre>

	<pre> "Status": "Stopped" }, { "Name": "NPI212611749M1_Simulation", "ComputerName": "NPI212611749M1", "Status": "Stopped" }, { "Name": "NPI212611749M1_Mqtt", "ComputerName": "NPI212611749M1", "Status": "Stopped" }] } </pre>
SAMPLE cURL COMMAND	<pre> curl -i -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://<historianservername>/historian-rest-a pi/v1/offlinecollectors </pre>

Table 126. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

Managing Data Stores

The Get Data Stores API

Using the Get Data Stores API, you can view the list of data stores in a system.

METHOD	GET
URI	<pre> https://<historianservername>/historian-rest-a pi/v1/datastores?dataStoreMask= </pre>

SAMPLE QUERY PARAM GET URL

```
https://<historianservername>/historian-rest-api/v1/datastores?dataStoreMask=*
```

SAMPLE RESPONSE

```
{  
  
  "ErrorCode": 0,  
  
  "ErrorMessage": null,  
  
  "Data": [  
  
    {  
  
      "Description": "The System Data  
Store.",  
  
      "Id":  
"D3C23639-81CD-40F7-9CB0-37484FC5190D",  
  
      "IsDefault": false,  
  
      "IsSystem": true,  
  
      "Name": "System",  
  
      "NumberOfTags": 0,  
  
      "State": 2,  
  
      "DHSStorageName": "System Storage",  
  
      "StorageType": 0,  
  
      "Links": [  
  
        {
```



```
        "Rel": "self",

        "Href": "/datastore/System"

    }

]

},

{

    "Description": "The Scada Buffer
Data Store.",

    "Id":
"39B39D42-DC7A-4048-9BA8-E4BAB4644B0C",

    "IsDefault": false,

    "IsSystem": false,

    "Name": "ScadaBuffer",

    "NumberOfTags": 0,

    "State": 2,

    "DHSStorageName": "xyz",

    "StorageType": 1,

    "Links": [

        {

            "Rel": "self",
```

```
        "Href":  
        "/datastore/ScadaBuffer"  
    }  
]  
  
},  
  
{  
  
    "Description": "The DHS System Data  
Store.",  
  
    "Id":  
    "56C1DFE9-D0BF-427F-B5D8-B127E38B5C11",  
  
    "IsDefault": false,  
  
    "IsSystem": false,  
  
    "Name": "DHSSystem",  
  
    "NumberOfTags": 0,  
  
    "State": 2,  
  
    "DHSSStorageName": "xyz",  
  
    "StorageType": 0,  
  
    "Links": [  
  
        {  
  
            "Rel": "self",
```

```
        "Href":  
        "/datastore/DHSSystem"  
    }  
]  
},  
{  
    "Description": "The User Data  
Store.",  
    "Id":  
    "33BA016D-B005-4702-96DB-42CF7238C8FF",  
    "IsDefault": true,  
    "IsSystem": false,  
    "Name": "User",  
    "NumberOfTags": 5,  
    "State": 2,  
    "DHSSStorageName": "xyz",  
    "StorageType": 0,  
    "Links": [  
    {  
        "Rel": "self",
```

	<pre> "Href": "/datastore/User" }] } } } } } } } } </pre>
<p>SAMPLE cURL COMMAND</p>	<pre> curl -i -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://<historianservername>/historian-rest-a pi/v1/datastores?dataStoreMask=* </pre>

Table 127. Query Parameters

Parameter	Description	Required?	Values
dataStoreMask	The value of the data store mask.	No	String

Table 128. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Get Data Stores of Storage API

Using the Get Data Stores of Storage API, you can view the list of data stores in a location.

METHOD	GET
URI	<pre>https://<historianservername>/historian-rest-api/v1/storage/datastores?storageName=</pre>
SAMPLE QUERY PARAM GET URL	<pre>https://<historianservername>/historian-resr-api/v1/storage/datastores?storageName=xx</pre>
SAMPLE RESPONSE	<pre>{ "ErrorCode": 0, "ErrorMessage": null, "Data": [{ "Description": "The Scada Buffer Data Store.", "Id": "39B39D42-DC7A-4048-9BA8-E4BAB4644B0C", "IsDefault": false, "IsSystem": false, "Name": "ScadaBuffer", "NumberOfTags": 0, "State": 2, "DHSStorageName": "xyz", "StorageType": 1, "Links": [</pre>

```
{
  "Rel": "self",
  "Href":
"/datastore/ScadaBuffer"
}
],
{
  "Description": "The DHS System Data
Store.",
  "Id":
"56C1DFE9-D0BF-427F-B5D8-B127E38B5C11",
  "IsDefault": false,
  "IsSystem": false,
  "Name": "DHSSystem",
  "NumberOfTags": 0,
  "State": 2,
  "DHSSStorageName": "xyz",
  "StorageType": 0,
  "Links": [
```

```
{
  "Rel": "self",
  "Href":
"/datastore/DHSSystem"
}
],
{
  "Description": "The User Data
Store.",
  "Id":
"33BA016D-B005-4702-96DB-42CF7238C8FF",
  "IsDefault": true,
  "IsSystem": false,
  "Name": "User",
  "NumberOfTags": 5,
  "State": 2,
  "DHSSStorageName": "xyz",
  "StorageType": 0,
  "Links": [
```

	<pre> { "Rel": "self", "Href": "/datastore/User" }] }] } </pre>
<p>SAMPLE cURL COMMAND</p>	<pre> curl -i -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://<historianservername>/historian-rest-a pi/v1/storage/datastores?storageName=xx </pre>

Table 129. Query Parameters

Parameter	Description	Required?	Values
storageName	The name of the location whose data stores you want to view.	Yes	String

Table 130. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Add Datastore API

Using the Add Datastore API, you can create a data store in a Historian server.



Note:

This API is applicable only to an Enterprise Historian.

METHOD	POST
URI	<pre>https://<historianservername>/historian-rest-api/v1/datastoretostorage</pre>
SAMPLE PATH PARAM GET URI	<pre>https://<historianservername>/historian-rest-api/v1/datastoretostorage Payload { "dataStoreName": "abc", "storageName": "storage1", "description": "test", "isDefault": true }</pre>
SAMPLE RESPONSE	<pre>{ "ErrorCode": 0, "ErrorMessage": null }</pre>
SAMPLE cURL COMMAND	<pre>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -d "{ \"dataStoreName \": \"name\", \" storageName \": \"sname\", \"description</pre>

```

\":" des\"," isDefault \":false}"
-X POST
https://<historianservername>/historian-rest-a
pi/v1/datastorestorage
    
```

Table 131. Query Parameters

Parameter	Description	Required?	Values
Payload	Contains the details of the data store in the JSON format.	Yes	Multiple

Table 132. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Delete Data Store API

Using the Delete Data Store API, you can delete a data store.

METHOD	DELETE
URI	https://<historianservername>/historian-rest-api/v1/datastore
SAMPLE URI	https://<historianservername>/historian-rest-api/v1/datastore
	Payload <pre>{ "dataStoreName": "testdatastore" }</pre>
SAMPLE RESPONSE	{

<p>SAMPLE cURL COMMAND</p>	<pre> "ErrorCode": 0, "ErrorMessage": null } curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -d "{ \"dataStoreName \": \"name\"}" -X DELETE https://<historianservername>/historian-rest-a pi/v1/datastore </pre>
----------------------------	--

Table 133. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Data Store Update API

Using the Data Store Update API, you can modify a data store.

METHOD	PUT
URI	<pre> https://<historianservername>/historian-rest-a pi/v1/dataStore/<data store name> </pre>
SAMPLE URI	<pre> https://<historianservername>/historian-rest-a pi/v1/dataStore/mirror1DS1 Payload { "Description": "testing", "IsDefault": true, </pre>

	<pre>"IsSystem": false, "Name": "mirror1DS1", "NumberOfTags": 0, "State": 2, "DHSSStorageName": "mirror1", "StorageType": 0, }</pre>
SAMPLE RESPONSE	<pre>{ "ErrorCode": 0, "ErrorMessage": null }</pre>
SAMPLE cURL COMMAND	<pre>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -d "{ \ Description\":"des\","IsDefault \": true, \ " IsSystem \":false, \ " Name\":"\ mirror1DS1\","NumberOfTags \":0,"State\":2, \ "DHSSStorageName\":"mirror1\","StorageType \ ":0,}" -X PUT https://<historianservername>/historian-rest-a pi/v1/dataStore/mirror1DS1</pre>

Table 134. Query Parameters

Parameter	Description	Required?	Values
Payload	Contains the values of the attributes of the data store that you want to change.	Yes	Multiple

Table 135. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Default Data Store Update API

Using the Default Data Store Update API, you can change the default data store.

