

Connector API

Table of Contents

1. Architecture	1
1. Implementation Steps	3
1.1. Step 1: Provide TrendMiner with your API version	3
1.2. Step 2: List your data sources	3
1.3. Step 3: Resolve tag metadata & values	3
1.4. Examples	3
1.4.1. Version	3
1.4.2. Data sources	4
1.4.3. Tag Metadata	4
1.4.4. Values	6
2. Analytics Optimization	9
2.1. Principles	9
2.2. Example	9
2.3. Between Values & Interpolation	10
2.4. Endpoint	11
3. Resources	13
3.1. Af	13
3.1.1. Returns the root assets with their details. WARNING - this endpoint is deprecated. Please use /assets.	13
3.1.2. Returns details for the specified asset. WARNING - this endpoint is deprecated. Please use /assets.	13
3.1.3. Returns details for the attribute of a specified asset. WARNING - this endpoint is deprecated. Please use /assets.	14
3.2. Assets	14
3.2.1. Returns details for the root assets in the hierarchy.	14
3.2.2. Returns details for the specified asset.	15
3.2.3. Returns details for the assets with the specified ids.	15
3.3. Configuration	16
3.3.1. Deletes the configured historian connection with the given ID.	16
3.3.2. Retrieve the configured historian connection with the given ID.	16
3.3.3. Update the configured historian connection with the given ID.	17
3.3.4. Test the historian connection with the given ID.	18
3.3.5. Creates a new historian connection configuration on the central connector.	18
3.3.6. Retrieve all the configured historian connections.	19
3.4. Context	19
3.4.1. Returns a list of changed context items in the source system as well as the type of change that happened.	19
3.4.2. Returns a list of the available fields in the source system.	20

3.4.3.	Returns a list of the available context items in the source system.	21
3.4.4.	Returns a specific context item by identifier.	22
3.4.5.	Returns a list of the available context items by identifiers.	22
3.4.6.	Returns a list of the available types in the source system.	23
3.4.7.	Returns a list of the available workflows in the source system.	24
3.5.	Ef.	24
3.5.1.	Returns a list of event frames found in the first configured asset framework. WARNING24 this endpoint is deprecated. Please use /context.	
3.6.	Informational	25
3.6.1.	Returns the version of the TrendMiner historian connector.	25
3.7.	Provider	25
3.7.1.	Returns provider capabilities.	25
3.8.	Support	26
3.8.1.	Deletes all files in the logs folder.	26
3.8.2.	Get the log file content.	26
3.8.3.	Changes log level, e.g. INFO, DEBUG, ERROR, WARN, TRACE.	26
3.9.	Tags	27
3.9.1.	List all digital states of a tag.	27
3.9.2.	Returns all tags for specified historian.	27
3.9.3.	Returns tag values for specified period for indexing purposes.	28
3.9.4.	Returns the last values of multiple tags for the last specified seconds.	29
3.9.5.	Returns tag values for specified period for plotting purposes.	29
3.9.6.	Returns tag raw values for specified period.	30
4.	Models	32
4.1.	<i>AFAAssetAttribute</i>	32
4.2.	<i>AFAAssetNode</i>	32
4.3.	<i>AFAAssetNodeAttribute</i>	32
4.4.	<i>AFAAssetNodeReference</i>	33
4.5.	<i>AssetNode</i>	33
4.6.	<i>AssetNodeReference</i>	34
4.7.	<i>Component</i>	34
4.8.	<i>ContextDataType</i>	34
4.9.	<i>ContextException</i>	34
4.10.	<i>ContextItem</i>	35
4.11.	<i>ContextItemChange</i>	35
4.12.	<i>ContextItemChangeStreamedResponse</i>	35
4.13.	<i>ContextItemListResponse</i>	36
4.14.	<i>ContextItemPagedResponse</i>	36
4.15.	<i>Event</i>	36
4.16.	<i>EventFrame</i>	36
4.17.	<i>Field</i>	37

4.18. <i>FieldPagedResponse</i>	37
4.19. <i>HistorianConnection</i>	37
4.20. <i>HistorianDatabaseProperty</i>	38
4.21. <i>InlineResponse200</i>	38
4.22. <i>InlineResponse400</i>	38
4.23. <i>Page</i>	38
4.24. <i>ProviderCapabilities</i>	39
4.25. <i>ProviderCapabilitiesProperty</i>	39
4.26. <i>ReferenceById</i>	39
4.27. <i>State</i>	39
4.28. <i>StringStreamedResponse</i>	40
4.29. <i>SystemException</i>	40
4.30. <i>Tag</i>	40
4.31. <i>TagDetails</i>	40
4.32. <i>TagLastValuesRequest</i>	41
4.33. <i>TemplateReference</i>	41
4.34. <i>TimeSeriesEntry</i>	41
4.35. <i>Type</i>	42
4.36. <i>TypePagedResponse</i>	42
4.37. <i>Workflow</i>	42
4.38. <i>WorkflowPagedResponse</i>	42

Building a custom time series connection? Connect your data archive, whether it's a Historian, IoT Platform, or Data Lake, to TrendMiner with the Connector API.

TrendMiner supports multiple sources for time series data. It can be uploaded as CSV or can be fetched from an external data source. The Connector API described here defines the API used by TrendMiner to connect to external data sources.

1. Architecture

A typical time series data flow has three components. Customer time series data is stored in an archive. This can be a Data Historian, an IoT Platform or a Data Lake. TrendMiner connects to an integration component, referred to as 'the connector', that implements the Connector API for the times series data archive.



