Time Series Data API

You can use the Time Series Data API microservice to create, read, update, and delete time series data. The time series data is stored against an entity instance and a property set name in the database. A time series record consists of a timestamp, one or more property values, and an optional quality indicator for each property. If there are multiple properties in the property set, the microservice expects them to arrive in the same payload. Writing a time series record with the same timestamp, entity instance, and property set as a record in the database overwrites the old record since there is no versioning of data.

You must specify the timestamp with millisecond precision. Measurement values must match the type that was defined in the property set. In the property set type definition, the qualityCode field indicates whether the quality code data accompanies the property time series data. If the qualityCode value is Y for a property, the quality code property name is the property name with a "_qc" suffix. For example, the quality code property for the "temperature" property would be "temperature_qc".

You can write multiple records for an entity instance and property set with a single call to the microservice. The maximum size of a write payload is 1 megabyte. Writes for an individual entity and property set are throttled to 100 kilobytes per second, with each record consuming at least 1 kilobyte. You can read and delete records by IDentifying the entity instance, property set, and time range. You can delete only 30 days of data at a time.

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

When writing or deleting data, the microservice queues the incoming request and performs physical writes to the database asynchronously. If you write time series data and immediately try to read it back, it may not yet be available in the database and may not return any data. If you delete data and then try to read that time range, the data may not have been physically deleted yet and can be returned in the response. The microservice processes the requests for an entity in the order that they were received.

The microservice performs the following validations:

- The application token "ten" value must be the owner of the entity specified
- The entity must exist and not be logically deleted
- The property set must exist on the entity's entity type
- The properties in the payload must exist on the property set type definition
- Quality code values are valid if the qualityCode field is set to Y for that property
- The property values must be consistent with the datatypes defined on the property set

Time Series Data API Details

GET

The GET method reads time series data for an entity, property set name, and a time range. The response message includes the requested data sorted by timestamp with the oldest timestamp first. You cannot search for time series data based on filters, such as asking for all values greater than X. You cannot request data for multiple entities or multiple property sets in one call.

Recent and future time series data is available in the hot storage, and older data is available in the cold storage. The GET method provides results regardless of the storage medium. You might experience slightly slower responses for requests that require retrieving data from the cold storage. A typical data retention period for hot storage is one to two weeks.

The GET method requires time series read permission (iot.tim.r).

GET Parameters

The following table describes the parameters of the GET method.

Parameter	Datatype	Description
entity	string	Unique identifier of the entity
propertysetname	string	The name of the property set
from	date-time	The beginning of the time range to read (exclusive)
to	date-time	The end of the time range to read (exclusive)
limit	integer	The maximum number of entries to read
select	string	Select fields to return

GET Responses

Response Code	Description
200	The array of time series
400	Bad request
401	Unauthorized
404	Entity not found
default	Unexpected error

GET Field Details

The entity ID is the ID value returned from the GET or POST requests to the Entity Service API. The property set name is the name of the property set as defined on the entity type, not the property set type name. The property set name is case insensitive. The from and to time parameters are specified in ISO8601 format, for example, "2017-09-21T14:08:22.345+02:00". The timestamps in the response will also be in ISO8601 format, using the UTC timezone. The service returns time series records with timestamps greater than the "from time" and less than or equal to the "to time". If successive time series requests are made for consecutive hours, each time series record is provided in one response. If no from and to parameter times are specified, the service returns the latest value. If the last value is from more than 12 months ago, it is not returned. For example, a request from 09:00 to 10:00 returns the records with timestamps greater than 09:00 and up to and including 10:00.

There is an optional limit parameter that prevents more than that number of records being returned in the response. The service never returns more than 2000 records in the response, regardless of the setting of this parameter.

There is an optional select parameter that defines the fields to return as a comma-separated list of property names. The parameter can include quality code property names, which are the same as the property name with a "_qc" suffix. For example, the "temperature" property has "temperature_qc" as the quality code property name. If you do not specify a select clause, the service returns all the properties, including quality code properties. The property names are case insensitive.