METHOD	PUT
URI	<code>https://<historianservername>/historian-rest-api/v1/storage/<location name></code>
SAMPLE URI	<pre> https://<historianservername>/historian-rest-api/v1/storage/NPI212611749M1 Payload { "StorageName": "NPI212611749M1", "StorageType": 0, "NumberOfDataStores": 5, "NumberOfArchivers": 1, "DataStores": [</pre>

```
        "User",  
  
        "testDS1",  
  
        "ScadaBuffer",  
  
        "testDS2",  
  
        "DHSSystem"  
  
    ],  
  
    "Id":  
    "9CD06AFB-1566-4CE6-99D4-B2F65857F33A",  
  
    "IsDefault": true,  
  
    "LastModifiedUser": null,  
  
    "LastModifiedTime":  
    "1970-01-01T00:00:00.000Z",  
  
    "ArchiverServices": [  
  
        "DataArchiver_NPI212611749M1",  
        "DataArchiver_distamchine1"  
  
    ]  
  
    }
```

SAMPLE RESPONSE

```
{  
  
    "ErrorCode": 0,  
  
    "ErrorMessage": null,
```

	}
SAMPLE cURL COMMAND	<pre>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -d "{ \"StorageName\": \"name\", \"StorageType\": 0, \"NumberOfDataStores\": 5, \" NumberOfArchivers\": 1, \"IsDefault\": true, \"ArchiverServices\": [\"DataArchiver_NPI212611749M1\", \"DataArchiver_distamchine1\"]}" -X PUT https://<historianservername>/historian-rest-a pi/v1/storage/NPI212611749M1</pre>

Table 136. Query Parameters

Parameter	Description	Required?	Values
Payload	Contains the values of the attributes of the default data store that you want to change.	Yes	Multiple

Table 137. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

Managing Tags

The Tags API

The Tags API retrieves the qualified tag name list by a given `nameMask`.



Note:

URI format supports asterisks (*) and question marks (?).

METHOD	GET
URI	<code>https://<historianservername>:443/historian-rest-api/v1/tags/{nameMask}/{maxNumber}</code>
SAMPLE URI	<code>https://<historianservername>:443/historian-rest-api/v1/tags?nameMask=*&maxNumber=100</code>
SAMPLE cURL COMMAND	<code>curl -i -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://<nodename>:443/historian-rest-api/v1/tags?nameMask=*&maxNumber=<Number_Of_Tags></code>

Table 138. Query Parameters

Parameter	Description	Required?	Values
<code>nameMask</code>	Tagmask that searches for all tags that match the mask and applies the remaining criteria to retrieve data. The mask can include wildcards, such as asterisks (*).	Optional	String
<code>maxNumber</code>	<p>Maximum tag number provides the limit while returning the results (0 by default). This means that for a query, if using 0, all tags are returned.</p> <p>If a negative number is used, then 0 is used for the maxNumber.</p> <p>If a positive number is used, then that number of tags is returned. In addition, an error number of +14 notifies the user that there are more than the requested number of tags in the system.</p>	Optional	<p>Integer</p> <p>0 by default</p>

Table 139. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Number	Yes	For example, ErrorCode = 0, which means the operation was successful.
ErrorMessage	String	Yes	For example, NULL.
tags	String	Yes	Includes the following: <ul style="list-style-type: none"> • ALT_SENSOR • tagName1 • tagName2

The Taglist API

The Tags List API `GET` method retrieves the list of tags.

When retrieving large tag lists from Historian, you can paginate the response, allowing you to get the next page, go the end, go back one page, and go to the beginning.

Request Parameters

You can use wildcards (*, &?) with string parameters for pattern matching. Results are sorted in ascending tag names. All parameters use the `AND` operator. The `OR` operator is not supported.

All request parameters are optional.

When there are NO wildcard characters (*, &?) with string parameters for pattern matching, then search would be a `contains` search

Example: "dog" pattern will match "dog1", "dog2","dogs", "dogx", "dog12", "dogs are faithful", "1dog1" and so on. When wildcards (*,& ?) are used in the search string parameters for pattern matching, then they work as per the wildcard character definition.

? - Single character matching

* - Multi character matching

Eg1: "dog?" pattern will match: "dog1", "dog2","dogs", "dogx" and so on but not "dog12" or "dogs are faithful"

Eg2: “dog*” pattern will match “dog1”, “dog2”, “dogs”, “dogx”, “dog12”, “dogs are faithful” and so on but not “1dog1”

Parameter Name	Data Type	Default	Description
<code>calctype</code>	Integer	-1	Returns exact match of calc type (0,1,2).
<code>collectiondisabled</code>	Boolean	If ignored, all types considered.	Must be only true / false, else error out.
<code>collectioninterval</code>	Integer	0 – means all intervals	If <code>collectorinterval</code> = 0 consider all intervals, else exact match.
<code>collectorcompression</code>	Boolean	*	Returns exact match of collector compression (true/false).
<code>collectorname</code>	String	*	Default * means consider all.
<code>collectortype</code>	Integer	0 – means consider all collector types	Returns exact match of collector type.
<code>comment</code>	String	*	Default * means consider all.
<code>data storename</code>	String	*	Default * means consider all.
<code>datatype</code>	Integer	0 – means consider all data types	Returns exact match of data type.
<code>description</code>	String	*	Default * means consider all.
<code>egudescription</code>	String	*	Default * means consider all.
<code>enumeratedset</code>	String	*	Default * means consider all.
<code>hasalias</code>	Boolean	If ignored, all types considered.	Must be only true / false, else error out.
<code>isstale</code>	Boolean	If ignored, all types considered.	Must be only true / false, else error out.

Parameter Name	Data Type	Default	Description
lastmodified	String	1970-01-01T00:00:00	is applied so that last modified tag is returned in the result set.
lastmodifieduser	String	*	Default * means consider all.
numberofelements	Integer	0	If 0, ignore this parameter else returns exact match of number of elements.
pageno	Integer	1 Must be > 1	If invalid, no data is returned.
pagesize	Integer	128 Max 512 Min 2	If out of range, returns error. Provide 0 to return all tags at once without any pagination.
sourceaddress	String	*	Default * means consider all.
tagname	String	*	Default * means consider all.
userdefinedtypename	String	*	Default * means consider all.

The Tags List Pagination Parameters

When retrieving large tag lists from Historian, you can paginate the response, allowing you to get the next page, go the end, and go back on page and to the beginning. Results with no errors return these pagination parameters:

Parameter	Value
pagesize	Current page size.
pageno	Current page number
totalcount	Total result other than current page.
Links to URLs	All URLs are part of the HTTP response headers.

Parameter	Value
	<ul style="list-style-type: none"> • <code>first</code> – First page <code>tags list</code> URL (can be null if count is 0). • <code>last</code> - Last page <code>tags list</code> URL (can be null if count is 0). • <code>prev</code> – Previous page <code>tags list</code> URL (can be null if current page is 1). • <code>Next</code> - Next page <code>tags list</code> URL (can be null if current page is last page).

