Grant Agreement N° 872592

PLATOON
Digital platform and analytic tools for energy

Deliverable D3.5
Marketplace demonstrator and report

Contractual delivery date:
M24

Actual delivery date:
31st December 2021

Responsible partner:
P4: IAIS, Germany

Project Title	PLATOON – Digital platform and analytic tools for energy
Deliverable number	D3.5
Deliverable title	Marketplace demonstrator and report
Author(s):	Tasneem Tazeen Rashid (IAIS) Tejas Morbagal Harish (IAIS) Erik Maqueda (TECN) Adelaida Lejarazu (TECN)
Responsible Partner:	P4 – IAIS
Date:	31.12.2021
Nature	R
Distribution level (CO, PU):	PU
Work package number	WP3 – Data Governance, Security and Privacy
Work package leader	IAIS, Germany
Abstract:	This deliverable demonstrates PLATOON Marketplace and interactions among the Metadata Registry, the Connectors,

	the Clearing House, the Vocabulary Provider components, and the Marketplace User Interface.
Keyword List:	Marketplace, Metadata Registry, User interface, Clearing House, Vocabulary Provider, DAPS

The research leading to these results has received funding from the European Community's Horizon 2020 Work Programme (H2020) under grant agreement no 872592.

Editor(s):	Tasneem Tazeen Rashid (IAIS) Tejas Morbagal Harish (IAIS) Najmeh Mousavi Nejad (IAIS)
Contributor(s):	Erik Maqueda (TECN) Adelaida Lejarazu (TECN)
Reviewer(s):	Martino Maggio (ENG) Philippe Calvez (ENGIE) Erik Maqueda (TECN)
Approved by:	Martino Maggio (ENG) Philippe Calvez (ENGIE) Erik Maqueda (TECN)
Recommended/mandatory readers:	WP2-WP6, WP8, WP9, Partners and Task Leaders

Document Description

Document Revision History

Version	Date	Modifications Introduced	
		Modification Reason	Modified by
0.1	02.11.2021	Creation of Table of Content	Tasneem Tazeen Rashid (IAIS)
0.2	20.11.2021	Chapter 1 -4 and 6	Tasneem Tazeen Rashid (IAIS) Tejas Morbagal Harish (IAIS)
0.3	28.11.2021	Chapter 5: Vocabulary Provider	Erik Maqueda (TECN)
0.4	28.11.2021	Update Chapter 4: User Interface	Tasneem Tazeen Rashid (IAIS) Tejas Morbagal Harish (IAIS)
0.5	1.12.2021	Update Marketplace Architecture	Erik Maqueda (TECN) Tasneem Tazeen Rashid (IAIS)
0.6	13.12.2021	Adaptation of the internal review	Tasneem Tazeen Rashid (IAIS)
0.7	21.12.2021	Adaptation of the review from Erik	Tasneem Tazeen Rashid (IAIS)
0.8	30.12.2021	Final Review	Erik Maqueda (TECN) Philippe Calvez (ENGIE)

Table of Contents

Terms and abbreviations

API	Application Programming Interface
CO	Confidential
DAPS	Dynamic Attribute Provisioning Service
DAT	Dynamic Attribute Token
DC	Data Consumer
DER	Distinguished Encoding Rules
DP	Data Provider
GA	Grant Agreement
GUI	Graphical User Interface
HTTP	HyperText Transfer Protocol
IDS	International Data Spaces
IDSA	International Data Spaces Association
JSON	JavaScript Object Notation
JSON-LD	JavaScript Object Notation for Linked Data
JWT	JSON Web Token
OWL	Web Ontology Language
PID	Process ID
PU	Public
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
REST	REpresentational State Transfer
RSA	Rivest-Shamir-Adleman encryption
SHACL	Shapes Constraint Language
SSL	Secure Sockets Layer
UI	User Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WP	Work Package

List of Figures

List of Tables

Executive Summary

This document provides technical information about the PLATOON Marketplace Components (Deliverable 3.5). The PLATOON Marketplace is a place to search and discover data assets and services. The Provider publishes the metadata about the data assets and tools it is interested in offering in PLATOON Metadata Registry. The publishing of metadata happens through the PLATOON Connector. PLATOON Marketplace will offer a User Interface where all the metadata will be visualized in a user-friendly manner. By browsing through the User Interface, one can become interested in acquiring the data assets or services and starting a transaction process by establishing a Contract Agreement. PLATOON Clearing House mediates between the Provider and Consumer by ensuring the contractual obligations are met. PLATOON Vocabulary Provider helps to annotate and describe the datasets in the Marketplace, enhancing the interoperability. This document includes architectural and functional information about Metadata Registry, Clearing House, Marketplace User Interface, and Vocabulary Provider.

1 Introduction

This deliverable reports the design of the marketplace module that implements a common endpoint to access the data and energy services based on the International Data Spaces reference architecture. The core purpose of the International Data Spaces is to enable controlled exchange and sharing of data between organizations – regardless of the type of data. In many use cases of the International Data Spaces, this is some form of structured data (e.g., measurement data, product data, or logistics data). Also, other types of (streaming) data are supported. The IDS Connector allows data owners and data providers to exchange and share their data with other participants in the IDS ecosystem, as shown in Figure 1, while data sovereignty is ensured at any time. For PLATOON, the Broker and AppStore are merged into one component called the Metadata Registry that contains the metadata for both datasets and apps/data analytics tools. The developed metadata registry supports both approaches IDS Apps (that are implemented directly into the IDS connectors) and the PLATOON Data Analytics defined in WP4 (that can be implemented on-premise or as a service). In addition, as part of this task, a Vocabulary Provider has been developed which manages the PLATOON Common Data Models and other standard ontologies to enable to understand the datasets/apps metadata. The rest of the components from the IDS ecosystem shown in Figure 1 have been implemented in PLATOON. in particular an Identity Provider and a Clearing House developed by Fraunhofer AISEC.

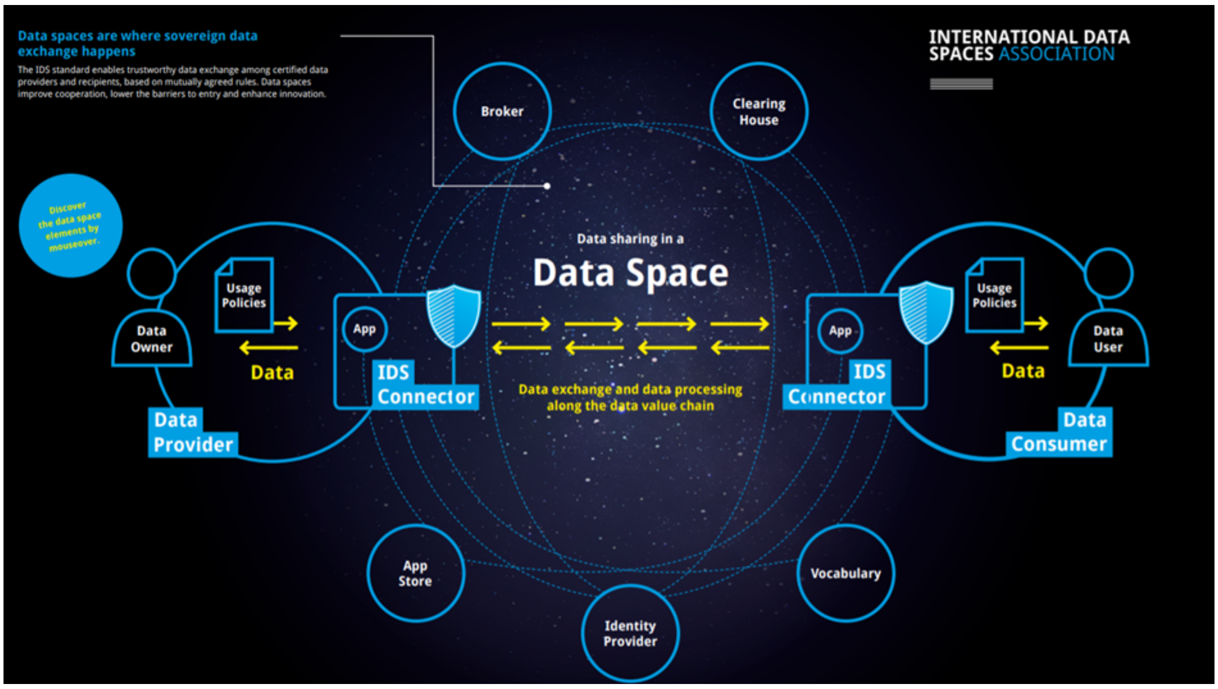

Data spaces are where sovereign data exchange happens
The IDS standard enables trustworthy data exchange among certified data providers and recipients, based on mutually agreed rules. Data spaces improve cooperation, lower the barriers to entry and enhance innovation.
INTERNATIONAL DATA SPACES ASSOCIATION
Discover the data space elements by mouseover.
Broker
Clearing House
Data sharing in a
Data Space
Data Owner
Usage Policies
Data
App
IDS Connector
Data Provider
Data exchange and data processing along the data value chain
App
IDS Connector
Usage Policies
Data User
Data
Data Consumer
App Store
Identity Provider
Vocabulary

Figure 1: IDS Ecosystem and components [1]

1.1 Marketplace Overview

The Marketplace is one common endpoint to access the data and energy services provided by a Marketplace participant to set the foundations for a functional Energy data Marketplace, open to all relevant stakeholders. This output layer of PLATOON allows the data flow to the stakeholders, offering different services to the rest of the community: data services (share of raw and process data) and energy services (energy-specific data analytics tools). The Marketplace is compliant with the specification defined by the IDS reference architecture. It will provide functionalities required by each use case based on their specific requirement.

Marketplace offers the opportunity to explore data exchange for innovative use-cases without exposing providers to risks of unauthorized data access.

PLATOON Marketplace comprehends the following IDS components:

- **Metadata Registry:** The Metadata Registry will receive and store metadata describing data assets (datasets or services) made available by providers.
- **User Interface:** The User Interface will visualize the metadata registered in the Metadata Registry user-friendly.
- **Clearing House:** The Clearing House mediates between the Data Provider and Data Consumer by logging the transactions structured into processes to ensure the contractual obligation are preserved by both parties.
- **Vocabulary Provider:** The Vocabulary Provider manages and offers vocabularies (ontologies, reference data models, metadata elements) that can be used to annotate and describe datasets and apps/services. In other words, it provides domain-specific vocabularies and the IDS Information Model[1].
- **DAPS:** Dynamic Attribute Provisioning Service (DAPS) in a given IDS ecosystem enables the enrichment of organizations' identities and Connectors with additional attributes. In Marketplace, all other components must incorporate DAPS[2]. DAPS issues Dynamic Attribute Token (DAT) to complete the verification process during communication among IDS components.

1.2 Marketplace Architecture

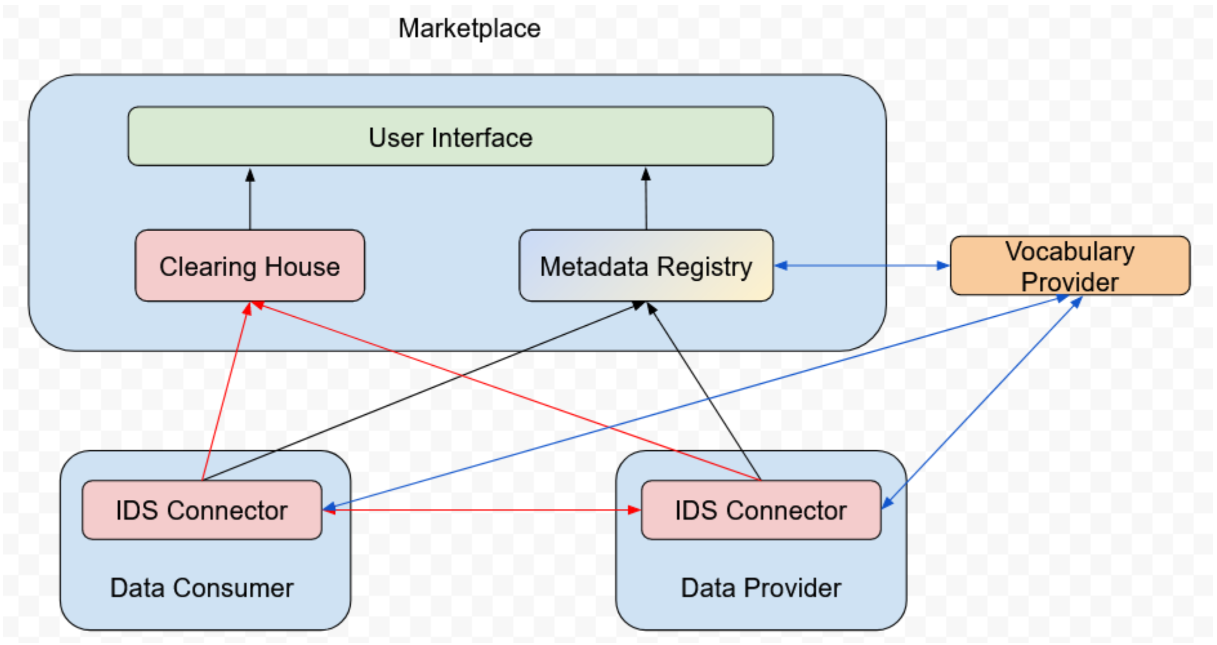


Figure 2: Marketplace Architecture

Figure 2 shows the architecture of the Marketplace. Presume, Company A has acquired some rare datasets and used them to invent a new technology to store electricity efficiently. Now, Company A wants to monetize the datasets with other companies without losing control over

[1] https://github.com/International-Data-Spaces-Association/InformationModel
[2] https://daps.aisec.fraunhofer.de/v2/token

the data. Company A will have to craft the metadata to represent the datasets as a Resource. Company A should first install a Connector on their IT environment and then register their Connector to the Metadata Registry. This registration includes information like who owns the Resource, where to download it, which contracts apply etc. User Interface (UI) reflects this information in a user-friendly manner. Now Company B, who already has registered their Connector to the Metadata Registry, browses through the UI and decides to acquire the dataset from Company A. In this scenario, Company A becomes Data Provider, and Company B becomes the Data Consumer in Figure 2. The negotiations between the two parties will be directly between the respective Connectors of Company A and Company B. Both parties log their settled Contract Agreement to the Clearing House during the data exchange process between the Data Provider and Data Connector. Vocabulary Provider is a special connector that exchanges metadata instead of exchanging data. So it will register itself in the Metadata Registry. Company A and Company B can send SPARQL queries from their connectors to Vocabulary Provider through IDS messages and can use the Vocabulary Provider to annotate and understand the corresponding data assets (datasets/services).

1.3 Information Model and Marketplace

The Information Model[3], an RDFS/OWL-ontology, is an essential agreement shared by the participants and components of the IDS, facilitating compatibility and interoperability. The primary purpose of this formal model is to enable (semi-)automated exchange of digital resources within a trusted ecosystem of distributed parties while preserving the data sovereignty of Data Owners. The Information Model, therefore, supports the description, publication, and identification of data products and reusable data processing software (both referred to hereinafter as "Digital Resources," or simply "Resources") in the PLATOON Marketplace. Once the relevant Resources are identified, they can be exchanged and consumed via semantically annotated, easily discoverable services. Apart from those core commodities, the Information Model describes essential constituents of the International Data Spaces, inevitably PLATOON Marketplace, its participants, its infrastructure components, and its processes. The ontology and its documentation are published at https://w3id.org/idsa/core.

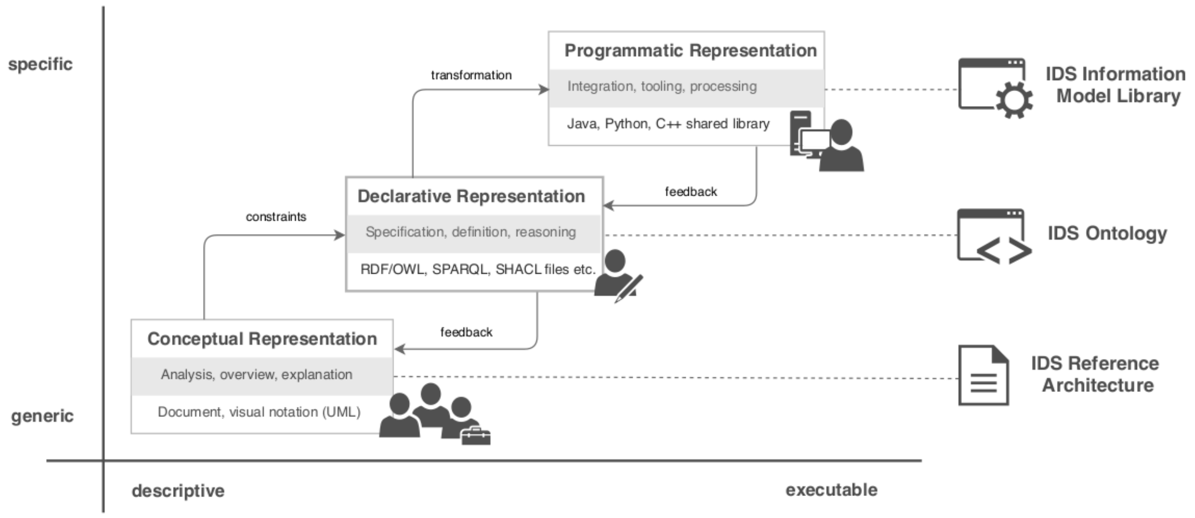

specific
generic
descriptive
executable
Programmatic Representation
Integration, tooling, processing
Java, Python, C++ shared library
transformation
feedback
Declarative Representation
Specification, definition, reasoning
RDF/OWL, SPARQL, SHACL files etc.
constraints
feedback
Conceptual Representation
Analysis, overview, explanation
Document, visual notation (UML)
IDS Information Model Library
IDS Ontology
IDS Reference Architecture

Figure 3: Representation of the Information Model [2]

All the IDS components need to follow this Information Model for IDS messaging and communication. Participants of the IDS use the RDF vocabulary provided by the Information Model as their common language within the IDS. To ensure the correct usage and understanding

[3] https://github.com/International-Data-Spaces-Association/InformationModel

of the vocabulary, validation structures are provided in the W3C Shapes Constraint Language (SHACL). These so-called SHACL shape graphs can be used to validate self-generated RDF statements against the Information Model and check if:

- a Connector's self-description is valid;
- a Resource is described using the correct metadata terms;
- an HTTP multipart message exchanged between IDS components provides the necessary information.

The SHACL shapes can be found in the testing subdirectory of the IDS Information Model[4].

1.3.1 Metadata Representation in Marketplace

In the use-case scenario described in Section 1.2, the Data Provider or Company A needs to design the Resource to represent their metadata in the Marketplace. The scope of Resources is defined in the Resource class[5] in the information model, and the Resource Shape class[6] maintains the SHACL validation. This Resource can be extended by adding more properties depending on the nature of the data. For example, adding title, language, the publication date of the metadata. The Data Provider must look at the file, provided here[7], that describes the Json terms to RDF terms of the IDS information model while creating the Resource. To get a more general idea of the horizons of creating Resources, we recommend checking these examples[8][9][10]. An example Resource will be described in Section 2.1. The metadata represented in the Marketplace will be in line with the metadata defined in tasks T2.4 and T5.3.

[4] https://github.com/International-Data-Spaces-Association/InformationModel/tree/develop/testing
[5] https://github.com/International-Data-Spaces-Association/InformationModel/blob/develop/model/content/Resource.ttl
[6] https://github.com/International-Data-Spaces-Association/InformationModel/blob/develop/testing/taxonomies/ResourceShape.ttl
[7] https://github.com/PLATOONProject/Metadata-Registry/blob/main/broker-core/src/main/resources/context.json
[8] https://github.com/International-Data-Spaces-Association/InformationModel/tree/develop/examples
[9] https://github.com/International-Data-Spaces-Association/InformationModel/blob/a611f476a9b68d0a7cd5f9e28fac38c413f13749/examples/TEXT_RESOURCE.jsonld
[10] https://github.com/International-Data-Spaces-Association/InformationModel/blob/a611f476a9b68d0a7cd5f9e28fac38c413f13749/examples/TEXT_RESOURCE.ttl

2 Metadata Registry

The Metadata Registry is a registry for datasets and apps/data analytics tools derived from the International Data Spaces (IDS) Metadata Broker. In contrast to the general IDS Metadata Broker, the Metadata Registry has been tailored to provide the main functionalities of metadata handling for Connectors, Data Resources, Apps, and querying for the metadata. This PLATOON component has adapted the functionality to store metadata of the App hosted IDS App Store. Combining the functionalities is that there are challenges regarding transferring a complex App to the App Store or exchanging Apps using Connector. Thus, only metadata of the App is registered in the Metadata Registry. The important point is that the Broker does not serve the datasets themselves: querying is performed on metadata only. The feature for handling App messages sent by the Connectors is added to the previous version of the Broker[11] (which was explained in Deliverable D3.3) to make it a Metadata Registry. The metadata of the Connectors, Data Resources, and App Resources are stored in an RDF Database as triples.

The Metadata Registry is the main component of Marketplace. It can be used to register, update, or unregister the Connector or resource (Resource or AppResource) metadata. Self-description of the Metadata Registry is also available in the Marketplace, and one can query for any information provided in it. The following list shows the main functions of the Metadata Registry:

- **Description Request**: Gets the self-description of the Metadata Registry.
- **Register/Update Connector:** Registers the Connector if it doesn't exist in the Metadata Registry or update the Connector if it exists. Users can also include Resources (metadata of the data) while registering a Connector.
- **Unregister Connector:** Unregisters the Connector from the Metadata Registry.
- **Update Resource:** Updates the information of a particular Resource of a Connector.
- **Unregister Resource:** Removes a particular Resource from a Connector.
- **Register App:** Registers metadata of an App in the Resource Catalog of the Connector.
- **Unregister App:** Unregisters metadata of an App from the Resource Catalog of the Connector.
- **Query:** Queries the SPARQL triples in Fuseki triple store.

Note that all data in the Metadata Registry is metadata only. The register, update and unregister of connectors, Resource, and AppResource mentioned above are also related to metadata.

[11] https://github.com/PLATOONProject/open-source-broker

2.1 Interaction with Metadata Registry

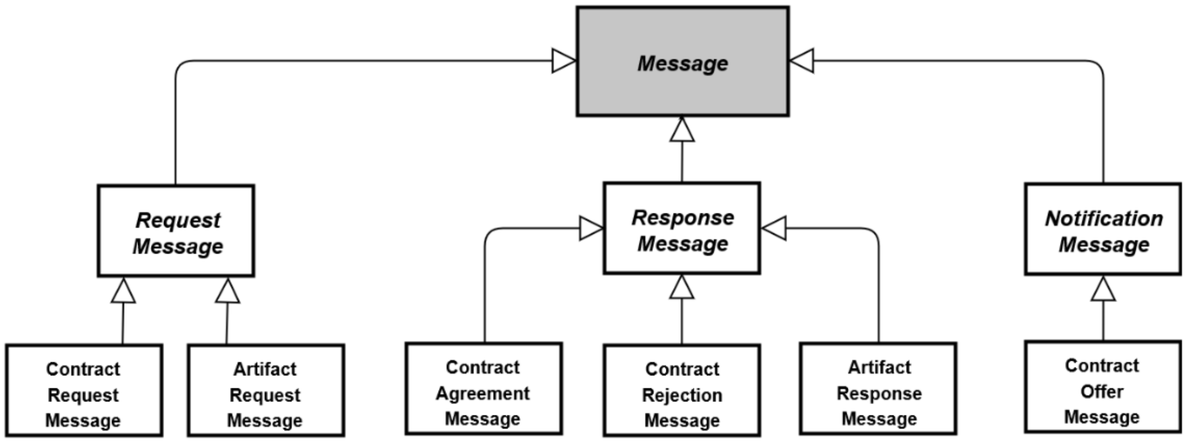
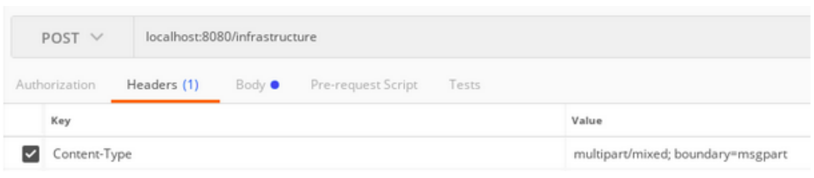


Figure 4: IDS Message taxonomy [2]

The Metadata Registry accepts and sends messages according to the IDS Information Model explained in Section 1.3. These messages are called IDS messages, and Figure 4 illustrates an excerpt of the message taxonomy. Request-response interactions between Marketplace components are reflected by the dedicated subclass of the RequestMessage and the RequestResponse type. The NotificationMessage subclasses reflect Event-like notifications. All messages in the Metadata Registry are JSON-LD formatted as HTTP Multipart messages. Table 1 enlists the possible response codes of the IDS messages. The multipart endpoint of Metadata Registry is "/infrastructure". If the Metadata Registry is running, an HTTP POST request can be sent to interact with it. The header should be "Content-Type," and the value should be "multipart/mixed; boundary=msgpart" as shown in Figure 5. Here the boundary value can be changed to any value, but it must be the same boundary in the request body.



Key	Value
☑ Content-Type	multipart/mixed; boundary=msgpart

Figure 5: Header of HTTP API request

The endpoint for the Fuseki server of the Metadata Registry is "/fuseki". This endpoint is important for the connection setup of the UI with the Metadata Registry described in Section 4.2.

In the use-case scenario mentioned in Section 1.2, Company A and Company B will use their corresponding Connectors to interact with the Metadata Registry. Subsections of Section 2.1 will show how to interact with the Metadata Registry, and they will cover all main functionalities provided by it. Note that the localhost environment acts as the IDS connector to interact with the Metadata Registry.

Table 1: Response codes of IDS messages

Status	Meaning	Description	Schema
200	OK	Successful Operation	ResultMessage
201	CREATED	Created	MessageProcessedNotificationMessage
400	Bad Request	Bad Request	RejectionMessage

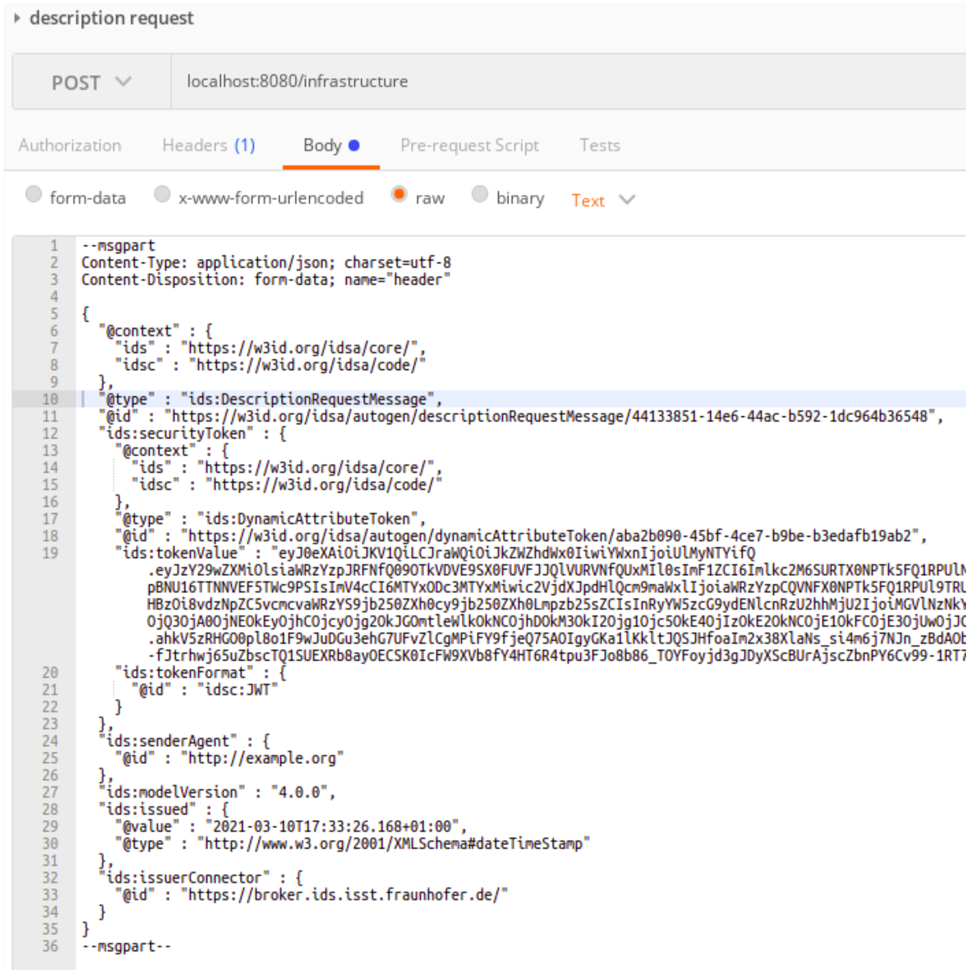
401	Unauthorized	Unauthorized	RejectionMessage
405	Method Not Allowed	Method Not Allowed	RejectionMessage
500	Internal Server Error	Internal Error	RejectionMessage

The Postman[12] tool is used to test these interactions with the Metadata Registry. The Postman collections provided here[13] can be imported to the tool. As mentioned earlier, it is a must to incorporate DAPS in each of the components; for successful communication with the Metadata Registry, a proper DAT is a must to include in the API requests. The following sections contain sample snippets of all the requests and their corresponding responses[14].

