Importing Data

There are several approaches for importing data into XINA, but for most projects we recommend the **XINA Import** utility.

XINA Import reads XINA API calls from JSON files and passes them to the XINA server. Each JSON file corresponds to a single API action, but may be paired with additional files of other types depending on the content.

For example, to upload some housekeeping data from a CSV requires two files. First, the JSON file:

```
{
 "action": "load",
 "database": "demo.model.data.hk.full",
 "columns": true,
 "delimit": ",",
 "line": "\n",
 "$object_id": "{local}/hk.csv"
}
```

The CSV file then looks like:

```
t,name,value
1602086313288000,SCAN_INDEX(Step),-1
1602086313288000,MO1_LD1_CURR(mA),0
1602086313288000,MO1_LD2_CURR(mA),0
1602086313288000,MO1_CASE_TEC(C),21.739
...
```

A couple notes on these:

- The `"line"` property in the JSON file must exactly match the new line character(s) in the CSV file. You can use `"\r\n"` if you prefer, but either way we recommend explicitly using one or the other when you write the files, as a general "print line" may use different output depending on the platform.
- The `"{local}"` in the `"$object_id"` property is a macro used by the XINA Import application to look for the CSV file in the same location as the JSON file. We recommend using this and keeping the files in the same location. If you need to separate them you can use a full path instead, but this is more fragile if folder organization needs to be changed.
- For best performance we recommend paging the CSV files so that each is 50MB or less.
- If you need to represent an empty value, you can either omit a data point or use `"NULL"` (without quotes). (`"NaN"` and `"Infinity"` are not supported at the database level.)

While this CSV approach is recommended for large data sets, data can alternatively be embedded directly in JSON files. For example, a file to insert a new instant might look like:

```
{
  "action": "insert",
  "database": "demo.model.data.ins",
  "records": [
    {
      "u_id": "58ea870a-52c3-33c7-b858-c20795ec3301",
      "p_id": 0,
      "s_id": 0,
      "type": 20,
      "level": 0,
      "t": 1606333792000000,
      "label": "SPECTRA_Startf-0_Stopf-1k",
      "content": "some additional text here...",
      "meta": {
        "Resolution Bandwidth": 2.07014,
        "Stop Frequency": 1000,
        "Average Factor": 30,
        "Start Frequency": 0
      }
    }
  ]
}
```


The full API reference can be found here.

Revision #3
Created 9 June 2022 19:21:04 by Nick Dobson
Updated 26 April 2024 15:24:54 by Bradley Tse