Sensor Object Library Documentation

Release 1

Thomas Watteyne, Sami Malek, Ziran Zhang, Keoma Brun

May 07, 2018

Contents

1 sol **3**
1.1 openhdlc module 3
1.2 Sol module 3
1.3 SolDefines module 5

2 Indices and tables **7**

Python Module Index **9**

Contents:

CHAPTER 1

sol

1.1 openhdlc module

`openhdlc.`**`dehdlcify`**(*fileName*, *fileOffset=0*, *maxNum=None*)

`openhdlc.`**`hdlcify`**(*inBuf*)

1.2 Sol module

exception `Sol.`**`SolDuplicateOapNotificationException`**

Bases: `exceptions.Exception`

`Sol.`**`bin_to_http`**(*sol_binl*)

Convert a list of binary SOL objects (compound or not) into a JSON string to be sent as HTTP payload to the server.

Parameters **`sol_binl`** (*`list`*) – a list of binary SOL Objects

Returns A JSON string to be sent to the server over HTTP.

Return type string

`Sol.`**`bin_to_json`**(*sol_bin*, *mac=None*)

Convert a binary SOL object into a JSON SOL Object.

Parameters

- **`sol_bin`** (*`list`*) – binary SOL object
- **`mac`** (*`list`*) – A list of byte containing the MAC address of the that created the object

Returns JSON SOL Objects

Rtpe list

`Sol.`**`dumpToFile`**(*sol_jsonl*, *file_name*)

`Sol.`**`dust_to_json`**(*dust_notif*, *mac_manager=None*, *timestamp=None*)
Convert a single Dust serial API notification into a list of JSON SOL Object.

Parameters

- **dust_notif** (*dict*) – The Dust serial API notification as a json object created by the JsonServer application
- **mac_manager** (*list*) – A list of byte containing the MAC address of the manager
- **timestamp** (*int*) – the Unix epoch of the message creation in seconds (UTC)

Returns A list of SOL Object in JSON format

Return type list

`Sol.`**`http_to_bin`**(*sol_http*)
Convert the JSON string contained in an HTTP request into a list of binary SOL objects (compound or not).

Parameters **sol_http** (*string*) – JSON string contained in an HTTP request

Returns list of binary SOL objects (compound or not)

Return type list

`Sol.`**`influxdb_to_json`**(*sol_influxdb*)
Converts an Influxdb query reply into a list of dicts.

Parameters **sol_influxdb** (*dict*) – the result of a database query (sush as SELECT * FROM)

Returns a list of JSON SOL objects

Return type list

`Sol.`**`json_to_bin`**(*sol_json*)
Convert a JSON SOL Object into a single binary SOL Object.

Parameters **sol_json** (*dict*) – a JSON SOL Object

Returns A single binary SOL Object

Return type list

`Sol.`**`json_to_influxdb`**(*sol_json*, *tags*)
Convert a JSON SOL object into a InfluxDB point