If you use the "+" sign in the timezone in the from and to parameters, you must convert the "+" character to "%2B" because it is a special character and used in the URL. For example, send "2017-09-21T14:08:00%2B02:00" instead of "2017-09-21T14:08:00+02:00.

GET Example

In this example, there is an entity type "forklift" with a property set "tireMonitor" with the properties "pressure", "temperature", and "treadDepth". The "pressure" property has a qualityCode value of "Y" while the others are "N". There is an instance of forklift with an entity ID of 802.

To read the time series data from 2017-05-01T00:09:00Z to 2017-05-01T00:12:00Z for the "tireMonitor" property set, make a call to:
https://{host}/api/iottimeseries/v3/timeseries/978528e7a124458f87c8f1d38fd9400f/tireMonitor?from=2017-05-01T00:09:00Z&to=2017-05-01T00:12:00Z

You receive the following response.


```
[
{
"pressure": 94,
"pressure_qc": 192,
"temperature": 45,
"treadDepth": 0.475,
"_time": "2017-05-01T00:10:00Z"
},
{
"pressure": 95,
"pressure_qc": 192,
"temperature": 45,
"treadDepth": 0.475,
"_time": "2017-05-01T00:11:00Z"
},
{
"pressure": 93,
"pressure_qc": 192,
"temperature": 44,
"treadDepth": 0.475,
"_time": "2017-05-01T00:12:00Z"
}
]
```

PUT

The PUT method writes time series data for an entity and property set name. You cannot write data for multiple entities or multiple property sets in one request. You can write an array of records for an entity and property set name with one request. The maximum size of a write payload is one megabyte. If data already exists for any part of the array, the service overwrites the old data with the new data. There is no versioning of time series data at this time. If a property set has multiple properties, the service expects to receive a complete set of data for all properties for a timestamp in a payload. If you provide only a subset of the properties, and the other properties arrive at a later time, the original properties are lost. Do not include those properties in the same property set if this is an issue.

Time series data can be of type int, long, double, boolean, string, big_string (blobs), or timestamp. The service supports timestamps at a millisecond resolution. The maximum sizes of strings and big strings are defined in the property set type and can be up to 255 and 100,000, respectively. The maximum size of a time series record is 200,000 bytes. The service validates strings and big strings with two bytes per character, integers with four bytes, longs and doubles with eight bytes, timestamps with eight bytes, and booleans with one byte.

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

When the service receives data, it is put in a queue and then stored. There is a delay before the data is available for reading, depending on the load on the system. The data is processed in the order that it was received. Therefore, if two records for the same timestamp are received for the same entity and property set, the last record overwrites the earlier record.

The service throttles the requests for an individual entity and property set to a maximum of 100 kilobytes per second, with each record consuming a minimum of one kilobyte. For small records (less than one kilobyte), the service can write 100 records in a second. For records larger than one kilobyte, the service rounds up the size of the record to the nearest kilobyte and consumes that number of kilobytes. For 20 kilobyte records, the service can write five records in a second. If the request rate is exceeded, the service will send a "429 Too Many Requests" response. The request should be retried, preferably after a short delay. For example, if a request with 100 one kilobyte records is sent, the service might throttle another PUT request for the same entity and property sent within the following second.

The PUT method requires time series write permission (iot.tim.w).

PUT Parameters

The following table describes the parameters of the GET method.

Parameter	Datatype	Description
entity	string	Unique identifier of the entity
propertysetname	string	The name of the property set
timeseries	array [object]	The time series data array.

PUT Responses

Response Code	Description
204	The time series has been written
400	Bad request
401	Unauthorized
404	Entity not found
429	Too many requests
default	Unexpected error

PUT Field Details

The entity ID is the ID value returned from the GET or POST requests to the Entity Service API. The property set name is the name of the property set as defined on the entity type, not the property set type name. The property names must match the property set type definition. Property set names and property names are case insensitive. Timestamps are specified in ISO8601 format, for example "2017-09-21T14:08:22.345+02:00". If the property set type definition has a qualityCode value of "Y" for a property, the parameter can include quality code property names, which are the same as the property name with a "_qc" suffix. For example, the "temperature" property has "temperature_qc" as the quality code property name. If you do not specify a select clause, the service returns all the properties, including quality code properties. The property names are case insensitive.

