

Open APIs  
for Open  
Minds

# SMART DATA MODELS UPDATES

## 13-6-2023

Slides : [https://bit.ly/SDM2\\_20230613](https://bit.ly/SDM2_20230613)

Alberto Abella

Data modelling expert

FIWARE Foundation

[alberto.abella@fiware.org](mailto:alberto.abella@fiware.org)



Smart  
Data Models

# Agenda

- Contribution
  - Contribution manual
  - Templates for contributing
  - Contribution with spreadsheet
  - Online editor
  - Validation of a data model
  - Contribution with a PR
  - Testing service
- Draft a data model based on examples
  - Draft a data model based on a Json example
  - Draft a data model based on a csv example
- Generating examples
  - Key values format
  - Normalized format
- Pysmartdatamodels
  - Examples of code
- Customizing @context
  - Merging several subjects in a single @context
  - Mapping @context with external ontologies
- Future
  - Integration into data spaces

Open APIs  
for Open  
Minds

Contribution

# Contribution manual

## Review of the contribution manual

1. [https://bit.ly/contribution\\_manual](https://bit.ly/contribution_manual)
  - **Schema.json**
  - **Examples of payloads**
  - **CONTRIBUTORS.yaml**
  - **ADOPTERS.yaml**
  - **Notes\_context.jsonld**
  - **Test you data model before contribution**

# Templates for contribution

Home → Tools → [Templates](#)

## 1. Data model for submission

- **schema.json**
- **ADOPTERS.yaml**
- **notes.yaml**
- **Examples**
  - **example.json**
  - **example.jsonld**
  - **example-normalized.json**
  - **example-normalized.jsonld**

# Contribution by spreadsheet

Home → Tools → [MasterSheet for simple contributions](#)

1. Oriented to very simple data models
2. Not capable of coding everything (array and objects should be added manually)
3. Limited validation
4. Requires a google account on drive (free)
5. Contributed to the repository drafted data models
6. Demo: [https://bit.ly/SDM\\_MasterSheet](https://bit.ly/SDM_MasterSheet)

- Make a local copy
- Authorize the code execution
- Include the example
- Include the global parameters
- Include the attributes and descriptions

id	Unique ID of the object
title	Title of the form element
subtitle	Description of the element
serviceElements	List of booking elements See section 3.4.4 of the original document <a href="https://acris.aero/publicdownloads/file/19/">https://acris.aero/publicdownloads/file/19/</a>
footerText	Footer Text

# Online editor

Home → Tools → [Data Model editor](#)

1. It is an external resource (free)
2. Oriented to a bit more complex data models
3. For users with limited knowledge of json schema
4. Demo: [https://bit.ly/SDM\\_DataModelEditor](https://bit.ly/SDM_DataModelEditor)
  - Copy text
  - Open online page
  - Edit the data model
  - Save the schema

# Validation of a data model

Home → Tools → [Validation of a payload \(external\)](#)

1. It is an external resource (free)
2. Located at <https://www.jsonschemavalidator.net/>
3. Allows the test of the validation between the schema and the keyvalues examples
4. Demo:
  - Choose a data model
  - Copy schema
  - Copy keyvalues example
  - Check that is validated



# Contribution with a PR

## Option 1: In the incubated repository

1. Once requested access.
2. Granted permissions. Specific directory
3. Appear in the frontpage
4. (Trello CRM in progress)



Start data model/s

Data Models in progress

## Option 2: In user local repository

5. Need to warn us at [info@smartdatamodels.org](mailto:info@smartdatamodels.org)
6. Likely to be moved to incubated
7. Self check of contribution (next slide)

# Testing service

Home → Tools → [Test you data model](#)

1. Just provide the root of the repository. It provides text feedback and more json detailed support
2. It is tested
  - That the schema.json is
    - Present in the right position
    - It is a valid json and it has the right structure (types and descriptions)
    - It has the right metadata, and the required section
    - External references are valid
    - The descriptions are included
  - That the examples are:
    - Present in the right position
    - Key values are validated by the schema
3. Notes.yaml and ADOPTERS.yaml are present in the right position
4. Welcomed new possible tests to be included
5. Demo:

Open APIs  
for Open  
Minds

**Draft a data model based on examples**

# Draft a data model based on a Json example

Home → tools → Draft a model based on examples → [Generator of data model from json example](#)

1. Include the payload and your mail (need for contact)
2. Identifies data types (mostly)
3. Identifies if the properties are already defined in the database

# Draft a data model based on a csv example

Home → tools → Draft a model based on examples → [Generator of data model from json example](#)

1. Include the name of the columns
2. Include a line of values separated by commas
3. Identifies data types (mostly)
4. Identifies if the properties are already defined in the database

Open APIs  
for Open  
Minds

**Generating examples**

# Generate a NGSI-LD key values payload

Home → Tools → Generator of examples based on schemas → [Generator of NGSI-LD key values examples](#)

1. Available in the readme of all data models
2. Also through a the Tools menu in the frontend
3. Also available as an API call
4. Values are compliant with types but nonsense
5. [Demo](#):

# Generate a NGSI-LD normalized payload

Home → Tools → Generator of examples based on schemas → [Generator of NGSI-LD normalized examples](#)

1. Available in the readme of all data models
2. Also through a the Tools menu in the frontend
3. Also available as an API call
4. Values are compliant with types but nonsense
5. [Demo](#)