[12] https://www.postman.com/
[13] https://www.getpostman.com/collections/597f48eb7bfca0604623
[14] Please import the Postman collection to test the APIs by yourself

2.1.1 Description Request

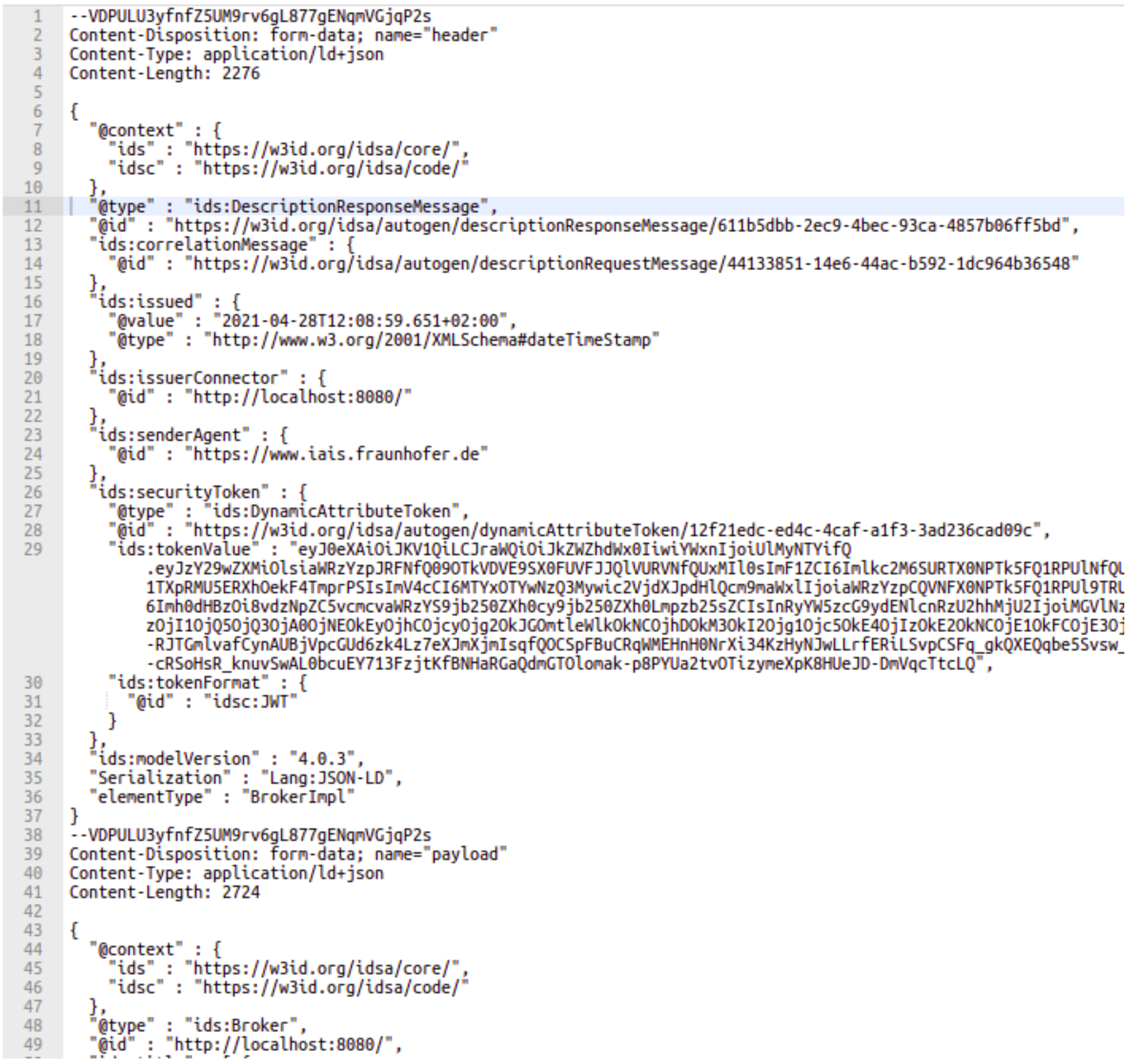
The multipart message header should be DescriptionRequestMessage, and the payload should be empty.



```
--msgpart
Content-Type: application/json; charset=utf-8
Content-Disposition: form-data; name="header"

{
  "@context" : {
    "ids" : "https://w3id.org/idsa/core/",
    "idsc" : "https://w3id.org/idsa/code/"
  },
  "@type" : "ids:DescriptionRequestMessage",
  "@id" : "https://w3id.org/idsa/autogen/descriptionRequestMessage/44133851-14e6-44ac-b592-1dc964b36548",
  "ids:securityToken" : {
    "@context" : {
      "ids" : "https://w3id.org/idsa/core/",
      "idsc" : "https://w3id.org/idsa/code/"
    },
    "@type" : "ids:DynamicAttributeToken",
    "@id" : "https://w3id.org/idsa/autogen/dynamicAttributeToken/aba2b090-45bf-4ce7-b9be-b3edafb19ab2",
    "ids:tokenValue" : "eyJ0eXAiOiJKV1QiLCJraWQiOiJkZWZhdWx0IiwiYWxnIjoiUlMyNTYifQ
        .eyJzY29wZXMiOlsiaWRzYzpJRFNfQ09OTkVDVE9SX0FUVFJJQlVURVNfQUxMIl0sImF1ZCI6Imlkc2M6SURTX0NPTk5FQ1RPUl
        pBNU16TTNNVEF5TWc9PSIsImV4cCI6MTYxODc3MTYxMiwic2VjdXJpdHlQcm9maWxlIjoiaWRzYzpCQVNFX0NPTk5FQ1RPUl9TR
        HBzOi8vdzNpZC5vcmcvaWRzYS9jb250ZXh0cy9jb250ZXh0Lmpzb25sZCIsInRyYW5zcG9ydENlcnRzU2hhMjU2IjoiMGVlNzNk
        OjQ3OjA0OjNEOkEyOjhCOjcyOjg2OkJGOmtleWlkOkNCOjhDOkM3OkI2Ojg1Ojc5OkE4OjIzOkE2OkNCOjE1OkFCOjE3OjUwOjJ
        .ahkV5zRHGO0pl8o1F9wJuDGu3ehG7UFvZlCgMPiFY9fjeQ75AOIgyGKa1lKkltJQSJHfoaIm2x38XlaNs_si4m6j7NJn_zBdAO
        -fJtrhwj65uZbscTQ1SUEXRb8ayOECSK0IcFW9XVb8fY4HT6R4tpu3FJo8b86_TOYFoyjd3gJDyXScBUrAjscZbnPY6Cv99-1RT
    "ids:tokenFormat" : {
      "@id" : "idsc:JWT"
    }
  },
  "ids:senderAgent" : {
    "@id" : "http://example.org"
  },
  "ids:modelVersion" : "4.0.0",
  "ids:issued" : {
    "@value" : "2021-03-10T17:33:26.168+01:00",
    "@type" : "http://www.w3.org/2001/XMLSchema#dateTimeStamp"
  },
  "ids:issuerConnector" : {
    "@id" : "https://broker.ids.isst.fraunhofer.de/"
  }
}
--msgpart--
```

Figure 6: Sample description request message of Metadata Registry

The response should be a DescriptionResponseMessage with the self-description of the Metadata Registry in the payload, as shown in Figure 7.

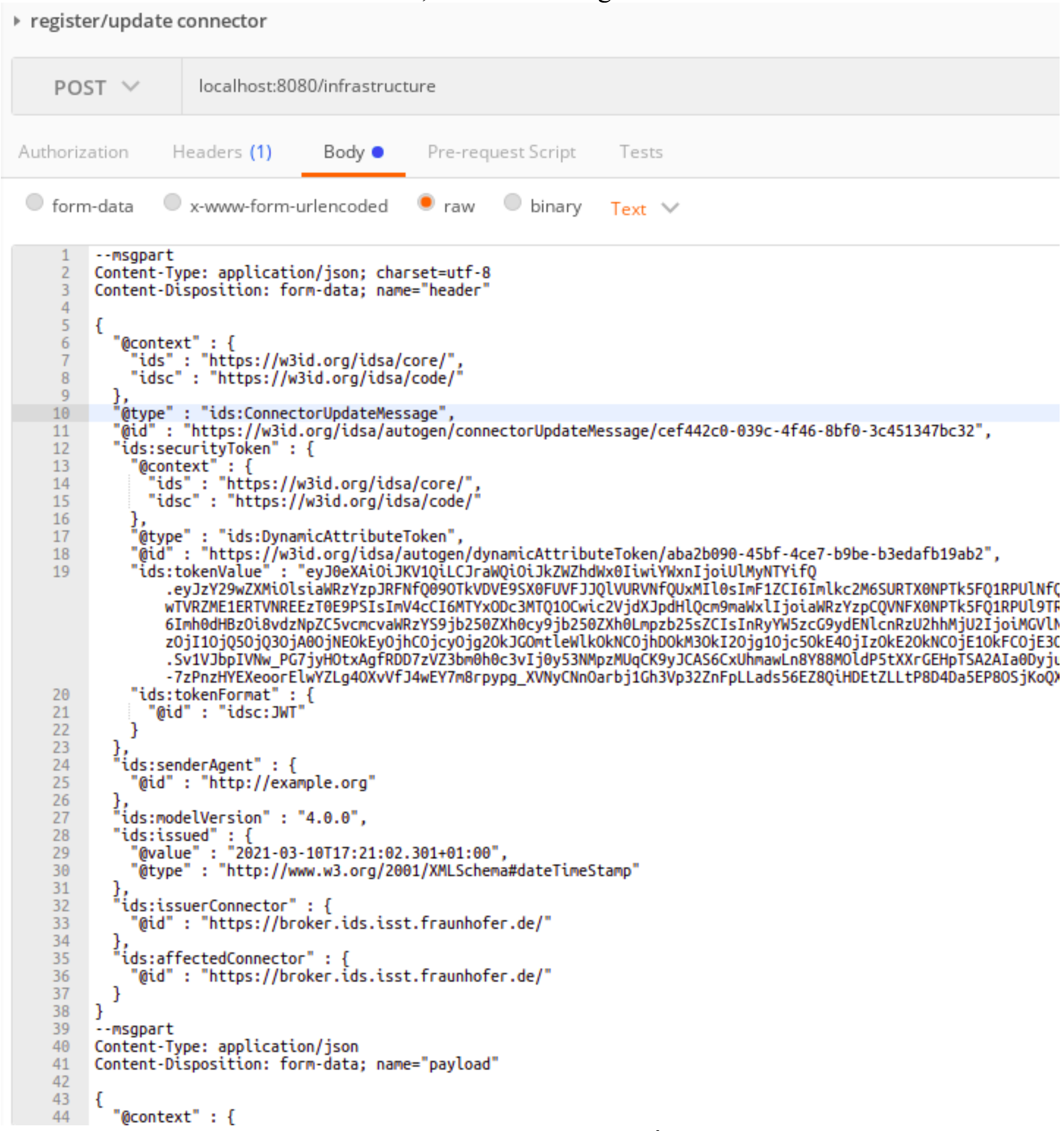


```
--VDPULU3yfnfZ5UM9rv6gL877gENqmVGjqP2s
Content-Disposition: form-data; name="header"
Content-Type: application/ld+json
Content-Length: 2276

{
  "@context" : {
    "ids" : "https://w3id.org/idsa/core/",
    "idsc" : "https://w3id.org/idsa/code/"
  },
  "@type" : "ids:DescriptionResponseMessage",
  "@id" : "https://w3id.org/idsa/autogen/descriptionResponseMessage/611b5dbb-2ec9-4bec-93ca-4857b06ff5bd",
  "ids:correlationMessage" : {
    "@id" : "https://w3id.org/idsa/autogen/descriptionRequestMessage/44133851-14e6-44ac-b592-1dc964b36548"
  },
  "ids:issued" : {
    "@value" : "2021-04-28T12:08:59.651+02:00",
    "@type" : "http://www.w3.org/2001/XMLSchema#dateTimeStamp"
  },
  "ids:issuerConnector" : {
    "@id" : "http://localhost:8080/"
  },
  "ids:senderAgent" : {
    "@id" : "https://www.iais.fraunhofer.de"
  },
  "ids:securityToken" : {
    "@type" : "ids:DynamicAttributeToken",
    "@id" : "https://w3id.org/idsa/autogen/dynamicAttributeToken/12f21edc-ed4c-4caf-a1f3-3ad236cad09c",
    "ids:tokenValue" : "eyJ0eXAiOiJKV1QiLCJraWQiOiJkZWZhdWx0IiwiYWxnIjoiUlMyNTYifQ
        .eyJzY29wZXMiOlsiaWRzYzpJRFNfQ09OTkVDVE9SX0FUVFJJQlVURVNfQUxMIl0sImF1ZCI6Imlkc2M6SURTX0NPTk5FQ1RPUlNfQL
        1TXpRMU5ERXhOekF4TmprPSIsImV4cCI6MTYxOTYwNzQ3Mywic2VjdXJpdHlQcm9maWxlIjoiaWRzYzpCQVNFX0NPTk5FQ1RPUl9TRL
        6Imh0dHBzOi8vdzNpZC5vcmcvaWRzYS9jb250ZXh0cy9jb250ZXh0Lmpzb25sZCIsInRyYW5zcG9ydENlcnRzU2hhMjU2IjoiMGVlNz
        zOjI1OjQ5OjQ3OjA0OjNEOkEyOjhCOjcyOjg2OkJGOmtleWlkOkNCOjhDOkM3OkI2Ojg1Ojc5OkE4OjIzOkE2OkNCOjE1OkFCOjE3Oj
        -RJTGmlvafCynAUBjVpcGUd6zk4Lz7eXJmXjmIsqfQOCSpFBuCRqWMEHnH0NrXi34KzHyNJwLLrfERiLSvpCSFq_gkQXEQqbe5Svsw_
        -cRSoHsR_knuvSwAL0bcuEY713FzjtKfBNHaRGaQdmGTOlomak-p8PYUa2tvOTizymeXpK8HUeJD-DmVqcTtcLQ",
    "ids:tokenFormat" : {
      "@id" : "idsc:JWT"
    }
  },
  "ids:modelVersion" : "4.0.3",
  "Serialization" : "Lang:JSON-LD",
  "elementType" : "BrokerImpl"
}
--VDPULU3yfnfZ5UM9rv6gL877gENqmVGjqP2s
Content-Disposition: form-data; name="payload"
Content-Type: application/ld+json
Content-Length: 2724

{
  "@context" : {
    "ids" : "https://w3id.org/idsa/core/",
    "idsc" : "https://w3id.org/idsa/code/"
  },
  "@type" : "ids:Broker",
  "@id" : "http://localhost:8080/",
```

Figure 7: Sample response message of self-description of the Metadata Registry

2.1.2 Register/Update Connector

The multipart message header should be ConnectorUpdateMessage, and the payload should be JSON-LD format connector metadata, as shown in Figure 8.

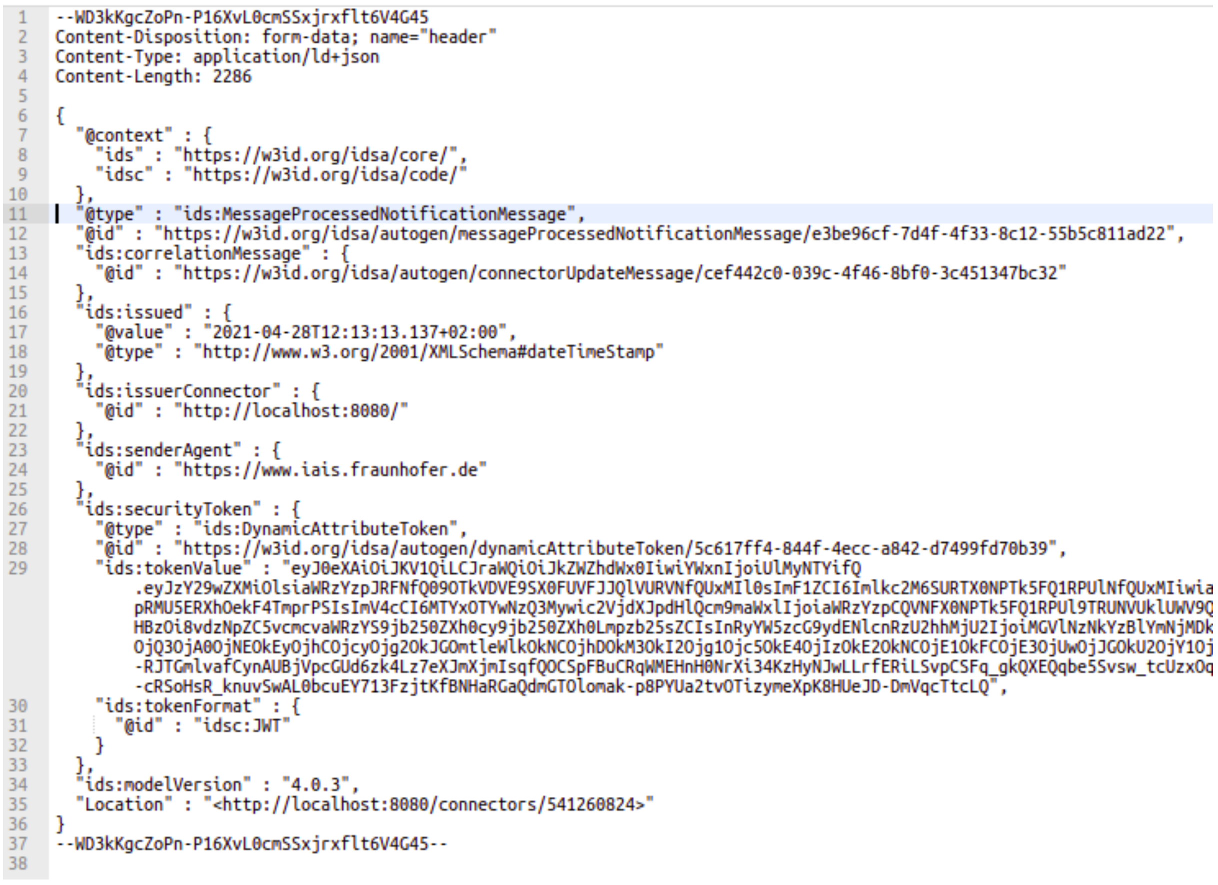


localhost:8080/infrastructure

```
--msgpart
Content-Type: application/json; charset=utf-8
Content-Disposition: form-data; name="header"

{
  "@context" : {
    "ids" : "https://w3id.org/idsa/core/",
    "idsc" : "https://w3id.org/idsa/code/"
  },
  "@type" : "ids:ConnectorUpdateMessage",
  "@id" : "https://w3id.org/idsa/autogen/connectorUpdateMessage/cef442c0-039c-4f46-8bf0-3c451347bc32",
  "ids:securityToken" : {
    "@context" : {
      "ids" : "https://w3id.org/idsa/core/",
      "idsc" : "https://w3id.org/idsa/code/"
    },
    "@type" : "ids:DynamicAttributeToken",
    "@id" : "https://w3id.org/idsa/autogen/dynamicAttributeToken/aba2b090-45bf-4ce7-b9be-b3edafb19ab2",
    "ids:tokenValue" : "eyJ0eXAiOiJKV1QiLCJraWQiOiJkZWZhdWx0IiwiYWxnIjoiUlMyNTYifQ
        .eyJzY29wZXMiOlsiaWRzYzpJRFNfQ09OTkVDVE9SX0FUVFJJQlVURVNfQUxMIl0sImF1ZCI6Imlkc2M6SURTX0NPTk5FQ1RPUlNf
        wTVRZME1ERTVNREEzT0E9PSIsImV4cCI6MTYxODc3MTQ1OCwic2VjdXJpdHlQcm9maWxlIjoiaWRzYzpCQVNFX0NPTk5FQ1RPUl9T
        6Imh0dHBzOi8vdzNpZC5vcmcvaWRzYS9jb250ZXh0cy9jb250ZXh0Lmpzb25sZCIsInRyYW5zcG9ydENlcnRzU2hhMjU2IjoiMGVl
        zOjI1OjQ5OjQ3OjA0OjNEOkEyOjhCOjcyOjg2OkJGOmtleWlkOkNCOjhDOkM3OkI2Ojg1Ojc5OkE4OjIzOkE2OkNCOjE1OkFCOjE3
        .Sv1VJbpIVNw_PG7jyHOtxAgfRDD7zVZ3bm0h0c3vIj0y53NMpzMUqCK9yJCAS6CxUhmawLn8Y88MOldP5tXXrGEHpTSA2AIa0Dyj
        -7zPnzHYEXeoorElwYZLg4OXvVfJ4wEY7m8rpypg_XVNyCNnOarbj1Gh3Vp32ZnFpLLads56EZ8QiHDEtZLLtP8D4Da5EP8OSjKoQ
    "ids:tokenFormat" : {
      "@id" : "idsc:JWT"
    }
  },
  "ids:senderAgent" : {
    "@id" : "http://example.org"
  },
  "ids:modelVersion" : "4.0.0",
  "ids:issued" : {
    "@value" : "2021-03-10T17:21:02.301+01:00",
    "@type" : "http://www.w3.org/2001/XMLSchema#dateTimeStamp"
  },
  "ids:issuerConnector" : {
    "@id" : "https://broker.ids.isst.fraunhofer.de/"
  },
  "ids:affectedConnector" : {
    "@id" : "https://broker.ids.isst.fraunhofer.de/"
  }
}
--msgpart
Content-Type: application/json
Content-Disposition: form-data; name="payload"

{
  "@context" : {
```

Figure 8: Sample of connector registration/update message

The response should be a MessageProcessedNotificationMessage without payload, as shown in Figure 9.

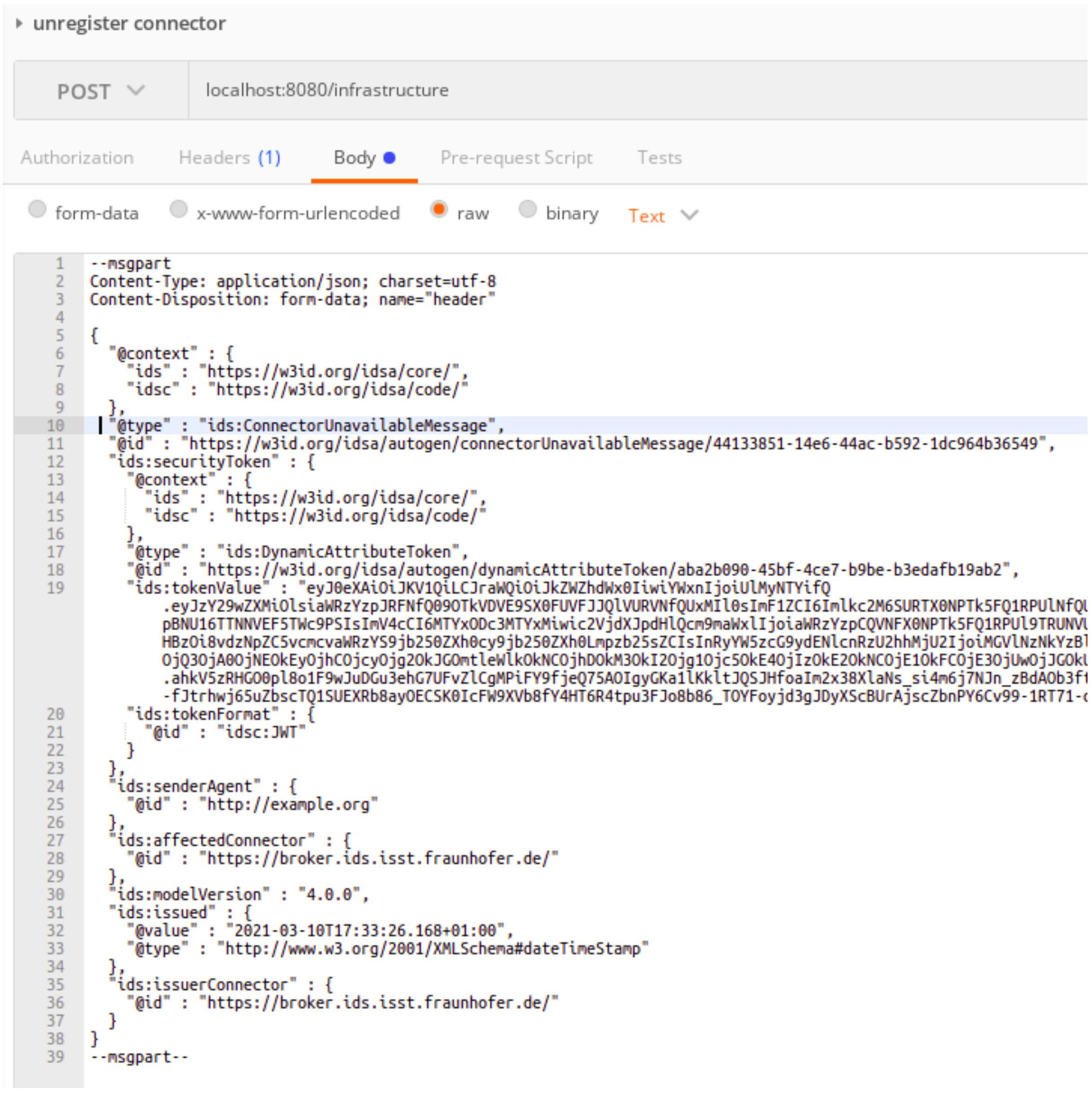


```
--WD3kKgcZoPn-P16XvL0cmSSxjrxflt6V4G45
Content-Disposition: form-data; name="header"
Content-Type: application/ld+json
Content-Length: 2286

{
  "@context" : {
    "ids" : "https://w3id.org/idsa/core/",
    "idsc" : "https://w3id.org/idsa/code/"
  },
  "@type" : "ids:MessageProcessedNotificationMessage",
  "@id" : "https://w3id.org/idsa/autogen/messageProcessedNotificationMessage/e3be96cf-7d4f-4f33-8c12-55b5c811ad22",
  "ids:correlationMessage" : {
    "@id" : "https://w3id.org/idsa/autogen/connectorUpdateMessage/cef442c0-039c-4f46-8bf0-3c451347bc32"
  },
  "ids:issued" : {
    "@value" : "2021-04-28T12:13:13.137+02:00",
    "@type" : "http://www.w3.org/2001/XMLSchema#dateTimeStamp"
  },
  "ids:issuerConnector" : {
    "@id" : "http://localhost:8080/"
  },
  "ids:senderAgent" : {
    "@id" : "https://www.iais.fraunhofer.de"
  },
  "ids:securityToken" : {
    "@type" : "ids:DynamicAttributeToken",
    "@id" : "https://w3id.org/idsa/autogen/dynamicAttributeToken/5c617ff4-844f-4ecc-a842-d7499fd70b39",
    "ids:tokenValue" : "eyJ0eXAiOiJKV1QiLCJraWQiOiJkZWZhdWx0IiwiYWxnIjoiUlMyNTYifQ
        .eyJzY29wZXMiOlsiaWRzYzpJRFNfQ09OTkVDVE9SX0FUVFJJQlVURVNfQUxMIl0sImF1ZCI6Imlkc2M6SURTX0NPTk5FQ1RPUlNfQUxMIiwia
        pRMU5ERXhOekF4TmprPSIsImV4cCI6MTYxOTYwNzQ3Mywic2VjdXJpdHlQcm9maWxlIjoiaWRzYzpCQVNFX0NPTk5FQ1RPUl9TRUNVUklUWV9Q
        HBzOi8vdzNpZC5vcmcvaWRzYS9jb250ZXh0cy9jb250ZXh0Lmpzb25sZCIsInRyYW5zcG9ydENlcnRzU2hhMjU2IjoiMGVlNzNkYzBlYmNjMDk
        OjQ3OjA0OjNEOkEyOjhCOjcyOjg2OkJGOmtleWlkOkNCOjhDOkM3OkI2Ojg1Ojc5OkE4OjIzOkE2OkNCOjE1OkFCOjE3OjUwOjJGOkU2OjY1Oj
        -RJTGmlvafCynAUBjVpcGUd6zk4Lz7eXJmXjmIsqfQOCSpFBuCRqWMEHnH0NrXi34KzHyNJwLLrfERiLSvpCSFq_gkQXEQqbe5Svsw_tcUzxOq
        -cRSoHsR_knuvSwAL0bcuEY713FzjtKfBNHaRGaQdmGTOlomak-p8PYUa2tvOTizymeXpK8HUeJD-DmVqcTtcLQ",
    "ids:tokenFormat" : {
      "@id" : "idsc:JWT"
    }
  },
  "ids:modelVersion" : "4.0.3",
  "Location" : "<http://localhost:8080/connectors/541260824>"
}
--WD3kKgcZoPn-P16XvL0cmSSxjrxflt6V4G45--