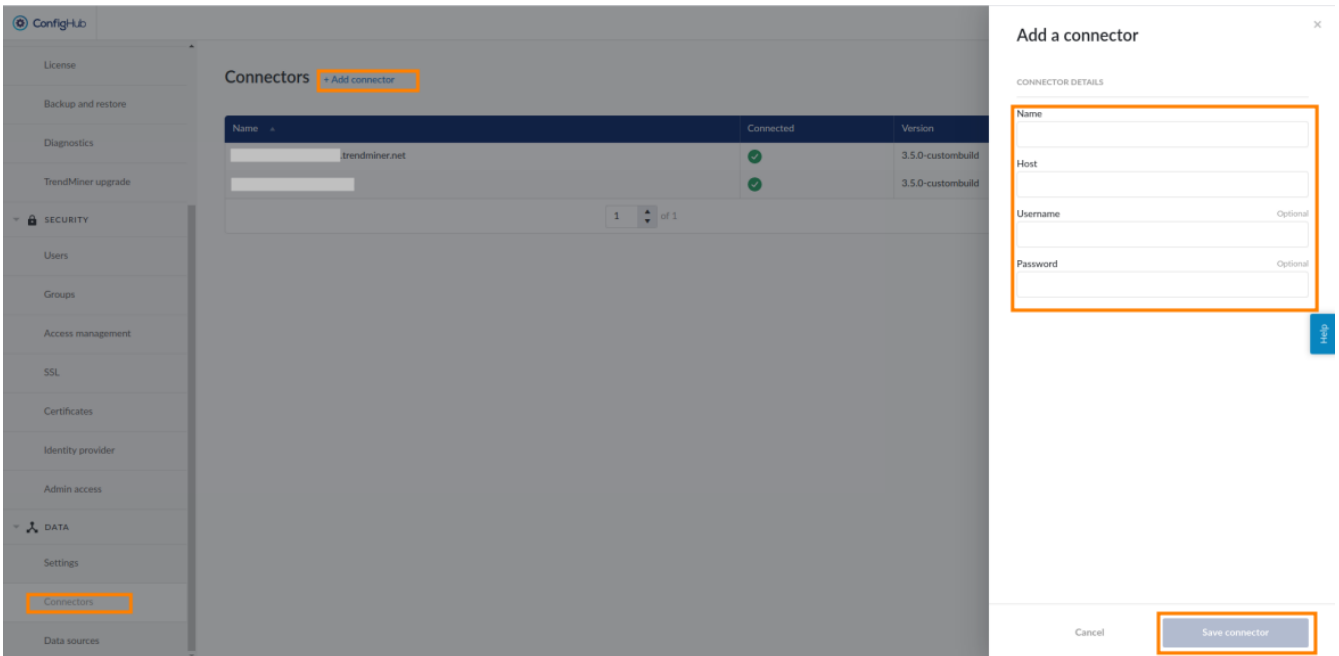
The TrendMiner Plant Integrations connector is one implementation of the Connector API. It is a .NET application, deployed in Microsoft IIS.

Plant Integrations supports the following time series data sources:

- OSIsoft PI Server
- AspenTech IP21
- Yokogawa Exaquantum
- Honeywell PHD
- GE Proficy
- Wonderware Historian
- Local SQLite
- Generic ODBC or OLEDB

Consult the [TrendMiner online help](#) for more information about ODBC and OLEDB connectivity.

One TrendMiner instance can connect to multiple Connector API implementations. Connectors are configured in TM ConfigHub:



When developing a custom implementation of the Connector API, several API endpoints need to be implemented. Others are optional. The next chapters examine the steps that are required, the optional steps and contain the reference documentation of the API.



Chapter 1. Implementation Steps

In order to create a custom TrendMiner time series connector, start by implementing the following 4 endpoints.

1.1. Step 1: Provide TrendMiner with your API version

```
GET /api/version/
```

1.2. Step 2: List your data sources

```
GET /api/database/
```

1.3. Step 3: Resolve tag metadata & values

```
GET /api/v2/tags/  
GET /api/v2/tags/indexvalues/
```

As soon as these are working properly, enter the connector details in ConfigHub. Authentication is optional and only necessary if the set-up requires it. Only Basic Authentication is supported.

**In this API documentation you will find more calls than necessary to build a new time series connector. Please do not hesitate to contact your API support team (support@trendminer.com) if you have any questions on that.*

**If the custom connector accepts datasource calls all API endpoints should have a trailing slash ("GET /api/version/"). Otherwise it is optional.*

1.4. Examples

1.4.1. Version

```
[admin@localmachine ~]$ curl https://tm-connector-ds-poc.azure-api.net/api/version/  
{"version": "3.0.5"}
```

Notes:

- TrendHub will show an error message when the minimal required version is not met.
- The version corresponds to the version of TrendMiner Plant Integrations that implements the minimal required version of the API.

1.4.2. Data sources

One instance of a Connector API implementation can connect to multiple data sources. A list of these sources needs to be returned.

Users can also select which source to query. Administrators can grant specific users permissions on one source only.

```
[admin@localmachine ~]$ curl https://tm-connector-ds-poc.azure-api.net/api/database/
[
  {
    "DbId": 1,
    "Name": "PoCHistorian",
    "Prefix": "",
    "TagFilter": "",
    "Provider": "HC",
    "DataSource": "https://tm-connector-ds-poc.azure-api.net/",
    "UserId": "",
    "Password": "*****",
    "Version": "other",
    "OnlySupportsRawValues": true
  }
]
```

Notes:

- All fields are required to be present, but most can be left blank.
- Fields follow the PascalCase naming convention, as is common in .NET.
- The **Name** free-text field needs to be unique per TrendMiner installation. The name is provided when querying the list of time series or data.
- The **OnlySupportsRawValues** boolean field indicates whether the analytics optimization described in the next chapter is supported.
- The **Provider** free-text field refers to the connection type used for this connection.
- The **DbId** field will only be used when updating the configuration settings from within ConfigHub with the **PUT**, **DELETE**, **TEST** `/api/database/{id}` calls. It is not required to support this.
- The other fields can be used freely to allow further configuration of the connection.

1.4.3. Tag Metadata

A 'tag' refers to one time series. TrendMiner requires the metadata of all tags to be known before they can be used.

All tags in one data source need to be returned in a flat list. To provide additional structure, the Assets endpoints can be implemented.

The query for metadata should accept one query parameter:

- `historianName` name of the time series data source that should be queried.

```
[admin@localmachine ~]$ curl https://tm-connector-ds-poc.azure-
api.net/api/v2/tags/?historianName=PoCHistorian/
[
  {
    "Name": "React5_temp001",
    "Description": "Reactor 5 - Cooling water temperature",
    "Units": "C",
    "Type": "ANALOG"
  },
  {
    "Name": "React5_press",
    "Description": "Reactor 5 - Top pressure",
    "Units": "",
    "Type": "ANALOG"
  },
  {
    "Name": "React5_phas",
    "Description": "Reactor 5 - Production Phases",
    "Units": "",
    "Type": "STRING"
  }
]
```

All fields are required. `Description` and `Units` can be left blank.

