

Data Warehouse Usage Documentation

Handshake Protocol

The **Handshake Protocol** is a process by which the client can define the structure and types of the data that they wish to store in the warehouse. It includes a structured json object, which contains required and optional fields in order to properly describe the data.

Handshake Request

Endpoint

`/api/prepare/`

Handshake Request JSON Payload Structure

The request will be a JSON object that has multiple nested object within the main object. The main object, or group object, will be as follows

Group object

- **group_name** - The name of the group that this data belongs to. **This is a required field.**
- **sources** - An array of source objects. **This is a required field.** for the structure of a [Source Object](#) see below
- **classification** - The classification of the data. This is an optional field.

Source Object

A source corresponds with a table in the datawarehouse. Each source should contain metrics that are relevant to that source, and should try to keep all metrics updated at the same time. For example, a GPS device could be a source, holding 3 metrics: X, Y, Z. or Latitude and Longitude etc.

- **name** - The name of the source. **This is a required field.** This should be unique per the group.
- **metrics** - An array of metric objects. **This is a required field.** for the structure of a [Metric Object](#) see below
- **tz_info** - The timezone relevant to the data. This is an optional field.

Metric Object

A metric corresponds to one column in the source's table. The metric is one measurement of a source. For example, if the source is a GPS device, it would have 3 metrics, X, Y, and Z. If the source were a water tower, it would probably hold metrics like 'Current Volume' or 'Temperature' or similar.

- **name** - The name of the metric. **This is a required field.** This should be unique per the source.
- **data_type** - The data type of the metric. **This is a required field.** This should follow the data types listed in the [Data Types](#) section.
- **units** - The units by which the metric is measured by. This is an optional field.
- **asc** - The sort direction of this metric. This is an optional field.

Example Handshake Request

Example Handshake Request

```
{
  "group_name": "My Group",
  "classification": "My Classification",
  "sources": [
    {
      "name": "My Source",
      "tz_info": "America/New_York",
      "metrics": [
        {
          "name": "My Metric",
          "data_type": 8,
          "units": "F",
          "asc": true
        }
      ]
    }
  ]
}
```

Handshake Response

The response object of the handshake request will contain all the same fields that were submitted, but with additional unique identifiers for each field. You will receive a **group_uid** and a **source_uid** for each source that you submitted. You will also receive a **metric_uid** for each metric that you submitted per source. **It is important that these unique identifiers are stored in your system, as they will be used to query your data.**

Example Handshake Response

Example Handshake Response

```
{
  "group_name": "My Group",
  "classification": "My Classification",
  "group_uid": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
  "sources": [
    {
      "name": "My Source",
      "tz_info": "America/New_York",
      "source_uid": "YYYYYYYY-YYYY-YYYY-YYYY-YYYYYYYYYYYY",
      "metrics": [
        {
          "name": "My Metric",
          "data_type": 8,
          "units": "F",
          "asc": true,
          "metric_uid": "ZZZZZZZZ-ZZZZ-ZZZZ-ZZZZ-ZZZZZZZZZZZZ"
        }
      ]
    }
  ]
}
```

Data Types

The following are the supported data types for the metrics. Instead of specifying the name of the metric's data type, you will instead pass in a data type code as an integer.

Data Type	Code
Text	1
TupleType	2
String	3
Integer	4
SmallInteger	5
BigInteger	6
Numeric	7
Float	8
DateTime	9
Date	10
Time	11
LargeBinary	12
Boolean	13

Data Type	Code
Unicode	14
UnicodeText	15
Interval	16

Submitting Data

Endpoint

`/api/store`

In order to submit your data to be stored, you must provide the unique identifiers provided in the [Handshake Request](#), structured in a specific way.

The structure of the request will be based on the source, and will reflect the following:

Example Data Storage Request Payload

[source, json]s

```
{
  "source_uid": "YYYYYYYY-YYYY-YYYY-YYYY-YYYYYYYYYYYY",
  "metrics": [
    {
      "metric_uid": "ZZZZZZZZ-ZZZZ-ZZZZ-ZZZZ-ZZZZZZZZZZZZ",
      "value": 1.0
    }
  ]
}
```

Fields

- `source_uid` - The unique identifier of the source that this data belongs to. This is a required field.
- `metrics` - An array of metric objects. This is a required field.
 - `metric_uid` - The unique identifier of the metric that this data belongs to. This is a required field.
 - `value` - The value of the metric. This is a required field.

Querying your Data

Querying a specific source

Endpoint

`/api/query/<your group id>/<desired source id>/`

Requirements

You must provide a valid pair of `group_uid` and `source_uid` in the url.

By default, this will return the 1000 most recently entered rows in this table, by timestamp. To specify how many rows you would like, use query string parameter `limit=<desired row number>`.

Example

```
/api/XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX/YYYYYYYY-YYYY-YYYY-YYYY-  
YYYYYYYYYYYY?limit=10
```

Specifying your query

In order to filter your data, you must provide a query string. This query string can be highly customized to finetune your query.

Supported Operators

Operation	Query String	Effect	Example
<	__<	Less than operation	?temperature__<=98.6
=<	__<=	Less than or equal to	?temperature__<=98.6
=	__eq	Equals this value	?temperature__eq=98.6
!=	__!	Not equals this value	?temperature__!=98.6
>	__>	Greater than operation	?temperature__>=98.6
>=	__>=	Greater than or equal to	?temperature__>=98.6
like	__like	String is like this value	?model_name__like=Civic
contains	__contains	String contains this value as substring	?model_name__contains=Civic
contained by	__contained_by	String is contained by this value	?car_brand__contained_by=2012% 20Honda%20Civic

These operators can be applied in combination to further filter your query results.

Example of combined query operators

?limit=100&temperature>100&volume<30 The above query string will filter the table where temperature is greater than 100, and the volume is less than 30, and will return only 100 rows at most.

To query a column, you can pass in a url_encoded column name, or the metric_uid you were given upon the handshake request. The metric_uid/column_name that you pass in must be a column or metric of the source.