```

Figure 9: Sample response of Connector registration/update message

2.1.3 Unregister Connector

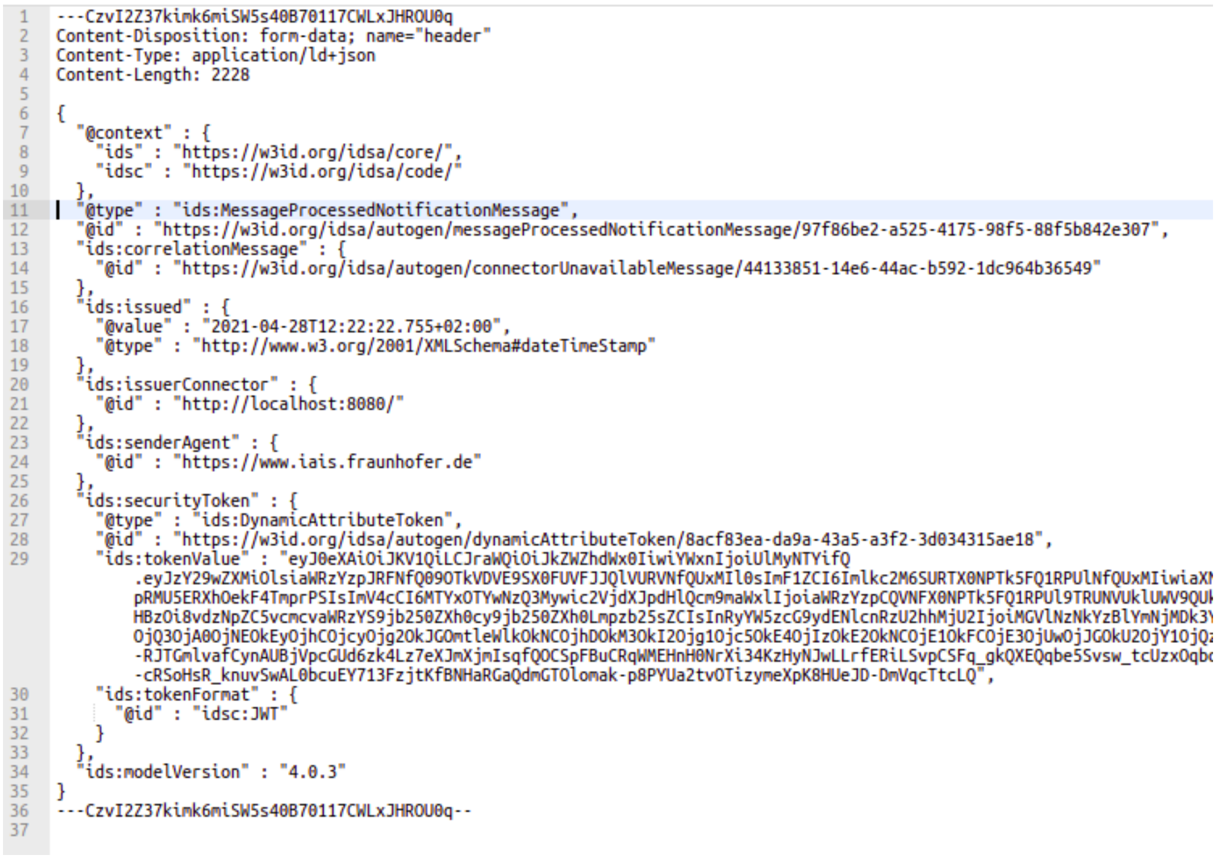
The multipart message header should be ConnectorUnavailableMessage, and the payload should be empty, as shown in Figure 10.



```
--msgpart
Content-Type: application/json; charset=utf-8
Content-Disposition: form-data; name="header"

{
  "@context" : {
    "ids" : "https://w3id.org/idsa/core/",
    "idsc" : "https://w3id.org/idsa/code/"
  },
  "@type" : "ids:ConnectorUnavailableMessage",
  "@id" : "https://w3id.org/idsa/autogen/connectorUnavailableMessage/44133851-14e6-44ac-b592-1dc964b36549",
  "ids:securityToken" : {
    "@context" : {
      "ids" : "https://w3id.org/idsa/core/",
      "idsc" : "https://w3id.org/idsa/code/"
    },
    "@type" : "ids:DynamicAttributeToken",
    "@id" : "https://w3id.org/idsa/autogen/dynamicAttributeToken/aba2b090-45bf-4ce7-b9be-b3edafb19ab2",
    "ids:tokenValue" : "eyJ0eXAiOiJKV1QiLCJraWQiOiJkZWZhdWx0IiwiYWxnIjoiUlMyNTYifQ
        .eyJzY29wZXMiOlsiaWRzYzpJRFNfQ09OTkVDVE9SX0FUVFJJQlVURVNfQUxMIl0sImF1ZCI6Imlkc2M6SURTX0NPTk5FQ1RPUlNfQU
        pBNU16TTNNVEF5TWc9PSIsImV4cCI6MTYxODc3MTYxMiwic2VjdXJpdHlQcm9maWxlIjoiaWRzYzpCQVNFX0NPTk5FQ1RPUl9TRUNVU
        HBzOi8vdzNpZC5vcmcvaWRzYS9jb250ZXh0cy9jb250ZXh0Lmpzb25sZCIsInRyYW5zcG9ydENlcnRzU2hhMjU2IjoiMGVlNzNkYzBl
        OjQ3OjA0OjNEOkEyOjhCOjcyOjg2OkJGOmtleWlkOkNCOjhDOkM3OkI2Ojg1Ojc5OkE4OjIzOkE2OkNCOjE1OkFCOjE3OjUwOjJGOkU
        .ahkV5zRHGO0pl8o1F9wJuDGu3ehG7UFvZlCgMPiFY9fjeQ75AOIgyGKa1lKkltJQSJHfoaIm2x38XlaNs_si4m6j7NJn_zBdAOb3f
        -fJtrhwj65uZbscTQ1SUEXRb8ayOECSK0IcFW9XVb8fY4HT6R4tpu3FJo8b86_TOYFoyjd3gJDyXScBUrAjscZbnPY6Cv99-1RT71-
    "ids:tokenFormat" : {
      "@id" : "idsc:JWT"
    }
  },
  "ids:senderAgent" : {
    "@id" : "http://example.org"
  },
  "ids:affectedConnector" : {
    "@id" : "https://broker.ids.isst.fraunhofer.de/"
  },
  "ids:modelVersion" : "4.0.0",
  "ids:issued" : {
    "@value" : "2021-03-10T17:33:26.168+01:00",
    "@type" : "http://www.w3.org/2001/XMLSchema#dateTimeStamp"
  },
  "ids:issuerConnector" : {
    "@id" : "https://broker.ids.isst.fraunhofer.de/"
  }
}
--msgpart--
```

Figure 10: Sample of Connector unregister message

The response should be a MessageProcessedNotificationMessage without payload, as shown in Figure 11.

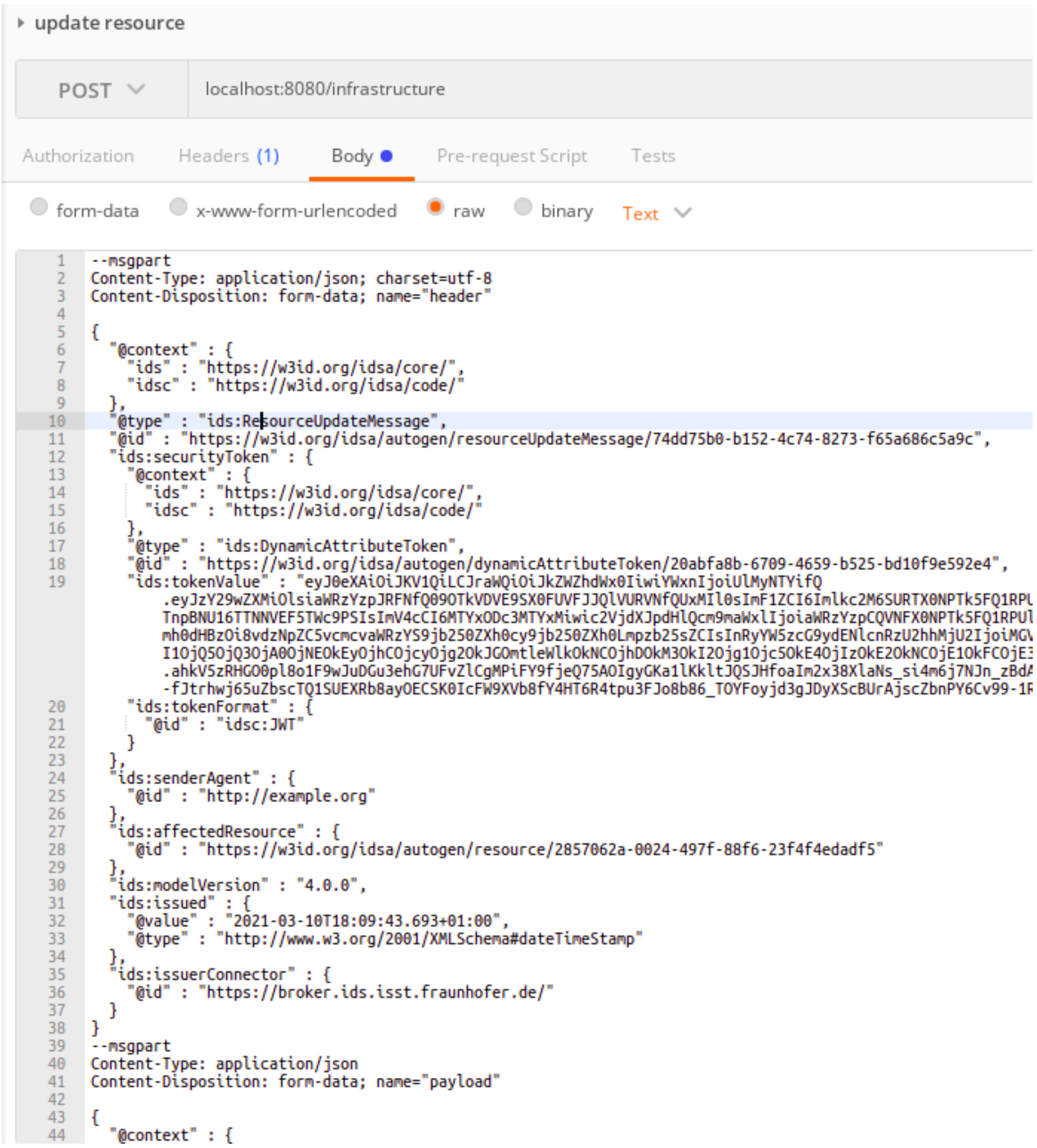


```
---CzvI2Z37kimk6miSW5s40B70117CWLxJHROU0q
Content-Disposition: form-data; name="header"
Content-Type: application/ld+json
Content-Length: 2228

{
  "@context" : {
    "ids" : "https://w3id.org/idsa/core/",
    "idsc" : "https://w3id.org/idsa/code/"
  },
  "@type" : "ids:MessageProcessedNotificationMessage",
  "@id" : "https://w3id.org/idsa/autogen/messageProcessedNotificationMessage/97f86be2-a525-4175-98f5-88f5b842e307",
  "ids:correlationMessage" : {
    "@id" : "https://w3id.org/idsa/autogen/connectorUnavailableMessage/44133851-14e6-44ac-b592-1dc964b36549"
  },
  "ids:issued" : {
    "@value" : "2021-04-28T12:22:22.755+02:00",
    "@type" : "http://www.w3.org/2001/XMLSchema#dateTimeStamp"
  },
  "ids:issuerConnector" : {
    "@id" : "http://localhost:8080/"
  },
  "ids:senderAgent" : {
    "@id" : "https://www.iais.fraunhofer.de"
  },
  "ids:securityToken" : {
    "@type" : "ids:DynamicAttributeToken",
    "@id" : "https://w3id.org/idsa/autogen/dynamicAttributeToken/8acf83ea-da9a-43a5-a3f2-3d034315ae18",
    "ids:tokenValue" : "eyJ0eXAiOiJKV1QiLCJraWQiOiJkZWZhdWx0IiwiYWxnIjoiUlMyNTYifQ
        .eyJzY29wZXMiOlsiaWRzYzpJRFNfQ09OTkVDVE9SX0FUVFJJQlVURVNfQUxMIl0sImF1ZCI6Imlkc2M6SURTX0NPTk5FQ1RPUlNfQUxMIiwiaX
        pRMU5ERXhOekF4TmprPSIsImV4cCI6MTYxOTYwNzQ3Mywic2VjdXJpdHlQcm9maWxlIjoiaWRzYzpCQVNFX0NPTk5FQ1RPUl9TRUNVUklUWV9QU
        HBzOi8vdzNpZC5vcmcvaWRzYS9jb250ZXh0cy9jb250ZXh0Lmpzb25sZCIsInRyYW5zcG9ydENlcnRzU2hhMjU2IjoiMGVlNzNkYzBlYmNjMDk3
        OjQ3OjA0OjNEOkEyOjhCOjcyOjg2OkJGOmtleWlkOkNCOjhDOkM3OkI2Ojg1Ojc5OkE4OjIzOkE2OkNCOjE1OkFCOjE3OjUwOjJGOkU2OjY1OjQ
        -RJTGmlvafCynAUBjVpcGUd6zk4Lz7eXJmXjmIsqfQOCSpFBuCRqWMEHnH0NrXi34KzHyNJwLLrfERiLSvpCSFq_gkQXEQbe5Svsw_tcUzxOqb
        -cRSoHsR_knuvSwAL0bcuEY713FzjtKfBNHaRGaQdmGTOlomak-p8PYUa2tvOTizymeXpK8HUeJD-DmVqcTtcLQ",
    "ids:tokenFormat" : {
      "@id" : "idsc:JWT"
    }
  },
  "ids:modelVersion" : "4.0.3"
}
---CzvI2Z37kimk6miSW5s40B70117CWLxJHROU0q--
```

Figure 11: Sample response of Connector unregister message

2.1.4 Update Resource

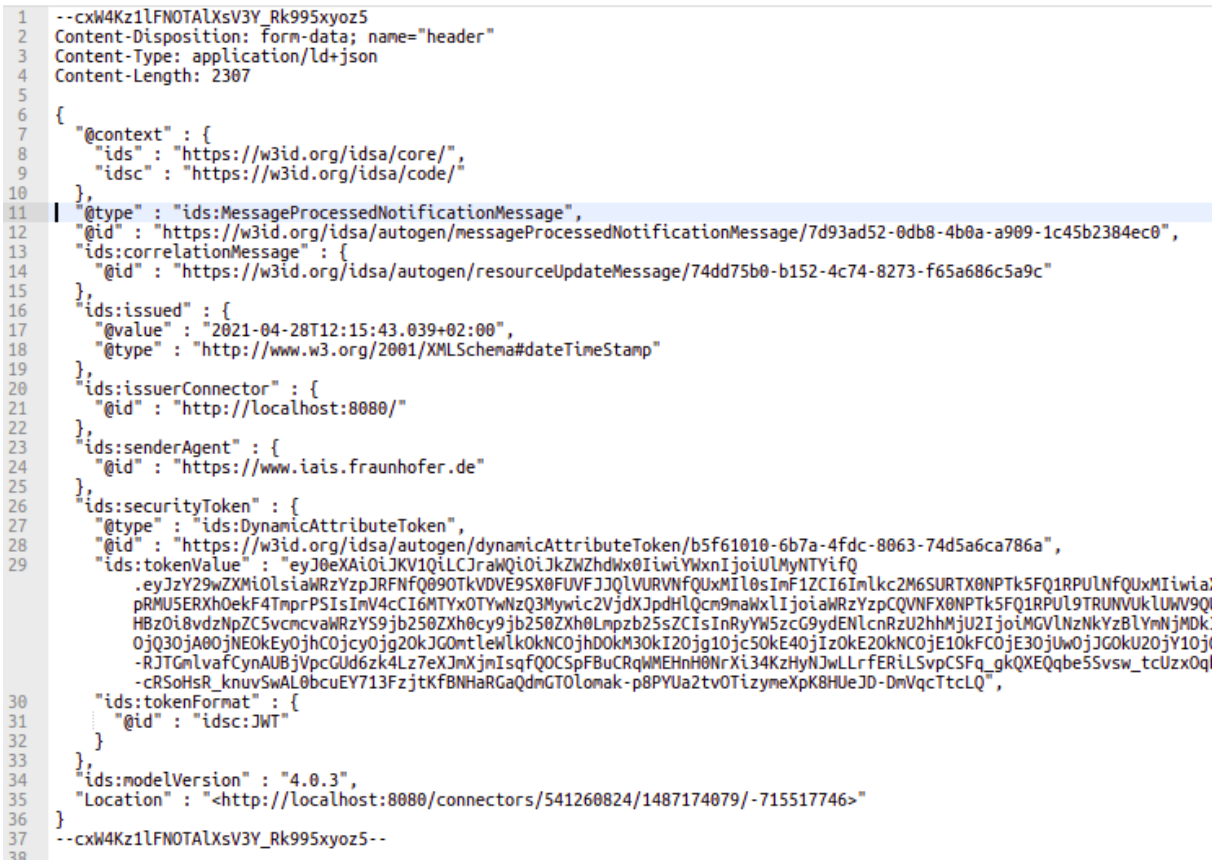
The multipart message header should be ResourceUpdateMessage, and the payload should be JSON-LD format resource metadata, as shown in Figure 12.



```
--msgpart
Content-Type: application/json; charset=utf-8
Content-Disposition: form-data; name="header"

{
  "@context" : {
    "ids" : "https://w3id.org/idsa/core/",
    "idsc" : "https://w3id.org/idsa/code/"
  },
  "@type" : "ids:ResourceUpdateMessage",
  "@id" : "https://w3id.org/idsa/autogen/resourceUpdateMessage/74dd75b0-b152-4c74-8273-f65a686c5a9c",
  "ids:securityToken" : {
    "@context" : {
      "ids" : "https://w3id.org/idsa/core/",
      "idsc" : "https://w3id.org/idsa/code/"
    },
    "@type" : "ids:DynamicAttributeToken",
    "@id" : "https://w3id.org/idsa/autogen/dynamicAttributeToken/20abfa8b-6709-4659-b525-bd10f9e592e4",
    "ids:tokenValue" : "eyJ0eXAiOiJKV1QiLCJraWQiOiJkZWZhdWx0IiwiYWxnIjoiUlMyNTYifQ
        .eyJzY29wZXMiOlsiaWRzYzpJRFNfQ090TkVDVE9SX0FUVFJJQlVURVNfQUxMIl0sImF1ZCI6Imlkc2M6SURTX0NPTk5FQ1RPU
        TnpBNU16TTNNVEF5TWc9PSIsImV4cCI6MTYxODc3MTYxMiwic2VjdXJpdHlQcm9maWxlIjoiaWRzYzpCQVNFX0NPTk5FQ1RPU
        mh0dHBzOi8vdzNpZC5vcmcvaWRzYS9jb250ZXh0cy9jb250ZXh0Lmpzb25sZCIsInRyYW5zcG9ydENlcnRzU2hhMjU2IjoiMG
        I1OjQ5OjQ3OjA0OjNEOkEyOjhCOjcyOjg2OkJGOmtleWlkOkNCOjhDOkM3OkI2Ojg1Ojc5OkE4OjIzOkE2OkNCOjE1OkFCOjE
        .ahkV5zRHGO0pl8o1F9wJuDGu3ehG7UFvZlCgMPiFY9fjeQ75AOIgyGKa1lKkltJQSJHfoaIm2x38XlaNs_si4m6j7NJn_zBd
        -fJtrhwj65uZbscTQ1SUEXRb8ayOECSK0IcFW9XVb8fY4HT6R4tpu3FJo8b86_TOYFoyjd3gJDyXScBUrAjscZbnPY6Cv99-1
    "ids:tokenFormat" : {
      "@id" : "idsc:JWT"
    }
  },
  "ids:senderAgent" : {
    "@id" : "http://example.org"
  },
  "ids:affectedResource" : {
    "@id" : "https://w3id.org/idsa/autogen/resource/2857062a-0024-497f-88f6-23f4f4edadf5"
  },
  "ids:modelVersion" : "4.0.0",
  "ids:issued" : {
    "@value" : "2021-03-10T18:09:43.693+01:00",
    "@type" : "http://www.w3.org/2001/XMLSchema#dateTimeStamp"
  },
  "ids:issuerConnector" : {
    "@id" : "https://broker.ids.isst.fraunhofer.de/"
  }
}
--msgpart
Content-Type: application/json
Content-Disposition: form-data; name="payload"

{
  "@context" : {
```

Figure 12: Sample of the resource update message

The response should be a MessageProcessedNotificationMessage without payload, as shown in Figure 13.



```
--cxW4Kz1lFNOTAlXsV3Y_Rk995xyoz5
Content-Disposition: form-data; name="header"
Content-Type: application/ld+json
Content-Length: 2307

{
  "@context" : {
    "ids" : "https://w3id.org/idsa/core/",
    "idsc" : "https://w3id.org/idsa/code/"
  },
  "@type" : "ids:MessageProcessedNotificationMessage",
  "@id" : "https://w3id.org/idsa/autogen/messageProcessedNotificationMessage/7d93ad52-0db8-4b0a-a909-1c45b2384ec0",
  "ids:correlationMessage" : {
    "@id" : "https://w3id.org/idsa/autogen/resourceUpdateMessage/74dd75b0-b152-4c74-8273-f65a686c5a9c"
  },
  "ids:issued" : {
    "@value" : "2021-04-28T12:15:43.039+02:00",
    "@type" : "http://www.w3.org/2001/XMLSchema#dateTimeStamp"
  },
  "ids:issuerConnector" : {
    "@id" : "http://localhost:8080/"
  },
  "ids:senderAgent" : {
    "@id" : "https://www.iais.fraunhofer.de"
  },
  "ids:securityToken" : {
    "@type" : "ids:DynamicAttributeToken",
    "@id" : "https://w3id.org/idsa/autogen/dynamicAttributeToken/b5f61010-6b7a-4fdc-8063-74d5a6ca786a",
    "ids:tokenValue" : "eyJ0eXAiOiJKV1QiLCJraWQiOiJkZWZhdWx0IiwiYWxnIjoiUlMyNTYifQ
        .eyJzY29wZXMiOlsiaWRzYzpJRFNfQ09OTkVDVE9SX0FUVFJJQlVURVNfQUxMIl0sImF1ZCI6Imlkc2M6SURTX0NPTk5FQ1RPUlNfQUxMIiwia
        pRMU5ERXhOekF4TmprPSIsImV4cCI6MTYxOTYwNzQ3Mywic2VjdXJpdHlQcm9maWxlIjoiaWRzYzpCQVNFX0NPTk5FQ1RPUl9TRUNVUklUWV9Q
        HBzOi8vdzNpZC5vcmcvaWRzYS9jb250ZXh0cy9jb250ZXh0Lmpzb25sZCIsInRyYW5zcG9ydENlcnRzU2hhMjU2IjoiMGVlNzNkYzBlYmNjMDk
        OjQ3OjA0OjNEOkEyOjhCOjcyOjg2OkJGOmtleWlkOkNCOjhDOkM3OkI2Ojg1Ojc5OkE4OjIzOkE2OkNCOjE1OkFCOjE3OjUwOjJGOkU2OjY1Oj
        -RJTGmlvafCynAUBjVpcGUd6zk4Lz7eXJmXjmIsqfQOCSpFBuCRqWMEHnH0NrXi34KzHyNJwLLrfERiLSvpCSFq_gkQXEQqbe5Svsw_tcUzxOq
        -cRSoHsR_knuvSwAL0bcuEY713FzjtKfBNHaRGaQdmGTOlomak-p8PYUa2tvOTizymeXpK8HUeJD-DmVqcTtcLQ",
    "ids:tokenFormat" : {
      "@id" : "idsc:JWT"
    }
  },
  "ids:modelVersion" : "4.0.3",
  "Location" : "<http://localhost:8080/connectors/541260824/1487174079/-715517746>"
}
--cxW4Kz1lFNOTAlXsV3Y_Rk995xyoz5--
```

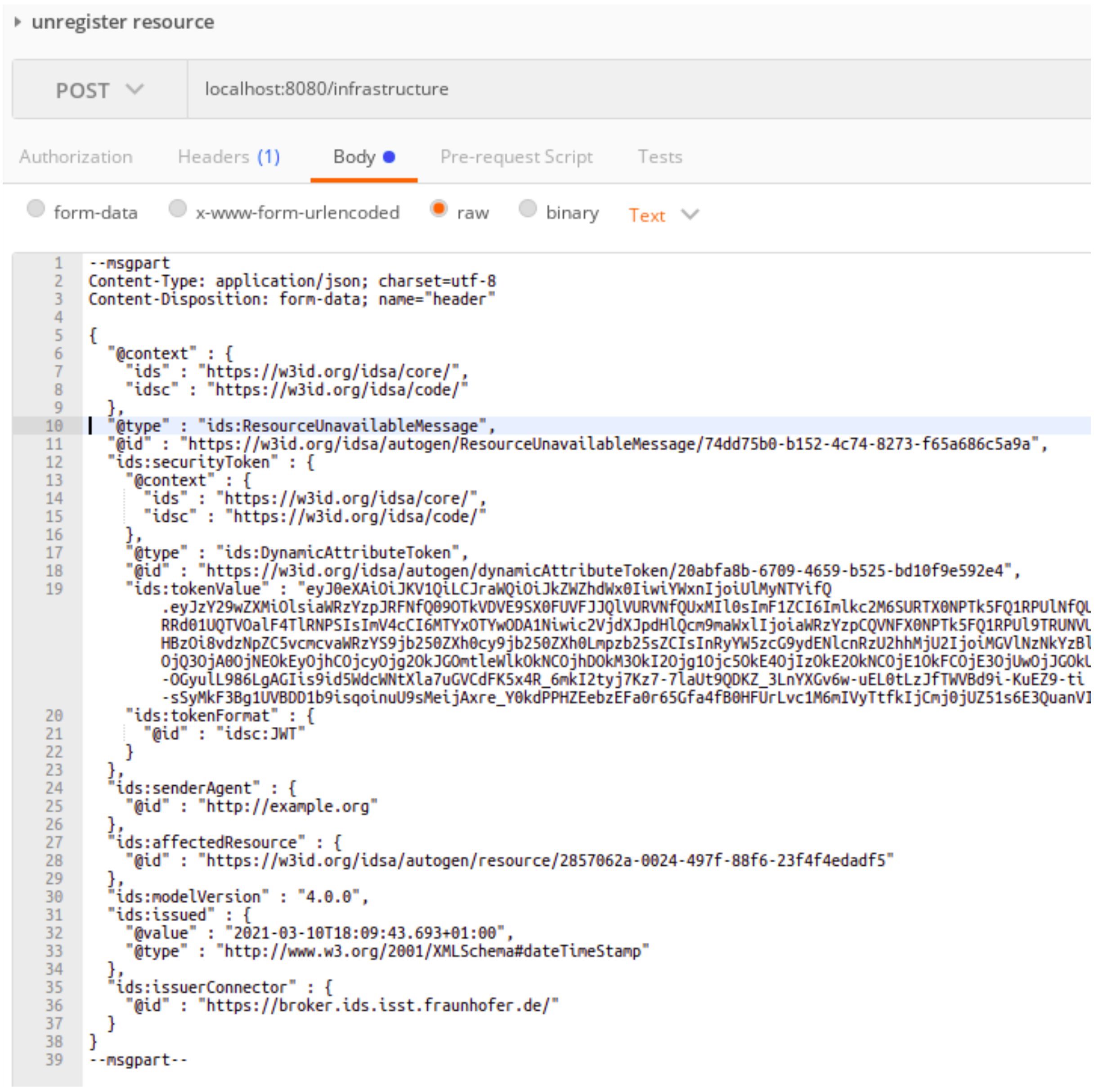
Figure 13: Sample response of Resource update message