Quality code values are integer values, where lower numbers indicate better quality. Generally, the values are based on the OPC UA standard. You can also use other schemes, but the good data must have lower quality code values than uncertain data, which is lower than the bad data. The aggregation process uses quality code data. The service ignores the bad data and counts the uncertain data separately and includes them in the results with the good data.

PUT Example

In this example, there is an entity type "forklift" with a property set "tireMonitor" with the properties "pressure", "temperature", and "treadDepth". The "pressure" property has a qualityCode value of "Y" while the others are "N". There is an instance of forklift with an entity ID of 978528e7a124458f87c8f1d38fd9400f.

To write three time series records for the "tireMonitor" property set, make a call to: https://{host}/api/iottimeseries/v3/timeseries/978528e7a124458f87c8f1d38fd9400f/tireMonitor and send the following payload:

```
[
{
"_time": "2017-05-01T00:10:00Z",
"pressure": 93,
"pressure_qc": 192,
"temperature": 44,
"treadDepth": 0.475,
},
{
```

```
"_time": "2017-05-01T00:11:00Z",
"pressure": 95,
"pressure_qc": 192,
"temperature": 45,
"treadDepth": 0.475,
},
{
"_time": "2017-05-01T00:12:00Z",
"pressure": 94,
"pressure_qc": 192,
"temperature": 45,
"treadDepth": 0.475,
}
]
```


DELETE

The DELETE method deletes time series data for an entity, property set name, and a time range. You cannot request deletes for multiple entities or multiple property sets in one call. The service deletes the data physically and is not recoverable. You can delete up to 30 days of data with one request.

Requests to delete data are put in a queue and then processed. Therefore, there can be a delay before a read request returns no data. The delay depends on the load on the system at the time of the request.

Time series data is stored in the hot storage for recent data and the cold storage for older data. The service deletes the data regardless of the storage medium. You might experience slightly slower responses for requests that require deleting the data from the cold storage. The typical data retention period for the hot storage is two weeks.

The DELETE method requires time series delete permission (iot.tim.d).

DELETE Parameters

The following table describes the parameters of the GET method.

Parameter	Datatype	Description
entity	string	Unique identifier of the entity
propertysetname	string	The name of the property set
from	date-time	The beginning of the time range to read (exclusive)
to	date-time	The end of the time range to read (exclusive)

DELETE Responses

Response Code	Description
204	The time series has been deleted
400	Bad request

Response Code	Description
401	Unauthorized
404	Entity not found
default	Unexpected error

DELETE Field Details

The entity ID is the ID value returned from the GET or POST requests to the Stack Service API. The property set name is the name of the property set as defined on the entity type, not the property set type name. The property set name is case insensitive. The from and to parameter times are specified in ISO8601 format, for example, "2017-09-21T14:08.345:18+02:00". The service deletes 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. You can delete up to 30 days of data with one request.

If you use the "+" sign in the time zone in the from and to parameters, you must convert the "+" character to "%2B" because it is a special character and used in the URL. For example, instead of sending "2017-09-21T14:08:00+02:00", send "2017-09-21T14:08:00%2B02:00".

DELETE Example

In this example, there is an entity type "forklift" with a property set "tireMonitor" with the properties "pressure", "temperature", and "treadDepth". There is an instance of forklift with an entity ID of 978528e7a124458f87c8f1d38fd9400f. To delete data from 2017-05-01T00:09:00Z to 2017-05-01T00:12:00Z for the "tireMonitor" property set, make a call to:
https://{host}/api/iottimeseries/v3/timeseries/978528e7a124458f87c8f1d38fd9400f/tireMonitor?from=2017-05-01T00:09:00Z&to=2017-05-01T00:12:00Z

This request does not delete a record with timestamp "2017-05-01T00:09:00Z", but deletes the record with the timestamp "2017-05-01T00:12:00Z".