TrendMiner supports 4 tag types:

- **ANALOG**: numerical sensor data with linear interpolation (temperature, pressure, flowrate, ...)
- **DISCRETE**: numerical value with stepped interpolation (value stays same till next datapoint)
- **DIGITAL**: a finite list of possible numerical values, mapped to text labels using the `/api/v2/tags/digitalstates/` call.
- **STRING**: string value with stepped interpolation (product name, batch number, production phase)

Notes:

- All numerical values in **ANALOG** and **DISCRETE** tags are **floats**. Discrete does **not** mean only integer values are allowed.
- **booleans** can be sent as **DISCRETE** values (0/1) or **STRING** (true/false). Prefer the **DISCRETE** type for efficiency reasons.
- **string** values have a maximum of 255 characters.
- **DIGITAL** tags can often be represented as **DISCRETE** or **STRING** tags. This avoids the need to implement the `/api/v2/tags/digitalstates/` endpoint.

1.4.4. Values

TrendMiner requests values for a given tag in a specific time period while indexing data.

Implementations of the Connector API can choose to either return raw values, or implement the analytics optimization as described in the next chapter. Implementing the analytics optimization is only recommended when the data source natively supports generating plots. TrendMiner will efficiently generate plots, even when only raw data is provided for indexing.

Four query parameters are important:

- **historianName** is the name of the time series data source that should be queried.
- **tagName** is the name of the tag.
- **startDate** is the RFC3339 start date of the time period. TrendMiner will always query using UTC timestamps.
- **endDate** is the RFC3339 end date of the time period. TrendMiner will always query using UTC timestamps.

The other query parameters are relevant for the plot optimized representation only.

```
[admin@localmachine ~]$ curl 'https://tm-connector-ds-poc.azure-
api.net/api/v2/tags/indexvalues?historianName=PoCHistorian&tagName=React5_temp001&tagT
ype=ANALOG&interpolationType=DEFAULT&startDate=2018-01-01T00:00:00Z&endDate=2018-01-
02T00:00:00Z&numberOfIntervals=25' /
[
  {
    "Ts": "2018-01-01T00:00:00.0000000Z",
    "Value": "73.1889496"
  },
  {
    "Ts": "2018-01-01T00:31:24.0000000Z",
    "Value": "85.7846146"
  },
  {
    "Ts": "2018-01-01T01:31:24.0000000Z",
    "Value": "98.4507751"
  },
  {
    "Ts": "2018-01-01T02:31:24.0000000Z",
    "Value": "98.1346054"
  },
  {
    "Ts": "2018-01-01T03:40:54.0000000Z",
    "Value": "81.8376465"
  },
  {
    "Ts": "2018-01-01T06:33:24.0000000Z",
    "Value": "13.6114807"
  },
  },
```

```
{
  "Ts": "2018-01-01T07:37:24.0000000Z",
  "Value": "0.9693058"
},
{
  "Ts": "2018-01-01T08:42:24.0000000Z",
  "Value": "3.3837295"
},
{
  "Ts": "2018-01-01T09:56:24.0000000Z",
  "Value": "23.6519966"
},
{
  "Ts": "2018-01-01T12:30:24.0000000Z",
  "Value": "85.4783173"
},
{
  "Ts": "2018-01-01T13:39:24.0000000Z",
  "Value": "99.194191"
},
{
  "Ts": "2018-01-01T14:44:24.0000000Z",
  "Value": "96.293663"
},
{
  "Ts": "2018-01-01T15:57:24.0000000Z",
  "Value": "75.9762878"
},
{
  "Ts": "2018-01-01T18:30:54.0000000Z",
  "Value": "14.3683872"
},
{
  "Ts": "2018-01-01T19:39:54.0000000Z",
  "Value": "0.7672905"
},
{
  "Ts": "2018-01-01T20:44:54.0000000Z",
  "Value": "3.7891531"
},
{
  "Ts": "2018-01-01T21:57:54.0000000Z",
  "Value": "24.2102547"
},
{
  "Ts": "2018-01-02T00:00:00.0000000Z",
  "Value": "73.1889496"
}
]
```

Each data point in the response contains two fields:

- **Ts** is the RFC3339 timestamp of the point.
- **Value** is a string, containing either:
 - a numerical value using the **.** as decimal separator,
 - or a string value.

Notes:

- The default time period used by TrendMiner to query data is one month. This is called the 'indexing granularity'. It is often helpful to adjust this setting in ConfigHub. For example, when a data source has one archive per day, set the granularity to one day. TrendMiner can request shorter time periods than the configured granularity, but will never request a longer range.

Chapter 2. Analytics Optimization

When connecting to data sources that have powerful and fast built-in aggregation capabilities, the data transfer overhead of sending all raw values can be reduced by implementing the 'analytics optimization'. Connector API implementations that want to enable this need to set the `OnlySupportsRawValues` flag in the connection details to `false` and implement the `/api/v2/tags/plotvalues/` endpoint next to `/api/v2/tags/indexvalues/`.

Note that this is absolutely not required to be implemented. TrendMiner works equally well with raw values as described earlier.

2.1. Principles

When the `OnlySupportsRawValues` flag is set to `false`, both the `/api/v2/tags/indexvalues/` and `/api/v2/tags/plotvalues/` endpoints need to implement the exact same logic as described here. Each request will include a `numberOfIntervals` query parameter. The requested time period should be divided in this number of equal-length intervals. For each interval, the response should include at most 4 data points.