Below is an example of a self-description of a Resource in JSON-LD format.

```
{
  "@context" : {
    "ids" : "https://w3id.org/idsa/core/",
    "idsc" : "https://w3id.org/idsa/code/"
  },
  "@type" : "ids:Resource",
  "@id" : "https://w3id.org/idsa/autogen/resource/2857062a-0024-497f-88f6-23f4f4edadf5",
  "ids:language" : [ {
    "@id" : "idsc:DE"
  }, {
    "@id" : "idsc:EN"
  } ],
  "ids:version" : "1.2",
  "ids:contentType" : {
    "@id" : "idsc:SCHEMA_DEFINITION"
  },
  "ids:description" : [ {
    "@value" : "Multiple numbers of resources could be offered by the connectors",
    "@type" : "http://www.w3.org/2001/XMLSchema#string"
  }, {
    "@value" : "provide more description of this resource",
    "@type" : "http://www.w3.org/2001/XMLSchema#string"
  } ],
  "ids:keyword" : [ {
    "@value" : "broker",
    "@type" : "http://www.w3.org/2001/XMLSchema#string"
  }, {
    "@value" : "metadata",
    "@type" : "http://www.w3.org/2001/XMLSchema#string"
  } ],
  "ids:title" : [ {
    "@value" : "Title of another resource offered by the Connector",
    "@type" : "http://www.w3.org/2001/XMLSchema#string"
  }, {
    "@value" : "Extending title",
    "@type" : "http://www.w3.org/2001/XMLSchema#string"
  } ]
}
```

2.1.5 Unregister Resource

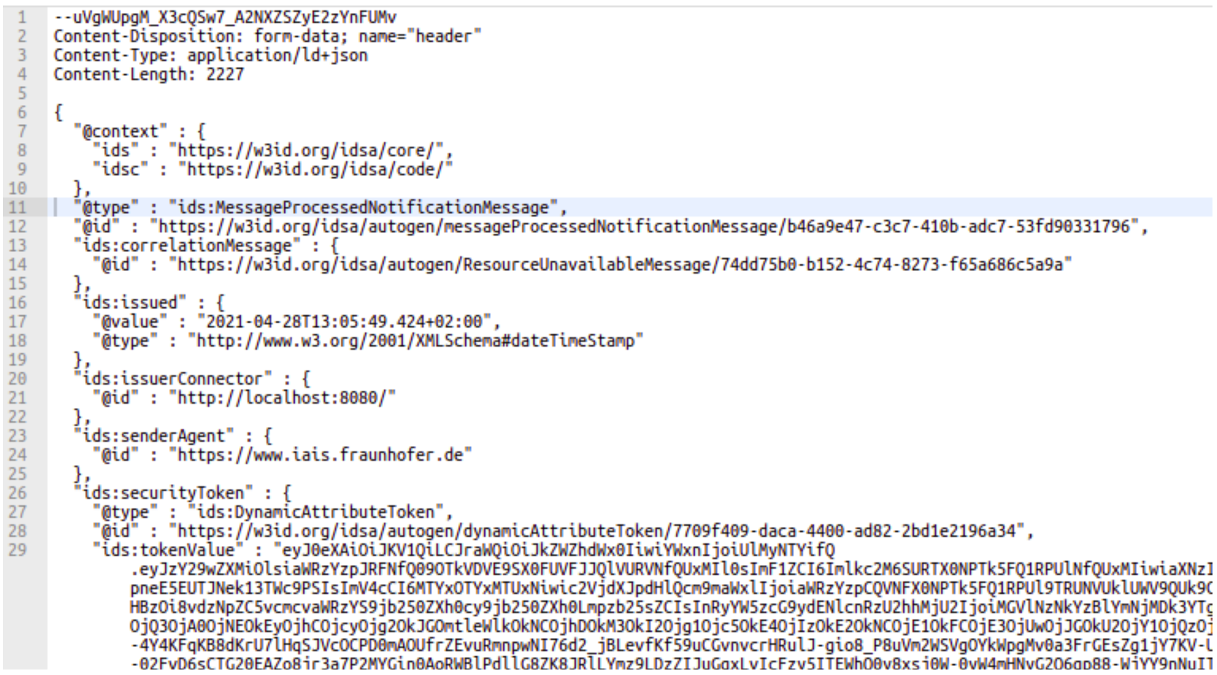
The multipart message header should be ResourceUnavailableMessage, and the payload should be empty, as shown in Figure 14.



```
--msgpart
Content-Type: application/json; charset=utf-8
Content-Disposition: form-data; name="header"

{
  "@context" : {
    "ids" : "https://w3id.org/idsa/core/",
    "idsc" : "https://w3id.org/idsa/code/"
  },
  "@type" : "ids:ResourceUnavailableMessage",
  "@id" : "https://w3id.org/idsa/autogen/ResourceUnavailableMessage/74dd75b0-b152-4c74-8273-f65a686c5a9a",
  "ids:securityToken" : {
    "@context" : {
      "ids" : "https://w3id.org/idsa/core/",
      "idsc" : "https://w3id.org/idsa/code/"
    },
    "@type" : "ids:DynamicAttributeToken",
    "@id" : "https://w3id.org/idsa/autogen/dynamicAttributeToken/20abfa8b-6709-4659-b525-bd10f9e592e4",
    "ids:tokenValue" : "eyJ0eXAiOiJKV1QiLCJraWQiOiJkZWZhdWx0IiwiYWxnIjoiUlMyNTYifQ
        .eyJzY29wZXMiOlsiaWRzYzpJRFNfQ09OTkVDVE9SX0FUVFJJQlVURVNfQUxMIl0sImF1ZCI6Imlkc2M6SURTX0NPTk5FQ1RPUlNfQL
        RRd01UQTVOalF4TlRNPSIsImV4cCI6MTYxOTYwODA1Niwic2VjdXJpdHlQcm9maWxlIjoiaWRzYzpCQVNFX0NPTk5FQ1RPUl9TRUNVL
        HBzOi8vdzNpZC5vcmcvaWRzYS9jb250ZXh0cy9jb250ZXh0Lmpzb25sZCIsInRyYW5zcG9ydENlcnRzU2hhMjU2IjoiMGVlNzNkYzBl
        OjQ3OjA0OjNEOkEyOjhCOjcyOjg2OkJGOmtleWlkOkNCOjhDOkM3OkI2Ojg1Ojc5OkE4OjIzOkE2OkNCOjE1OkFCOjE3OjUwOjJGOkL
        -OGyulL986LgAGIis9id5WdcWNtXla7uGVCdFK5x4R_6mkI2tyj7Kz7-7laUt9QDKZ_3LnYXGv6w-uEL0tLzJfTWVBd9i-KuEZ9-ti
        -sSyMkF3Bg1UVBDD1b9isqoinuU9sMeijAxre_Y0kdPPHZEebzEFa0r65Gfa4fB0HFUrLvc1M6mIVyTtfkIjCmj0jUZ51s6E3QuanVI
    "ids:tokenFormat" : {
      "@id" : "idsc:JWT"
    }
  },
  "ids:senderAgent" : {
    "@id" : "http://example.org"
  },
  "ids:affectedResource" : {
    "@id" : "https://w3id.org/idsa/autogen/resource/2857062a-0024-497f-88f6-23f4f4edadf5"
  },
  "ids:modelVersion" : "4.0.0",
  "ids:issued" : {
    "@value" : "2021-03-10T18:09:43.693+01:00",
    "@type" : "http://www.w3.org/2001/XMLSchema#dateTimeStamp"
  },
  "ids:issuerConnector" : {
    "@id" : "https://broker.ids.isst.fraunhofer.de/"
  }
}
--msgpart--
```

Figure 14: Sample of resource unregistration message

The response should be a MessageProcessedNotificationMessage without payload, as shown in Figure 15.

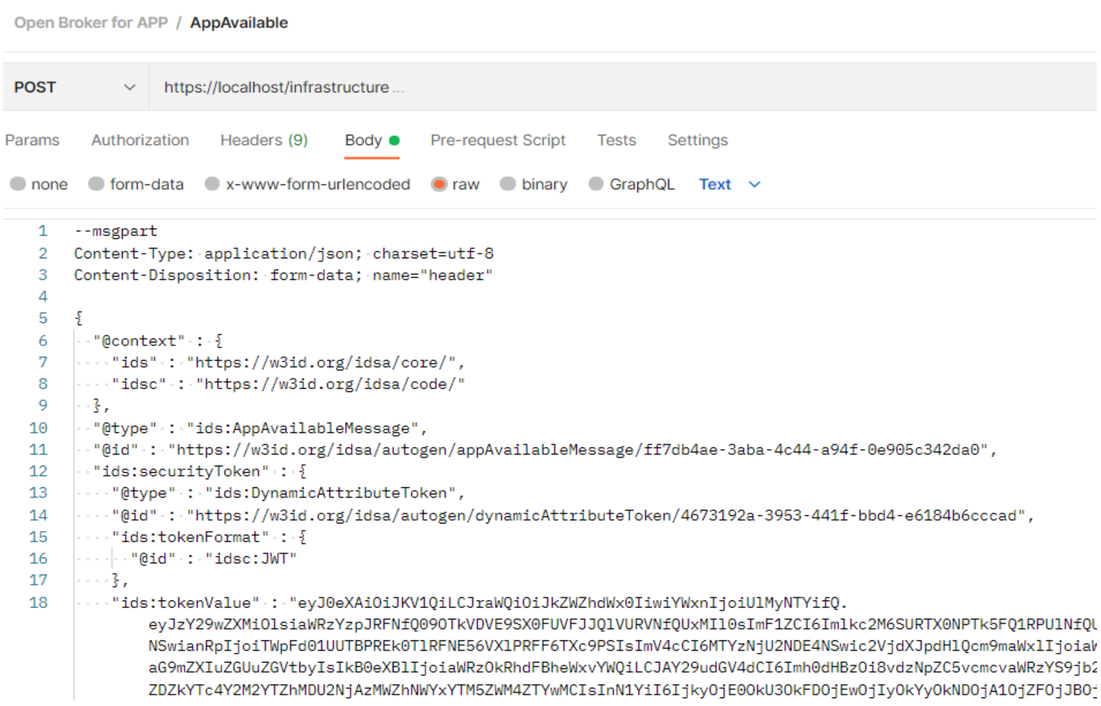


```
--uVgWUpgM_X3cQSw7_A2NXZSZyE2zYnFUMv
Content-Disposition: form-data; name="header"
Content-Type: application/ld+json
Content-Length: 2227

{
  "@context" : {
    "ids" : "https://w3id.org/idsa/core/",
    "idsc" : "https://w3id.org/idsa/code/"
  },
  "@type" : "ids:MessageProcessedNotificationMessage",
  "@id" : "https://w3id.org/idsa/autogen/messageProcessedNotificationMessage/b46a9e47-c3c7-410b-adc7-53fd90331796",
  "ids:correlationMessage" : {
    "@id" : "https://w3id.org/idsa/autogen/ResourceUnavailableMessage/74dd75b0-b152-4c74-8273-f65a686c5a9a"
  },
  "ids:issued" : {
    "@value" : "2021-04-28T13:05:49.424+02:00",
    "@type" : "http://www.w3.org/2001/XMLSchema#dateTimeStamp"
  },
  "ids:issuerConnector" : {
    "@id" : "http://localhost:8080/"
  },
  "ids:senderAgent" : {
    "@id" : "https://www.iais.fraunhofer.de"
  },
  "ids:securityToken" : {
    "@type" : "ids:DynamicAttributeToken",
    "@id" : "https://w3id.org/idsa/autogen/dynamicAttributeToken/7709f409-daca-4400-ad82-2bd1e2196a34",
    "ids:tokenValue" : "eyJ0eXAiOiJKV1QiLCJraWQiOiJkZWZhdWx0IiwiYWxnIjoiUlMyNTYifQ
        .eyJzY29wZXMiOlsiaWRzYzpJRFNfQ09OTkVDVE9SX0FUVFJJQlVURVNfQUxMIl0sImF1ZCI6Imlkc2M6SURTX0NPTk5FQ1RPUlNfQUxMIiwiaXNzI
        pneE5EUTJNek13TWc9PSIsImV4cCI6MTYxOTYxMTUxNiwic2VjdXJpdHlQcm9maWxlIjoiaWRzYzpCQVNFX0NPTk5FQ1RPUl9TRUNVUklUWV9QUk9C
        HBzOi8vdzNpZC5vcmcvaWRzYS9jb250ZXh0cy9jb250ZXh0Lmpzb25sZCIsInRyYW5zcG9ydENlcnRzU2hhMjU2IjoiMGVlNzNkYzBlYmNjMDk3YTg
        0jQ3OjA0OjNEOkEyOjhCOjcyOjg2OkJGOmtleWlkOkNCOjhDOkM3OkI2Ojg1Ojc5OkE4OjIzOkE2OkNCOjE1OkFCOjE3OjUwOjJGOkU2OjY1OjQzOj
        -4Y4KFqKB8dKrU7lHqSJVcOCPD0mAOUfrZEvuRmnpwNI76d2_jBLevfKf59uCGvnvcrHRulJ-gio8_P8uVm2WSVgOYkWpgMv0a3FrGEsZg1jY7KV-L
        -02FvD6sCTG20EAZo8ir3a7P2MYGin0AoRWBlPdllG8ZK8JRlLYmz9LDzZIJuGqxLvIcFzv5ITEWhO0v8xsj0W-0vW4mHNvG2O6qp88-WjYY9nNuIT
```

Figure 15: Sample response of Resource unregistration message

2.1.6 Register App

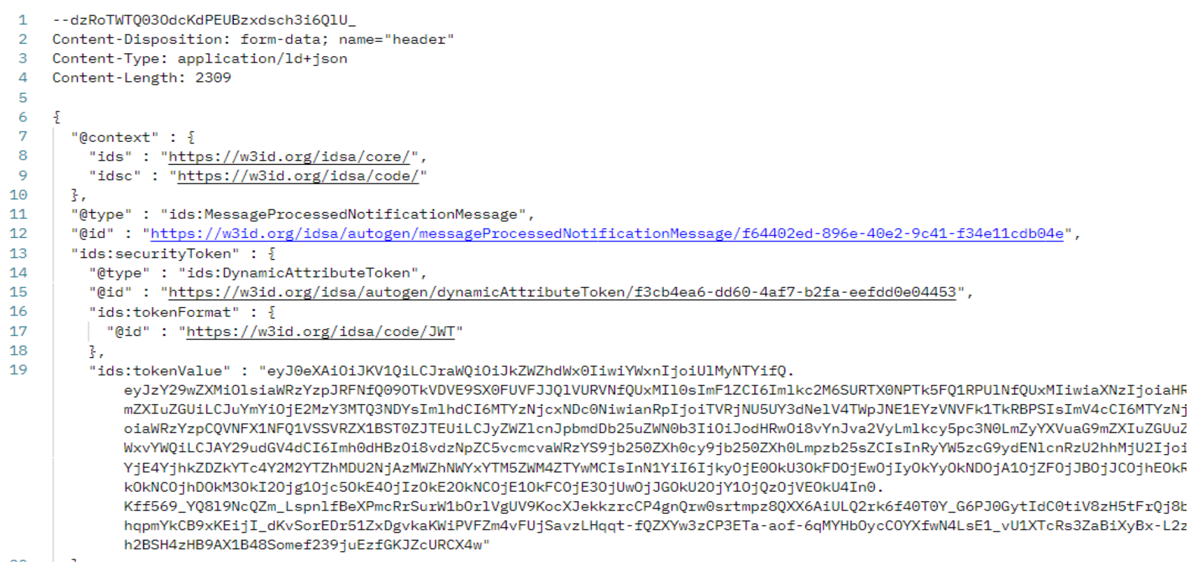
Participants need to register a Connector before registering an App. The App will be there in the resource catalog of the Connector. The multipart message header should be AppAvailableMessage, and the payload should be JSON-LD format AppResource metadata, as shown in Figure 16.



```
--msgpart
Content-Type: application/json; charset=utf-8
Content-Disposition: form-data; name="header"

{
  "@context" : {
    "ids" : "https://w3id.org/idsa/core/",
    "idsc" : "https://w3id.org/idsa/code/"
  },
  "@type" : "ids:AppAvailableMessage",
  "@id" : "https://w3id.org/idsa/autogen/appAvailableMessage/ff7db4ae-3aba-4c44-a94f-0e905c342da0",
  "ids:securityToken" : {
    "@type" : "ids:DynamicAttributeToken",
    "@id" : "https://w3id.org/idsa/autogen/dynamicAttributeToken/4673192a-3953-441f-bbd4-e6184b6cccad",
    "ids:tokenFormat" : {
      "@id" : "idsc:JWT"
    },
    "ids:tokenValue" : "eyJ0eXAiOiJKV1QiLCJraWQiOiJkZWZhdWx0IiwiYWxnIjoiUlMyNTYifQ.
        eyJzY29wZXMiOlsiaWRzYzpJRFNfQ09OTkVDVE9SX0FUVFJJQlVURVNfQUxMIl0sImF1ZCI6Imlkc2M6SURTX0NPTk5FQ1RPUlNfQU
        NSwianRpIjoiTWpFd01UUTBPREk0TlRFNE56VXlPRFF6TXc9PSIsImV4cCI6MTYzNjU2NDE4NSwic2VjdXJpdHlQcm9maWxlIjoiaW
        aG9mZXIuZGUuZGVtbyIsIkB0eXBlIjoiaWRzOkRhdFBheWxvYWQiLCJAY29udGV4dCI6Imh0dHBzOi8vdzNpZC5vcmcvaWRzYS9jb2
        ZDZkYTc4Y2M2YTZhMDU2NjAzMWZhNWYxYTM5ZWM4ZTYwMCIsInN1YiI6IjkyOjE0OkU3OkFDOjEwOjIyOkYyOkNDOjA1OjZFOjBO
```

Figure 16: Sample of App registration message

The response should be a MessageProcessedNotificationMessage without payload, as shown in Figure 17.



```
--dzRoTWTQ03OdcKdPEUBzxdsch3i6QlU_
Content-Disposition: form-data; name="header"
Content-Type: application/ld+json
Content-Length: 2309

{
  "@context" : {
    "ids" : "https://w3id.org/idsa/core/",
    "idsc" : "https://w3id.org/idsa/code/"
  },
  "@type" : "ids:MessageProcessedNotificationMessage",
  "@id" : "https://w3id.org/idsa/autogen/messageProcessedNotificationMessage/f64402ed-896e-40e2-9c41-f34e11cdb04e",
  "ids:securityToken" : {
    "@type" : "ids:DynamicAttributeToken",
    "@id" : "https://w3id.org/idsa/autogen/dynamicAttributeToken/f3cb4ea6-dd60-4af7-b2fa-eefdd0e04453",
    "ids:tokenFormat" : {
      "@id" : "https://w3id.org/idsa/code/JWT"
    },
    "ids:tokenValue" : "eyJ0eXAiOiJKV1QiLCJraWQiOiJkZWZhdWx0IiwiYWxnIjoiUlMyNTYifQ.
        eyJzY29wZXMiOlsiaWRzYzpJRFNfQ09OTkVDVE9SX0FUVFJJQlVURVNfQUxMIl0sImF1ZCI6Imlkc2M6SURTX0NPTk5FQ1RPUlNfQUxMIiwiaXNzIjoiaH
        mZXIuZGUiLCJuYmYiOjE2MzY3MTQ3NDYsImlhdCI6MTYzNjcxNDc0NiwianRpIjoiTVRjNU5UY3dNelV4TWpJNE1EYzVNVFk1TkRBPSIsImV4cCI6MTYzNj
        oiaWRzYzpCQVNFX1NFQ1VSSVRZX1BST0ZJTEUiLCJyZWZlcnJpbmdDb25uZWN0b3IiOiJodHRwOi8vYnJva2VyLmlkcy5pc3N0LmZyYXVuaG9mZXIuZGUu
        WxvYWQiLCJAY29udGV4dCI6Imh0dHBzOi8vdzNpZC5vcmcvaWRzYS9jb250ZXh0cy9jb250ZXh0Lmpzb25sZCIsInRyYW5zcG9ydENlcnRzU2hhMjU2Ijoi
        YjE4YjhkZDZkYTc4Y2M2YTZhMDU2NjAzMWZhNWYxYTM5ZWM4ZTYwMCIsInN1YiI6IjkyOjE0OkU3OkFDOjEwOjIyOkYyOkNDOjA1OjZFOjBBOjJCOjhEOk
        kOkNCOjhDOkM3OkI2Ojg1Ojc5OkE4OjIzOkE2OkNCOjE1OkFCOjE3OjUwOjJGOkU2OjY1OjQzOjVEOkU4In0.
        Kff569_YQ8l9NcQZm_LspnlfBeXPmcRrSurW1bOrlVgUV9KocXJekkzrcCP4gnQrw0srtmpz8QXX6AiULQ2rk6f40T0Y_G6PJ0GytIdC0tiV8zH5tFrQj8
        hqpmYkCB9xKEijI_dKvSorEDr51ZxDgvkaKWiPVFZm4vFUjSavzLHqqt-fQZXYw3zCP3ETa-aof-6qMYHbOycCOYXfwN4LsE1_vU1XTcRs3ZaBiXyBx-L2z
        h2BSH4zHB9AX1B48Somef239juEzfGKJZcURCX4w"
```

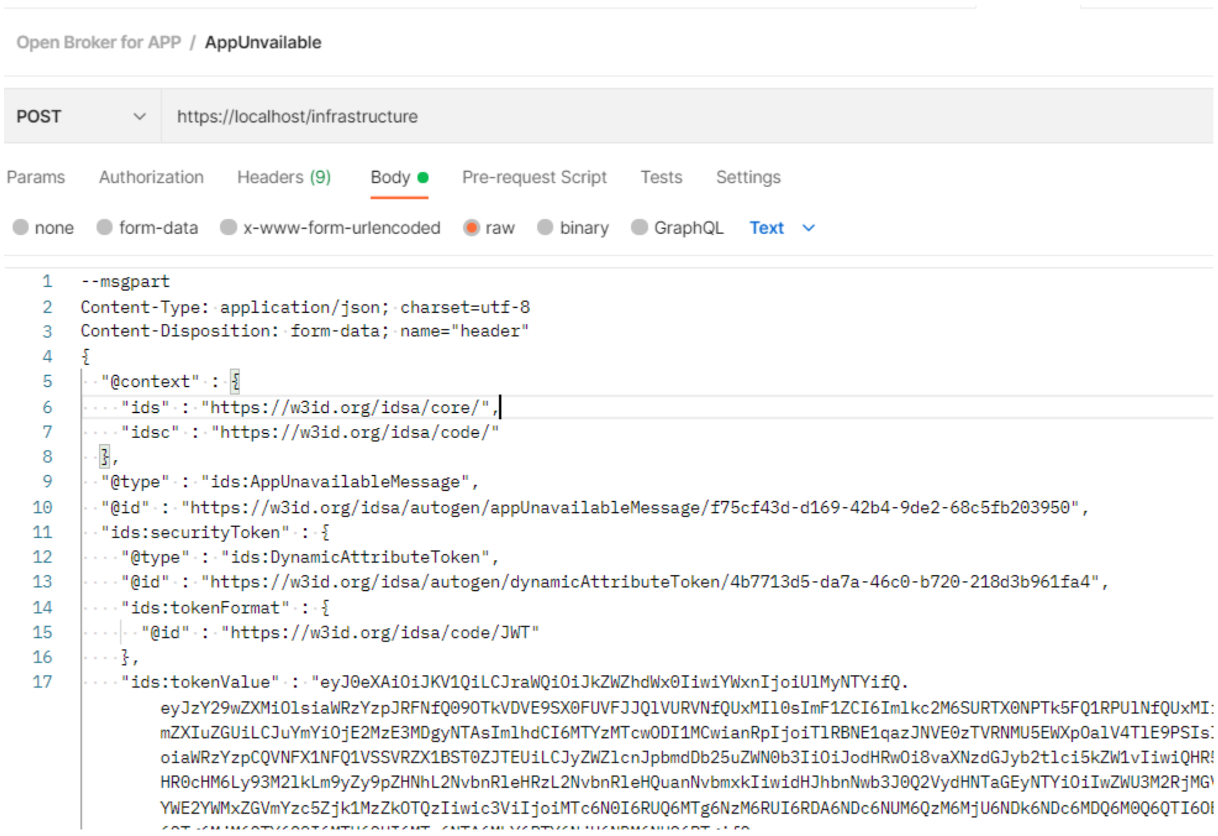
Figure 17: Sample response of App registration message

Below is the example of a self-description of an AppResource in JSON-LD format.

```
{
“@context” : {
“ids” : “https://w3id.org/idsa/core/”,
“idsc” : “https://w3id.org/idsa/code/”
  },
“@type” : “ids:AppResource”,
“@id” : “https://example.org/app2”,
“ids:language” : [ {
“@id” : “idsc:DE”
  }, {
"@id ": "idsc:EN "
  } ],
„ids:version‟ : „1.2‟,
„ids:contentType‟ : {
„@id‟ : „idsc:SCHEMA_DEFINITION‟
  },
“ids:description” : [ {
"@value": "Multiple numbers of resources could be offered by the connectors",
"@type": "http://www.w3.org/2001/XMLSchema#string"
  }, {
"@value": "provide more description of this resource",
"@type": "http://www.w3.org/2001/XMLSchema#string"
  } ],
“ids:keyword” : [ {
“@value” : “broker”,
“@type” : “http://www.w3.org/2001/XMLSchema#string”
  }, {
“@value” : “metadata”,
“@type” : “http://www.w3.org/2001/XMLSchema#string”
  } ],
“ids:title” : [ {
"@value": "Title of another resource offered by the Connector",
"@type": "http://www.w3.org/2001/XMLSchema#string"
  }, {
"@value": "Extending title",
"@type": "http://www.w3.org/2001/XMLSchema#string"
  } ],
“ids:representation” : [ {
“@type” : “ids:AppRepresentation” ,
“ids:mediaType” : { “@id” : “https://www.iana.org/assignments/media-types/application/zip” },
“ids:dataAppDistributionService” : { “@id” : “https://example.com” },
“ids:dataAppRuntimeEnvironment” : “Docker” ,
“ids:dataAppInformation” : {
“@type” : “ids:SmartDataApp” ,
“ids:appDocumentation” : “App-related human-readable documentation.” ,
"ids:appEnvironmentVariables" : "$Env1 = environment variable 1, $Env2 = environment variable
2" ,
"ids:appStorageConfiguration" : "1 Docker volume required, e.g., -v /data" ,
"ids:appEndpoint" : {
"@type" : "ids:AppEndpoint" ,
"ids:appEndpointInformation" : "I am an app endpoint. I do endpoint things.",
"ids:appEndpointDocumentation" : { "@id" : "https://app.swaggerhub.com/apis/app/1337" },
"ids:appEndpointType" : "idsc:INPUT_ENDPOINT"  ,
"ids:appEndPointPort"  :  5000 ,
"ids:path" : "/input" ,
"ids:appEndpointMediaType"    :    {    "@id"    :    "https://www.iana.org/assignments/media-
types/application/json" },
"ids:appEndpointProtocol" : "HTTP/1.1"
            }
               }
     }]
}
```

2.1.7 Unregister App

The multipart message header should be AppUnavailableMessage, and the payload should be empty, as shown in Figure 18.

