DOME 4.0

Deliverable D3.5 - "Reference Data Connector and B2B data connectors"

Responsible Partner:	UCL	2023.05.31
Contributor(s):	Adham Hashibon, Mike Wang (UCL)	2023.03.20
Reviewer(s):	Kostas Sipsas (INTRA), Amit Bhave (CMCL)	2023.05.30
Coordinator:	CMCL Innovations	2023.05.31
Dissemination Level:	Public	
Due Date:	M30	
Submission Date:	31.05.2023	

Project Profile

Programme	**Horizon 2020**
Call	**H2020-NMBP-TO-IND-2020-twostage**
Topic	**DT-NMBP-40-2020** **Creating an open marketplace for industrial data (RIA)**
Project number	**953163**
Acronym	**DOME 4.0**
Title	**Digital Open Marketplace Ecosystem 4.0**
Start Date	**December 1st, 2020**
Duration	**48 months**

This document is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953163. It is the property of the DOME 4.0 consortium and do not necessarily reflect the views of the European Commission.

Document History

Version	Date	Author	Remarks
V0.1	20.3.2023	Mike Wang	Initial Version
V0.2	18.5.2023	Adham Hashibon	Final draft
V1	31.05.2023	Technical coordination team	Finalising document

Executive Summary

Reference connector service is vital for DOME 4.0 platform to extend its reach to other external platforms and enrich the users' search and discover ability. In addition, it is also a key to fulfill the requirements of connecting DOME 4.0 with various showcase partners. In this development, a docker-container based service has been built that enables the functions of i) automatic generation of query API's based on the search strings and keywords from the frontend service and the common trends among various platforms. ii) Automatic extraction of the required DOME 4.0 standard metadata from the data set ontology from the returned external data records. iii) communicating with the frontend to enable any "get" requests for the selected databases. The service itself is based on a template defined via use of Abstract Base Classes (ABC) available in python and this template would allow the future integration of additional other platform API's and make them available for searching on DOME 4.0. Currently, the external databases that can be queried using the reference connector template are:

- Materials Project
- Crystallography
- Theoretical crystallography open database
- The open quantum materials
- Novel Materials Discovery NOWAD
- Chemeo
- PUBCHEM

Table of Contents

1. Introduction

1.1 Reference Connector

The reference connector template is an abstract base class (ABC) based python class that is embedded with a wide range of functions to help with converting search strings into the suitable format, generating APIs based on the platform names, the final APIs based on the platform name and search string, extracting the metadata from the returned results with respect to the keywords identified in the data set ontology. This service is built compatible with DOME 4.0 and has various API's that would communicate with various parts of the available services such as broker and the frontend on DOME 4.0. The development of functions to generate API's and extract metadata is based on the consideration of data platforms given below;

- Materials Project
- Crystallography
- Theoretical crystallography open database
- The open quantum materials
- Novel Materials Discovery NOWAD
- Chemeo
- PUBCHEM

This serves as a foundation on what functions the reference connector should contain and the current running service demonstrates what platforms we can connect to so far. However, this shall be further developed to fit with any future goals and new platforms with different API and data models. The code repository for this service is given in the GitHub page: https://github.com/DOME-4-0/Connector-service.

Data Connectors are the enablers of semantic data exchange on the DOME 4.0 platform. Based on the design in D1.3 we have in this task implemented a reference Data Connector as an orchestrated container service using docker-compose. It is providing the common, re-usable functionalities needed for platform interfacing, semantic API, administration/configuration web front-end and automation API. This is mainly facilitated currently by the extensive use of ABC in the implementation. A re-usable repository of adapters that can read/write data from common data sources (including file stores, SQL databases, document/object stores as well as other repositories) will be created as needed based on this template, simplifying future development of data connectors. The reference data connector and the repository of adapters support and strengthen the onboarding activities in T1.5 as they provide means to rapidly on board (add) new data consumers and providers because of the developed new connectors. This task has been released first as internal deliverable at M12 and continuedly improved and developed towards the final release at M30 (D3.5). However, following the nature of an open-source project we expect improvements and bug fixed, and especially adaptation to other developments within the platform in response to new requirements rising from new platforms will be conducted.

The list of functions/methods that are covered inside of the reference connector scripts are given below along with their discussions:

Function 1: search_api(platform_name):

This function allows to pass through a platform name and generate the respective API based on the given name. The platform names are passed down from the broker which was extracted from the ecosystem ontology developed in task 3.2. This function includes the template formats of API for various OPTIMADE supported platforms. Once the platform names are passed down as a parameter, this will automatically select the format of the query API for the selected platform and return the base API. The API generation service covers most of the OPTIMADE supported platforms such as the materials project, theoretical crystallography open database and other platforms such as chemeo and pubchem. Other platforms will be added as the need for the use cases arises. In general, further work is needed to create an easy plug and play approach for adding new systems, however, this requires colaboration of the platform owners. Hence currently we are limited to this initial set mentioned.

