Time Series Data Services

Table of Contents

Overview

The time series data service REST APIs are used to create, read, update, and delete instantaneous and aggregate (summary) time series data. The PUT method handles the create and update functions as upserts. If the record exists, it is updated. If it does not exist, it is created. A time series record consists of a timestamp, one reading (measured quantity) value, and an optional measurement qualifier for reading. Multi-valued readings are not supported. Writing a time series record with the same timestamp, entity instance, and measurement type as a record in the database completely overwrites the old record. There is no versioning of the data.

The timestamp supports up to millisecond precision. The reading value datatype must match the type that was defined in the measurement type `DATA_TYPE_CD` field. The `INST_VALIDATOR_CHAIN` and `AGGR_VALIDATOR_CHAIN` fields specify the validations to be performed when reading data.

The measurement qualifier is a qualifier string for the time series data values being measured, for example RAW, CLEAN, and so on. If a measurement qualifier is not specified, DEFAULT is used. It is case-insensitive and saved in the database in uppercase.

You can write multiple records for an entity instance and measurement type with one call to the service. You can read and delete records by identifying the entity instance, measurement type, and time range.

You must have the appropriate permissions to read, write, and delete time series data. You can only interact with entity instances that belong to your tenant.

When writing or deleting data, the data service writes data synchronously to database. You can enable an option to queue the incoming request and perform the physical writes to the database asynchronously by sending a parameter in the header of the request. If you write time series data and immediately try to read it back, it may not yet be present in the database and may not be returned. If you delete data and then read that time range, the data may not have been physically deleted yet, and can be returned in the response. Requests for an entity are processed in the order they were received.

Data services can perform the following validations:

- The entity must exist and not be logically deleted.
- The measurement type must exist on the entity's entity type.
- The relation between entity and measurement type must exist.
- The reading values must be consistent with the datatypes defined on the measurement type.

Common HTTP Codes

The following table lists the HTTP response codes the time series and aggregate APIs can return.

Code	Model	Description
200	The response model depends on the type of service and is covered in the details for the specific service.	Success.
204	The response body is empty.	Success: data written.
400	**Bad Request** `{ "id" : "string", "message" : "The request is not valid." }`	The request is not well-formed, and the server does not understand it.
401	**Unauthorized** `{ "id" : "string", "message" : "Not authorized to access this resource." }`	The user does not have access to the resource.
404	**Not Found** `{ "id" : "string", "message" : "Resource not found." }`	The specified resource does not exist.
422	**Unprocessable** `{ "id" : "string", "message" : "The request is unprocessable ." }`	The server understands the request but is unable to process it as it is invalid.
500	**Error** `{ "id" : "string", "message" : "string" }`	A runtime exception occurred.

HTTP Headers

The following table lists the HTTP headers the time series and aggregate APIs use.

Header Type	Name	Description	Type
Request header	async	Set this parameter to true for an asynchronous write to the database. The value of this optional header can be either true or false. If any other value is provided, it is an error condition. Default value is false.	String
Response header	hasMore	Indicates if there are more records present in the database. Value can be either true or false. If true, further calls are required to fetch more records, else if false no more records are present in the database.	String

API Details

Time Series Resource

GET - Time Series Data

The GET method is used to read the latest time series data for an entity and all its associated measurement types. There is no ability to search for time series data based on filters, such as asking for all values greater than "X". There is no ability to request data for multiple entities in one call. The response message includes the latest reading of each measurement type associated with the entity.

This method requires time series read permission (ds.ts.r).

GET Field Details - Time Series Data

The entity referred to in the API URLs is a unique identifier of an entity. The measurement type is the ID of the measurement type, not the measurement type name. The measurement type is case sensitive. The timestamps in the response are in ISO8601 format, using the UTC time zone. The service returns the latest value only if a record exists for the current month and the previous month in the database. For example, if a request is made on 2018-08-20 and the last record that was inserted for any measurement type was on or after 2018-07-01, it will be returned, otherwise no value will be returned.

```
http://{host:port}/em-timeseriessvcs/api/emtimeseries/v1/timeseries/SVC_PT:2
```

The following request has an entity type DEVICE with no measurement type and an entity ID of 1.

```
http://{host:port}/em-timeseriessvcs/api/emtimeseries/v1/timeseries/DEVICE:1
```

The `${HOST_NAME}:{PORT}` represents the host and port of the em-timeseriessvcs application instance or load balancer (if multiple em-timeseriessvcs instances are deployed behind a load balancer) in EnergyIP.