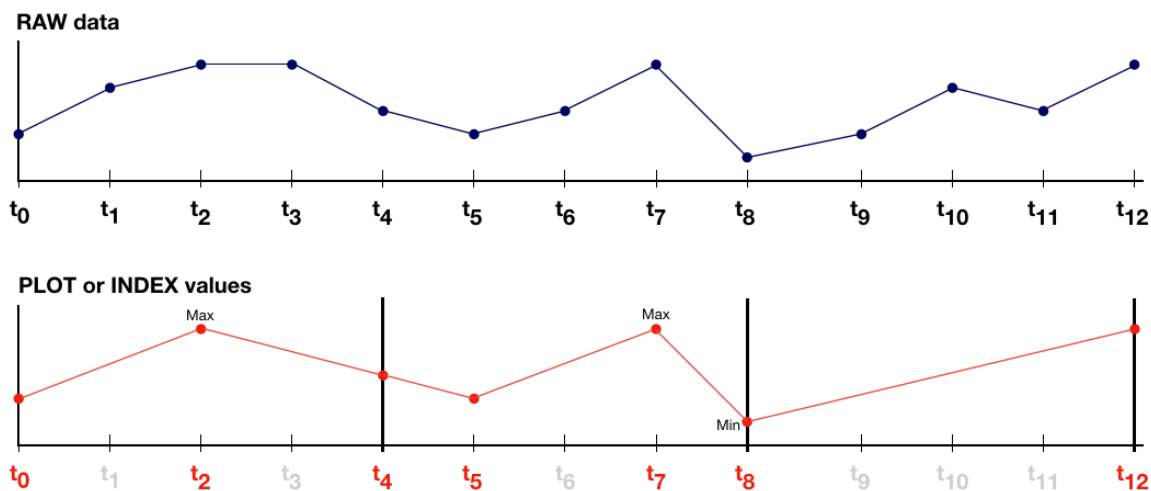
- first value within each interval
- last value within each interval
- minimum value within an interval, if lower than first/last
- maximum value within an interval, if higher than first/last

Intervals that do not contain data at all do not result in data points in the response. Intervals that contain less than 4 data points, result in only as many data points as they contain.

Data points at the edges of the requested time period should also be returned or interpolated.

2.2. Example

Here is a simplified example for the call `startDate=t0&endDate=t12&numberOfIntervals=3`:

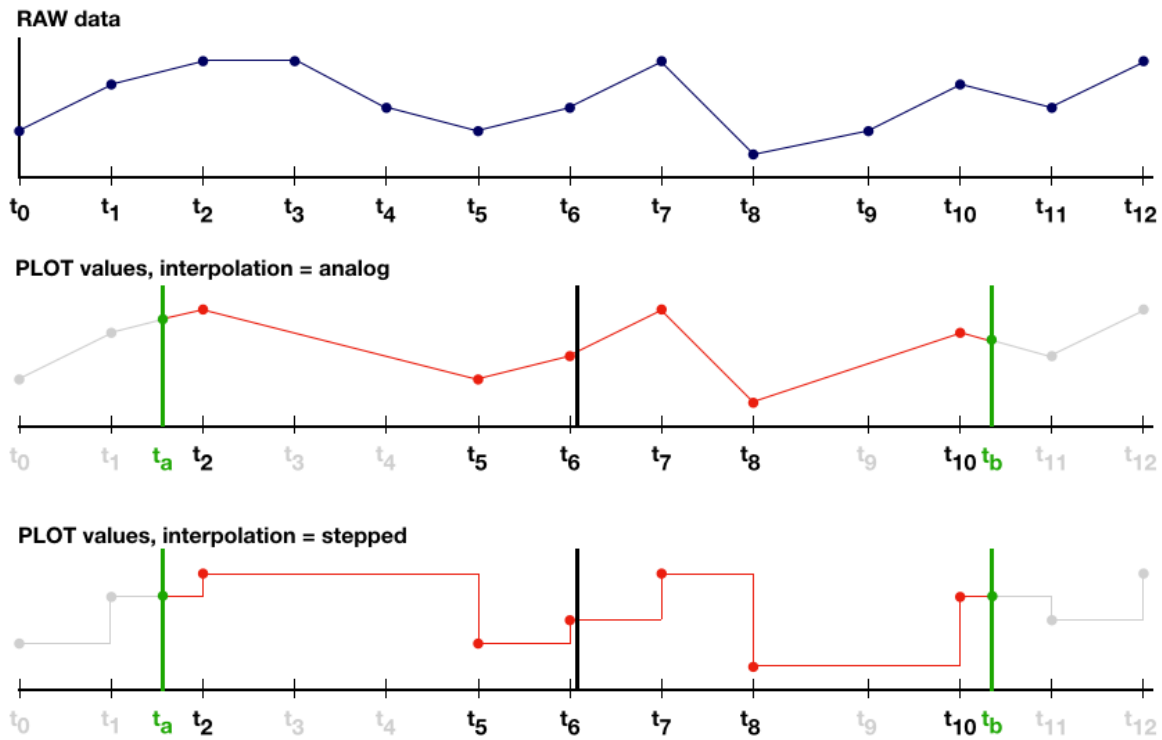


The response in this example would be:

```
[
  {
    "Ts": "t0",
    "Value": "2"
  },
  {
    "Ts": "t2",
    "Value": "5"
  },
  {
    "Ts": "t4",
    "Value": "3"
  },
  {
    "Ts": "t5",
    "Value": "2"
  },
  {
    "Ts": "t7",
    "Value": "5"
  },
  {
    "Ts": "t8",
    "Value": "1"
  },
  {
    "Ts": "t12",
    "Value": "5"
  }
]
```

2.3. Between Values & Interpolation

In the real world it is rare to see such a coincidence as in the previous example where the `startDate`, the `endDate`, and all the intervals fall exactly on values within the raw data. In the example below we've used the same raw data but requested a `startDate=ta` and `endDate=tb` that fall in between raw values in the archive. Even the divide between two intervals does not fall in line with a specific value.



The response in this case looks like this:

	interval 1	interval 2
first value	t2	t7
last value	t6	t10
min	t5	t8
max	-	-

On the edges of a `plotvalues` call, TrendMiner also expects to get the exact values **on** the `startDate` and the `endDate`, respectively according to the type of interpolation. This in order to create a complete view in the graphical representation (trend). In this case both `ta` and `tb` 's values have to be added to the response array.

Retrieving the analog value for `ta` and `tb` can be done through linear interpolation. For example:

$$V_{ta} = V_{t1} + (((ta-t1) * (V_{t2}-V_{t1})) / (t2-t1))$$

Since `t1` and `t11` may not be known in each data source, one possible solution could be to add another interval before and after and respectively use their `lastValue` (for `t1`) and `firstValue` (for `t11`).

2.4. Endpoint

The `/api/v2/tags/(plot|index)values/` endpoints receive a few query parameters that are helpful when implementing the analytics optimization.

- `tagType` is the type of the tag as known by TrendMiner based on the metadata response. This is passed to avoid the need to query the source to find the type of tag in case the implementation

branches between for example `STRING` and `ANALOG` tags.