Table 140. Sample cURL commands

METHOD	GET
SAMPLE cURL COM- MAND: [lastmodi- fied]	curl -i -H "Accept: application/json" -H "Authorization:Bearer <TOKEN>" http://<nodename>:443/historian-rest-api/v1/tagslist?lastmodified=2017-05-01T00:00:00.00Z
SAMPLE cURL COM- MAND: [pageno=0]	curl -i -H "Accept: application/json" -H "Authorization:Bearer <TOKEN>" http://<nodename>:443/historian-rest-api/v1/tagslist?pageno=0
SAMPLE cURL COM- MAND: [pageno=1]	curl -i -H "Accept: application/json" -H "Authorization:Bearer <TOKEN>" http://<nodename>:443/historian-rest-api/v1/tagslist?pageno=1
SAMPLE cURL COM- MAND: [complete tagslist]	curl -i -H "Accept: application/json" -H "Authorization:Bearer <TOKEN>" http://<nodename>:443/historian-rest-api/v1/tagslist

Example Queries

The following request returns first page as `pageno` is ignored and `pagesize` is defaulted to 128, all tags are considered:

```
<baseurl>/v1/tagslist
```

The following request returns first page as `pageno` is ignored and `pagesize` is defaulted to 128, all tags are considered that are modified after `2017-05-01T00:00:00.00Z`.

```
<baseurl>/v1/tagslist?lastmodified=2017-05-01T00:00:00.00Z
```

Example Results

The following info is returned for each tag from the criteria provided in the request as an array of tag info.

- `tagid` - String
- `tagname` - String
- `description` - String
- `datatype` - Integer
- `collectorname` - String
- `collectortype` - Integer
- `data storename` - String
- `egudescription` - String
- `comment` - String
- `sourceaddress` - String
- `sourceaddress` - String
- `collectioninterval` - Integer
- `collectorcompression` - Boolean
- `lastmodifieduser` - String
- `enumeratedset` - String
- `userdefinedtypename` - String
- `calctype` - Integer
- `isstale` - Boolean
- `lastmodified` - Long
- `lastmodified` - Long
- `lastmodifiedString` - String - In readable format
- `has alias` - Boolean
- `numberofelements` - Integer
- `collectiondisabled` - Boolean

Example:

```
{
  "TotalCount": 1031,
  "Page": 1,
  "PageSize": 4,
  "Tags": [
    {
      "Tagid": "adb70ebf-978f-46dd-ac6f-5e863cdb0739",
      "Tagname": "-anilgwxb.Constant",
```

```

"Description": "anilgwxb.Constant",
"DataType": 3,
"CollectorName": "ANILGWXB_Simulation",
"CollectorType": 2,
"DataStoreName": "User",
"EngineeringUnits": "",
"Comment": "",
"SourceAddress": "$Constant",
"CollectionInterval": 1000,
"CollectorCompression": false,
"LastModifiedUser": null,
"EnumeratedSetName": "",
"UserDefinedTypeName": "",
"CalcType": 0,
"IsStale": false,
"HasAlias": false,
"NumberOfElements": 0,
"CollectionDisabled": false,
"LastModified": 1496992712,
"LastModifiedString": "2017-06-09T07:18:32Z"
},
{
"Tagid": "88elf448-643f-465a-95c2-d2bd08870547",
"TagName": "anilgwxb.Constant_1%Noise",
"Description": "anilgwxb.Constant_1%Noise",
"DataType": 3,
"CollectorName": "ANILGWXB_Simulation",
"CollectorType": 2,
"DataStoreName": "User",
"EngineeringUnits": "",
"Comment": "",
"SourceAddress": "$Constant_1%Noise",
"CollectionInterval": 1000,
"CollectorCompression": false,
"LastModifiedUser": null,
"EnumeratedSetName": "",
"UserDefinedTypeName": "",

```

```

"CalcType": 0,

"IsStale": false,

"HasAlias": false,

"NumberOfElements": 0,

"CollectionDisabled": false,

"LastModified": 1496992712,

"LastModifiedString": "2017-06-09T07:18:32Z"

},

<SNIP>

],

"Links": {

"first": "https://anilgwxb:443/historian-rest-api/v1/tagslist?pageno=1&pagesize=4",

"last": "https://anilgwxb:443/historian-rest-api/v1/tagslist?pageno=258&pagesize=4",

"prev": null,

"next": "https://anilgwxb:443/historian-rest-api/v1/tagslist?pageno=2&pagesize=4"

}

}

```

The Raw Data API

The Raw Data API queries raw data, such as a number of samples or the time range for a list of tags. If the count is not zero, then the API service returns the number of raw samples taken beginning from the start time. If the count is zero, then the service returns the raw samples taken between the start time and the end time.

METHOD:	GET, POST
URI:	<p>GET</p> <p>https://<historianservername>:443/historian-rest-api/v1/datapoints/raw/{tagNames}/{start}/{end}/{direction}/{count}</p> <p>POST</p> <p>https://<historianservername>:443/historian-rest-api/v1/datapoints/raw/{start}/{end}/{direction}/{count}</p>
SAMPLE GET URI:	Raw By Number

	<p>Count value is a non-zero positive number, and end time is greater than start time.</p> <pre>https://<historianservername>:443/historian-rest-api/datapoints/raw/tagName1/2013-10-02T11:30:00.111Z/2013-10-02T11:31:11.111Z/0/100https://<historianservername>:443/historian-rest-api/datapoints/raw/tagNames=tagName1&start=2013-10-02T11:30:00.111Z&end=2013-10-02T11:31:11.111Z&count=100&direction=0</pre> <p>Raw By Time</p> <p>The count value equals 0.</p> <pre>https://<historianservername>:443/historian-rest-api/datapoints/raw/tagName1/2013-10-02T11:30:00.111Z/2013-10-02T11:31:11.111Z/0/0https://<historianservername>:443/historian-rest-api/datapoints/raw/tagNames=tagName1&start=2013-10-02T11:30:00.111Z&end=2013-10-02T11:31:11.111Z&count=0&direction=0</pre>
<p>SAMPLE POST URI:</p>	<p>Raw By Number</p> <p>Count value is a non-zero positive number, and end time is greater than start time.</p> <pre>https://<historianservername>:443/historian-rest-api/datapoints/raw/2013-10-02T11:30:00.111Z/2013-10-02T11:31:11.111Z/0/100https://<historianservername>:443/historian-rest-api/datapoints/raw/start=2013-10-02T11:30:00.111Z&end=2013-10-02T11:31:11.111Z&count=100&direction=0</pre> <p>Raw By Time</p> <p>The count value equals 0.</p> <pre>https://<historianservername>:443/historian-rest-api/datapoints/raw/2013-10-02T11:30:00.111Z/2013-10-02T11:31:11.111Z/0/0https://<historianservername>:443/historian-rest-api/datapoints/raw/</pre>

	<code>start=2013-10-02T11:30:00.111Z&end=2013-10-02T11:31:11-111Z&count=0&direction=0</code>
SAMPLE cURL COMMAND (GET): [Raw By Number]	<code>curl -i -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" http://<nodename>:443/historian-rest-api/v1/ datapoints/raw/<tagName>/<start time>/<end time>/<direction>/<count></code>
SAMPLE cURL COMMAND (GET): [Raw By Time]	<code>curl -i -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" http://<nodename>:443/historian-rest-api/v1/ datapoints/raw/<tagName>/<start time>/<end time>/<direction>/0</code>
SAMPLE cURL COMMAND (POST): [Raw By Number]	<code>curl -X POST -i -H "Content-Type: application/json" -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" -d "{\"tagNames\": \"<tagName>;<tagName>\"}" http:// <nodename>/ historian-rest-api/v1/ datapoints/raw/ <start time>/<end time>/<direction>/<count></code>
SAMPLE cURL COMMAND (POST): [Raw By Time]	<code>curl -X POST -i -H "Content-Type: application/json" -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" -d "{\"tagNames\": \"<tagName>;<tagName>\"}" http:// <nodename>/ historian-rest-api/v1/ datapoints/raw/ start=<start time>&end=<end time>&direction=<direction>&count=<count></code>

Table 141. Query Parameters

Parameter	Description	Required?	Values
TagNames	Queries the specified tag names.	Yes	String
Start	Start time of the query, in ISO data format (such as YYYY-MM-DDTHH:m:m:ss.SSSZ).	Yes	DateTime
End	End time of the query, in ISO data format (such as YYYY-MM-DDTHH:m:m:ss.SSSZ).	Yes	DateTime
Direction	Specifies the direction (Forward or Backward	Yes	Integer, with a value such as 1.

Table 141. Query Parameters (continued)

Parameter	Description	Re-quired?	Values
	from the starting time) of data sampling from the archive. The default value is Forward (1).		
Count	Count of archive values within each calculation interval.	Yes	Integer, with a value such as 100.