Both cases will return a response like the following one:

```
[{

"readTime": "2018-06-15T00:56:35.363Z",
"measType": "xyz",
"readings":[
{
"value": "10211",
"name": null,
"flags": "zvy",
"datatype": "Int"
```

```
}
],
"attributes":{
"additionalProp1": "string1",
"additionalProp2": "string",
"additionalProp3": "string"
}
}]
```

GET - Time Series Data for a Measurement Type

The GET method is used to read time series data for an entity and its associated measurement type in a given time range. There is no ability to search for time series data based on filters, such as asking for all values greater than "X". There is no ability to request data for multiple entities in one call. The response message includes the latest reading of each measurement type associated with the entity.

This method requires time series read permission (ds.ts.r).

GET Field Details - Time Series Data for a Measurement Type

The measurement type is the ID of the measurement type, not the measurement type name. The measurement type is case sensitive. The timestamps in the response are in ISO8601 format, using the UTC time zone.

When specifying a date range, both "from" and "to" must be specified, and the maximum range of records that can be returned is the month containing the "to" date plus the previous month. For example, if the "from" date is 2018-06-20 and the "to" date is 2018-08-20, then the records from 2018-07-01 to 2018-08-20 will be returned. If no date range is specified (neither "from" nor "to" are specified), the service returns the latest available record, provided a record exists in the current or previous month. For example, if a request is made on 2018-08-20 and the last record that was inserted for any measurement type was on or after 2018-07-01, it will be returned otherwise no value will be returned.

There is an optional select parameter that defines which data to return. The value of the select parameter is a comma-separated list of requested data. The field name can be one of the following:

- readtime
- readings
- attributes

If no select parameter is specified, all the fields will be returned.

The datatype field in the readings array of a response is the value of the `DATA_TYPE_CD` for the associated measurement type. It can be used to typecast the reading value into a specific datatype in client applications.

For single-valued readings, the name field will always be null as that field is not saved in the database.

GET Example - Time Series Data for a Measurement Type

In this example, the entity type SVC_PT has a measurement type set EQI which has ID as 1 and there is an instance of forklift with an entity ID of 2.

To read the time series data from 2018-05-17T03:56:35.362Z to 2018-08-30T03:56:33.363Z for the EQI measurement type, make the following call:

```
http://{host:port}/em-
timeseriessvcs/api/emtimeseries/v1/timeseries/SVC_PT:2/1?from=2018-05-
17T03:56:35.362Z&to=2018-08-30T03:56:33.363Z
```

This returns the following response:

```
[{

"readTime": "2018-06-15T00:56:35.363Z",
"readings":[
{
"value": "10211",
"name": null,
"flags": "zvy",
"datatype": "Int"
}
],
"attributes":{
"additionalProp1": "string1",
"additionalProp2": "string",
"additionalProp3": "string"
}
}]
```

PUT - Time Series Data

The PUT method is used to write time series data for an entity and measurement type. There is no ability to write data for multiple entities or multiple measurement types in one request. An array of records for an entity and measurement type can be written with one request. The maximum size of a write payload is 1 megabyte. If data already exists for any part of the array, the old data is overwritten by the new data. There is no versioning of time series data.

Timestamps are supported at a millisecond resolution. Time series data can be of type int, long, double, and string. Strings are validated using two bytes per character, integers use four bytes, longs and doubles use eight bytes.

The maximum range for a double is -9.99999999999999999999999999999999999E+125 to 9.99999999999999999999999999999999999E+125.

Data is processed in the order that it is received, so if two records for the same timestamp are received for the same entity and attributes, the last record to be sent overwrites the earlier record.

This method requires time series write permission (ds.ts.w).

PUT Field Details - Time Series Data

The measurement type is the ID of the measurement type, not the measurement type name. The request fields name is case insensitive. By default any put request to the API will write directly to the database without any buffering, but with the async header parameter set to true, buffering of requests can be done. Timestamps are specified in ISO8601 format, for example 2017-09-21T14:08:22.345+02:00.