- `interpolationType` is either `DEFAULT`, `LINEAR` or `STEPPED`. This parameter can be ignored. TrendMiner allows overriding the interpolation type for IP21 data sources only.
- `numberOfIntervals` is the number of intervals to divide the time interval in.

This in addition to the parameters described earlier, for reference:

- `historianName` is the name of the time series data source that should be queried.
- `tagName` is the name of the tag.
- `startDate` is the RFC3339 start date of the time period. TrendMiner will always query using UTC timestamps.
- `endDate` is the RFC3339 end date of the time period. TrendMiner will always query using UTC timestamps.

Notes:

- In contrast to the case of `OnlySupportsRawValues=false`, TrendMiner will not use fixed intervals (or granularity) when requesting data. The user can request plots of any time period.

Chapter 3. Resources

3.1. Af

3.1.1. Returns the root assets with their details. WARNING - this endpoint is deprecated. Please use /assets.

GET /af

Parameters

Type	Name	Description	Schema
Query <i>optional</i>	databaseName	The name of the AF database to connect to.	

Responses

HTTP Code	Description	Schema
200	OK	List[AFAssetNode]
500	Server Error	SystemException

Produces

- application/json

3.1.2. Returns details for the specified asset. WARNING - this endpoint is deprecated. Please use /assets.

GET /af/{id}

Parameters

Type	Name	Description	Schema
Path required	id	The id of the asset.	String
Query <i>optional</i>	databaseName	The name of the AF database to connect to.	

Responses

HTTP Code	Description	Schema
200	OK	AFAssetNode
500	Server Error	SystemException

Produces

- application/json

3.1.3. Returns details for the attribute of a specified asset. **WARNING - this endpoint is deprecated. Please use /assets.**

GET /af/{nodeId}/attribute/{attributeId}

Parameters

Type	Name	Description	Schema
<i>Path required</i>	nodeId	The id of the asset.	String
<i>Path required</i>	attributeId	The id of the attribute.	String
Query <i>optional</i>	databaseName	The name of the AF database to connect to.	

Responses

HTTP Code	Description	Schema
200	OK	AFAssetAttribute
500	Server Error	SystemException

Produces

- application/json

3.2. Assets

3.2.1. Returns details for the root assets in the hierarchy.

GET /assets

Parameters

Type	Name	Description	Schema
Query <i>required</i>	databaseName	The name of the Assets database to connect to.	

Responses

HTTP Code	Description	Schema
200	OK	List[AssetNode]
400	Bad Request. When validation errors occur.	

404	Not Found. When the specified database does not exist.	
500	Server Error	SystemException
502	Bad Gateway. The connector received an invalid response from the historian.	

Produces

- application/json

3.2.2. Returns details for the specified asset.

GET /assets/{id}

Parameters

Type	Name	Description	Schema
Path required	id	The id of the asset.	String
Query required	databaseName	The name of the Assets database to connect to.	

Responses

HTTP Code	Description	Schema
200	OK	AssetNode
400	Bad Request. When validation errors occur.	
404	Not Found. When the specified database or asset does not exist.	
500	Server Error	SystemException
502	Bad Gateway. The connector received an invalid response from the historian.	

Produces

- application/json

3.2.3. Returns details for the assets with the specified ids.

POST /assets

Parameters

Type	Name	Description	Schema
Query required	databaseName	The name of the Assets database to connect to.	

Body <i>required</i>	ids	The ids of the assets.	[string]
--------------------------------	------------	------------------------	----------

Responses

HTTP Code	Description	Schema
200	Returns an array with the details of the specified assets.	List[AssetNode]
400	Bad Request. When validation errors occur.	
404	Not Found. When the specified database does not exist.	
500	Server Error	SystemException
502	Bad Gateway. The connector received an invalid response from the historian.	

Produces

- application/json

3.3. Configuration

3.3.1. Deletes the configured historian connection with the given ID.

DELETE /database/{id}

Parameters

Type	Name	Description	Schema
Path required	id	The ID of the group to delete.	String

Responses

HTTP Code	Description	Schema
200	OK. Deleted.	String
401	Unauthorized	
403	Forbidden	
404	Not Found	
500	Server Error	SystemException

Produces

- application/json

3.3.2. Retrieve the configured historian connection with the given ID.

GET /database/{id}

Parameters

Type	Name	Description	Schema
Path required	id	The ID of the connection to retrieve.	String

Responses

HTTP Code	Description	Schema
200	OK	HistorianConnection
400	Bad Request. When validation errors occur.	
401	Unauthorized	
403	Forbidden	
404	Not Found	
500	Server Error	SystemException

Produces

- application/json

3.3.3. Update the configured historian connection with the given ID.

PUT /database/{id}

Parameters

Type	Name	Description	Schema
Path required	id	The ID of the group to update.	String
Body required	connection	The connection configuration to update. You can provide an ID, but this will be ignored, the ID specified in the path will be used.	HistorianConnection

Responses

HTTP Code	Description	Schema
200	OK	String
400	Bad Request. When validation errors occur.	
401	Unauthorized	
403	Forbidden	
404	Not Found	
500	Server Error	SystemException

Produces

- application/json

3.3.4. Test the historian connection with the given ID.

GET /database/{id}/test

Parameters

Type	Name	Description	Schema
Path required	id	The ID of the connection to retrieve.	String

Responses

HTTP Code	Description	Schema
200	OK	HistorianConnection
400	Error opening configured connection.	inline_response_400
401	Unauthorized	
403	Forbidden	
404	Not Found	
500	Server Error	SystemException

Produces

- application/json

3.3.5. Creates a new historian connection configuration on the central connector.

POST /database

Parameters

Type	Name	Description	Schema
Body required	historianConnection	The connection to add to the connector API. You can provide an ID, but this will be ignored. A unique ID is assigned and returned by the API.	HistorianConnection

Responses

HTTP Code	Description	Schema
200	Created. Returns the ID of the new connection configuration.	String

400	Bad Request. When validation errors occur.	
401	Unauthorized	
403	Forbidden	
404	Not Found	
500	Server Error	SystemException

Produces

- text/plain
- application/json

3.3.6. Retrieve all the configured historian connections.

GET /database

Parameters

Type	Name	Description	Schema
------	------	-------------	--------

Responses

HTTP Code	Description	Schema
200	OK	List[HistorianConnection]
401	Unauthorized	
403	Forbidden	
404	Not Found	
500	Server Error	SystemException

Produces

- application/json

3.4. Context

3.4.1. Returns a list of changed context items in the source system as well as the type of change that happened.

GET /context/changes

Parameters

Type	Name	Description	Schema
Query <i>required</i>	databaseName	The name of the Context database to connect to.	