Table 142. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, 0.
ErrorMessage	String	Yes	For example, NULL.
Data	String	Yes	<p>The object container for the following parameters:</p> <p>DataType</p> <p>DoubleFloat, which stores decimal values up to 15 places.</p> <p>ErrorCode</p> <p>Value is 0, which means the operation was successful.</p> <p>TagName</p> <p>Example: TagName1.</p> <p>Samples</p> <p>Provides TimeStamp, Value, Quality, and DataAttributes for each sample. Where, DataAttributes is</p>

Table 142. Response Parameters (continued)

Parameter	Data Type	Required?	Description
			<p>the detailed information regarding the Quality.</p> <p>For example,</p> <pre> "TimeStamp": "2013-10-02T11:30:00.111Z", "Value": 34.26155", "Quality": 3, "DataAttributes": [] </pre>

The Interpolated Data API

The Interpolated Data API queries interpolated values for a list of tags. If the start time equals the end time, the request returns one sample.

METHOD:	GET, POST
URI:	<p>GET</p> <pre> https://<historianservername>:443/historian-rest-api/v1/datapoints/interpolated/{tag-Names}/{start}/{end}/{count}/{intervalMs} </pre> <p>POST</p> <pre> https://<historianservername>:443/historian-rest-api/v1/datapoints/interpolated/{start}/{end}/{count}/{intervalMs} </pre>
SAMPLE GET URI:	<pre> https://<historianservername>:443/historian-rest-api/v1/datapoints/interpolated/tag-Name1/2013-10-02T11:30:00.111111Z/2013-10-02T11:31:11.111Z/100/10000 </pre>
SAMPLE POST URI:	<pre> https://<historianservername>:443/historian-rest-api/v1/datapoints/interpolated/2013-10-02T11:30:00.111111Z/2013-10-02T11:31:11.111Z/100/10000 </pre>
SAMPLE cURL COMMAND (GET):	<pre> curl -i -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" http://<nodename>:443/historian-rest-api/v1/ </pre>

	<code>datapoints/interpolated/<tagName>/<start time>/<end time>/<count>/<intervalMS></code>
SAMPLE cURL COMMAND (POST):	<code>curl -i -X POST -H "Content-Type: application/json" -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" -d "{\"tagName\":\"<tagName>\"}" http://<nodename>:443/historian-rest-api/v1/datapoints/interpolated/<start time>/<end time>/<count>/<intervalMS></code>

Table 143. Query Parameters

Parameter	Description	Required?	Values
TagName	Queries the tag names specified.	Yes	String
Start	Start time of the query, in ISO data format (such as YYYY-MM-DDTHH:m-m:ss.SSSZ).	Yes	DateTime
End	End time of the query, in ISO data format (such as YYYY-MM-DDTHH:m-m:ss.SSSZ).	Yes	DateTime
Count	Count of archive values within each calculation interval.	Yes	Integer, with a value such as 100.
intervalMS	Interval in milliseconds.	Yes	64-bit signed integer, with a value such as 10000.

Table 144. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, 0.
ErrorMessage	String	Yes	For example, NULL.
Data	String	Yes	The object container for the following parameters:

Table 144. Response Parameters (continued)

Parameter	Data Type	Required?	Description
			<p>DataType</p> <p>DoubleFloat, which stores decimal values up to 15 places.</p> <p>ErrorCode</p> <p>Value is 0, which means the operation was successful.</p> <p>TagName</p> <p>Example is TagName1.</p> <p>Samples</p> <p>Provides TimeStamp, Value and Quality for each sample. For example, TimeStamp = 2013-10-02T11:30:00.111Z, Value = 34.26155, and Quality = 3.</p>

The Current Value API

The Current Value API queries the current value data and reads the current values for a list of tags. If the start time is equal to end time, the request returns one sample.

METHOD:	GET, POST
URI:	<p>GET</p> <p><code>https://<historianservername>:443/historian-rest-api/v1/datapoints/raw/{tagNames}/{start}/{end}/{direction}/{count}</code></p> <p>POST</p> <p><code>https://<historianservername>:443/historian-rest-api/v1/datapoints/currentvalue</code></p>

SAMPLE GET URI:	<code>https://<historianservername>:443/historian-rest-api/v1/datapoints/currentvalue?tagNames=tagName1</code>
SAMPLE POST URI:	<code>https://<historianservername>:443/historian-rest-api/v1/datapoints/currentvalue</code>
SAMPLE cURL COMMAND (GET):	<code>curl -i -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" http://<nodename>:443/historian-rest-api/v1/datapoints/currentvalue/<tagName></code>
SAMPLE cURL COMMAND (POST):	<code>curl -i -X POST -H "Content-Type: application/json" -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" -d "{\"tagNames\": \"<tagName>\"}" http://<nodename>:443/historian-rest-api/v1/datapoints/currentvalue</code>

Table 145. Query Parameters

Parameter	Description	Required?	Values
TagNames	Queries the specified tag names.	Yes	String

Table 146. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, 0.
ErrorMessage	String	Yes	For example, NULL.
Data	String	Yes	<p>The object container for the following parameters:</p> <p>DataType</p> <p>DoubleFloat, which stores decimal values up to 15 places.</p> <p>ErrorCode</p> <p>Value is 0, which means the operation was successful.</p>

Table 146. Response Parameters (continued)

Parameter	Data Type	Required?	Description
			<p>TagName</p> <p>Example is TagName1.</p> <p>Samples</p> <p>Provides TimeStamp, Value and Quality for each sample. For example, TimeStamp = 2014-01-01T12:00:00Z, Value = 34.26155, and Quality = 3.</p>

The Calculated Data API

The Calculated Data API queries the calculated data for a list of tags. Data can be requested using a number of samples or a time range for a list of tags. If the count is not zero, the service returns the number of raw samples beginning from the start time. If the count is zero, the services uses the interval, start time, and end time to calculate the required sample number.

METHOD:	GET, POST
URI:	<p>GET</p> <p><code>https://<historianservername>:443/historian-rest-api/v1/datapoints/calculated/{tagNames}/{start}/{end}/{calculationMode}/{count}/{intervalMs}</code></p> <p>POST</p> <p><code>https://<historianservername>:443/historian-rest-api/v1/datapoints/calculated/{start}/{end}/{calculationMode}/{count}/{intervalMs}</code></p>
SAMPLE GET URI:	<p>Number of Samples</p> <p><code>https://<historianservername>:443/historian-rest-api/v1/datapoints/calculated/tagName1/2013-10-02T11:30:00.111Z/2013-10-02T11:31:11.111Z/1/100/1000</code></p>

	<p>Time Range for List of Tags</p> <pre>https://<historianservername>:443/historian-rest-api/v1/datapoints/calculated?tagNames=tagName1&start=2013-10-02T11:30:00.111Z&end=2013-10-02T11:31:11.111Z&count=100&calculationMode=1&intervalMs=1000</pre>
<p>SAMPLE POST URI:</p>	<p>Number of Samples</p> <pre>https://<historianservername>:443/historian-rest-api/v1/datapoints/calculated/2013-10-02T11:30:00.111Z/2013-10-02T11:31:11.111Z/1/100/1000</pre> <p>Time Range for List of Tags</p> <pre>https://<historianservername>:443/historian-rest-api/v1/datapoints/calculated?start=2013-10-02T11:30:00.111Z&end=2013-10-02T11:31:11.111Z&count=100&calculationMode=1&intervalMs=1000</pre>
<p>SAMPLE cURL COMMAND (GET):</p>	<pre>curl -i -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" http://<nodename>:8843/historian-rest-api/v1/datapoints/calculated/<tagName>/<start time>/<end time>/<count>/<calculation mode>/<intervalMS></pre>
<p>SAMPLE cURL COMMAND (POST):</p>	<pre>curl -i -X POST -H "Content-Type: application/json" -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" -d "{\"tagNames\": \"<tagName>\"}" http://<nodename>:8843/historian-rest-api/v1/datapoints/calculated/<start time>/<end time>/<count>/<calculationmode>/<intervalMS></pre>

Table 147. Query Parameters

Parameter	Description	Required?	Values
TagNames	GE identifier for a location.	Yes	1000000106
Start	Start time of the query, in ISO data format (such as YYYY-MM-DDTHH:m:m:ss.SSSZ).	Yes	DateTime

Table 147. Query Parameters (continued)

Parameter	Description	Re-quired?	Values
End	End time of the query, in ISO data format (such as YYYY-MM-DDTHH:m-m:ss.SSSZ).	Yes	DateTime
Count	Count of archive values within each calculation interval.	Yes	Integer, with a value such as 100.
Calculation Mode		Yes	Integer, with a value such as 1. For more information on the calculation modes, refer to calculation modes for REST API.
IntervalMS	Interval in milliseconds.		64-bit signed integer, with a value such as 1000.