Function 2. optimade_search_string_split (search_string):

This function allows the split of the search strings passed from the frontend into the optimade supported format. The returned results are the final search strings that can be used either as a parameter to be passed down using the API generated or be integrated into the API itself.

Function 3. results_request(self):

An abstract method that allows programmer to flexibly request the results, based on the generated APIs from above, and retrieve the results back from the platform.

Function 4. dome_results_template(self):

An abstract method function converts the returned results from those platforms into DOME 4.0 compatible format based on the CUDS data strucrtures (se D3.6). This includes the information about various metadata and the actual data from the quired APIs. An example of the scripts is included in the DomeConnector child class that demonstrates the execution of the API generation function, posting the get request to retrieve the initial raw data and metadata extraction method (extract metadata). Since this is an abstract method, it allows the future developer to write their own compatible scripts to for new databases.

In the following sections, functions are given to retrieve the respective keywords according to the data set ontology. These functions are all written in the abstract method class which again allows the developer to write their own scripts to extract the respective information when the returned data model is different to what has been considered so far.

Function 5. hastitledata(self):

This returns the title of the data and is presented in the string format.

Function 6. dataset(self):

This returns the name of the dataset that has been retrieved.

Function 7. hasKeywords(self):

This function returns the keywords that are used to described the returned data.

Function 8. hasdatacreator(self):

This function returns the data creator for the data that has been retrieved.

Function 9. hasdatapublisher(self):

This function returns the data publisher for the data that has been retrieved.

Function 10. hasissuedate(self):

This function returns the issue date for the data that has been retrieved.

Function 11. haslicense(self):

This function returns the license of the data that has been retrieved.

Function 12. hasurl(self):

This function returns the url for the data that has been retrieved.

This metadata information is displayed to the users when they search for a string. Upon approving the request, the actual data will be displayed by DOME 4.0.

1.1.1 Available platforms

In order to develop the reference connector template, investigations were made into a few available data platforms. This includes the optimade supported platforms and a few showcase platforms such as chemeo and pubchem. Optimade formalizes the the APIs for various platforms and allows users to query various onboarded platforms using similar API format and obtain the returned data in a similar data model.

1.1.2 Keywords extraction

The keywords extraction and mapping from the returned data results from the optimade platforms is investigated based on the returned results. The codes are developed for executing such purposes are given in the main repository. The aim of this mapping is to convert the often syntactic, third party metadata schema into the one based on DOME 4.0 ontology. This is done currently on a case by case basis, and future work will rely on using the new semantic backend knowledge service based on the SimPhoNy OSP.

2. Conclusions / Next steps

A simple and efficient connector abstract base class (ABC) templates have been developed with support for many emerging data platforms. Development will continue in the form of maintenance, bug fixes, adding more features as needed, and refactoring the code into further abstraction layers as new platforms are added.

3. Lessons learnt

The connector service is not trivial in that it requires three distinct and highly specific levels:

1. Understanding the third-party API and consequently creating the API generators and integrating with DOME 4.0
2. Understanding (often reverse engineering) the format of the third party platform and creating the mapping to the DOME 4.0 ontology and
3. Catering for the display of this data on DOME 4.0 and linking the user to the actual platform.

Such a process can be simplified if indeed a common API and data standard existed, which is in essence one of the goals of this project.

4. Deviations from Annex 1

There are no deviations from Annex 1.

5. Acknowledgement

The author(s) would like to thank the partners in the project for their valuable comments on previous drafts and for performing the review.

Project partners:

#	Type	Partner	Partner full name
1	SME	CMCL	Computational Modelling Cambridge Limited
2	Research	FHG	Fraunhofer Gesellschaft zur Förderung der Angewandten Forschung E.V.
3	Research	INTRA	Intrasoft International SA
4	University	UNIBO	Alma Mater Studiorum – Universita di Bologna
5	University	EPFL	Ecole Polytechnique Federale de Lausanne
6	Research	UKRI	United Kingdom Research and Innovation
7	Large Industry	SISW	Siemens Industry Software NV
8	Large Industry	BOSCH	Robert Bosch GmbH
9	SME	UNR	Uniresearch B.V.
10	Research	SINTEF	SINTEF AS
11	SME	CNT	Cambridge Nanomaterials Technology LTD
12	University	UCL	University College London

This document is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953163. It is the property of the DOME 4.0 consortium and do not necessarily reflect the views of the European Commission.