Query <i>optional</i>	since	The start date of the query period. Internet DateTime format, see RFC3339 e.g.: '2012-01-01T00:00:00Z'.	
Query <i>optional</i>	page	The page to get. This is a cookie that is returned by a previous call to the /changes endpoint.	
Query <i>optional</i>	size	The maximum number of returned items.	

Responses

HTTP Code	Description	Schema
200	OK	ContextItemChangeStreamResponse
400	Bad Request. When validation errors occur.	ContextException
404	Not Found. When the specified database name cannot be found.	ContextException
500	Server Error	SystemException
502	Bad Gateway. The connector received an invalid response from the historian.	ContextException

Produces

- application/json

3.4.2. Returns a list of the available fields in the source system.

GET /context/field

Parameters

Type	Name	Description	Schema
Query <i>required</i>	databaseName	The name of the Context database to connect to.	
Query <i>optional</i>	page	The page number.	
Query <i>optional</i>	size	The size of the page.	

Responses

HTTP Code	Description	Schema
-----------	-------------	--------

200	OK	FieldPagedResponse
500	Server Error	SystemException

Produces

- application/json

3.4.3. Returns a list of the available context items in the source system.

GET /context/

Parameters

Type	Name	Description	Schema
Query <i>required</i>	databaseName	The name of the Context database to connect to.	
Query <i>required</i>	startDate	The start date of the query period. Internet DateTime format, see RFC3339 e.g.: '2012-01-01T00:00:00Z'.	
Query <i>required</i>	endDate	The end date of the query period. Internet DateTime format, see RFC3339 e.g.: '2012-01-01T00:00:00Z'.	
Query <i>optional</i>	createdAfter	Getting only items created after this date. Internet DateTime format, see RFC3339 e.g.: '2012-01-01T00:00:00Z'.	
Query <i>optional</i>	page	The page number.	
Query <i>optional</i>	size	The size of the page.	

Responses

HTTP Code	Description	Schema
200	OK	ContextItemPagedResponse
400	Bad Request. When validation errors occur.	ContextException
404	Not Found. When the specified af database name or context items cannot be found.	ContextException
500	Server Error	SystemException

502	Bad Gateway. The connector received an invalid response from the historian.	ContextException
-----	---	------------------

Produces

- application/json

3.4.4. Returns a specific context item by identifier.

GET /context/{id}

Parameters

Type	Name	Description	Schema
Path <i>required</i>	id	The ID of the connection to retrieve.	String
Query <i>required</i>	databaseName	The name of the Context database to connect to.	

Responses

HTTP Code	Description	Schema
200	OK	ContextItem
400	Bad Request. When validation errors occur.	ContextException
404	Not Found. When the specified af database name or context items cannot be found.	ContextException
500	Server Error	SystemException
502	Bad Gateway. The connector received an invalid response from the historian.	ContextException

Produces

- application/json

3.4.5. Returns a list of the available context items by identifiers.

POST /context/

Parameters

Type	Name	Description	Schema
Query <i>required</i>	databaseName	The name of the Context database to connect to.	

Body <i>required</i>	ids	The ids of the context items.	[string]
--------------------------------	------------	-------------------------------	----------

Responses

HTTP Code	Description	Schema
200	OK	ContextItemListResponse
400	Bad Request. When validation errors occur.	ContextException
404	Not Found. When the specified af database name or context items cannot be found.	ContextException
500	Server Error	SystemException
502	Bad Gateway. The connector received an invalid response from the historian.	ContextException

Produces

- application/json

3.4.6. Returns a list of the available types in the source system.

GET /context/type

Parameters

Type	Name	Description	Schema
Query <i>required</i>	databaseName	The name of the Context database to connect to.	
Query <i>optional</i>	page	The page number.	
Query <i>optional</i>	size	The size of the page.	

Responses

HTTP Code	Description	Schema
200	OK	TypePagedResponse
500	Server Error	SystemException

Produces

- application/json

3.4.7. Returns a list of the available workflows in the source system.

GET /context/workflow

Parameters

Type	Name	Description	Schema
Query <i>required</i>	databaseName	The name of the Context database to connect to.	
Query <i>optional</i>	page	The page number.	
Query <i>optional</i>	size	The size of the page.	

Responses

HTTP Code	Description	Schema
200	OK	WorkflowPagedResponse
500	Server Error	SystemException

Produces

- application/json

3.5. Ef

3.5.1. Returns a list of event frames found in the first configured asset framework. WARNING - this endpoint is deprecated. Please use /context.

GET /ef

Parameters

Type	Name	Description	Schema
Query <i>optional</i>	startDate	The start date of the query period, defaults to one day ago.	
Query <i>optional</i>	endDate	The end date of the query period, defaults to now.	
Query <i>optional</i>	databaseName	The name of the AF database to connect to.	

Responses

HTTP Code	Description	Schema
200	OK	List[EventFrame]
500	Server Error	SystemException

Produces

- application/json

3.6. Informational

3.6.1. Returns the version of the TrendMiner historian connector.

GET /version

Parameters

Type	Name	Description	Schema
------	------	-------------	--------

Responses

HTTP Code	Description	Schema
200	OK	inline_response_200
401	Unauthorized	
403	Forbidden	
404	Not Found	
0	Server Error	SystemException

Produces

- application/json

3.7. Provider

3.7.1. Returns provider capabilities.

GET /provider

Parameters

Type	Name	Description	Schema
------	------	-------------	--------

Responses

HTTP Code	Description	Schema
-----------	-------------	--------

200	OK	ProviderCapabilities
-----	----	----------------------

Produces

- application/json

3.8. Support

3.8.1. Deletes all files in the logs folder.

DELETE /log

Parameters

Type	Name	Description	Schema
------	------	-------------	--------

Responses

HTTP Code	Description	Schema
204	Sucessfully deleted all files on the logs folder.	
500	Server Error	SystemException

Produces

- /

3.8.2. Get the log file content.

GET /log

Parameters

Type	Name	Description	Schema
------	------	-------------	--------

Responses

HTTP Code	Description	Schema
200	Log file as attachment.	
500	Server Error	SystemException

Produces

- text/plain

3.8.3. Changes log level, e.g. INFO, DEBUG, ERROR, WARN, TRACE.

PUT /log

Parameters

Type	Name	Description	Schema
Query <i>required</i>	level	NLog Level name.	