Table 148. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, 0.
ErrorCode	String	Yes	For example, NULL.
Data	String	Yes	<p>The object container for the following parameters:</p> <p>DataType</p> <p>DoubleFloat, which stores decimal values up to 15 places.</p> <p>ErrorCode</p> <p>Value is 0, which means the operation was successful.</p> <p>TagName</p>

Table 148. Response Parameters (continued)

Parameter	Data Type	Required?	Description
			<p>Example is TagName1.</p> <p>Samples</p> <p>Provides TimeStamp, Value and Quality for each sample. For example, TimeStamp = 2013-10-02T11:30:00.111Z, Value = 34.26155, and Quality = 3.</p>

The Sampled Data API

The Sampled Data API queries the sampled data for a list of tags. Data can be requested using a number of samples or a time range for a list of tags. If the count is not zero, the service returns the number of raw samples beginning from the start time. If the count is zero, the services uses the interval, start time, and end time to calculate the required sample number.



Note:

For the query, you can also use optional parameters such as FilterMode and ReturnDataFields. Unused parameters can be omitted.

METHOD:	GET, POST
URI:	<p>GET</p> <p><code>https://<historianservername>:443/historian-rest-api/v1/datapoints/sampled</code></p> <p>POST</p> <p><code>https://<historianservername>:443/historian-rest-api/v1/datapoints/sampled</code></p>
SAMPLE GET URI:	<code>https://<historianservername>:443/historian-rest-api/v1/datapoints/sampled?tagNames=tagName1&start=2013-10-02T11:30:00.111Z&end=2013-10-02T11:31:11-</code>

	<code>.111Z&samplingMode=1&calculationMode=1&direction=0&count=0&intervalMs=1000</code>
SAMPLE POST URI:	<code>https://<historianservername>:443/historian-rest-api/v1/datapoints/sampled</code>
SAMPLE cURL COMMAND (GET):	<code>curl -i -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" http://<nodename>:443/historian-rest-api/v1/datapoints/sampled/<tagName>/<start time>/<end time>/<direction>/<count>/<intervalMS></code>
SAMPLE cURL COMMAND (POST):	<code>curl -i -X POST -H "Content-Type: application/json" -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" -d "{ \"tagNames\": \"<tagName>\", \"start\": \"<start>\", \"end\": \"<end>\", \"samplingMode\": <samplingMode>, \"calculationMode\": <calculationMode>, \"direction\": <direction>, \"count\": <count>, \"returnDataFields\": <returnDataFields>, \"intervalMs\": <intervalMs>, \"queryModifier\": <queryModifier>, \"filterMode\": <filterMode>, \"filterExpression\": \"<filterExpression>\"}" http://<nodename>:443/historian-rest-api/v1/datapoints/sampled</code>

Table 149. Query Parameters

Parameter	Description	Required?	Values
TagNames	Queries the tag names specified.	Yes	String
Start	Start time of the query, in ISO data format (such as YYYY-MM-DDTHH:m-m:ss.SSSZ).	Yes	DateTime
End	End time of the query, in ISO data format (such as YYYY-MM-DDTHH:m-m:ss.SSSZ).	Yes	DateTime
Sampling-Mode	Also known as SamplingModeType.	Optional	Integer, with a value such as 1.

Table 149. Query Parameters (continued)

Parameter	Description	Re-quired?	Values
Calculation-Mode	Also known as CalculationModeType.	Optional	Integer, with a value such as 1.
Direction	Specifies the direction (Forward or Backward from the starting time) of data sampling from the archive. The default value is Forward (1).	Optional	Integer, with a value such as 1.
Count	The count of archive values within each calculation interval.	Optional	Integer, with a value such as 0.
IntervalMS	Interval in milliseconds.	Optional	64-bit signed integer, with a value such as 1000.

Table 150. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, 0.
ErrorCode	String	Yes	For example, NULL.
Data	String	Yes	<p>The object container for the following parameters:</p> <p>DataType</p> <p>DoubleFloat, which stores decimal values up to 15 places.</p> <p>ErrorCode</p> <p>Value is 0, which means the operation was successful.</p> <p>TagName</p>

Table 150. Response Parameters (continued)

Parameter	Data Type	Required?	Description
			<p>Example is TagName1.</p> <p>Samples</p> <p>Provides TimeStamp, Value and Quality for each sample. For example, TimeStamp = 2013-10-02T11:30:00.111Z, Value = 34.26155, and Quality = 3.</p>

The Trend Data API

The Trend Data API queries the trend data for a list of tags.



Note:

For the query, you can also use optional parameters such as FilterMode and StatisticsItemFilter. Unused parameters can be omitted.

METHOD:	GET, POST
URI:	<p>GET</p> <p><code>https://<historianservername>:443/historian-rest-api/v1/datapoints/trend</code></p> <p>POST</p> <p><code>https://<historianservername>:443/historian-rest-api/v1/datapoints/trend</code></p>
SAMPLE GET URI:	<pre>https://<historianservername>:443/historian-rest-api/v1/datapoints/trend?tagNames=tagName1&start=2013-10-02T11:30:00.111Z&end=2013-10-02T11:31:11.111Z&samplingMode=1&calculationMode=1&direction=0&count=0&intervalMs=1000</pre>
SAMPLE POST URI:	<pre>https://<historianservername>:443/historian-rest-api/v1/datapoints/trend</pre>

SAMPLE cURL COMMAND (GET):	<pre>curl -i -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" http://<nodename>:443/historian-rest-api/v1/datapoints/trend/<tagName>/<start time>/<end time>/<samplingMode>/<calculationMode>/<direction>/<count>/<intervalMS></pre>
SAMPLE cURL COMMAND (POST):	<pre>curl -i -X POST -H "Content-Type: application/json" -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" -d "{ \"tagNames\": \"<tagName>\", \"start\": \"<start>\", \"end\": \"<end>\", \"samplingMode\": <samplingMode>, \"calculationMode\": <calculationMode>, \"direction\": <direction>, \"count\": <count>, \"returnDataFields\": <returnDataFields>, \"intervalMs\": <intervalMs>, \"queryModifier\": <queryModifier>, \"filterMode\": <filterMode>, \"filterExpression\": \"<filterExpression>\"}" http://<nodename>:443/historian-rest-api/v1/datapoints/trend</pre>

Table 151. Query Parameters

Parameter	Description	Required?	Values
TagNames	Queries the tag names specified.	Yes	String
Start	Start time of the query, in ISO data format (such as YYYY-MM-DDTHH:m-m:ss.SSSZ).	Yes	DateTime
End	End time of the query, in ISO data format (such as YYYY-MM-DDTHH:m-m:ss.SSSZ).	Yes	DateTime
Sampling-Mode	Also known as SamplingModeType.	Optional	Integer, with a value such as 1.
Calculation-Mode	Also known as CalculationModeType.	Optional	Integer, with a value such as 1.
Direction	Specifies the direction (Forward or Backward	Optional	Integer, with a value such as 1.

Table 151. Query Parameters (continued)

Parameter	Description	Re-quired?	Values
	from the starting time) of data sampling from the archive. The default value is Forward (1).		
Count	The count of archive values within each calculation interval.	Optional	Integer, with a value such as 0.
IntervalMS	Interval in milliseconds.	Optional	64-bit signed integer, with a value such as 1000.