The following validations are applied to the request:

- The entity must exist and not be logically deleted.
- The measurement type relation with the entity must exist, that is, the read time must lie within the effective period of the measurement type relation with the entity.
- The reading values must be consistent with the datatypes defined in the measurement type.

PUT - Data Type Handling

Storage

For storage of read values (value field in reading object of request), the value in the request is mapped to one of three Cassandra datatypes -- BigInt, Double or Text. The data type to which the read value should be mapped is defined in reference data for the measurement type (device or svc_pt) of the entity in request. The `DATA_TYPE_CD` column of the `MEAS_TYPE` table is used for this purpose. The following values are allowed in the `DATA_TYPE_CD` field. This field is case insensitive.

DATA_TYPE_CD	Cassandra Data Type
Short, Integer, Int, Long	BigInt
Decimal, Float, Double	Double
Bigint, BigInteger, BigDecimal, Char, Character, String, Clob, Text, Timestamp, Date, DateTime, Duration, Boolean	Text

Validation

If the `INST_VALIDATOR_CHAIN` column of the `MEAS_TYPE` table specifies that the instDataTypeCheck validation should be performed, then the format of the Short, Integer, Int, BigInt, BigInteger, Long, Decimal, BigDecimal, Float, and Double data types is checked by

casting to its corresponding Java type when reading data. BigInt and BigInteger both map to the BigInteger Java type.

The Max/Min value and format restrictions are the same as that of the corresponding Java types.

There is no format or Max/Min value check for Char, Character, String, Clob, Text, Timestamp, Date, DateTime, Duration, or Boolean data types.

PUT Example - Time Series Data

In this example, the entity type, SVC_PT, has a measurement type set EQI, having ID as 1.

To write one time series records for the EQI, make the following call:

```
http://{host:port}/em-
timeseriessvcs/api/emtimeseries/v1/timeseries/SVC_PT:2/1
```

This will return the following payload:

```
[
  {
    "readTime": "2017-11-25T03:56:35.363Z",
    "measQualifier": "string",
    "readings": [
      {
        "value": "1316",
        "name": "a",
        "flags": "zvy"
      }
    ],
    "attributes": {
      "additionalProp1": "string1",
      "additionalProp2": "string",
      "additionalProp3": "string"
    }
  }
]
```

DELETE - Time Series Data

The DELETE method is used to delete time series data for an entity, measurement type, and time range. There is no ability to request deletes for multiple entities or multiple measurement type in one call. Data is physically deleted and is not recoverable. There is no versioning of time series data.

The request for instantaneous data can only be triggered for a maximum of 730 days, that is the difference between the "from" and "to" dates of the request cannot be more than 730 days.

This method requires time series delete permission (ds.ts.d).

DELETE Field Details - Time Series Data

The measurement type is the ID of the measurement type, not the measurement type name. The "from" and "to" parameter times are specified in ISO8601 format, for example 2017-09-21T14:08.345:18+02:00. The service will delete the time series records with timestamps greater than the "from" time and less than or equal to the "to" time. For example, a request from 09:00 to 10:00 will delete the records with timestamps greater than 09:00 and up to and including 10:00.

DELETE Example - Time Series Data

In this example, the entity type SVC_PT has a measurement type set EQI, and there is an instance of forklift with an entity ID of 2 and a measurement type ID as 1.

To delete data from 2018-06-17T03:56:35.363Z to 2018-06-18T03:56:35.364Z for the EQI measurement type, make the following call:

```
http://{host:port}/em-
timeseriessvcs/api/emtimeseries/v1/timeseries/SVC_PT:2/1?from=2018-06-
17T03:56:35.363Z&to=2018-06-18T03:56:35.364Z
```

Note that this request would not delete a record with timestamp 2018-06-17T03:56:35.363Z, but would delete a record with timestamp 2018-06-18T03:56:35.364Z.

Time Series Aggregate Values Resource

GET - Time Series Aggregate Data

The GET method is used to read the latest time series aggregate data for an entity and all its associated measurement types. The API does not do aggregation on its own; it is expected that all the aggregation is done on the client side. The API will return whatever is put into it. There is no ability to search for time series data based on filters, such as asking for all values greater than "X".