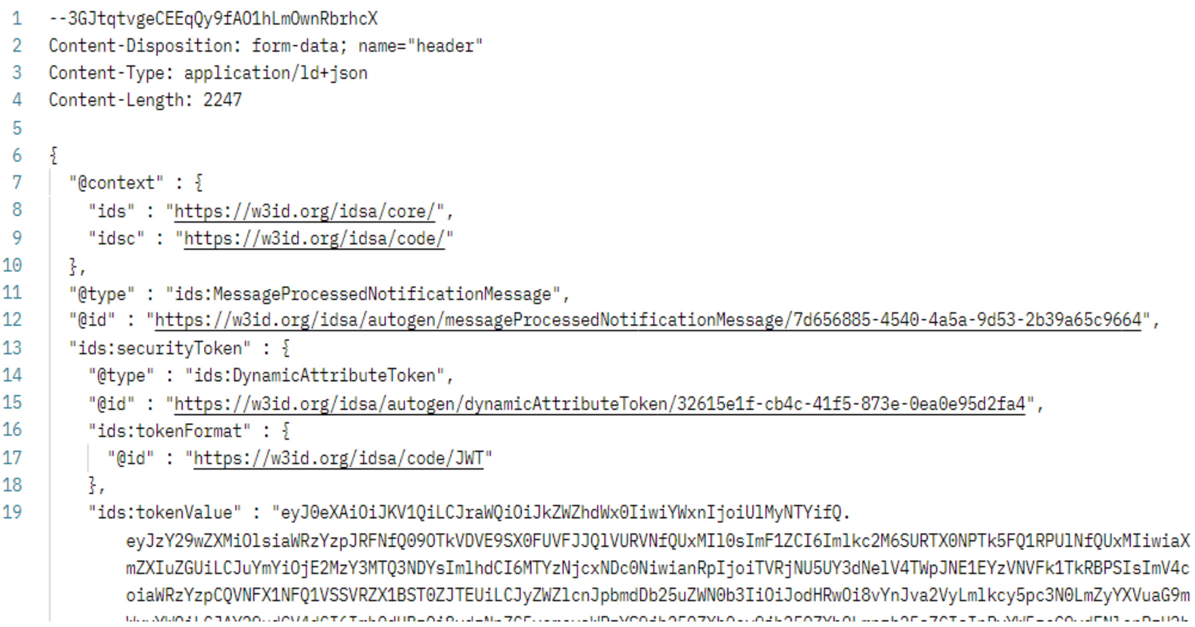


POST https://localhost/infrastructure

```
--msgpart
Content-Type: application/json; charset=utf-8
Content-Disposition: form-data; name="header"
{
  "@context" : {
    "ids" : "https://w3id.org/idsa/core/",
    "idsc" : "https://w3id.org/idsa/code/"
  },
  "@type" : "ids:AppUnavailableMessage",
  "@id" : "https://w3id.org/idsa/autogen/appUnavailableMessage/f75cf43d-d169-42b4-9de2-68c5fb203950",
  "ids:securityToken" : {
    "@type" : "ids:DynamicAttributeToken",
    "@id" : "https://w3id.org/idsa/autogen/dynamicAttributeToken/4b7713d5-da7a-46c0-b720-218d3b961fa4",
    "ids:tokenFormat" : {
      "@id" : "https://w3id.org/idsa/code/JWT"
    },
    "ids:tokenValue" : "eyJ0eXAiOiJKV1QiLCJraWQiOiJkZWZhdWx0IiwiYWxnIjoiUlMyNTYifQ.
        eyJzY29wZXMiOlsiaWRzYzpJRFNfQ09OTkVDVE9SX0FUVFJJQlVURVNfQUxMIl0sImF1ZCI6Imlkc2M6SURTX0NPTk5FQ1RPUlNfQUxMI
        mZXIuZGUiLCJuYmYiOjE2MzE3MDgyNTAsImlhdCI6MTYzMTcwODI1MCwianRpIjoiTlRBNE1qazJNVE0zTVRNMU5EWXpOalV4TlE9PSIs
        oiaWRzYzpCQVNFX1NFQ1VSSVRZX1BST0ZJTEUiLCJyZWZlcnJpbmdDb25uZWN0b3IiOiJodHRwOi8vaXNzdGJyb2tlci5kZW1vIiwiQHR
        HR0cHM6Ly93M2lkLm9yZy9pZHNhL2NvbnRleHRzL2NvbnRleHQuanNvbmxkIiwidHJhbnNwb3J0Q2VydHNTaGEyNTYiOiIwZWU3M2RjMG
        YWE2YWMxZGVmYzc5Zjk1MzZkOTQzIiwic3ViIjoiMTc6N0I6RUQ6MTg6NzM6RUI6RDA6NDc6NUM6QzM6MjU6NDk6NDc6MDQ6M0Q6QTI6O
```

Figure 18: Sample of App unregistration message

The response should be a MessageProcessedNotificationMessage without payload, as shown in Figure 19.

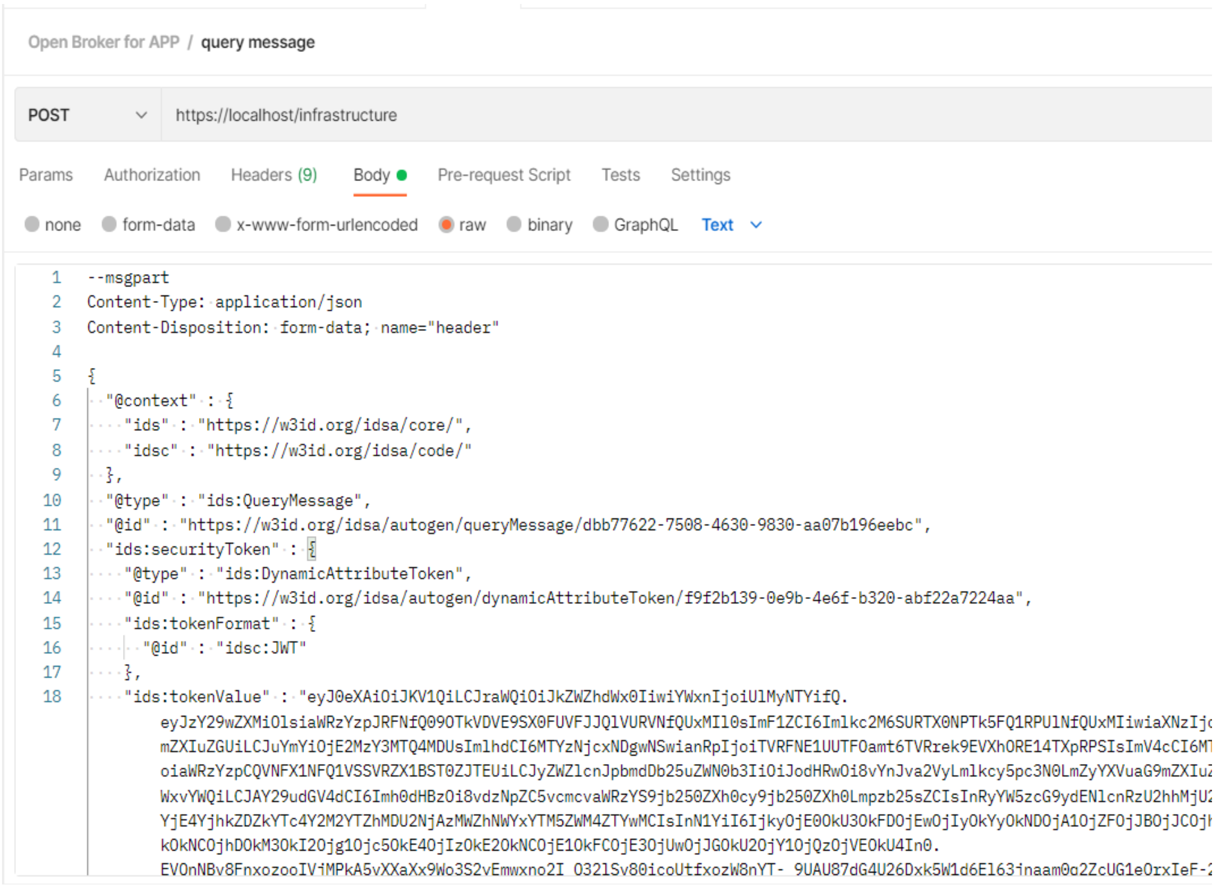


```
--3GJtqtvgeCEEqQy9fA01hLmOwnRbrhcX
Content-Disposition: form-data; name="header"
Content-Type: application/ld+json
Content-Length: 2247

{
  "@context" : {
    "ids" : "https://w3id.org/idsa/core/",
    "idsc" : "https://w3id.org/idsa/code/"
  },
  "@type" : "ids:MessageProcessedNotificationMessage",
  "@id" : "https://w3id.org/idsa/autogen/messageProcessedNotificationMessage/7d656885-4540-4a5a-9d53-2b39a65c9664",
  "ids:securityToken" : {
    "@type" : "ids:DynamicAttributeToken",
    "@id" : "https://w3id.org/idsa/autogen/dynamicAttributeToken/32615e1f-cb4c-41f5-873e-0ea0e95d2fa4",
    "ids:tokenFormat" : {
      "@id" : "https://w3id.org/idsa/code/JWT"
    },
    "ids:tokenValue" : "eyJ0eXAiOiJKV1QiLCJraWQiOiJkZWZhdWx0IiwiYWxnIjoiUlMyNTYifQ.
        eyJzY29wZXMiOlsiaWRzYzpJRFNfQ09OTkVDVE9SX0FUVFJJQlVURVNfQUxMIl0sImF1ZCI6Imlkc2M6SURTX0NPTk5FQ1RPUlNfQUxMIiwiaX
        mZXIuZGUiLCJuYmYiOjE2MzY3MTQ3NDYsImlhdCI6MTYzNjcxNDc0NiwianRpIjoiTVRjNU5UY3dNelV4TWpJNE1EYzVNVFk1TkRBPSIsImV4c
        oiaWRzYzpCQVNFX1NFQ1VSSVRZX1BST0ZJTEUiLCJyZWZlcnJpbmdDb25uZWN0b3IiOiJodHRwOi8vYnJva2VyLmlkcy5pc3N0LmZyYXVuaG9m
```

Figure 19: Sample response of App unregistration message

2.1.8 Query

The multipart message header should be QueryMessage, and the payload should be a SPARQL query.



Open Broker for APP / query message

POST https://localhost/infrastructure

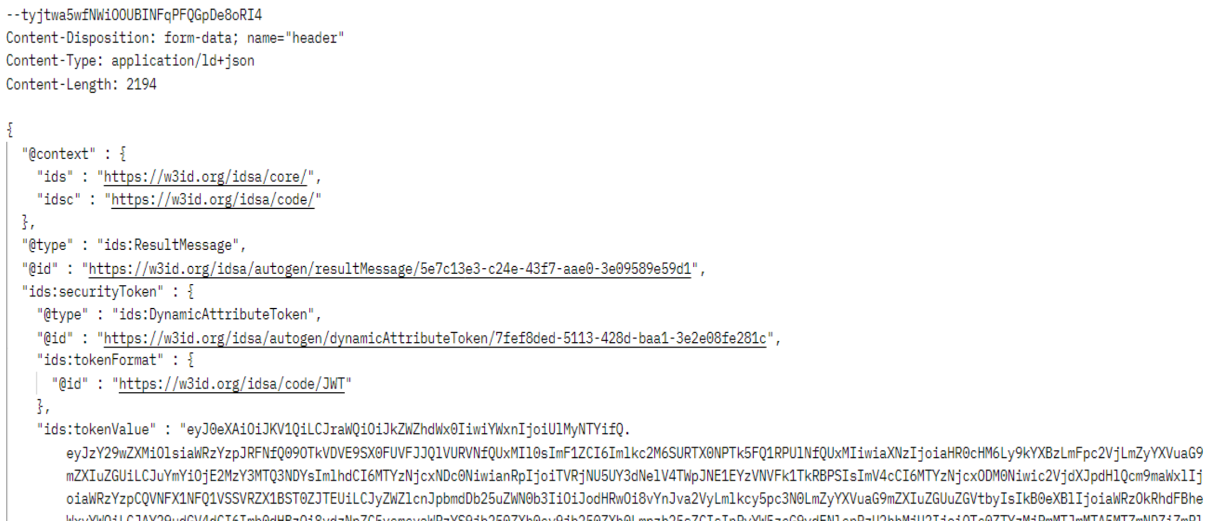
Params | Authorization | Headers (9) | Body ● | Pre-request Script | Tests | Settings

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL Text

```
--msgpart
Content-Type: application/json
Content-Disposition: form-data; name="header"

{
  "@context" : {
    "ids" : "https://w3id.org/idsa/core/",
    "idsc" : "https://w3id.org/idsa/code/"
  },
  "@type" : "ids:QueryMessage",
  "@id" : "https://w3id.org/idsa/autogen/queryMessage/dbb77622-7508-4630-9830-aa07b196eebc",
  "ids:securityToken" : {
    "@type" : "ids:DynamicAttributeToken",
    "@id" : "https://w3id.org/idsa/autogen/dynamicAttributeToken/f9f2b139-0e9b-4e6f-b320-abf22a7224aa",
    "ids:tokenFormat" : {
      "@id" : "idsc:JWT"
    },
    "ids:tokenValue" : "eyJ0eXAiOiJKV1QiLCJraWQiOiJkZWZhdWx0IiwiYWxnIjoiUlMyNTYifQ.
        eyJzY29wZXMiOlsiaWRzYzpJRFNfQ090TkVDVE9SX0FUVFJJQlVURVNfQUxMIl0sImF1ZCI6Imlkc2M6SURTX0NPTk5FQ1RPUlNfQUxMIiwiaXNzIjo
        mZXIuZGUiLCJuYmYiOjE2MzY3MTQ4MDUsImlhdCI6MTYzNjcxNDgwNSwianRpIjoiTVRFNE1UUTFOamt6TVRrek9EVXhORE14TXpRPSIsImV4cCI6MT
        oiaWRzYzpCQVNFX1NFQ1VSSVRZX1BST0ZJTEUiLCJyZWZlcnJpbmdDb25uZWN0b3IiOiJodHRwOi8vYnJva2VyLmlkcy5pc3N0LmZyYXVuaG9mZXIuZ
        WxvYWQiLCJAY29udGV4dCI6Imh0dHBzOi8vdzNpZC5vcmcvaWRzYS9jb250ZXh0cy9jb250ZXh0Lmpzb25sZCIsInRyYW5zcG9ydENlcnRzU2hhMjU
        YjE4YjhkZDZkYTc4Y2M2YTZhMDU2NjAzMWZhNWYxYTM5ZWM4ZTYwMCIsInN1YiI6IjkyOjE0OkU3OkFDOjEwOjIyOkYyOkNDOjA1OjZFOjBBOjJCOj
        kOkNCOjhDOkM3OkI2Ojg1Ojc5OkE4OjIzOkE2OkNCOjE1OkFCOjE3OjUwOjJGOkU2OjY1OjQzOjVEOkU4In0.
        EVOnNBv8FnxozooIViMPkA5vXXaXx9Wo3S2vEmwxno2I_0321Sv80icoUtfxozW8nYT-_9UAU87dG4U26Dxk5W1d6El63inaam0g2ZcUG1eOrxIeF-
```

Figure 20: Sample SPARQL query

The Response should be a ResultMessage, and the payload should be the result of the SPARQL query, as shown in Figure 21.

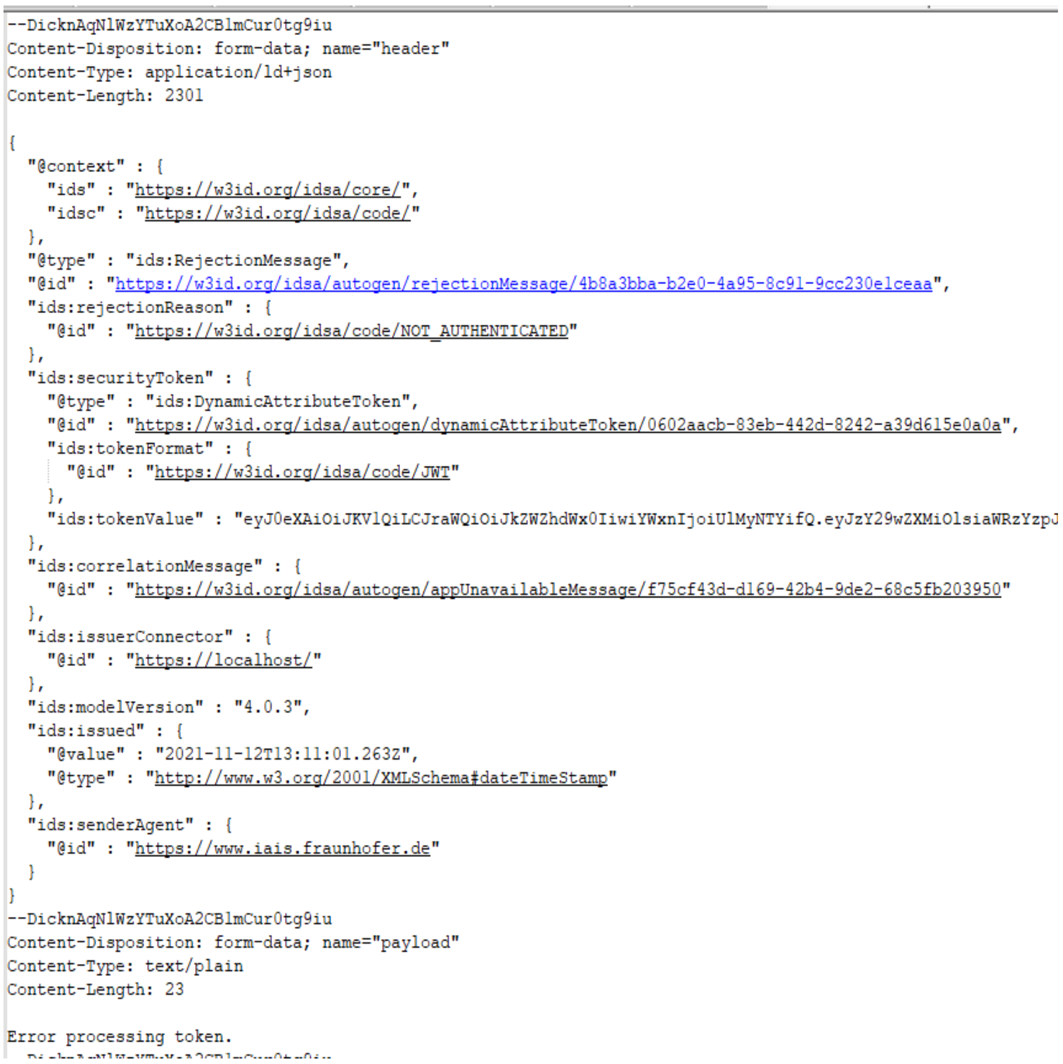


```
--tyjtwa5wfNWiOOUBINFqPFQGpDe8oRI4
Content-Disposition: form-data; name="header"
Content-Type: application/ld+json
Content-Length: 2194

{
  "@context" : {
    "ids" : "https://w3id.org/idsa/core/",
    "idsc" : "https://w3id.org/idsa/code/"
  },
  "@type" : "ids:ResultMessage",
  "@id" : "https://w3id.org/idsa/autogen/resultMessage/5e7c13e3-c24e-43f7-aae0-3e09589e59d1",
  "ids:securityToken" : {
    "@type" : "ids:DynamicAttributeToken",
    "@id" : "https://w3id.org/idsa/autogen/dynamicAttributeToken/7fef8ded-5113-428d-baa1-3e2e08fe281c",
    "ids:tokenFormat" : {
      "@id" : "https://w3id.org/idsa/code/JWT"
    },
    "ids:tokenValue" : "eyJ0eXAiOiJKV1QiLCJraWQiOiJkZWZhdWx0IiwiYWxnIjoiUlMyNTYifQ.
        eyJzY29wZXMiOlsiaWRzYzpJRFNfQ090TkVDVE9SX0FUVFJJQlVURVNfQUxMIl0sImF1ZCI6Imlkc2M6SURTX0NPTk5FQ1RPUlNfQUxMIiwiaXNzIjoiaHR0cHM6Ly9kYXBzLmFpc2VjLmZyYXVuaG9
        mZXIuZGUiLCJuYmYiOjE2MzY3MTQ3NDYsImlhdCI6MTYzNjcxNDc0NiwianRpIjoiTVRjNU5UY3dNelV4TWpJNE1EYzVNVFk1TkRBPSIsImV4cCI6MTYzNjcxODM0Niwic2VjdXJpdHlQcm9maWxlIj
        oiaWRzYzpCQVNFX1NFQ1VSSVRZX1BST0ZJTEUiLCJyZWZlcnJpbmdDb25uZWN0b3IiOiJodHRwOi8vYnJva2VyLmlkcy5pc3N0LmZyYXVuaG9mZXIuZGUuZGVtbyIsIkB0eXBlIjoiaWRzOkRhdFBhe
```

Figure 21: Sample response of the SPARQL Query

2.1.9 Rejection

Rejection messages are specialized response messages that notify the sender of a message (i.e., The connectors) that the processing of this message has failed. Table 1 shows the list of possible Rejection message codes.
Below is the example of a rejection message which says, "Error processing the token". This indicates that the DAT token used to send the request message to the Metadata Registry is invalid.



```
--DicknAqNlWzYTuXoA2CBlmCur0tg9iu
Content-Disposition: form-data; name="header"
Content-Type: application/ld+json
Content-Length: 2301

{
  "@context" : {
    "ids" : "https://w3id.org/idsa/core/",
    "idsc" : "https://w3id.org/idsa/code/"
  },
  "@type" : "ids:RejectionMessage",
  "@id" : "https://w3id.org/idsa/autogen/rejectionMessage/4b8a3bba-b2e0-4a95-8c91-9cc230e1ceaa",
  "ids:rejectionReason" : {
    "@id" : "https://w3id.org/idsa/code/NOT_AUTHENTICATED"
  },
  "ids:securityToken" : {
    "@type" : "ids:DynamicAttributeToken",
    "@id" : "https://w3id.org/idsa/autogen/dynamicAttributeToken/0602aacb-83eb-442d-8242-a39d615e0a0a",
    "ids:tokenFormat" : {
      "@id" : "https://w3id.org/idsa/code/JWT"
    },
    "ids:tokenValue" : "eyJ0eXAiOiJKV1QiLCJraWQiOiJkZWZhdWx0IiwiYWxnIjoiUlMyNTYifQ.eyJzY29wZXMiOlsiaWRzYzpJ
  },
  "ids:correlationMessage" : {
    "@id" : "https://w3id.org/idsa/autogen/appUnavailableMessage/f75cf43d-d169-42b4-9de2-68c5fb203950"
  },
  "ids:issuerConnector" : {
    "@id" : "https://localhost/"
  },
  "ids:modelVersion" : "4.0.3",
  "ids:issued" : {
    "@value" : "2021-11-12T13:11:01.263Z",
    "@type" : "http://www.w3.org/2001/XMLSchema#dateTimeStamp"
  },
  "ids:senderAgent" : {
    "@id" : "https://www.iais.fraunhofer.de"
  }
}
--DicknAqNlWzYTuXoA2CBlmCur0tg9iu
Content-Disposition: form-data; name="payload"
Content-Type: text/plain
Content-Length: 23

Error processing token.
```

Figure 22: Sample rejection message

2.1.10 REST Endpoints

The Metadata Registry facilitates optional REST endpoints also. The change in these Endpoints is that we put the mandatory fields of the multipart header to the header of the REST request. These mandatory fields include ids-securityToken, ids-senderAgent, ids-modelVersion, ids-issued, ids-issuerConnector, etc., and keep the body of the multipart request in the body of the REST request.

2.2 Deployment of Metadata Registry

The code of the Open-source Metadata Registry is hosted in the PLATOON Git repository: https://github.com/PLATOONProject/Metadata-Registry
The development environment of Open-source Broker consists of two main components:

- Fuseki triple store: The database storing all the metadata
- Metadata Registry: The core component

2.2.1 Prerequisites

Prerequisite to run the Metadata Registry:

- Docker
- Docker Compose
- Java
- Maven
- OpenSSL

2.2.2 Structure of Metadata Registry

Docker images are created for each Metadata Registry component and run the corresponding docker containers in the same docker-compose environment so that each component can communicate internally.
The Metadata Registry consists of three components:

- **broker-core:** The broker-core component is the main component of the Metadata Registry, a Java environment running the Maven package of our code. The broker-core component will take requests from broker-reverseporxy, handle requests using the same handlers we mentioned in Section 2.1, and use APIs that broker-fuseki provides to either read or write in the Fuseki triple store.
- **broker-fuseki:** The broker-fuseki component is an instance of the Fuseki triple store, which hosts all metadata of the Metadata Registry.
- **broker-reverseproxy:** The broker-reverseproxy is an NGINX reverse proxy instance, which hosts the certificate to provide a secure connection to the Metadata Registry. It also acts as the gateway to redirect requests to the broker-core component.

2.2.3 Creation of SSL Certificates

A valid X.509 certificate, signed by a trusted certification authority, is strongly recommended to avoid warnings about insecure HTTPS connections. The certificate needs to be of .crt format and must have the name server.crt. In case your certificate is of .pem format, it can be converted with the following commands, which require OpenSSL to be installed:

```
OpenSSL x509 -in mycert.pem -out server.crt
OpenSSL RSA -in mycert.pem -out server.key
mkdir cert
mv server.crt cert/
mv server.key cert/
```

2.2.4 Configuring the Docker-compose File

The docker-compose file is located in path docker/composefiles/broker-localhost. The most crucial part of adapting the configuration is to provide the correct location of the X.509 certificate in the broker-reverseproxy service. Assuming the location of the certificate is "/home/ids/cert," the corresponding configuration is:

```
services:
broker-reverseproxy:
image: registry.gitlab.cc-asp.fraunhofer.de/eis-ids/broker-open/reverseproxy
volumes:
- /home/ids/cert:/etc/cert/
[…]
```

2.2.4.1 Download Docker Images

All of the Metadata Registry docker images are currently hosted at the GitLab of Fraunhofer IAIS. No credentials are needed to download the images. The following command is for pulling all docker images (in path docker/composefiles/broker-localhost):

```
docker-compose pull
```

Note that the docker images will be hosted in a PLATOON image registry in the future and how to download the image may change afterward.

2.2.4.2 Run Application

To start up the Metadata Registry, run the following command inside the directory of the docker-compose.yml file (in path docker/composefiles/broker-localhost):

```
docker-compose up -d
```

This process can take several minutes to complete. You can test whether the Broker has successfully started by opening https://localhost. The result should be a JSON document, providing general metadata about the Metadata Registry.

2.2.4.3 Update

To update an existing installation of Metadata Registry, first, repeat the steps explained in Section 2.2.4.1. Containers can be either hot updated or restarted to apply the changes. To hot update a container, run the following command:

```
docker-compose up -d—no-deps—build <container name>
```

Alternatively, one can restart the entire service by running:

```
docker-compose down
docker-compose up –d
```

3 Clearing House

The Clearing House [3] acts as an intermediary in the IDS ecosystem as well as PLATOON Marketplace. This means that the Clearing House mediates between a Data Provider (DP) and a Data Consumer (DC), ensuring both parties meet their contractual obligations. Such as:

- The DP sharing data with the DC according to Usage Contracts and Data Usage Policies defined
- The DC using data according to Usage Contracts and Data Usage Policies defined and effecting payment to the DP as agreed.

For each data exchange transaction, the DP attaches metadata to the data requested by the DC, specifying data usage restrictions, pricing information, payment entitlement, time of validity, etc. This way, the DP can establish a Data Usage Policy as deemed appropriate, ensuring data sovereignty is guaranteed. The Clearing House in the Marketplace reduces risk and uncertainty on both sides as it is a trusted partner of both the DP and the DC. Since they don't know each other and/or haven't done data exchange transactions so far, the DP and DC may not trust each other in many cases. For example, if the DP agrees to provide data to the DC based on a Usage Contract, no one can verify the transaction. One party could not fulfill the contract, for example, by not delivering the data as agreed or providing data of poor quality on the DP side, or by misusing the data or failing with payment on the DC side. Such transactional risk can be mitigated by involving the Clearing House. There are two functionalities of the Clearing House:

- **Log Message:** Each party in a data exchange process logs their contractual obligation to the Clearing House. These logs are structured as Processes and have a Process ID.
- **Query Message:** Each party can query the Clearing House to ensure the contractual obligation is preserved.

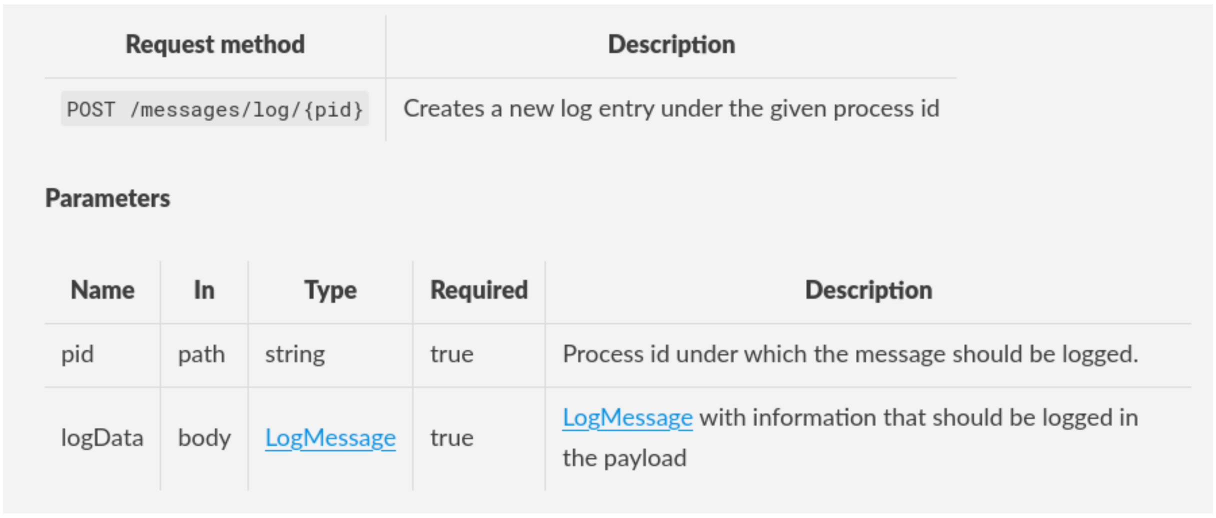
The Clearing House is developed by Fraunhofer AISEC and in PLATOON, it is hosted as provided.

3.1 Interaction With Clearing House

The IDS Clearing House structures its logs into processes and requires that all messages are logged under a process ID. Processes are meant to represent agreed-upon data exchanges that include the storage of the contract agreement and all messages that document the data exchange. Any IDS connector can create processes with a valid DAT issued by DAPS, who becomes the owner of the process. Only the owner of the process may read or write data in the process. All the possible response statuses of codes are provided in Table 1.