Table 152. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, 0.
ErrorMessage	String	Yes	For example, NULL.
Data	String	Yes	<p>The object container for the following parameters:</p> <p>TagName Name of the tag, such as ahistfile.Simulation00001.</p> <p>TagSource Location where tags are being searched for.</p> <p>Data Type Float, which stores decimal values up to 6 places.</p> <p>Trend Provides TimeStamp, Value and Quality for each sample. For ex-</p>

Table 152. Response Parameters (continued)

Parameter	Data Type	Required?	Description
			ample, TimeStamp = 2016-03-15T04:53:17.000Z, Value = 170903.6563, and Quality = True.

The Add Single Tag API

For the Add Single Tag API, you can add a new tag to Historian, and the tag name and data type must be provided in the payload (parameter) of the method. All other tags are optional. If a property is provided, the respective validation is performed at the server end. If the tag exists, then any new properties provided in the payload are applied to the existing tag.

METHOD:	POST
URI:	<code>https://<historianservername>:443/historian-rest-api/v1/tags/addtag</code>
SAMPLE URI:	<pre>https://<historianservername>:443/historian-rest-api/v1/tags/addtag Payload: { "Name" : "SampleTag", "DataType" : 3 }</pre>
SAMPLE cURL COMMAND:	<code>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -d "{ \"Name\": \"Sampletag\", \"DataType\": 3}" -X POST https://<historianservername>:443/historian-rest-api/v1/tags/addtag</code>

Table 153. Query Parameters

Parameter	Description	Required?	Values
Payload	JSON array of Property-Name and PropertyValue.	Yes. "Name" and "DataType" properties are required.	Multidata types. See Payload Parameter (on page 22) for a list of tag prop-

Table 153. Query Parameters (continued)

Parameter	Description	Required?	Values
		All other properties are optional.	Properties used to update a tag configuration.

Sample Response

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, 0.
ErrorMessage	String	Yes	For example, NULL.

The Add Bulk Tags API

For the Add Bulk Tags API, you can add new tags to Historian using an array, and the tag names and data types must be provided in the payload (parameter) of the method. All other tags are optional. If a property is provided, the respective validation is performed at the server end. If the tags exist, then any new properties provided in the payload are applied to the existing tags. The payload is be an array of tags defined.

METHOD:	POST
URI:	<code>https://<historianservername>:443/historian-rest-api/v1/tags/addtags</code>
SAMPLE URI:	<pre> https://<historianservername>:443/historian-rest-api/v1/tags/addtags Payload: [{ "Name" : "SampleTag1", "DataType" : 3 }, { "Name" : "SampleTag2", "DataType" : 3 }]</pre>

SAMPLE cURL COMMAND:	<pre>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -d "{ \"Name\": \"Sampletag1\", { \"Name\": \"Sampletag2\"}]" -X POST https://<historianservername>:443/historian-rest-api/v1/tags/addtags</pre>
-----------------------------	---

Table 154. Query Parameters

Parameter	Description	Required?	Values
Payload	JSON array tags with individual tags of PropertyName and PropertyValue.	Yes. "Name" and "DataType" properties are required. All other properties are optional.	Multidata types. See Payload Parameter (on page 22) for a list of tag properties used to update a tag configuration.

Table 155. Response Parameters

Parameter	Data Type	Exists?	Description
TagName	String	Yes	Tag name.
ErrorCode	Integer	Yes	For example, 0.
ErrorMessage	String	Yes	For example, NULL.

The Update Tag Configuration API

The Update Tag Configuration API allows you to set or modify any tag property values. You cannot, however, rename a tag using this API.

METHOD:	PUT
URI:	https://<historianservername>:443/historian-rest-api/v1/tags/properties/tagName
SAMPLE DELETE URI:	<pre>https://<historianservername>:443/historian-rest-api/v1/tags/properties/tagName</pre> <p>Payload:</p> <pre>{ "PropertyName" : "PropertyValue" }</pre>

SAMPLE cURL COMMAND:	<pre>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -d "{ \"Description\": \"SampleDesc\"}" -X PUT https://<historianservername>:443/historian-rest-api/v1/tags/properties/tagName</pre>
-----------------------------	---

Table 156. Query Parameters

Parameter	Description	Required?	Values
tagName	Tag name for which properties need to be set or modified.	Yes	String
Payload	JSON array of Property-Name and Property-Value.	At least one property must be provided.	Multidata types. See Payload Parameter (on page 22) for a list of tag properties used to update a tag configuration.

Table 157. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, 0.
ErrorMessage	String	Yes	For example, NULL.

The Get Tag Properties API

You can use this API to specify which properties are required for retrieval. If no property names are provided, then all properties are retrieved. When using the Get Tag Properties method, requesting a non-existent tag name returns an error.

METHOD:	GET / POST
URI: (GET)	<pre>https://<historianservername>:443/historian-rest-api/v1/tags/properties/tagName</pre> <p>This URI returns all tag properties.</p>
URI: (POST)	<pre>https://<historianservername>:443/historian-rest-api/v1/tags/properties/tagName</pre>

	<pre>Payload { "PropertyName1" : 1, "PropertyName2" : 1 }</pre>
SAMPLE GET URI:	<code>https://<historianservername>:443/historian-rest-api/v1/tags/properties/tagName</code>
SAMPLE POST URI:	<pre>https://<historianservername>:443/historian-rest-api/v1/tags/properties/tagName Payload: { "Description" : 1 }</pre>
SAMPLE cURL GET COMMAND:	<code>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -X GET https://<historianservername>:443/historian-rest-api/v1/tags/properties/tagName</code>
SAMPLE cURL POST COMMAND:	<code>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -d "{ \"Description\": 1}" -X POST https://<historianservername>:443/historian-rest-api/v1/tags/properties/tagName</code>

Table 158. Query Parameters

Parameter	Description	Required?	Values
tagName	Tag name for which properties need to be retrieved.	Yes	String
Payload	JSON array of Property-Name and boolean (true/false).	At least one property must be provided.	Multi data types. See Payload Parameter (on page 22) for a list of tag properties used to update a tag configuration.

**Note:**

The query payload contains all the tag properties you want returned from the server. In the Update Tag Config method, you need to provide the actual tag property value. However, in the Get Tag Properties method, you need to provide the property and a value of 1 (true), to allow it to be read from the server and returned.

Table 159. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, 0.
ErrorMessage	String	Yes	For example, NULL.
Name	String	Optional	If no error, then the tag name of query is returned and all requested parameters.

The Delete Tag API

The Delete Tag API provides the ability to delete an existing tag from the Historian server.

Its URI format supports question marks (?).

METHOD:	DELETE
URI:	<code>https://<historianservername>:443/historian-rest-api/v1/tags/tag-Name?{permanentDelete}</code>
SAMPLE DELETE URI:	<code>https://<historianservername>:443/historian-rest-api/v1/tags/tag-Name?permanentDelete=true</code>
SAMPLE CURL COMMAND:	<code>curl -i -H "Authorization: Bearer <TOKEN>" -X DELETE https://<historianservername>:443/historian-rest-api/v1/tags/tagName?permanentDelete=<true false></code>

Table 160. Query Parameters

Parameter	Description	Re-quired?	Values
tagName	Name of the tag to be deleted.	Yes	String

Table 160. Query Parameters (continued)

Parameter	Description	Re-quired?	Values
permanent-Delete	Deletes the tag permanently from the Historian server if the value passed in is true. If the parameter is not provided, then permanent-Delete is assumed to be false.	Optional (false is default)	Boolean, true or false

Table 161. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Number	Yes	For example, ErrorCode=0, which means the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

The Query Results API

The Query Results API enables you to include the number of samples required, by providing an end point to configure query results.

The minimum number of samples should be 1000.

METHOD:	PUT
URI:	<code>https://<historianservername>:443/historian-rest-api/v1/data-points/ configuration/{maxDataQueryResultSize}</code>
SAMPLE URI:	<code>https://<historianservername>:443/historian-rest-api/v1/data-points/ configuration?maxDataQueryResultSize=6000</code>
SAMPLE CURL COMMAND:	<code>curl -i -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://<nodename>:443/ historian-rest-api/v1/data-points/configuration? maxDataQueryResultSize=<Number_Of_Query_Results></code>

Table 162. Query Parameters

Parameter	Description	Re-quired?	Values
maxDataQueryResultSize	Maximum samples that should be configured as part of Query Results.	Yes	Integer

Table 163. Response Parameters

Parameter	Data Type	Required?	Description
Error Code	Integer	Yes	For example, 0.
Error Message	String	Yes	For example, NULL.
Maximum DataQueryResultSize	Integer	Yes	Returns the number of samples that were configured as part of query. For example, based on the sample URI, this parameter will be 6000.