The aggregation interval duration is specified by the following interval values (in integers) and units:

Interval Values	Interval Units
1-60	minutes
1-24	hourly

1 or higher	daily
1 or higher	monthly

The response message includes the latest reading of each measurement type associated with the entity for the current year.

This method requires time series read permission (ds.tsa.r).

GET Field Details - Time Series Aggregate Data

The entity referred to in an API URL is a unique identifier of an entity. The measurement type is the ID of the measurement type, not the measurement type name. The measurement type is case sensitive. The timestamps in the response are in ISO8601 format, using the UTC time zone. The service returns the latest value only if a record exists for the current month and the previous month in the database. For example, if a request is made on 2018-08-20 and the last record that was inserted for any measurement type was on or after 2018-07-01, it will be returned otherwise no value will be returned.

The entity ID is the ID value returned from GET or POST requests to the Entity Service API. The intervalValue field is an integer value indicating how many units (intervalUnit) are requested for the interval duration. The intervalUnit field can be minute, hour, day, or month.

GET Example - Time Series Aggregate Data

In this example, the entity type SVC_PT has no measurement type.

To read the aggregate data for the all the measurement types associated with service point for the particular period, make the following call:

```
http://{host:port}/em-
timeseriessvcs/api/emtimeseries/v1/aggregates/SVC_PT:2?intervalValue=15&inter
valUnit=hourly
```

This will return the following response:

```
[{
"startTime": "2018-07-01T00:05:17.526Z",
"endTime": "2018-07-01T00:05:17.526Z",
"aggregates":[
{
"name": null,
"count": 11,
"sum": 202.2,
"min": 33.3,
"max": 44.4,
```

```
"first": 55.5,
"last": 66.6,
"weightedAvg": 77.7,
"flags": "xyz1"
}
],
"attributes":{
"additionalProp11": "string1",
"additionalProp22": "string2",
"additionalProp33": "string3"
}
}]
```


GET - Time Series Aggregate Data for a Measurement Type

This method is used to read aggregated data for an entity, measurement type, and time range. The API does not do aggregation on its own, it is expected that all the aggregation is done on the client side. The API will return whatever is put into it.

The aggregation interval duration is specified by the following interval values (in integers) and units:

Interval Values	Interval Units
1-60	minutes
1-24	hourly
1 or higher	daily
1 or higher	monthly

The response message includes the latest reading of each measurement type associated with the entity.

This method requires time series read permission (ds.tsa.r).

GET Field Details - Time Series Aggregate Data for a Measurement Type

The entity referred to in API URLs is a unique identifier of an entity. The measurement type is the ID of the measurement type, not the measurement type name. The measurement type is case sensitive. The timestamps in the response are in ISO8601 format, using the UTC time zone. The service returns the latest value only if a record exists for the current month and the previous month in the database. For example, if a request is made on 2018-08-20 and the last record that was inserted for any measurement type was on or after 2018-07-01, it will be returned otherwise no value will be returned.

The entity ID is the ID value returned from GET or POST requests to the Entity Service API. The intervalValue field is an integer value indicating how many units (intervalUnit) are requested for the interval duration. The intervalUnit field can be minute, hour, day, or month.

There is an optional select parameter that defines which data to return. The value of the select parameter is a comma-separated list of requested data. The field name can be one of the following:

- starttime
- endtime
- aggregates
- attributes

If no select parameter is specified, all the fields will be returned.

GET Example - Time Series Aggregate Data for a Measurement Type

In this example, the entity type SVC_PT has a measurement type set EQI, and there is an instance of forklift with an entity ID of 2 and a measurement type ID as 1.

To read the time series data from 2018-05-17T03:56:35.362Z to 2018-08-30T03:56:33.363Z for the EQI measurement type, make the following call:

```
http://{host:port}/em-
timeseriessvcs/api/emtimeseries/v1/aggregates/SVC_PT:2/1?intervalValue=15&int
ervalUnit=hourly&from=2018-05-12T01:08:17.525Z&to=2018-08-12T01:08:17.526Z
```

This will return the following response:

```
[{
 "startTime": "2018-07-01T00:05:17.526Z", "endTime": "2018-07-
01T00:05:17.526Z", "aggregates":[ { "name": null, "count": 11, "sum": 202.2,
"min": 33.3, "max": 44.4, "first": 55.5, "last": 66.6, "weightedAvg": 77.7,
"flags": "xyz1" } ], "attributes":{ "additionalProp11": "string1",
"additionalProp22": "string2", "additionalProp33": "string3" } }]
```

PUT - Time Series Aggregate Data

The PUT method is used to write time series data for an entity and measurement type. There is no ability to write data for multiple entities or multiple measurement types in one request. An array of records for an entity and measurement type can be written with one request. The maximum size of a write payload is 1 megabyte. If data already exists for any part of the array, the old data is overwritten by the new data. There is no versioning of time series data.