Open APIs  
for Open  
Minds

**pysmartdatamodels**

# Python package pysmartdatamodels

Home → Tools → python package pysmartdatamodels → [pysmartdatamodels documentation](#)

1. Integration of the smart data models in other tools
2. Install: *pip install pysmartdatamodels*
3. 15 functions
  - Load all datamodels in a dict like the official list. Function `load_all_datamodels()`
  - Load all attributes in a dict like the official export of attributes. Function `load_all_attributes()`
  - List all data models. Function `list_all_datamodels()`
  - List all subjects. Function `list_all_subjects()`
  - List the data models of a subject. Function `datamodels_subject(subject)`
  - List description of an attribute. Function `description_attribute(subject, datamodel, attribute)`
  - List data-type of an attribute. Function `datatype_attribute(subject, datamodel, attribute)`
  - Give reference model for an attribute. Function `model_attribute(subject, datamodel, attribute)`
  - Give reference units for an attribute. Function `attributes_datamodel(subject, datamodel)`
  - List the attributes of a data model. Function `attributes_datamodel(subject, datamodel)`
  - List the NGSI type (Property, Relationship or Geoproperty) of the attribute. Function `ngsi_datatype_attribute(subject, datamodel, attribute)`
  - Print a list of data models attributes separated by a separator. Function `print_datamodel(subject, datamodel, separator, meta_attributes)`
  - Returns the link to the repository of a subject. Function `subject_repolink(subject)`
  - Returns the links to the repositories of a data model name. Function `datamodel_repolink(datamodel)`
  - **Update the official data model list or the database of attributes from the source. Function `update_data()`**

# Python package pysmartdatamodels

Home → Tools → python package pysmartdatamodels → [pysmartdatamodels code examples](#)

## 1. Code example:

```
from pysmartdatamodels import pysmartdatamodels as sdm
subject = "dataModel.Weather"
dataModel = "WeatherForecast"
attribute = "precipitation"
a = input("Load all data models. Continue?")
print(sdm.load all datamodels())
a = input("Load all attributes. Continue?")
print(len(sdm.load all attributes())) # there is more than 20.000 to get all listed
a = input("List all data models. Continue?")
print(sdm.list all datamodels())
a = input("List all subjects. Continue?")
print(sdm.list all subjects())
a = input("List data models of the subject "+ subject + ". Continue?")
print(sdm.datamodels subject["dataModel.Weather"])
a = input("List descripton of the attribute "+ attribute + ". Continue?")
print(sdm.description attribute(subject, dataModel, attribute))
a = input("List the type of the attribute "+ attribute + ". Continue?")
print(sdm.datatype attribute(subject, dataModel, attribute))
a = input("List the model of the attribute "+ attribute + ". Continue?")
print(sdm.model attribute(subject, dataModel, attribute))
a = input("List the units of the attribute "+ attribute + ". Continue?")
print(sdm.units attribute(subject, dataModel, attribute))
a = input("List the attributes of the data model "+ dataModel + ". Continue?")
print(sdm.attributes datamodel(subject, dataModel))
a = input("List the link to the repository of the subject "+ subject + ". Continue?")
print(sdm.subject repolink(subject))
a = input("List the subjects where this data model "+ dataModel + " appears. Continue?")
print(sdm.datamodel repolink(dataModel))
a = input("Print formatted the data model "+ dataModel + " of the subject "+ subject + ". Continue?")
print(sdm.print_datamodel(subject, dataModel,"", ["property", "type", "dataModel", "repoName", "description", "typeNGSI", "modelTags", "format", "units",
"model", ]))
a = input("Update all data models from the central repository. Continue?")
sdm.update data()
print("Thanks for attending this session")
```

Open APIs  
for Open  
Minds

Customizing context

# Merging several subjects' contexts

Home → Tools → subjects' [@context merger](#)

1. Multisubject requirements
2. Merge all non-repeated terms
3. Identify clashing elements and it should be solved manually
4. Demo:

# Mapping @context with external ontologies

Home → Tools → [Mapper @context to external ontologies](#)

1. When SDM IRIS are not the right ones for the project
2. Currently all terms and entity names have their automatic SDM URL (providing an actual web page)
3. List of mapped ontologies (on demand extended)

[https://github.com/smart-data-models/data-models/tree/master/context/ontologies\\_files](https://github.com/smart-data-models/data-models/tree/master/context/ontologies_files)

4. Configuration file:

```
{ "etsi": "https://raw.githubusercontent.com/smart-data-models/data-models/master/context/ontologies_files/ngsi-ld-core-context.jsonld#1.4",  
  "schema.org": "https://raw.githubusercontent.com/smart-data-models/data-models/master/context/ontologies_files/schema.org.json#1.0",  
  "iudx": "https://raw.githubusercontent.com/smart-data-models/data-models/master/context/ontologies_files/iudx.json#1.0",  
  "saref": "https://raw.githubusercontent.com/smart-data-models/data-models/master/context/ontologies_files/saref.json#3.1.1"  
}
```

Open APIs  
for Open  
Minds

Future

# Future. Other improvements

## Other minor improvements in the queue

- Automatic update to contributors when new versions / updates
- Video supporting the different services available
- Certificate to contributors (recognition)
- Visualization of data models structure
- Import metadata from CKAN to draft data models
- Improve conversion of examples' types
  - Will be able to use AI to answer?