The Tag Rename API

This API allows the administrator to rename tags.

METHOD:	PUT
URI	<code>https://<historianservername>:443/historian-rest-api/v1/tags/tagrename/oldtagname/newtagname?{truerename}</code>
SAMPLE URI	<code>https://<historianservername>:443/historian-rest-api/v1/tags/tagrename/GDW14NV2E.Simulation0000101/GDW14NV2E.Simulation0000101new-name?truerename= <true false></code>
SAMPLE CURL COMMAND	<code>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json"-H "Authorization: Bearer <TOKEN> -X PUT https://<historianservername>:443/historian-rest-api/v1/tags/tagrename/<oldtagname>/<newtagname>?truerename=<true false></code>

Table 164. Query Parameters

Parameter	Description	Required?	Values
oldtagname	Tag which is to be re-named.	Yes	String
newtagname	New name for the selected tag.	Yes	String
truerename	Renames the tag permanently if the value entered is true. Creates an alias if the value entered is false.	Optional (false is default)	Boolean (true or false)

Table 165. Response Parameters

Parameter	Data Type	Required?	Description
Error Code	Integer	Yes	For example, 0.
Error Message	String	Yes	For example, NULL.
Data	List	Yes	Returns all the properties of the tag.

The Write Tag API

Write Tag Data API enables you to create data for tags. You can write data to a tag for different data types such as integer, float, array, multifield and so on. Once created, you can view the data using other end points. Only REST API Administrator and users with write permission can perform this operation.

Method	POST
URI	<code>https://<historianservername>:443/historian-rest-api /v1/datapoints/create</code>
SAMPLE URI	<code>https://<historianservername>:443/historian-rest-api /v1/datapoints/create</code>

Method	POST
	<pre> Payload { "TagName": "GDW14NV2E.Simulation00015", "samples": [{ "TimeStamp": "2019-09-17T15:58:00.000Z", "Value": "1", "Quality": 3 }] } </pre>
SAMPLE RESPONSE	<pre> { "ErrorCode": 0, "ErrorMessage": "" } </pre>
SAMPLE CURL COMMAND	<pre> curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer <TOKEN>" -d "{ \"TagName\": \"GDW14NV2E.Simula- tion00015\", \"samples\": [{ \"TimeStamp\": \" 2019-09-17T15:58:00.000Z\", \"Val- ue\": \"1\", \"Quality\": 3}]}\" -X POST https://<historianservername>:443/histo- rian-rest-api/v1/datapoints/create </pre>

Table 166. Query Parameters

Parameter	Description	Required?	Values
Payload	JSON format of Property Name and Property Value.	Yes	Multi-data types. It can have integer, float, array, multifield data types.

Table 167. Response Parameters

Parameter	Data Type	Required?	Description
Error Code	Integer	Yes	For example, ErrorCode = 0, which means the operation was successful.
Error Message	String	Yes	For example, NULL.

Calculation Modes for REST API

Calculation modes are used on retrieval of data that has already been collected to the archive. Calculation modes are used when the sampling mode is set to "Calculated". The table below lists the different calculation modes and their corresponding value that you can pass as part of the query parameters to manage tags.

Calculation Mode	Value
Average	1
StandardDeviation	2
Total	3
Minimum	4
Maximum	5
Count	6
RawAverage	7
RawStandardDeviation	8
RawTotal	9

Calculation Mode	Value
MinimumTime	10
MaximumTime	11
TimeGood	12
StateCount	13
StateTime	14
OPCAnd	15
OPCOr	16
FirstRawValue	17
FirstRawTime	18
LastRawValue	19
LastRawTime	20
TagStats	21
DeltaPos	22
DeltaNeg	23
Delta	24

Managing Alarms and Events

Overview

Historian includes Alarms and Events (A&E) archiving to provide the ability to retrieve and store Alarms and Events Data from any OPC-compliant A&E server through the OPC Classic Alarms and Events collector.

Alarms are generally defined as tags going into an abnormal condition. For example, an alarm could be set on a boiler when it reaches a specified temperature. Alarms usually have a well-defined life cycle, which is defined by the individual data sources the alarms are collected from (iFIX, for example). They enter an alarm state, are generally acknowledged, and then return to normal.

Events are generally defined as activities in a system that occur only once. For example, a user logging on to a device is an event. When viewing this data in Historian, each event is

returned as a record. The Historian REST API provide API's to store, retrieve, backup, restore, and delete alarms and events data.

Add Alarm or Event

Add alarm API allows you to create alarm in Historian Server.

METHOD	POST
URI	<pre>https://<historianservername>:443/historian-rest-api/v1/alarms/create</pre>
SAMPLE URI	<pre>https://<historianservername>:443/historian-rest-api/v1/alarms/create</pre> <p>Payload</p> <pre>{ "DataSource": "Simulation", "AlarmsInfo": [{ "ItemId": "GF7DQL63E_Simulation.Simulation00001", "Source": "Simulation", "DataSourceName": "Simulation", "Tagname": "GF7DQL63E_Simulation.Simulation00001", "AlarmType": 4, "EventCategory": "SimCategory", "ConditionName": "Simulation", "SubConditionName": "Simulation", "StateTransitionTime": "2022-09-21T5:58:00.000Z", "StartTime": "2022-09-21T5:58:00.000Z", "EndTime": "2022-09-21T6:58:00.000Z", "AckTime": "2022-09-21T6:58:00.000Z", "TimeStamp": "2022-09-21T5:58:00.000Z", "Message": "This is simulation alarm",</pre>

	<pre> "Severity":100, "Actor":"Simulator", "Quality":3 }] } </pre>
<p>SAMPLE CURL COMMAND</p>	<pre> curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer " -d "{ \"DataSource\": \"Simulation\", \"AlarmsInfo\": [{ \"ItemId\": \"GF7DQL63E_Simulation.Simulation00001\", \"Source\": \"Simulation\", \"DataSourceName\": \"Simulation\", \"TagName\": \"GF7DQL63E_Simulation.Simulation00001\", \"AlarmType\": 4, \"EventCategory\": \"SimCategory\", \"ConditionName\": \"Simulation\", \"SubConditionName\": \"Simulation\", \"StateTransitionTime\": \"2022-09-21T5:58:00.000Z\", \"StartTime\": \"2022-09-21T5:58:00.000Z\", \"EndTime\": \"2022-09-21T6:58:00.000Z\", \"AckTime\": \"2022-09-21T6:58:00.000Z\", \"TimeStamp \": \"2022-09-21T5:58:00.000Z\", \"Message\": \"This is simulation alarm\", </pre>

```

        \Severity\":100,
        \Actor\": \"Simulator\",
        \Quality\":3
    }
}
} -X PUT https://:443/historian-rest-api/v1/
alarms/create
    
```

Table 168. Query Parameters

Parameter	Data Type	Description
DataSource	VT_- BSTR	The collector interface name associated with the alarm or event.
ItemId	VT_- BSTR	The OPC ItemID of the alarm. This contains the source address of the data access tag with which the alarm is associated. This can contain a NULL value if an alarm is not associated with a tag.
Source	VT_- BSTR	The unique identifier used by the OPC A&E Collector for the alarm or event.
Data-SourceName	VT_- BSTR	The collector interface name associated with the alarm or event.
Tagname	VT_- BSTR	The Historian tag name associated with the alarm. This value is NULL unless the tag is also collected by Historian
AlarmType	VT_- BSTR	The alarm type: <ul style="list-style-type: none"> Alarms: In Historian, the full life cycle of an alarm is stored as a single record in the alarm archive. Alarm_History: The separate transitions for all alarms. One row per transition is returned. Events: The simple and tracking events.
EventCategory	VT_- BSTR	The OPC event category of the alarm or event.
ConditionName	VT_- BSTR	The OPC condition of the alarm. Does not apply to event data. This value combined with the Source value comprises an alarm.