Responses

HTTP Code	Description	Schema
200	Successfully changed log level.	
500	Server Error	SystemException

Produces

- text/plain

3.9. Tags

3.9.1. List all digital states of a tag.

GET /v2/tags/digitalstates

Parameters

Type	Name	Description	Schema
Query <i>required</i>	historianName	The historian name to which the tag belongs to.	
Query <i>required</i>	tagName	The tag name.	

Responses

HTTP Code	Description	Schema
200	OK	List[State]
500	Server Error	SystemException

Produces

- application/json

3.9.2. Returns all tags for specified historian.

GET /v2/tags

Parameters

Type	Name	Description	Schema
Query <i>required</i>	historianName	The name of the historian to list tags of.	

Responses

HTTP Code	Description	Schema
200	OK	List[TagDetails]
500	Server Error	SystemException

Produces

- /

3.9.3. Returns tag values for specified period for indexing purposes.

GET /v2/tags/indexvalues

Parameters

Type	Name	Description	Schema
Query <i>required</i>	historianName	The name of the historian of the tag to fetch values.	
Query <i>required</i>	tagName	The name of the tag to search in.	
Query <i>required</i>	tagType	The type of tag.	
Query <i>required</i>	interpolationType	The interpolation type of the tag.	
Query <i>required</i>	startDate	The start date of the period to search. Internet DateTime format, see RFC3339 e.g.: '2012-01-01T00:00:00Z'.	
Query <i>required</i>	endDate	The end date of the period to search. Internet DateTime format, see RFC3339 e.g.: '2012-01-01T00:00:00Z'.	
Query <i>required</i>	numberOfIntervals	The number of intervals to return data for, this determines the resolution of the data.	

Responses

HTTP Code	Description	Schema
200	OK	List[TimeSeriesEntry]
500	Server Error	SystemException

Produces

- /

3.9.4. Returns the last values of multiple tags for the last specified seconds.

POST /v2/tags/lastvalues

Parameters

Type	Name	Description	Schema
Query <i>required</i>	historianName	The historian name to which the tag belongs to.	
Body <i>required</i>	tagRequest	The tags to fetch the last values.	TagLastValuesRequest

Responses

HTTP Code	Description	Schema
200	OK	List[Tag]
500	Server Error	SystemException

Produces

- application/json

3.9.5. Returns tag values for specified period for plotting purposes.

GET /v2/tags/plotvalues

Parameters

Type	Name	Description	Schema
Query <i>required</i>	historianName	The name of the historian of the tag to fetch values.	
Query <i>required</i>	tagName	The name of the tag to search in.	

Query <i>required</i>	tagType	The type of tag.	
Query <i>required</i>	interpolationType	The interpolation type of the tag.	
Query <i>required</i>	startDate	The start date of the period to search. Internet DateTime format, see RFC3339 e.g.: '2012-01-01T00:00:00Z'.	
Query <i>required</i>	endDate	The end date of the period to search. Internet DateTime format, see RFC3339 e.g.: '2012-01-01T00:00:00Z'.	
Query <i>required</i>	numberOfIntervals	The number of intervals to return data for, this determines the resolution of the data.	

Responses

HTTP Code	Description	Schema
200	OK	List[TimeSeriesEntry]
500	Server Error	SystemException

Produces

- /

3.9.6. Returns tag raw values for specified period.

GET /v2/tags/rawvalues

Parameters

Type	Name	Description	Schema
Query <i>required</i>	historianName	The name of the historian of the tag to fetch values.	
Query <i>required</i>	tagName	The name of the tag to search in.	
Query <i>required</i>	startDate	The start date of the period to search. Internet DateTime format, see RFC3339 e.g.: '2012-01-01T00:00:00Z'.	

Query <i>required</i>	endDate	The end date of the period to search. Internet DateTime format, see RFC3339 e.g.: '2012-01-01T00:00:00Z'.	
---------------------------------	----------------	---	--

Responses

HTTP Code	Description	Schema
200	OK	List[TimeSeriesEntry]
500	Server Error	SystemException

Produces

- /

Chapter 4. Models

4.1. *AFAssetAttribute*

An object that represents an asset attribute.

Field Name	Required	Type	Description	Format
Id	X	String		
Name	X	String		
Description		String		
Template		AFAssetNodeReference		
Attributes		List of AFAssetNodeAttribute		

4.2. *AFAssetNode*

An object that represents an asset node details.

Field Name	Required	Type	Description	Format
Id	X	String		
Name	X	String		
Description		String		
Children		List of AFAssetNodeReference		
Template		AFAssetNodeReference		
Attributes		List of AFAssetNodeAttribute		

4.3. *AFAssetNodeAttribute*

An object that represents an asset attribute.

Field Name	Required	Type	Description	Format
Description		String		

Field Name	Required	Type	Description	Format
Template		AFAssetNodeReference		
PIPoint	X	String		
Id	X	String		
Name	X	String		

4.4. AFAssetNodeReference

An object that represents an asset child.

Field Name	Required	Type	Description	Format
Id	X	String		
Name	X	String		

4.5. AssetNode

An object that represents an asset node details.

Field Name	Required	Type	Description	Format
identifier		String		
parentIdentifier		String		
type		String		Enum: ASSET, ATTRIBUTE,
name		String		
description		String		
children		List of AssetNodeReference		
template		TemplateReference		
data		String		
dataType		String		Enum: DATAREFERENC E, NUMERIC, STRING,

4.6. *AssetNodeReference*

An object that represents a reference to an asset.

Field Name	Required	Type	Description	Format
identifier		String		
type		String		<i>Enum:</i> ASSET, ATTRIBUTE,

4.7. *Component*

An object that represents a component.