Parameters

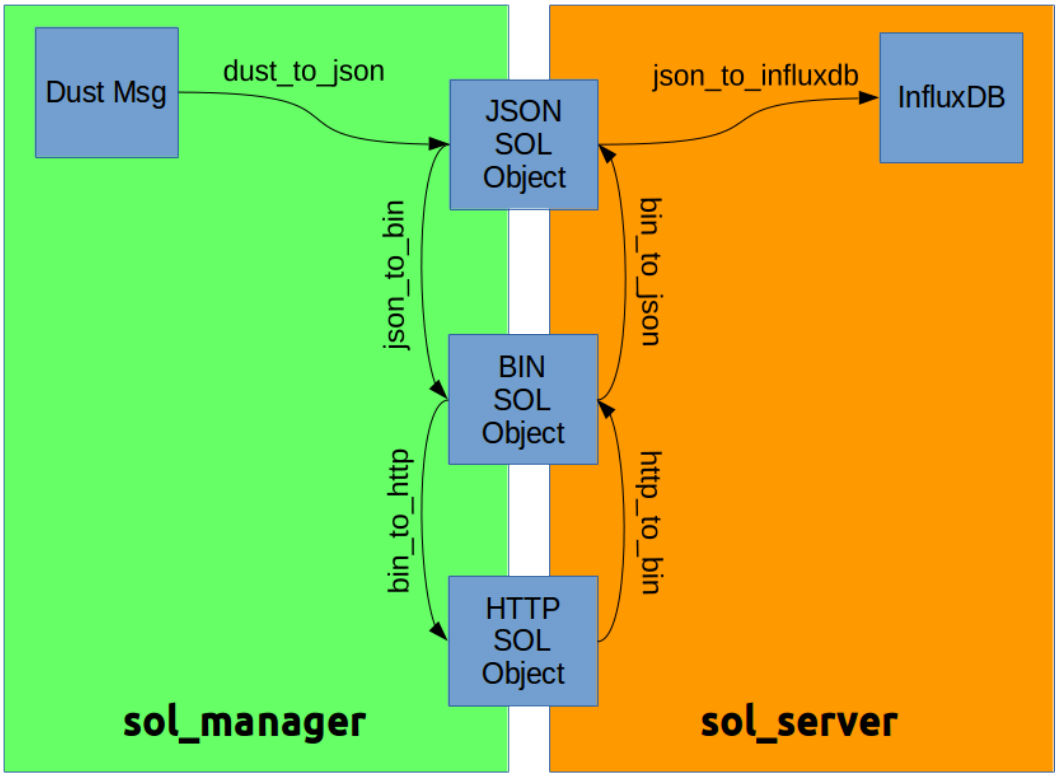
- **sol_json** (*dict*) – JSON SOL object
- **tags** (*dict*) – A dictionary of tags

Returns InfluxDB point

Rtpe list

`Sol.`**`loadFromFile`**(*file_name*, *start_timestamp=None*, *end_timestamp=None*)

`Sol.`**`version`**()



1.3 SolDefines module

`SolDefines.solStructure(type_id)`

Return the SOL structure according to the given type id If the element is not found, it raises a ValueError.

Parameters **type_id** (*int | str*) –

Returns a dictionary that contains the following keys: type, description, structure, fields

`SolDefines.sol_name_to_type(type_name)`

`SolDefines.sol_type_to_type_name(type_id)`

This file describes the SOL Objects structure.

1.3.1 How to add an Object structure

1. Create an issue with name: "Adding YOUR_OBJECT_NAME structure". This will create a issue number like #49.
   2. Create a new branch with the name: `develop_<your issue number>`

      ex: *develop_49*
   3. Add the object type in the list at the top of the SolDefines.py file.
   - Prepend the string "SOL_TYPE" to your object name.
   - Increment the last number of the list to get an object id

Refer to the other object if you are not sure.

4. Add the object structure at the bottom of the SolDefines.py file.

Refere to the python structure to know which field to set: https://docs.python.org/2/library/struct.html

5. Run the the `registry_gen.py` script. That will update the `registry.md` file.
6. Commit your changes starting with the issue number. Commit message example: "`#49 adding YOUR_OBJECT_NAME structure`".
7. Push your changes to the repo: `git push origin develop_49`
8. Create a merge request on branch develop using GitHub UI.

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

O

openhdlc, 3

S

Sol, 3
SolDefines, 5

Index

B

bin_to_http() (in module Sol), 3
bin_to_json() (in module Sol), 3

D

dehdlcify() (in module openhdlc), 3
dumpToFile() (in module Sol), 3
dust_to_json() (in module Sol), 3

H

hdlcify() (in module openhdlc), 3
http_to_bin() (in module Sol), 4

I

influxdb_to_json() (in module Sol), 4

J

json_to_bin() (in module Sol), 4
json_to_influxdb() (in module Sol), 4

L

loadFromFile() (in module Sol), 4

O

openhdlc (module), 3

S

Sol (module), 3
sol_name_to_type() (in module SolDefines), 5
sol_type_to_type_name() (in module SolDefines), 5
SolDefines (module), 5
SolDuplicateOapNotificationException, 3
solStructure() (in module SolDefines), 5

V

version() (in module Sol), 4