Timestamps are supported at a millisecond resolution. Time series data can be of type int, long, double, string. Strings are validated using two bytes per character, integers use four bytes, longs and doubles use eight bytes.

The maximum range for a double is -9.99999999999999999999999999999999999E+125 to 9.99999999999999999999999999999999999E+125.

Data is processed in the order that it was received, so if two records for the same timestamp are received for the same entity and measurement type, the last record to be sent overwrites the earlier record.

This method requires time series write permission (ds.tsa.w).

PUT Field Details - Time Series Aggregate Data

The measurement type is the ID of the measurement type, not the measurement type name. The request fields name is case insensitive. By default, any PUT request to the API will directly write to the database without any buffering, requests can be buffered by setting the async header parameter to true. Timestamps are specified in ISO8601 format, for example 2017-09-21T14:08:22.345+02:00.

The following validations are applied to the request:

The entity must exist and not be logically deleted.

The measurement type relation with the entity must exist, that is, the read time must lie within the effective period of the measurement type relation with entity.

PUT Example - Time Series Aggregate Data

In this example, the entity type SVC_PT has a measurement type set EQI having ID as 1.

To write one time series record for the EQI, make the following call:

```
http://{host:port}/em-
timeseriessvcs/api/emtimeseries/v1/aggregates/SVC_PT:2/1
```

This will return the following payload:

```
[
  {
    "startTime": "2016-05-30T00:08:17.526Z",
    "endTime": "2016-05-30T00:08:17.526Z",
        "measQualifier": "string",
    "intervalValue": 15,
    "intervalUnit": "hourly",
    "aggregates": [
      {
        "name": "abc",
        "count": 800,
        "sum": 20.2,
        "min": 30.3,
        "max": 40.4,
```

```
        "first": 50.5,
        "last": 60.6,
        "weightedAvg": 70.7,
        "flags": "xyz"
      }
    ]
  }]
```

DELETE - Time Series Aggregate Data

The DELETE method is used to delete time series data for an entity, measurement type, and time range. There is no ability to request deletes for multiple entities or multiple measurement type in one call. Data is physically deleted and is not recoverable. There is no versioning of time series data.

This method requires time series delete permission (ds.tsa.d).

DELETE Field Details - Time Series Aggregate Data

The measurement type is the ID of the measurement type, not the measurement type name. The "from" and "to" parameter times are specified in ISO8601 format, for example 2017-09-21T14:08.345:18+02:00. The service will delete the time series records with timestamps greater than the "from" time and less than or equal to the "to" time. For example, a request from 2017-09-21T14:08.345:18+02:00 09:00 to 10:00 will delete the records with timestamps greater than 09:00 and up to and including 10:00.

DELETE Example - Time Series Aggregate Data

In this example, the entity type SVC_PT has a measurement type set EQI with measurement type ID as 1.

To delete data from 2018-06-17T03:56:35.363Z to 2018-06-18T03:56:35.364Z for the EQI measurement type, make the following call:

```
http://{host:port}/em-
timeseriessvcs/api/emtimeseries/v1/aggregates/SVC_PT:2/1?from=2018-06-
17T03:56:35.363Z&to=2018-06-18T03:56:35.364Z
```

Note that this request would not delete a record with timestamp 2018-06-17T03:56:35.363Z, but would delete a record with timestamp 2018-06-18T03:56:35.364Z.

The Time Series Data Service uses a cache to get the entity information. By default, entities are stored in the cache for 1 hour after they are first requested. However, you can configure the `timeToLive` property, which is used to specify the refresh rate of the EDIG cache, to change that default minimum time. If the entity is activated or deactivated through Asset Services, it could take the time specified in the `timeToLive` property to reflect in the cache.