3.1.1 Logging messages

The information that should be logged in the Clearing House (logData) is sent in the payload of a LogMessage, as shown in Figure 23. The log entry stored in the Clearing House consists of the payload and meta-data from the LogMessage and stored under the given PID. This IDS message returns the newly created ID of the log entry together with status information.



Request method	Description
`POST /messages/log/{pid}`	Creates a new log entry under the given process id

Parameters

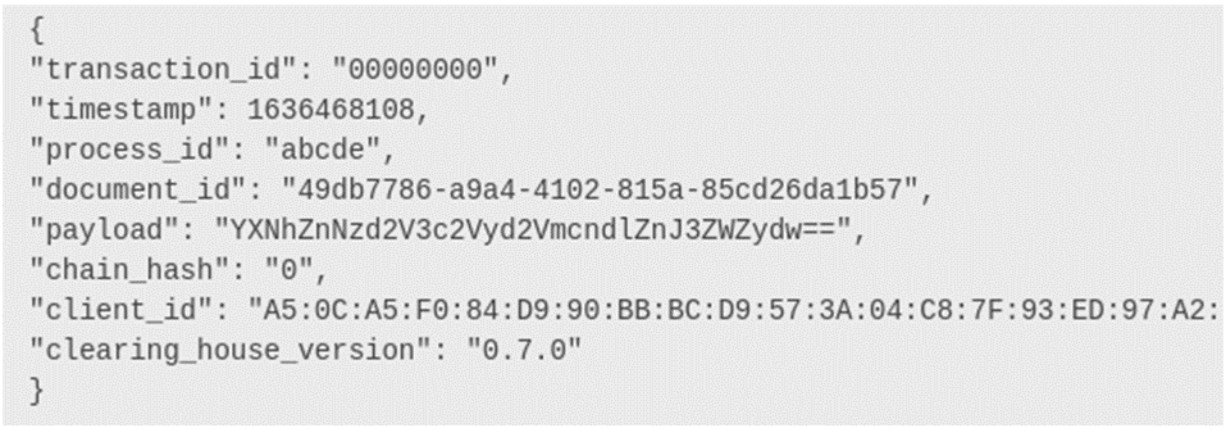
Name	In	Type	Required	Description
pid	path	string	true	Process id under which the message should be logged.
logData	body	LogMessage	true	LogMessage with information that should be logged in the payload

Figure 23: Request method of logging message in the Clearing House

The communication between an IDS connector and the Clearing House can be described on the application level with the flow of Infomodel messages between both entities. The IDS connector sends a LogMessage to the Clearing House to log information in the Clearing House, as provided below:

```
POST /logs/messages/test-process HTTP/1.1
Host: ch-ids.aisec.fraunhofer.de
Content-Type: multipart/form-data; boundary=X-TEST-REQUEST-BOUNDARY
Accept: */*

--X-TEST-REQUEST-BOUNDARY
Content-Disposition: form-data; name="header"
Content-Type: application/json
{
  "@context" : {
    "ids" : "https://w3id.org/idsa/core/",
    "idsc" : "https://w3id.org/idsa/code/"
  },
  "@type" : "ids:LogMessage",
  "@id" : "https://w3id.org/idsa/autogen/logMessage/c6c15a90-7799-4aa1-ac21-9323b87a7xv9",
  "ids:securityToken" : {
    "@type" : "ids:DynamicAttributeToken",
    "@id"     :      "https://w3id.org/idsa/autogen/dynamicAttributeToken/6378asd9-480d-80df-
c5cb02e4e260",
    "ids:tokenFormat" : {
      "@id" : "idsc:JWT"
    },
    "ids:tokenValue" : "eyJ0eXAiOiJKV1QiLCJraWQiOiJkZWZhdWx0IiwiYWxnIjoi....."
  },
  "ids:senderAgent" : "http://example.org",
  "ids:modelVersion" : "4.0.0",
  "ids:issued" : {
    "@value" : "2020-12-14T08:57:57.057+01:00",
    "@type" : "http://www.w3.org/2001/XMLSchema#dateTimeStamp"
  },
  "ids:issuerConnector" : {
    "@id" : "https://companyA.com/connector/59a68243-dd96-4c8d-88a9-0f0e03e13b1b"
  }
}
--X-TEST-REQUEST-BOUNDARY
Content-Disposition: form-data; name="payload"
Content-Type: application/json
{
  "@context" : "https://w3id.org/idsa/contexts/context.jsonld",
  "@type" : "ids:ConnectorUpdateMessage",
  "id"  :  "http://industrialdataspace.org/connectorAvailableMessage/34d761cf-5ca4-4a77-a7f4-
b14d8f75636a",
  "issued" : "2019-12-02T08:25:08.245Z",
  "modelVersion" : "4.0.0",
  "issuerConnector" : "https://companyA.com/connector/59a68243-dd96-4c8d-88a9-0f0e03e13b1b",
  "securityToken" : {
    "@type" : "ids:DynamicAttributeToken",
    "tokenFormat" : "https://w3id.org/idsa/code/tokenformat/JWT",
    "tokenValue" : "eyJhbGciOiJSUzI1NiIsInR5cCI..."
}

--X-TEST-REQUEST-BOUNDARY--
```

The response comes in a JWT format containing both data and metadata stored so that the recipient can easily verify that the signature on the JWT is veiled. The decode version of the JWT is shown in Figure 24. This indicates that the data with the given metadata was indeed logged in the Clearing House.



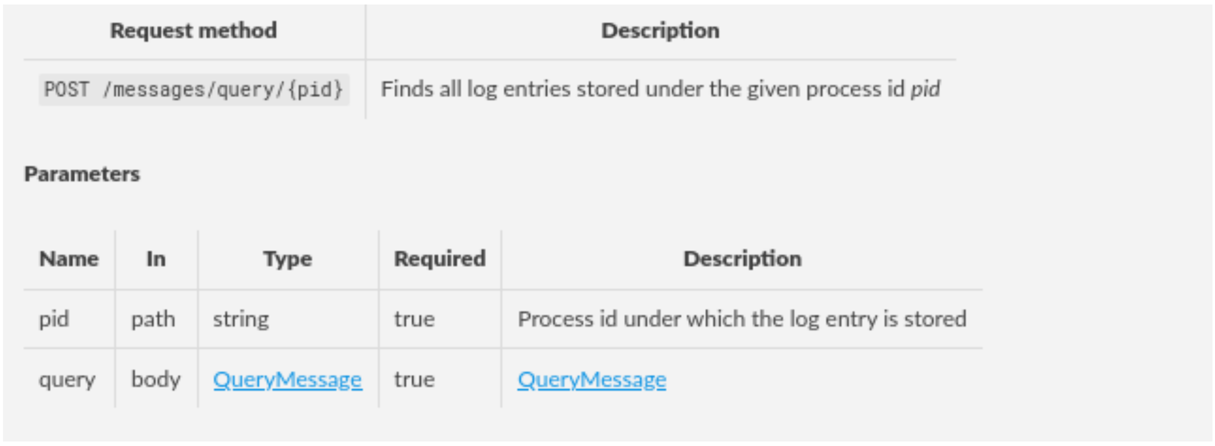
```
{
"transaction_id": "00000000",
"timestamp": 1636468108,
"process_id": "abcde",
"document_id": "49db7786-a9a4-4102-815a-85cd26da1b57",
"payload": "YXNhZnNzd2V3c2Vyd2VmcndlZnJ3ZWZydw==",
"chain_hash": "0",
"client_id": "A5:0C:A5:F0:84:D9:90:BB:BC:D9:57:3A:04:C8:7F:93:ED:97:A2:
"clearing_house_version": "0.7.0"
}
```

Figure 24: Sample decoded response from Log Message in the Clearing House

The *document_id* represents the log that should be used to query this specific log in the Clearing House. This ID will be used to query a message under a process as mentioned in Section 3.1.3. The *client_id* is the user ID of the client making the log request. It's taken from the DAT, so the CH logs whose certificate/key was used to get the DAT for the call. Theoretically, only the owner of the certificate/key should have been able to get the DAT. The *chain_hash* is for data immutability. More details on this will be provided in the next version of the deliverable.

3.1.2 Query all messages of a process

Retrieves all log entries stored under the given PID, as shown in Figure 25, in the Clearing House. The Clearing House answers the request with a ResultMessage that contains as the payload all log entries found. Each log entry is returned as a LogMessage, i.e., the payload of the ResultMessage contains a JSON array of LogMessage. The Connector receives a ResultMessage as a result from the Clearing House. The payload contains the found log entries.



Request method	Description
`POST /messages/query/{pid}`	Finds all log entries stored under the given process id *pid*

Parameters

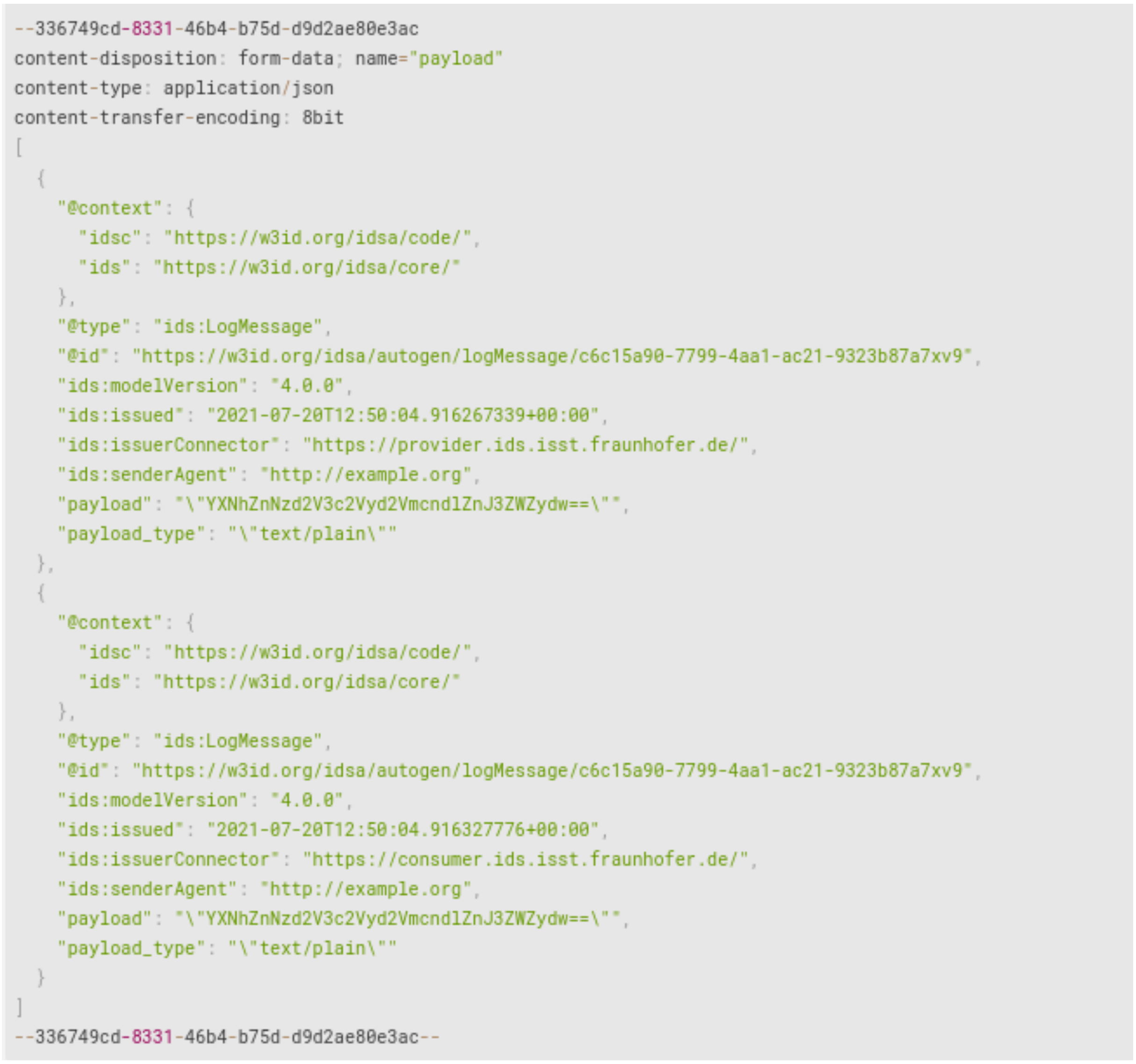
Name	In	Type	Required	Description
pid	path	string	true	Process id under which the log entry is stored
query	body	QueryMessage	true	QueryMessage

Figure 25: Request method of querying all messages of a process in the Clearing House

Infomodel messages do not model the payload of a message, but the Clearing House logs both the Infomodel messages (LogMessage) as meta-information and the payload as shown below:

```
HTTP/1.1 200 OK
Server: nginx/1.21.1
Date: Tue, 20 Jul 2021 12:50:05 GMT
Content-Type: multipart/form-data; boundary=336749cd-8331-46b4-b75d-d9d2ae80e3ac
Transfer-Encoding: chunked
Connection: keep-alive
Accept: */*
permissions-policy: interest-cohort=()
x-content-type-options: nosniff
x-frame-options: SAMEORIGIN

--336749cd-8331-46b4-b75d-d9d2ae80e3ac
content-disposition: form-data; name="header"
content-type: application/json; charset=UTF-8
content-transfer-encoding: 8bit
{
  "@context": {
    "ids": "https://w3id.org/idsa/core/",
    "idsc": "https://w3id.org/idsa/code/"
  },
  "@type": "ids:ResultMessage",
  "@id": "https://w3id.org/idsa/autogen/ResultMessage/cfe8bd4f-c8b5-42e4-aaae-1f594fa80719",
  "ids:issued": {
    "@value": "2021-07-20T12:50:04.916Z",
    "@type": "http://www.w3.org/2001/XMLSchema#dateTimeStamp"
  },
  "ids:issuerConnector": {
    "@id": "https://clearinghouse.aisec.fraunhofer.de/"
  },
  "ids:recipientConnector": [
    {
      "@id": "https://broker.ids.isst.fraunhofer.de/"
    }
  ],
  "ids:securityToken": {
    "@type": "ids:DynamicAttributeToken",
    "@id":     "https://w3id.org/idsa/autogen/dynamicAttributeToken/d9bd7b02-89ca-4e08-be3c-
f3c149d62a62",
    "ids:tokenValue": "eyJ0eXAiOiJKV1QiLCJraWQiOi...",
    "ids:tokenFormat": {
      "@id": "idsc:JWT"
    }
  },
  "ids:senderAgent": {
    "@id": "https://clearinghouse.aisec.fraunhofer.de"
  },
  "ids:recipientAgent": [
    {
      "@id": "http://example.org"
    }
  ],
  "ids:correlationMessage": {
    "@id": "https://w3id.org/idsa/autogen/logMessage/c6c15a90-7799-4aa1-ac21-9323b87a7xv9"
  },
  "ids:modelVersion": "4.0.0"
}

--336749cd-8331-46b4-b75d-d9d2ae80e3ac
content-disposition: form-data; name="payload"
content-type: application/json
content-transfer-encoding: 8bit
[
  {
    "@context": {
      "idsc": "https://w3id.org/idsa/code/",
      "ids": "https://w3id.org/idsa/core/"
    },
    "@type": "ids:LogMessage",
    "@id": "https://w3id.org/idsa/autogen/logMessage/c6c15a90-7799-4aa1-ac21-9323b87a7xv9",
    "ids:modelVersion": "4.0.0",
    "ids:issued": "2021-07-20T12:50:04.916267339+00:00",
    "ids:issuerConnector": "https://provider.ids.isst.fraunhofer.de/",
    "ids:senderAgent": "http://example.org",
    "payload": "\"YXNhZnNzd2V3c2Vyd2VmcndlZnJ3ZWZydw==\"",
    "payload_type": "\"text/plain\""
  },
  {
    "@context": {
      "idsc": "https://w3id.org/idsa/code/",
      "ids": "https://w3id.org/idsa/core/"
    },
    "@type": "ids:LogMessage",
    "@id": "https://w3id.org/idsa/autogen/logMessage/c6c15a90-7799-4aa1-ac21-9323b87a7xv9",
    "ids:modelVersion": "4.0.0",
    "ids:issued": "2021-07-20T12:50:04.916327776+00:00",
    "ids:issuerConnector": "https://consumer.ids.isst.fraunhofer.de/",
    "ids:senderAgent": "http://example.org",
    "payload": "\"YXNhZnNzd2V3c2Vyd2VmcndlZnJ3ZWZydw==\"",
    "payload_type": "\"text/plain\""
  }
]
--336749cd-8331-46b4-b75d-d9d2ae80e3ac--
```

The result of a query to the Clearing House will also return both meta-information and payload of the previously logged information. As stated above, the answer of the Clearing House to a QueryMessage is a ResultMessage, as shown in Figure 26, that contains an array of all log entries found. A LogMessage represents each log entry with two additional fields: payload and payload_type.

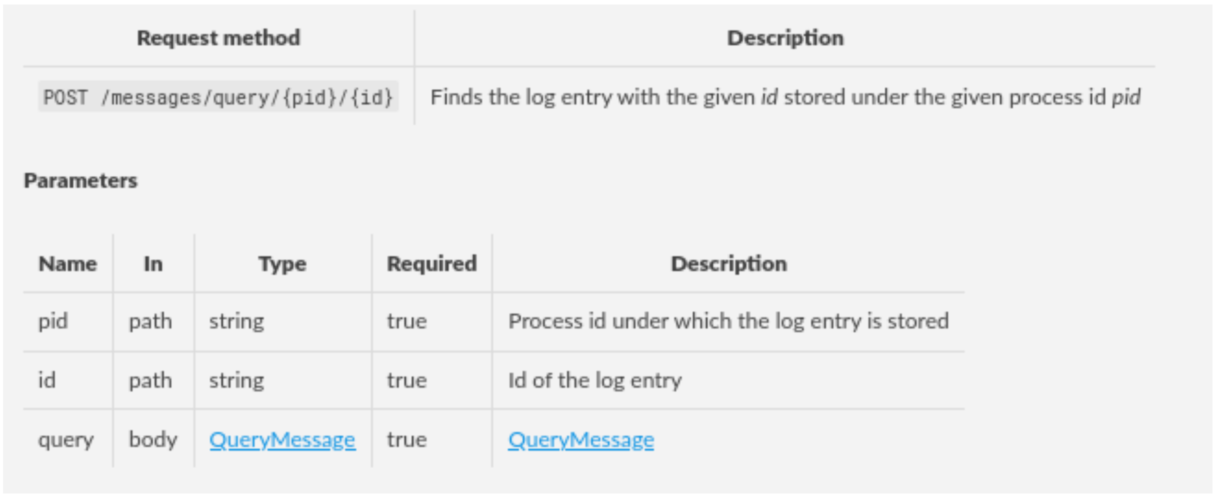


```
--336749cd-8331-46b4-b75d-d9d2ae80e3ac
content-disposition: form-data; name="payload"
content-type: application/json
content-transfer-encoding: 8bit
[
  {
    "@context": {
      "idsc": "https://w3id.org/idsa/code/",
      "ids": "https://w3id.org/idsa/core/"
    },
    "@type": "ids:LogMessage",
    "@id": "https://w3id.org/idsa/autogen/logMessage/c6c15a90-7799-4aa1-ac21-9323b87a7xv9",
    "ids:modelVersion": "4.0.0",
    "ids:issued": "2021-07-20T12:50:04.916267339+00:00",
    "ids:issuerConnector": "https://provider.ids.isst.fraunhofer.de/",
    "ids:senderAgent": "http://example.org",
    "payload": "\"YXNhZnNzd2V3c2Vyd2VmcndlZnJ3ZWZydw==\"",
    "payload_type": "\"text/plain\""
  },
  {
    "@context": {
      "idsc": "https://w3id.org/idsa/code/",
      "ids": "https://w3id.org/idsa/core/"
    },
    "@type": "ids:LogMessage",
    "@id": "https://w3id.org/idsa/autogen/logMessage/c6c15a90-7799-4aa1-ac21-9323b87a7xv9",
    "ids:modelVersion": "4.0.0",
    "ids:issued": "2021-07-20T12:50:04.916327776+00:00",
    "ids:issuerConnector": "https://consumer.ids.isst.fraunhofer.de/",
    "ids:senderAgent": "http://example.org",
    "payload": "\"YXNhZnNzd2V3c2Vyd2VmcndlZnJ3ZWZydw==\"",
    "payload_type": "\"text/plain\""
  }
]
--336749cd-8331-46b4-b75d-d9d2ae80e3ac--
```

Figure 26: Sample response message of querying all the messages under a PID in the Clearing House

3.1.3 Query a particular messages of a process

This IDS message retrieves the log entry stored under the given ID and PID, as shown in Figure 27, in the Clearing House. The Clearing House answers the request with a ResultMessage that contains as the payload the log entry found. The log entry is returned as a LogMessage, i.e., the payload of the ResultMessage contains a LogMessage. ID in this message request is acquired from the *document_id* as mentioned in the decoded response discussed in Section 3.1.1.



Request method	Description
`POST /messages/query/{pid}/{id}`	Finds the log entry with the given *id* stored under the given process id *pid*

Parameters

Name	In	Type	Required	Description
pid	path	string	true	Process id under which the log entry is stored
id	path	string	true	Id of the log entry
query	body	QueryMessage	true	QueryMessage

Figure 27: Request method of querying a message of a process in the Clearing House

Querying a message of a process behaves similarly to Section 3.1.2. Instead of a list of all the logs under a PID, the ResultMessage shows the specific log with respect to the ID.

3.2 Deployment

The code of Open-source Clearing House is available in the IDSA Git repository: https://github.com/International-Data-Spaces-Association/ids-clearing-house-service

3.2.1 Prerequisites

Prerequisite to host Clearing House:

- OpenSSL
- MongoDB
- Docker

Additionally, the Clearing House App depends on two microservices from the Clearing House Core[15]:

- **Document API:** Responsible for storing the data.
- **Keyring API:** Provides cryptographic support for encryption and decryption of the stored data.

The Clearing House Service API requires a Trusted Connector for the deployment of Clearing House.

3.2.2 Clearing House Core Configuration

Document API

The Document API is responsible for storing the data and performs basic encryption and decryption, for which it depends on the Keyring API. It is configured using the configuration file Rocket.toml, which must specify a set of configuration options, such as the correct URLs of the database and other service APIs:

- daps_api_url: Specifies the URL of the DAPS Service. Required to validate DAPS token
- keyring_api_url: Specifies the URL of the Keyring API

[15] https://github.com/International-Data-Spaces-Association/ids-clearing-house-core

- database_url: Specifies the URL of the database to store the encrypted documents. Currently, only MongoDB is supported, so URL is supposed to be MongoDB://<host>:<port>
- clear_db: true or false indicates if the database should be cleared when starting the Service API or not. If true, a restart will wipe the database! Creating the Service API on a clean database will initialize the database.

When starting the Clearing House Service API, it also needs the following environment variables set:

- API_LOG_LEVEL: Allowed log levels are: Off, Error, Warn, Info, Debug, Trace

Keyring API

The Keyring API is responsible for creating keys and the actual encryption and decryption of stored data. It is configured using the configuration file Rocket.toml, which must specify a set of configuration options, such as the correct URLs of the database and other service APIs:

- daps_api_url: Specifies the URL of the DAPS Service. Required to validate DAPS token
- database_url: Specifies the URL of the database to store document types and the master key. Currently, only MongoDB is supported, so URL is supposed to be MongoDB://<host>:<port>
- clear_db: true or false indicates if the database should be cleared when starting the Service API or not. If true, a restart will wipe the database! Creating the Service API on a clean database will initialize the database.

When starting the Clearing House Service API, it also needs the following environment variables set:

- API_LOG_LEVEL: Allowed log levels are: Off, Error, Warn, Info, Debug, Trace

The Keyring API requires that its database contains the acceptable document types. Currently, only the IDS_MESSAGE type is supported and must be present in the database for the Keyring API to function correctly. The database will be populated with an initial document type that needs to be located in init_db/default_doc_type.json.

DAPS

Both Document API and Keyring API need to be able to validate the certificate used by the DAPS. If the DAPS uses a self-signed certificate, the certificate needs to be added in two places:

- /server/certs: The microservice will load certificates in this folder in the container and use them for validation. The certificate needs to be in DER format.
- /usr/local/share/ca-certificates: The microservice relies on OpenSSL for parts of the validation, and OpenSSL will not trust a self-signed certificate unless it was added in this folder and update-ca-certificates was called in the docker container. Once this is done, the container might need to be restarted.

If you use these dockerfiles and daps.aisec.fraunhofer.de as the DAPS, you only need to follow the first step.

Docker Containers

There are two types of dockerfiles:

- Simple builds (e.g., dockerfile) that require you to build the Service APIs yourself using Rust

- Multistage builds (e.g., dockerfile) that have a stage for building the rust code

To build the containers, check out the repository, and in the main directory, execute

docker build -f docker/<dockerfile> . -t <image-name>

Before using docker run or docker-compose, please read the Configuration section of the Service API you are trying to run. All Containers built with the provided dockerfiles need two volumes:

- The configuration file Rocket.tomlis expected at /server/Rocket.toml
- The folder containing the daps certificate is expected at /server/certs

Containers of the Keyring API require an additional volume:

- /server/init_db needs to contain the default_doc_type.json

3.2.3 Clearing House Service Configuration

Once Document API and Keyring API are configured following the instruction mentioned in Section 3.2.2, the following configuration should be followed to install Clearing House Service.

Clearing House App Configuration

The Clearing House App is configured using the configuration file Rocket.toml, which must specify a set of configuration options, such as the correct URLs of the database and other service APIs:

- daps_api_url: Specifies the URL of the DAPS Service. Required to validate DAPS token
- keyring_api_url: Specifies the URL of the Keyring API
- document_api_url: Specifies the URL of the Document API
- database_url: Specifies the URL of the database to store process information. Currently, only MongoDB is supported, so URL is supposed to be MongoDB://<host>:<port>
- infomodel_version: Specifies which The Clearing House uses the version of the InfoModel. Currently: 4.0.0
- connector_name: Needed for IDS Messages as specified by the InfoModel
- server_agent: Needed for IDS Messages specified by the InfoModel
- clear_db: true or false indicates if the database should be cleared when starting the Service API or not. If true, a restart will wipe the database! Starting the Service API on a clean database will initialize the database.
- Signing key: Location of the private key (DER format) used for signing the Receipts. Clearing House uses the PS512 algorithm for signing.

When starting the Clearing House Service API, it also needs the following environment variables set:

- API_LOG_LEVEL: Allowed log levels are: Off, Error, Warn, Info, Debug, Trace

Signing Key

The Clearing House API sends a signed receipt as a response to a logging request. The key can be created using OpenSSL:

OpenSSL genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:4096 -outform der -out private_key.der

Please note that the Clearing House requires the key to be in DER format. It must be available to the Clearing House App under the path configured in Rocket.toml, e.g.,/server/keys/private_key.der.

DAPS
The Clearing House needs to be able to validate the certificate used by the DAPS. If the DAPS uses a self-signed certificate, the certificate needs to be added in two places:

- /server/certs: The Clearing House App will load certificates in this folder and use them for validation. The certificate needs to be in DER format.
- /usr/local/share/ca-certificates: The Clearing House App relies on OpenSSL for parts of the validation, and OpenSSL will not trust a self-signed certificate unless it was added in this folder and update-ca-certificates was called. Once this is done, the container might need to be restarted.

If you use these dockerfiles and daps.aisec.fraunhofer.de as the DAPS, you only need to follow the first step.

Docker Containers
The Clearing House App can be built using Dockerfiles that are located here. There are two types of dockerfiles:

- Simple builds (e.g., dockerfile) that require you to build the Clearing House APIs yourself using Rust
- Multistage builds (e.g., dockerfile) that have a stage for building the rust code

To build the containers, check out the repository, and in the main directory, execute

docker build -f docker/<dockerfile> . -t <image-name>

Please read the Clearing House App Configuration section before using docker run or docker-compose. Containers built with the provided dockerfiles need three volumes:

- The configuration file Rocket.tomlis expected at /server/Rocket.toml
- The folder containing the signing key needs to match the path configured for the signing key in Rocket.toml, e.g.,/sever/keys
- The folder containing the daps certificate is expected at /server/certs

The Clearing House Processors are not run as docker containers. The Clearing House Processors are needed to configure the Trusted Connector.