Field Name	Required	Type	Description	Format
identifier		String		
type		String		<i>Enum:</i> ASSET, ATTRIBUTE, TAG,

4.8. *ContextDataType*

An object that represents an embedded context data type.

Field Name	Required	Type	Description	Format
identifier		String		
name		String		
workflowIdentifier		String		

4.9. *ContextException*

A generic exception object used when a context error occurs.

Field Name	Required	Type	Description	Format
message		String		
errorCode		String		

Field Name	Required	Type	Description	Format
timestamp		String		
httpStatus		BigDecimal		int32
details		List of [string]		

4.10. *ContextItem*

An object that represents a context item details.

Field Name	Required	Type	Description	Format
identifier		String		
name		String		
description		String		
events		List of Event		
components		List of Component		
type		ContextDataType		
fields		List of [map]		
keywords		List of [string]		
lastModifiedDate		String		

4.11. *ContextItemChange*

An object that represents a context item change details.

Field Name	Required	Type	Description	Format
changedItem		ContextItem		
changeType		String		<i>Enum:</i> ADD, MODIFY, DELETE,

4.12. *ContextItemChangeStreamedResponse*

An object that represents a list of context item changes along with next page cookie.

Field Name	Required	Type	Description	Format
content		List of ContextItemChange		

Field Name	Required	Type	Description	Format
nextPage		String		

4.13. *ContextItemListResponse*

An object that represents a list of context items.

Field Name	Required	Type	Description	Format
items		ContextItem		

4.14. *ContextItemPagedResponse*

An object that represents a list of context items along with page information.

Field Name	Required	Type	Description	Format
content		List of <i>ContextItem</i>		
page		Page		

4.15. *Event*

An object that represents an event.

Field Name	Required	Type	Description	Format
identifier		String		
occured		Date		date-time
state		String		

4.16. *EventFrame*

An event frame.

Field Name	Required	Type	Description	Format
Id		String		
Name		String		
Template		String		
ElementId		String		

Field Name	Required	Type	Description	Format
StartDate		Date		date-time
EndDate		Date		date-time

4.17. Field

An object that represents a field.

Field Name	Required	Type	Description	Format
identifier		String		
name		String		
type		String		<i>Enum:</i> STRING, ENUMERATION, NUMERIC,
placeholder		String		
options		List of [string]		

4.18. FieldPagedResponse

An object that represents a list of fields along with page information.

Field Name	Required	Type	Description	Format
content		List of Field		
page		Page		

4.19. HistorianConnection

All the configuration parameters for a historian connection defined for the TrendMiner central connector API.

Field Name	Required	Type	Description	Format
DbId		BigDecimal		int32
Name	X	String		
Prefix	X	String		
Provider	X	String		

Field Name	Required	Type	Description	Format
DataSource	X	String		
UserId		String		
Password		String		
Version		String		
TagFilter		String		regex
OnlySupportsRaw Values		Boolean		
Properties		List of HistorianDatabase Property		

4.20. *HistorianDatabaseProperty*

The configuration properties.

Field Name	Required	Type	Description	Format
key	X	String		
value		String		

4.21. *InlineResponse200*

Field Name	Required	Type	Description	Format
version	X	String	The version string of the connector.	

4.22. *InlineResponse400*

Field Name	Required	Type	Description	Format
message	X	String	The error message, describes details of why the connection failed.	

4.23. *Page*

An object that represents a page information.

Field Name	Required	Type	Description	Format
size		BigDecimal		int32
totalElements		BigDecimal		int32

Field Name	Required	Type	Description	Format
totalPages		BigDecimal		int32
number		BigDecimal		int32

4.24. *ProviderCapabilities*

An object that represents a provider capabilities.

Field Name	Required	Type	Description	Format
provider		String		
experimental		Boolean		
capabilities		List of [string]		
properties		List of ProviderCapabilitiesProperty		

4.25. *ProviderCapabilitiesProperty*

An object that represents a provider capabilities property.

Field Name	Required	Type	Description	Format
name		String		
type		String		
capabilities		List of [string]		
required		Boolean		

4.26. *ReferenceById*

An object that represents a reference by id.

Field Name	Required	Type	Description	Format
identifier		String		

4.27. *State*

A State

Field Name	Required	Type	Description	Format
Code	X	BigDecimal		
Offset	X	BigDecimal		
Name	X	String		

4.28. *StringStreamedResponse*

An object that represents a list of strings along with next page cookie.

Field Name	Required	Type	Description	Format
content		List of [string]		
nextPage		String		

4.29. *SystemException*

A generic exception object used when an unspecified server error occurs.

Field Name	Required	Type	Description	Format
Message		String		
ExceptionMessage		String		
ExceptionType		String		
StackTrace		String		

4.30. *Tag*

A response object for a single tag values.

Field Name	Required	Type	Description	Format
Name	X	String		
Historian	X	String		
Entries	X	List of TimeSeriesEntry		

4.31. *TagDetails*

All details for a tag.

Field Name	Required	Type	Description	Format
Name	X	String		
Description	X	String		
Historian	X	BigDecimal		int32
Units	X	String		
Type	X	String		Enum: ANALOG , DIGITAL , DISCRETE, STRING,

4.32. TagLastValuesRequest

A request object to retrieve values from tags for the specified last seconds.

Field Name	Required	Type	Description	Format
LastSeconds	X	BigDecimal		int32
TagNames	X	List of [string]		

4.33. TemplateReference

An object that represents a reference to a template.

Field Name	Required	Type	Description	Format
identifier		String		
name		String		

4.34. TimeSeriesEntry

A single timeseries entry.

Field Name	Required	Type	Description	Format
Ts	X	Date		date-time
Value	X	String		

4.35. *Type*

An object that represents a context data type.

Field Name	Required	Type	Description	Format
identifier		String		
name		String		
workflow		ReferenceById		
color		String		
fields		List of ReferenceById		

4.36. *TypePagedResponse*

An object that represents a list of context data types along with page information.

Field Name	Required	Type	Description	Format
content		List of Type		
page		Page		

4.37. *Workflow*

An object that represents a workflow.

Field Name	Required	Type	Description	Format
identifier		String		
name		String		
startState		String		
endState		String		
states		List of [string]		

4.38. *WorkflowPagedResponse*

An object that represents a list of workflows along with page information.

Field Name	Required	Type	Description	Format
content		List of Workflow		
page		Page		