Table 168. Query Parameters (continued)

Parameter	Data Type	Description
SubConditionName	VT_- BSTR	The OPC subcondition of the alarm. Does not apply to event data. This value represents the state of the alarm.
StateTransitionTime	VT_- DB- Time- S- tamp	The time the state of the tag was generated.
StartTime	VT_- DB- Time- S- tamp	The start time or timestamp of the alarm or event.
EndTime	VT_- DB- Time- S- tamp	The end time of the alarm. Does not apply to event data.
AckTime	VT_- DB- Time- S- tamp	The time the alarm was acknowledged. Does not apply to event data.
TimeStamp	VT_- DB- Time- S- tamp	The time the alarm was generated.
Message	VT_- BSTR	The message attached to the alarm or event.

Table 168. Query Parameters (continued)

Parameter	Data Type	Description
Severity	VT_- I4	The severity of the alarm or event. Stored as an integer value with a range of 1–1000.
Actor	VT_- BSTR	The operator who acknowledged the alarm, or caused the tracking event.
Quality	VT_- I4	The quality of the alarm or event.

Table 169. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

Query Alarm or Event

The following table contains the query parameters for alarms and events data.

METHOD	POST
URI	<code>https://<historianservername>:443/historian-rest-api/v1/alarms/query</code>
SAMPLE URI	<p><code>https://<historianservername>:443/historian-rest-api/v1/alarms/query</code></p> <p>Payload:</p> <pre>{ "AlarmType": "4", "StartTime": "2022-10-30T3:30:00.000Z", "EndTime": "2022-10-31T4:30:00.000Z", "DataSourceName": "GF7DQL63E_Simulation", "MaxRecords": 50 }</pre> <p>Response:</p>

	<pre>{ "ErrorCode": 0, "ErrorMessage": null, "Data": [{ "Description": "This is a simulated alarm.", "Source": "Simulation00001_ALM", "StartTime": "2022-10-30T03:30:00.000Z", "EndTime": "2022-10-30T03:30:02.000Z", "Condition": "Simulated", "Sub Condition": "OK", "EventCategory": "SimCategory", "Severity": 0, "ModifiedTime": "2022-10-30T03:30:02.000Z", "Quality": 3 }] }</pre>
<p>SAMPLE CURL COMMAND</p>	<pre>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer " -d "{ \"AlarmType\": \"4\", \"StartTime\": \"2022-10-30T3:30:00.000Z\", \"EndTime\": \"2022-10-31T4:30:00.000Z\", \"DataSourceName\": \"GF7DQL63E_Simulation\", \"MaxRecords\": 50 }" -X PUT https://:443/historian-rest-api/v1/alarms/ query</pre>

Table 170. Query Parameters

Parameter	Description	Required?	Values
DataSourceName	The collector interface name associated with the alarm or event.	false	
AlarmType	Classifies this AlarmInfo as an alarm or an event. Enter 1 for an event and 4 for an alarm.	yes	
StartTime	The start time or time stamp of the alarm or event.	yes	
EndTime	The end time of the alarm. This does not apply to event data.	yes	
MaxRecords	Maximum records returned	false	

Table 171. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, ErrorCode = 0 implies the operation was successful.
ErrorMessage	String	Yes	For example, NULL.

Delete Alarms or Events

This REST API allows to purge the alarms between time frame.

METHOD	POST
URI	<code>https://<historianservername>:443/historian-rest-api/v1/alarms/delete</code>
SAMPLE URI	<p><code>https://<historianservername>:443/historian-rest-api/v1/alarms/delete</code></p> <p>Payload</p> <pre>https://<historianservername>:443/historian-rest-api/v1/alarms/delete Payload: { "StartTime": "2022-09-21T5:58:00.000Z", "EndTime": "2022-09-21T6:58:00.000Z" }</pre>
SAMPLE CURL COMMAND	<pre>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer " -d "{ \"StartTime\": \"2022-09-21T5:58:00.000Z\", \"EndTime\": \"2022-09-21T6:58:00.000Z\" }" -X PUT https://:443/historian-rest-api/v1/alarms/delete</pre>

Table 172. Query Parameters

Parameter	Description	Required?	Values
StartTime	The start time or time stamp of the alarm or event.	Yes	
EndTime	The end time of the alarm or event.	Yes	

Table 173. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, -26 A&E Server not connected
ErrorMessage	String	Yes	For example, NULL.

Backup Alarms or Events

This REST API allows to create a copy of the alarm data in an offline file that can be restored later. The alarms are not removed from the online system.

METHOD	POST
URI	<code>https://<historianservername>:443/historian-rest-api/v1/alarms/backup</code>
SAMPLE URI	<p><code>https://<historianservername>:443/historian-rest-api/v1/alarms/backup</code></p> <p>Payload</p> <pre>{ "BackupFileName": "C:\\Proficy Historian Data\\Archives\\abc.zip, "StartTime": "2022-09-21T5:58:00.000Z", "EndTime": "2022-09-21T6:58:00.000Z" }</pre>

<p>SAMPLE CURL COMMAND</p>	<pre>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer " -d "{ \"BackupFileName\": \"C:\\Proficy Historian Data\\Archives\\abc.zip\", \"StartTime\": \"2022-09-21T5:58:00.000Z\", \"EndTime\": \"2022-09-21T6:58:00.000Z\" }" -X PUT https://:443/historian-rest-api/v1/alarms/ backup</pre>
-----------------------------------	---

Table 174. Query Parameters

Parameter	Description	Required?	Values
BackupFileName	Path of the backup file.	Yes	C:\\Proficy Historian Data\\Archives\\abc.zip
StartTime	The start time or time stamp of the alarm or event.	Yes	
EndTime	The end time of the alarm or event.	Yes	

Table 175. Response Parameters

Parameter	Data Type	Required?	Description
ErrorCode	Integer	Yes	For example, -26 A&E Server not connected.
ErrorMessage	String	Yes	For example, NULL.

Restoring Alarms or Events

Restoring alarms to a running system makes them available for query and analysis. You can restore alarms that have been backed up previously..

METHOD	POST
--------	------

URI	<code>https://<historianservername>:443/historian-rest-api/v1/alarms/restore</code>
SAMPLE URI	<p><code>https://<historianservername>:443/historian-rest-api/v1/alarms/restore</code></p> <p>Payload:</p> <p><code>C:\\Proficy Historian Data\\Archives\\abc.zip</code></p>
SAMPLE cURL COMMAND	<pre>curl -i -H "Accept: application/json" -i -H "Content-Type: application/json" -H "Authorization: Bearer " -d "C:\\Proficy Historian Data\\Archives\\abc.zip" -X PUT https://:443/historian-rest-api/v1/alarms/ restore</pre>

Table 176. Query Parameters

Parameter	Description	Required?	Values
<code>C:\\Proficy Historian Data\\Archives\\abc.zip</code>	Path of the restore file.	Yes	<code>C:\\Proficy Historian Data\\Archives\\abc.zip</code>

Table 177. Response Parameters

Parameter	Data Type	Required?	Description
<code>ErrorCode</code>	Integer	Yes	For example, 10 invalid parameter.
<code>ErrorMessage</code>	String	Yes	For example, NULL.

Swagger Documentation

You can now access Historian REST APIs using Swagger UI. This tool enables you to visualize and interact with the API's resources without having any of the implementation logic in place establishing a fully interactive documentation experience using Swagger.

Install web-based clients or ensure web-based clients are installed.

This topic describes how to access the Historian REST APIs.

1. Access a node on which Historian web-based clients are installed.
2. Access the following URL: `https://<web-based client machine name>/historian-rest-api/swagger-ui.html`

**Note:**

To access the link, you must provide the machine name where web-based clients are installed. `https://<machinename>/historian-rest-api/swagger-ui.html`

The Swagger UI appears.

3. Select **Authorize**. The Available authorizations window appears.
4. In the Available authorizations window, scroll down to the **oauth2schema** (OAuth2, password) section, enter the following values, and then select **Authorize**:

Field	Description
username	Enter the web-based client login username. Default <machinename>.admin
password	Enter the web-based client login password.
client_id	Enter client id: historian_public_rest_api
client_secret	Enter publicapisecret as client secret.

5. You can now access the REST APIs for Historian.