4 Marketplace User Interface (UI)

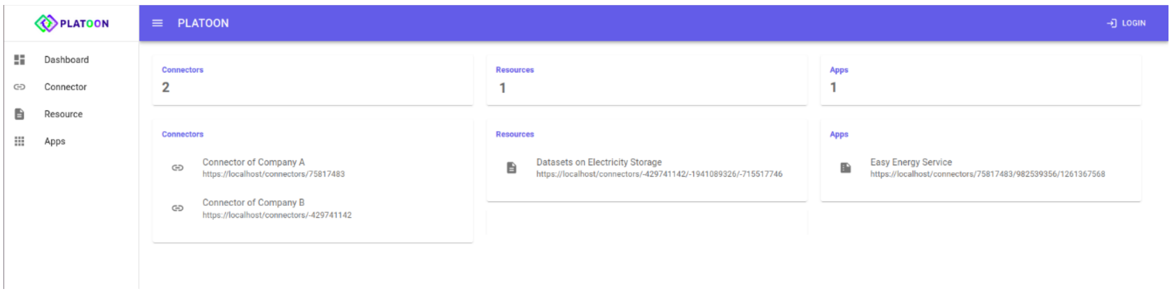
In PLATOON Marketplace, User Interface will present the metadata information of registered connectors, Resources, and Apps. The connectors register themselves and the resources and Apps it offers with the Metadata Registry (through the multipart/REST endpoints). The UI will fetch the information stored in the Fuseki Database by using SPARQL queries. An important point to note is that the UI will not provide the functionality to register connectors. The Connector should register itself and its resources by forming the appropriate IDS messages and sending them to the proper endpoints of the Metadata Registry. Once the Connectors registers itself along with the resources it has to offer, it will be displayed in the UI.

The initial idea was to develop a CKAN-based data market. However, CKAN can't handle IDS messages. There are no RDF/triple store connections. Furthermore, the amount of effort to overcome these issues is calculated to be high. Thus, now PLATOON will offer a Marketplace UI with the Metadata Registry (first version), illustrated in this deliverable. The potential second version of the UI will have more functionality, address the usage of FIWARE Business Framework and the API Ecosystem, and incorporate the Catalogues of Datasets and Data Analytics tools.

4.1 Interaction with Marketplace UI

The User Interface of the Marketplace comprises of four windows:

- **Dashboard:** The Dashboard shows the summary of all the registered Connectors, Resources, and Apps (services) in the Metadata Registry. As shown in Figure 28, the UI reflects two registered Connectors from Company A and Company B. One Resource and one App (metadata) are stored in the UI. With respect to the use-case mentioned in Section 1.2, Company A, acting as a Data Provider, has registered the dataset's metadata as Resource in the Metadata Registry. Also, Company A has one offered service called "Easy Energy Service."



Connectors
2

Resources
1

Apps
1

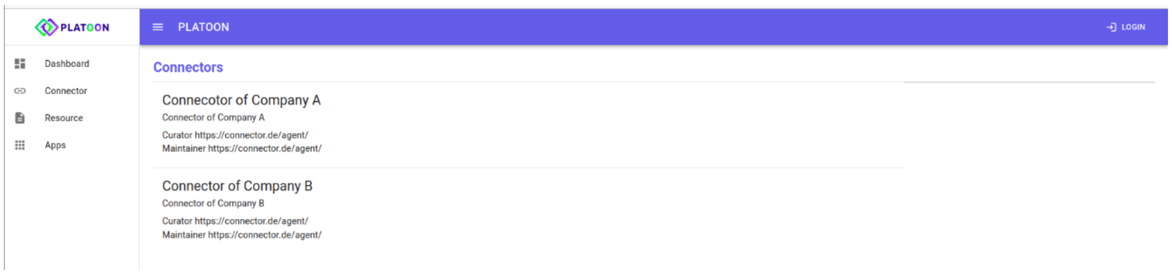
Connectors
Connector of Company A
https://localhost/connectors/75817483
Connector of Company B
https://localhost/connectors/-429741142

Resources
Datasets on Electricity Storage
https://localhost/connectors/-429741142/-1941089326/-715517746

Apps
Easy Energy Service
https://localhost/connectors/75817483/982539356/1261367568

Figure 28: Dashboard of the UI

- **Connector:** This window shows the detailed list of all the Connectors in the UI, as shown in Figure 29.



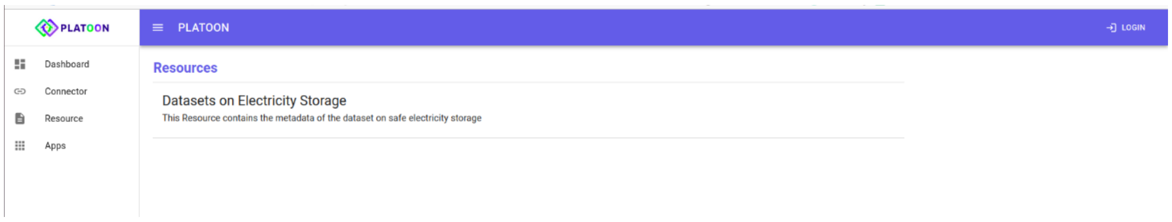
Connectors

Connecotor of Company A
Connector of Company A
Curator https://connector.de/agent/
Maintainer https://connector.de/agent/

Connector of Company B
Connector of Company B
Curator https://connector.de/agent/
Maintainer https://connector.de/agent/

Figure 29: List of Connectors in the UI

- **Resource:** This window shows the list of all the Resources present in the UI, as shown in Figure 30.

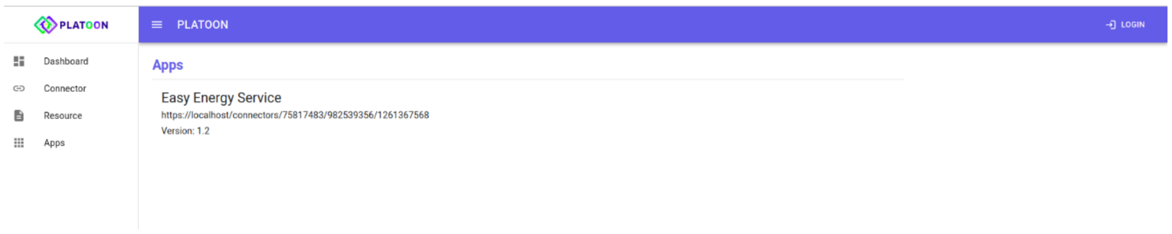


Resources

Datasets on Electricity Storage
This Resource contains the metadata of the dataset on safe electricity storage

Figure 30: List of Resources in the UI

- App: This window shows the list of the Metadata of the Apps in the UI, as shown in Figure 31.



Apps

Easy Energy Service
https://localhost/connectors/75817483/982539356/1261367568
Version: 1.2

Figure 31: List of the Apps(metadata) in the UI

4.2 Deployment

The GitHub repository of the UI is here: https://github.com/PLATOONProject/Marketplace-UI

4.2.1 Prerequisites

- NodeJS
- Metadata Registry

4.2.2 Configuration

Steps to configure the UI[16]:

- The Metadata Registry[17] needs to be installed.
- Navigate to ../src/urlConfig.js and set the link to the Fuseki server of the Metadata Registry. If the URL of the Fuseki server of the Metadata Registry is https://localhost/fuseki, then the JSON file will look like this:

[16] The current version of the User Interface is not dockerized yet. Thus, this deliverable includes the developer's guide, in case the reader needs to test the UI. For the latest installation guide, please refer to the README file.
[17] https://github.com/PLATOONProject/Metadata-Registry

```
export const baseURL = "https://localhost/fuseki"
```

4.2.3 Run Application

- Navigated to the main folder
- Execute the following command
  - npm install
  - npm start

Once the UI is running, all the interaction between a Connector and the Metadata Registry will be reflected in the UI.

5 Vocabulary Provider

IDS Vocabulary Provider is a central IDS component that manages all the vocabularies (ontologies, reference data models, metadata elements), which can annotate and describe datasets and data analytics tools. This includes the IDS information model and domain-specific vocabularies.

The IDS Vocabulary Provider plays an essential role within the PLATOON reference architecture defined in D2.1. It is the link between the Data Governance, Security, Privacy and Sovereignty layer (based on IDS reference architecture) and the Interoperability layer formed. The vocabulary provider provides direct Machine to Machine communication allowing to query and exchange metadata according to the PLATOON Data Models defined in D2.3 through the IDS connectors. In addition, the Vocabulary Provider has a Graphical User Interface (GUI), where users can manage vocabularies (upload/upgrade/delete), search for specific terms, visualize the vocabularies in a network graph and execute SPARQL queries.

The IDS Vocabulary Provider enhances the capabilities of the PLATOON Marketplace regarding interoperability. It allows the data/data analytics tools users/consumers to easily understand the data and data analytics tools published in the Marketplace, which facilitates the implementation and integration of these datasets and data analytics tools.

Currently, there is no open-source version of an IDS Vocabulary Provider. There is only an open-source Vocabulary Manager called Vocol [18], but the link to the IDS reference architecture is missing. The latter is offered as a service called Vocoreg [19], but it is not open source. Thus, PLATOON has developed a complete open-source IDS Vocabulary Provider based on Vocol and extended it by adding an extra layer to make it compatible with the IDS ecosystem. The design and implementation of the developed IDS Vocabulary Provider are explained in the following sections.

Figure 32 shows the high-level architecture for the IDS Vocabulary Provider as indicated in the IDS Reference Architecture [4].

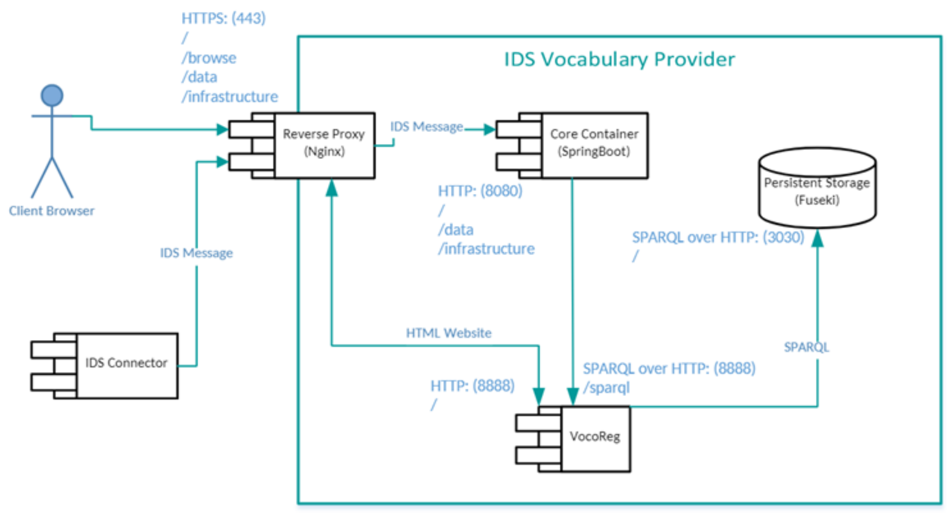


Figure 32: IDS Vocabulary High-level architecture according to IDS reference architecture

[18] https://vocol.iais.fraunhofer.de/
[19] https://www.vocoreg.com/

PLATOON has developed a complete open-source IDS Vocabulary based on this high-level architecture by reusing existing open-source components and extending them to add IDS capabilities.

For the Reverse proxy, a Multipart endpoint was developed using Apache Tomcat[20]. Apache Tomcat is a free and open-source implementation of the Jakarta Servlet, Jakarta Expression Language, and WebSocket technologies [21]. Apache Tomcat provides a "pure Java" HTTP web server environment in which Java code can run.

For the Persistent storage, Apache Jena Fuseki [22] was used. Apache Jena Fuseki is a SPARQL server. It can run as an operating system service, a Java web application (WAR file), and a standalone server. Fuseki comes in two forms, a single system, "webapp", combined with a UI for admin and query, and as "main", a server suitable to run as part of a larger deployment, including with Docker or running embedded. In this case, the latter was used, and it was embedded into a Docker. Fuseki provides the SPARQL 1.1 protocols for query and update as well as the SPARQL Graph Store protocol. Fuseki is tightly integrated with TDB to provide a robust, transactional persistent storage layer and incorporates Jena text query.

For the Vocabulary Management, instead of Vocoreg, VoCol [23] was used. VoCol is open-source software that allows managing (upload/modify/remove) ontologies using version control systems such as Git and repository hosting platforms like Github. VoCol provides a human-readable presentation of the vocabularies by means of a user-friendly Graphical User Interface (GUI). The GUI allows one to navigate through the metadata of the ontologies. The different functionalities are explained in more detail in the following paragraphs.

Figure 33 shows the home screen for the PLATOON IDS Vocabulary Provider where you can select one of the existing ontologies in the vocabulary provider.

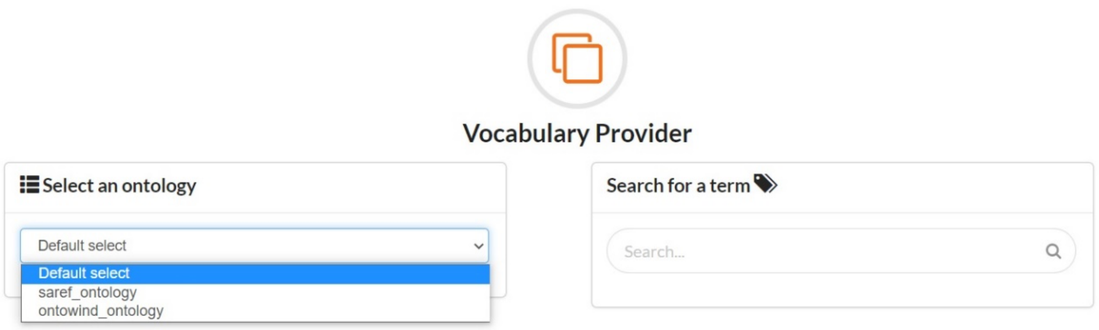
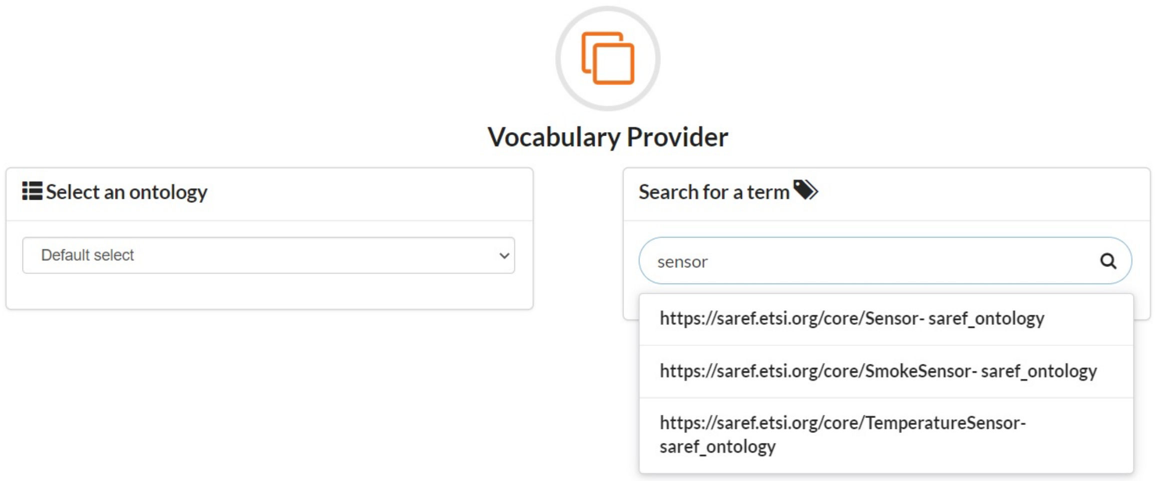


Figure 33:PLATOON IDS Vocabulary Provider - Home Screen - Select an Ontology

In addition, it allows searching for a specific term within the set of ontologies, as shown in Figure 34.

[20] http://tomcat.apache.org/
[21] https://projects.eclipse.org/projects/ee4j.jakartaee-platform
[22] https://jena.apache.org/documentation/fuseki2/
[23] https://github.com/vocol/vocol

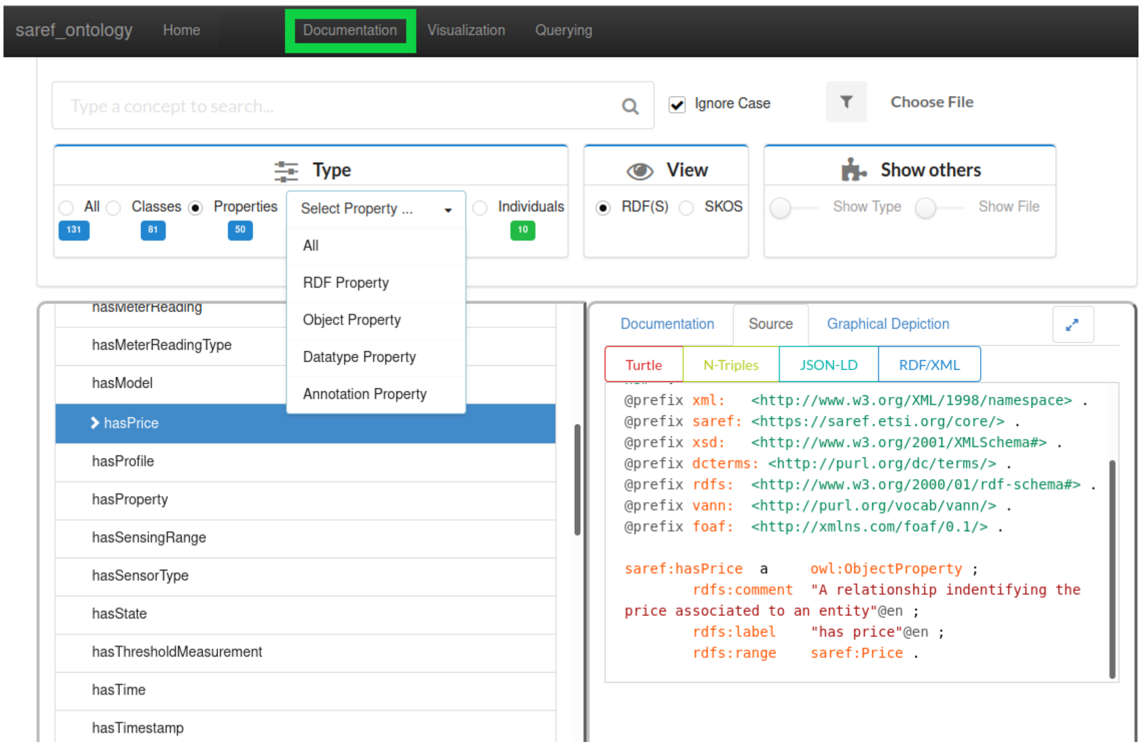


https://saref.etsi.org/core/Sensor- saref_ontology
https://saref.etsi.org/core/SmokeSensor- saref_ontology
https://saref.etsi.org/core/TemperatureSensor- saref_ontology

Figure 34: PLATOON IDS Vocabulary Provider - Home Screen – Search for a Term

Once the specific ontology has been selected, it allows to navigate through different options:

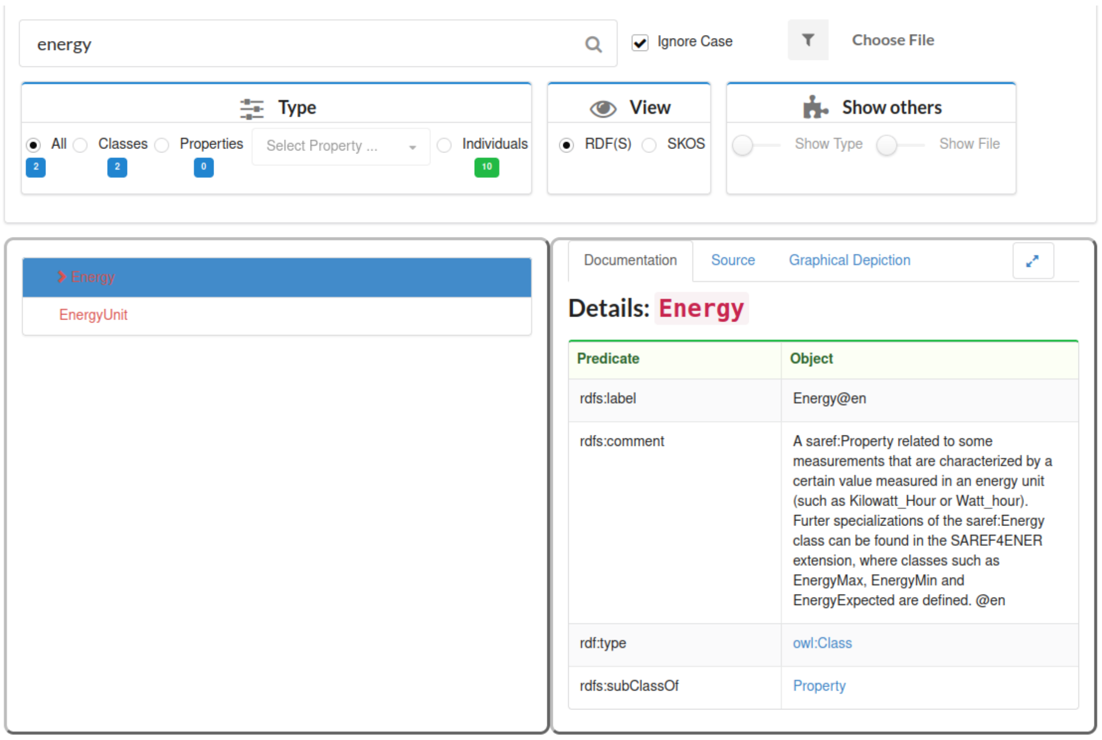
On the one hand, the "Documentation" tab allows to search and display detailed information of the different classes and properties. It allows selecting classes and properties and filter by property type. When selecting a specific class or property, more detailed information is shown on the right.



```
@prefix xml:   <http://www.w3.org/XML/1998/namespace> .
@prefix saref: <https://saref.etsi.org/core/> .
@prefix xsd:   <http://www.w3.org/2001/XMLSchema#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
@prefix vann:  <http://purl.org/vocab/vann/> .
@prefix foaf:  <http://xmlns.com/foaf/0.1/> .

saref:hasPrice  a     owl:ObjectProperty ;
        rdfs:comment  "A relationship indentifying the
price associated to an entity"@en ;
        rdfs:label    "has price"@en ;
        rdfs:range    saref:Price .
```

Figure 35: PLATOON IDS Vocabulary Provider - Documentation - Search for Classes and Properties

Furthermore, it allows searching for a specific term, as shown in Figure 36.



Details: Energy

Predicate	Object
rdfs:label	Energy@en
rdfs:comment	A saref:Property related to some measurements that are characterized by a certain value measured in an energy unit (such as Kilowatt_Hour or Watt_hour). Furter specializations of the saref:Energy class can be found in the SAREF4ENER extension, where classes such as EnergyMax, EnergyMin and EnergyExpected are defined. @en
rdf:type	owl:Class
rdfs:subClassOf	Property

Figure 36: PLATOON IDS Vocabulary Provider - Documentation - Search for a Term

It also allows selecting the right different ways to visualize the detailed information.

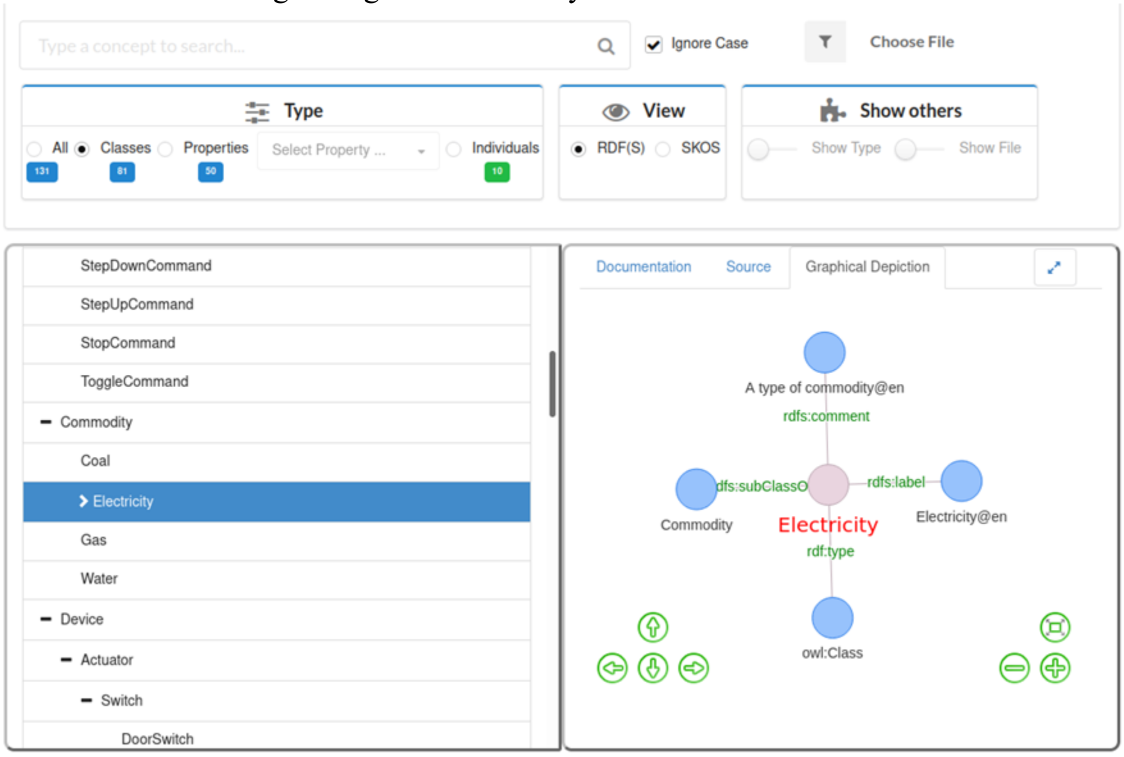


Figure 37: PLATOON IDS Vocabulary Provider - Documentation - Different Ways to show detailed information

On the other hand, the "Visualisation" tab allows visualizing the ontologies by representing graphically the relationships between the different terms through a network diagram, as shown in Figure 38

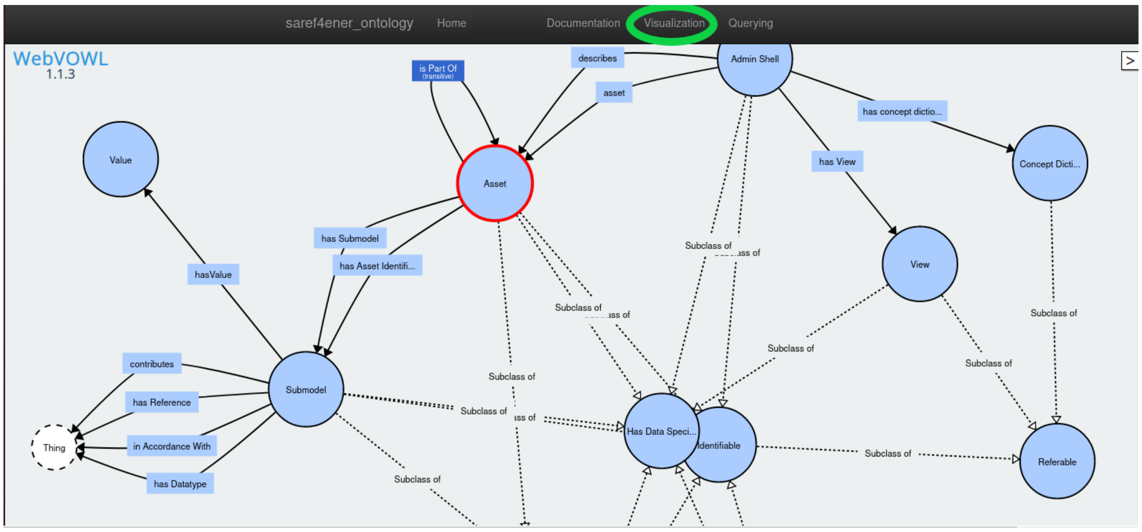
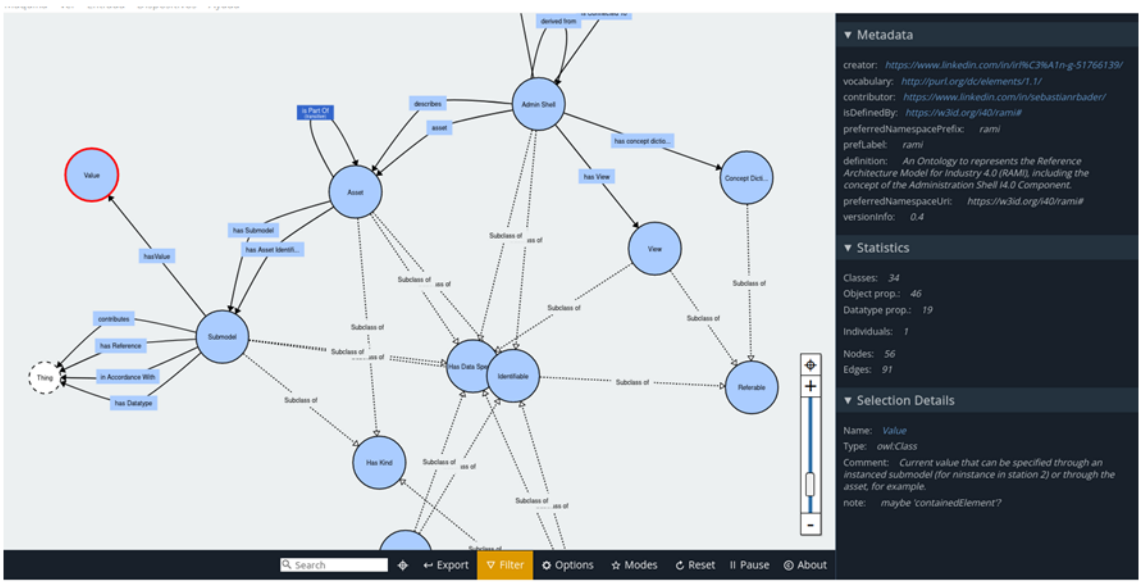


Figure 38: PLATOON IDS Vocabulary Provider – Visualisation – Network Diagram

When clicking on any of the terms or properties, more detailed information is shown in the right of the screen as represented in the next figure:



▾ Metadata

creator: https://www.linkedin.com/in/irl%C3%A1n-g-51766139/
vocabulary: http://purl.org/dc/elements/1.1/
contributor: https://www.linkedin.com/in/sebastianrbader/
isDefinedBy: https://w3id.org/i40/rami#
preferredNamespacePrefix: rami
prefLabel: rami
definition: An Ontology to represents the Reference Architecture Model for Industry 4.0 (RAMI), including the concept of the Administration Shell I4.0 Component.
preferredNamespaceUri: https://w3id.org/i40/rami#
versionInfo: 0.4

▾ Statistics

Classes: 34
Object prop.: 46
Datatype prop.: 19
Individuals: 1
Nodes: 56
Edges: 91

▾ Selection Details

Name: Value
Type: owl:Class
Comment: Current value that can be specified through an instanced submodel (for ninstance in station 2) or through the asset, for example.
note: maybe 'containedElement'?

Figure 39: PLATOON IDS Vocabulary Provider – Visualisation - Detailed information of terms

Also, several functionalities at the bottom allow to zoom in or out, search for a term, and filter by specific properties.

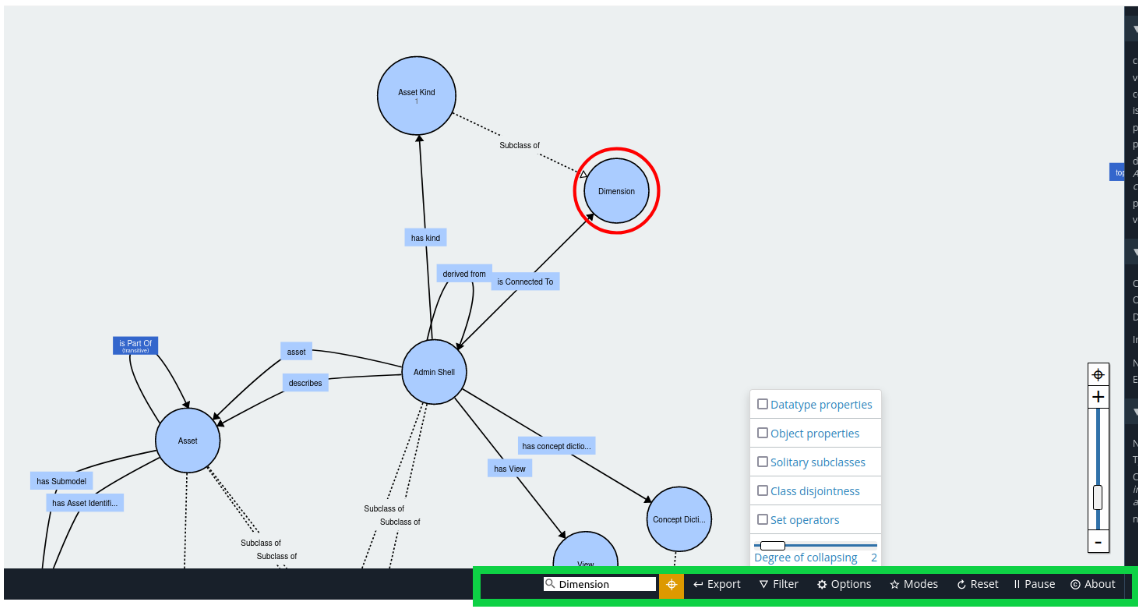
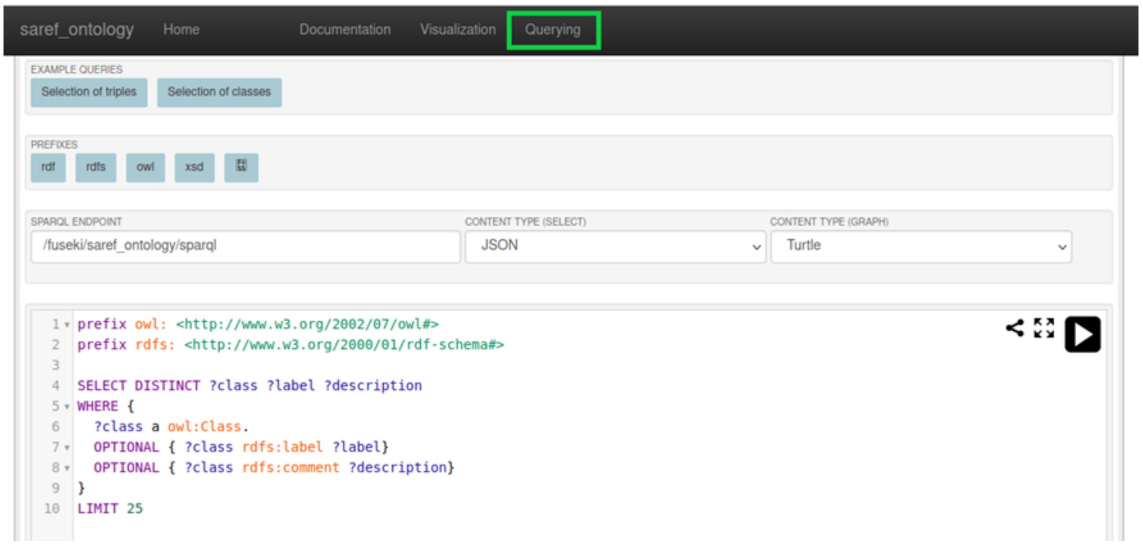


Figure 40: PLATOON IDS Vocabulary Provider – Visualisation - Additional Searching and Filtering Functionalities

Moreover, the "Querying" tab allows to directly run SPARQL queries on the selected ontology.



SPARQL ENDPOINT: /fuseki/saref_ontology/sparql
CONTENT TYPE (SELECT): JSON
CONTENT TYPE (GRAPH): Turtle

```
prefix owl: <http://www.w3.org/2002/07/owl#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?class ?label ?description
WHERE {
  ?class a owl:Class.
  OPTIONAL { ?class rdfs:label ?label}
  OPTIONAL { ?class rdfs:comment ?description}
}
LIMIT 25
```

Figure 41: PLATOON IDS Vocabulary Provider – Querying - SPARQL query execution

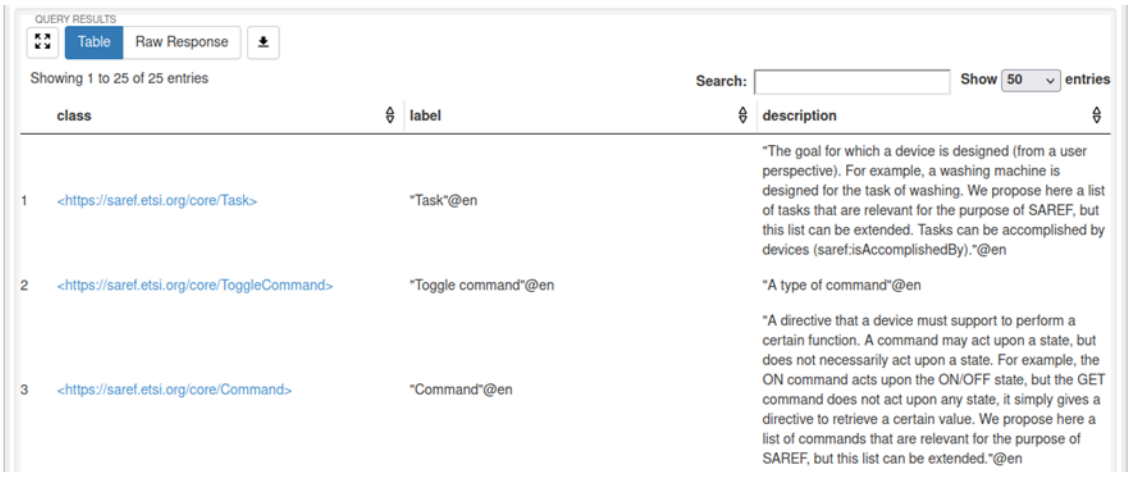

QUERY RESULTS

Table | Raw Response

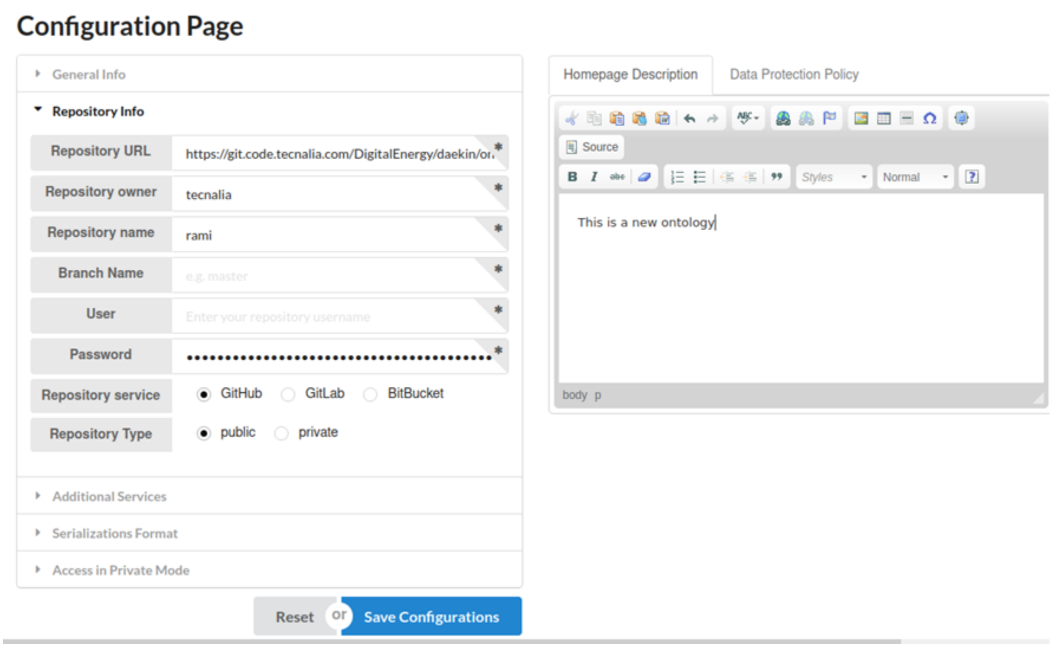
Showing 1 to 25 of 25 entries

Search: Show 50 entries

	class	label	description
1	<https://saref.etsi.org/core/Task>	"Task"@en	"The goal for which a device is designed (from a user perspective). For example, a washing machine is designed for the task of washing. We propose here a list of tasks that are relevant for the purpose of SAREF, but this list can be extended. Tasks can be accomplished by devices (saref:isAccomplishedBy)."@en
2	<https://saref.etsi.org/core/ToggleCommand>	"Toggle command"@en	"A type of command"@en
3	<https://saref.etsi.org/core/Command>	"Command"@en	"A directive that a device must support to perform a certain function. A command may act upon a state, but does not necessarily act upon a state. For example, the ON command acts upon the ON/OFF state, but the GET command does not act upon any state, it simply gives a directive to retrieve a certain value. We propose here a list of commands that are relevant for the purpose of SAREF, but this list can be extended."@en

Figure 42: PLATOON IDS Vocabulary Provider – Querying - SPARQL query result

Finally, the GUI has an administration section that allows you to insert new ontologies in the Platoon IDS Vocabulary Provider:


Configuration Page

General Info

Repository Info

Repository URL	https://git.code.tecnalia.com/DigitalEnergy/daekin/o
Repository owner	tecnalia
Repository name	rami
Branch Name	e.g. master
User	Enter your repository username
Password	••••••••••••••••••••••••••••••••••••••
Repository service	[✓ GitHub] GitLab BitBucket
Repository Type	[✓ public] private

Additional Services

Serializations Format

Access in Private Mode

Reset or Save Configurations

Homepage Description | Data Protection Policy

This is a new ontology

Figure 43: PLATOON IDS Vocabulary Provider – Configuration – Upload new Ontologies

5.1 Interaction With Vocabulary Provider

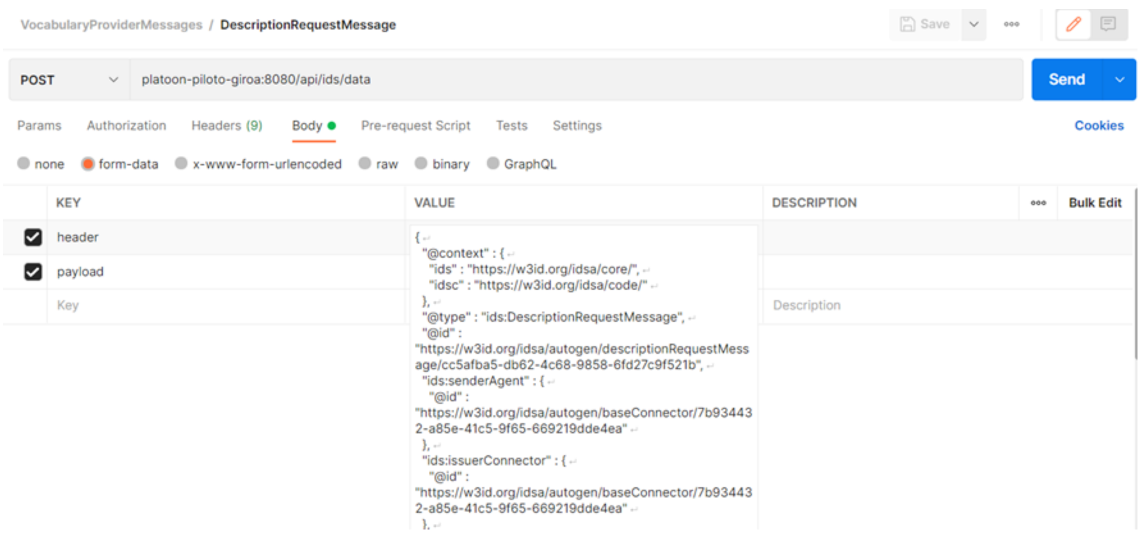
Apart from the GUI, the PLATOON IDS Vocabulary supports machine-to-machine communication allowing to query the different ontologies directly through an IDS connector. To do this, a REST API service was built on top of Vocol to allow the exchange of information between Vocol, Fuseki, and the IDS Core Container.

The IDS Core Container was developed using the SpringBoot library. A number of IDS messages have been implemented. Some of the messages allow IDS connectors from the pilots to communicate with the Platoon IDS Vocabulary Provider. Other messages allow the Platoon IDS Vocabulary Provider to communicate with other components of the IDS ecosystem, such as the Broker. In the following sections each of the messages is explained in more detail. Also the complete text of the messages is included in Appendix A.

5.1.1 DescriptionRequestMessage

This message allows a connector to call the PLATOON IDS Vocabulary Provider and obtain generic information from it; specifically, a "config.json" file is returned. It is a multipart POST message that will contain a header and a payload. In the header, the content of the message is displayed. In this case, the payload is empty.

URL to call: SERVER-URL:8080\api\ids\data



POST platoon-piloto-giroa:8080/api/ids/data

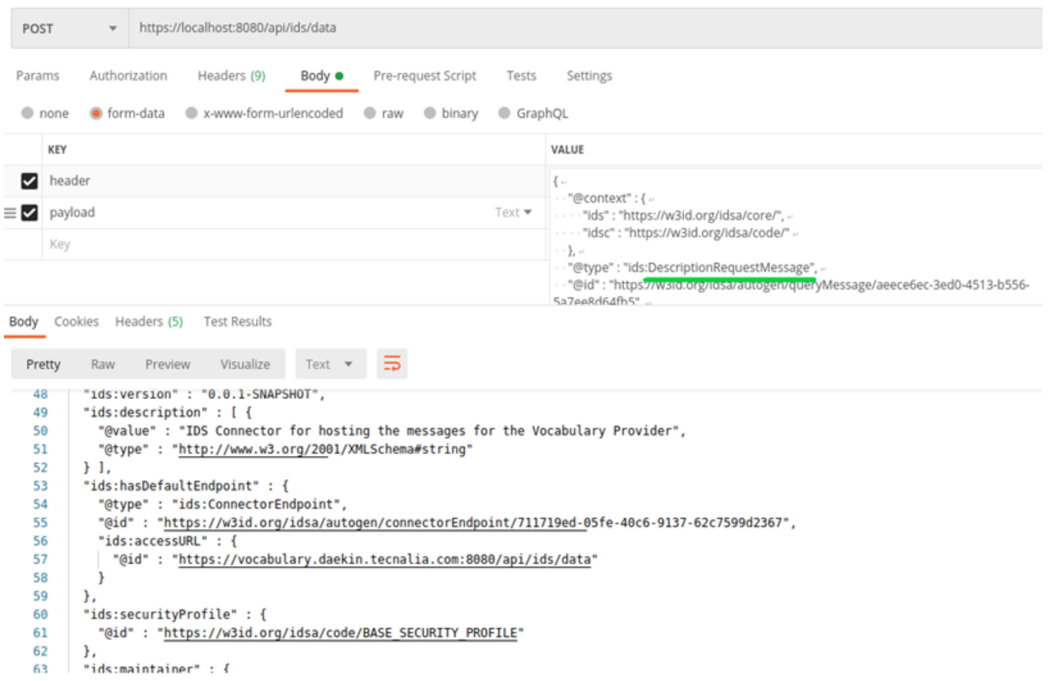
KEY	VALUE	DESCRIPTION
✓ header	(see below)	
✓ payload		
Key		Description

```
{
  "@context" : {
    "ids" : "https://w3id.org/idsa/core/",
    "idsc" : "https://w3id.org/idsa/code/"
  },
  "@type" : "ids:DescriptionRequestMessage",
  "@id" :
"https://w3id.org/idsa/autogen/descriptionRequestMess
age/cc5afba5-db62-4c68-9858-6fd27c9f521b",
  "ids:senderAgent" : {
    "@id" :
"https://w3id.org/idsa/autogen/baseConnector/7b93443
2-a85e-41c5-9f65-669219dde4ea"
  },
  "ids:issuerConnector" : {
    "@id" :
"https://w3id.org/idsa/autogen/baseConnector/7b93443
2-a85e-41c5-9f65-669219dde4ea"
  },
```

Figure 44: PLATOON IDS Vocabulary Provider - IDS Messages - Description Request Message

5.1.2 DescriptionResponseMessage

This message is the corresponding reply message to the description request message that contains in the payload the "config.json" file.



POST https://localhost:8080/api/ids/data

	KEY	VALUE
✓	header	
✓	payload	

```
{
  "@context" : {
    "ids" : "https://w3id.org/idsa/core/",
    "idsc" : "https://w3id.org/idsa/code/"
  },
  "@type" : "ids:DescriptionRequestMessage",
  "@id" : "https://w3id.org/idsa/autogen/queryMessage/aeece6ec-3ed0-4513-b556-5a7ee8d64fb5"
```

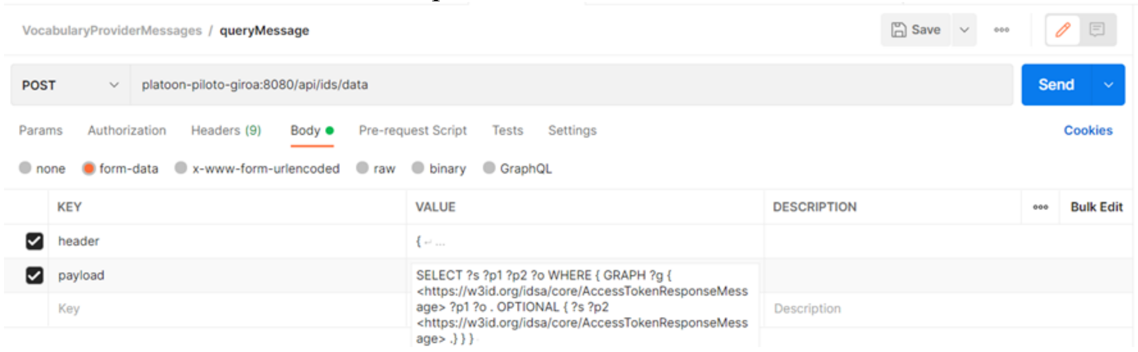
```
"ids:version" : "0.0.1-SNAPSHOT",
"ids:description" : [ {
  "@value" : "IDS Connector for hosting the messages for the Vocabulary Provider",
  "@type" : "http://www.w3.org/2001/XMLSchema#string"
} ],
"ids:hasDefaultEndpoint" : {
  "@type" : "ids:ConnectorEndpoint",
  "@id" : "https://w3id.org/idsa/autogen/connectorEndpoint/711719ed-05fe-40c6-9137-62c7599d2367",
  "ids:accessURL" : {
    "@id" : "https://vocabulary.daekin.tecnalia.com:8080/api/ids/data"
  }
},
"ids:securityProfile" : {
  "@id" : "https://w3id.org/idsa/code/BASE_SECURITY_PROFILE"
},
"ids:maintainer" : {
```

Figure 45: PLATOON IDS Vocabulary Provider - IDS Messages - Description Response Message

5.1.3 QueryMessage

This message allows an IDS connector to send SPARQL queries directly to the PLATOON IDS Vocabulary Provider. It is a multipart POST message that contains a header and a payload. The header contains the content of the message, and the payload contains the corresponding SPARQL query.

URL call: SERVER-URL:8080\api\ids\data



VocabularyProviderMessages / queryMessage

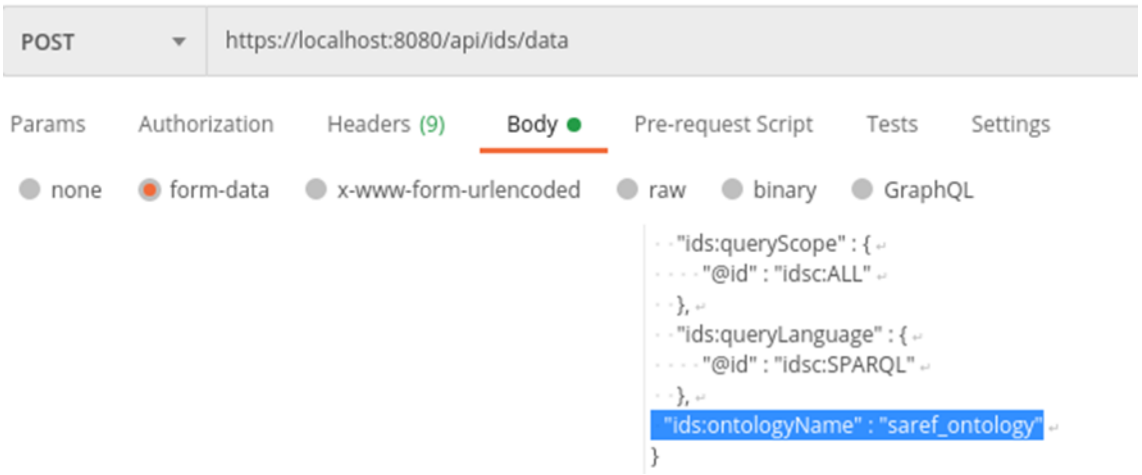
POST platoon-piloto-giroa:8080/api/ids/data

	KEY	VALUE	DESCRIPTION
✓	header	{ ...	
✓	payload	SELECT ?s ?p1 ?p2 ?o WHERE { GRAPH ?g { <https://w3id.org/idsa/core/AccessTokenResponseMessage> ?p1 ?o . OPTIONAL { ?s ?p2 <https://w3id.org/idsa/core/AccessTokenResponseMessage> .} } }	

Figure 46: PLATOON IDS Vocabulary Provider - IDS Messages - Query Message

As mentioned, the header contains the content of the message. This message contains a series of fields defined by the IDS reference architecture, which will vary depending on the type of message. The specific fields are the type of message, context, token, and information model used. In addition, in the case of the PLATOON IDS Vocabulary Provider, two new fields have been included that provide more flexibility to the queries. These fields are as follows:

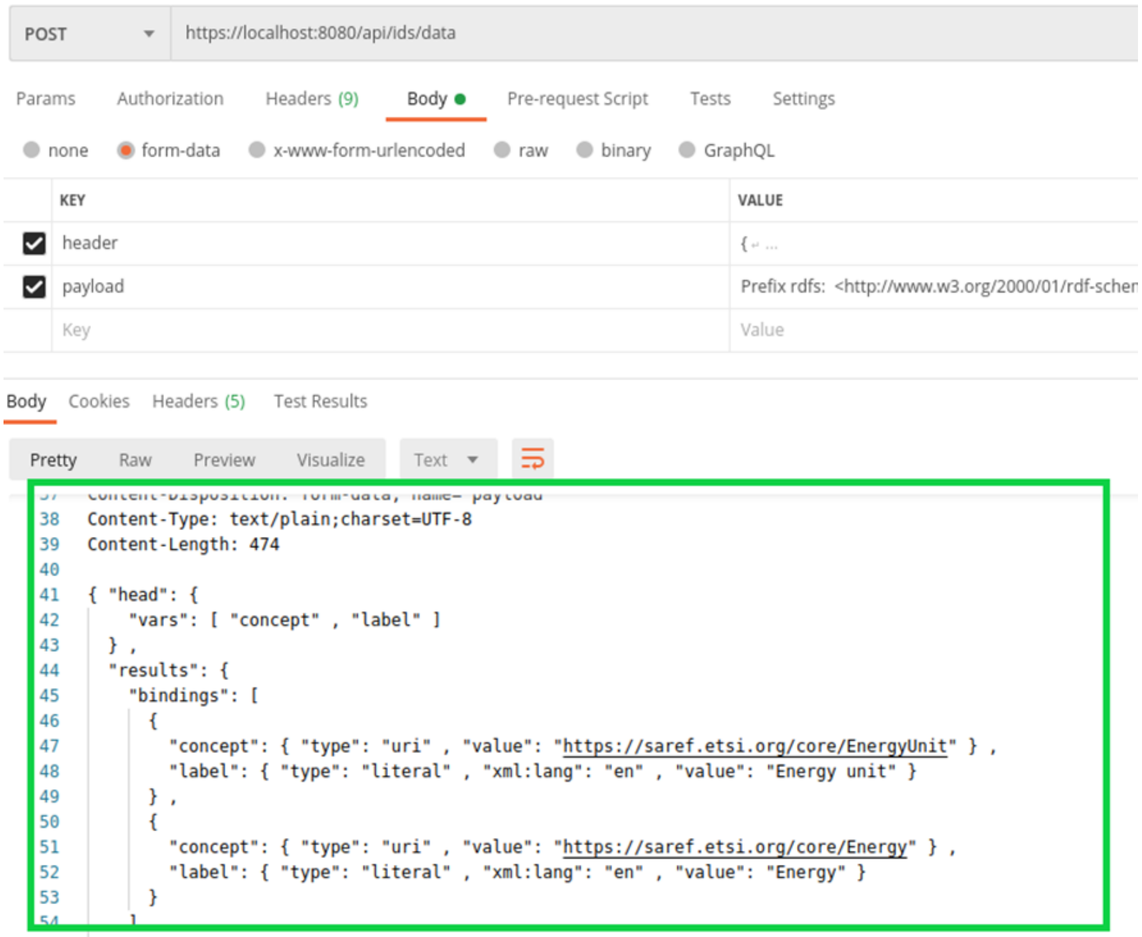
- *ids:ontologyName*: the name of the specific ontology to execute the query.



```
  "ids:queryScope" : {
    "@id" : "idsc:ALL"
  },
  "ids:queryLanguage" : {
    "@id" : "idsc:SPARQL"
  },
  "ids:ontologyName" : "saref_ontology"
}
```

Figure 47: PLATOON IDS Vocabulary Provider - IDS Messages - Query Message - Header Fields- ontologyName

If everything has gone correctly, a ResponseMessage will be returned with the results of the query specified in the payload field as shown in the following Figure:

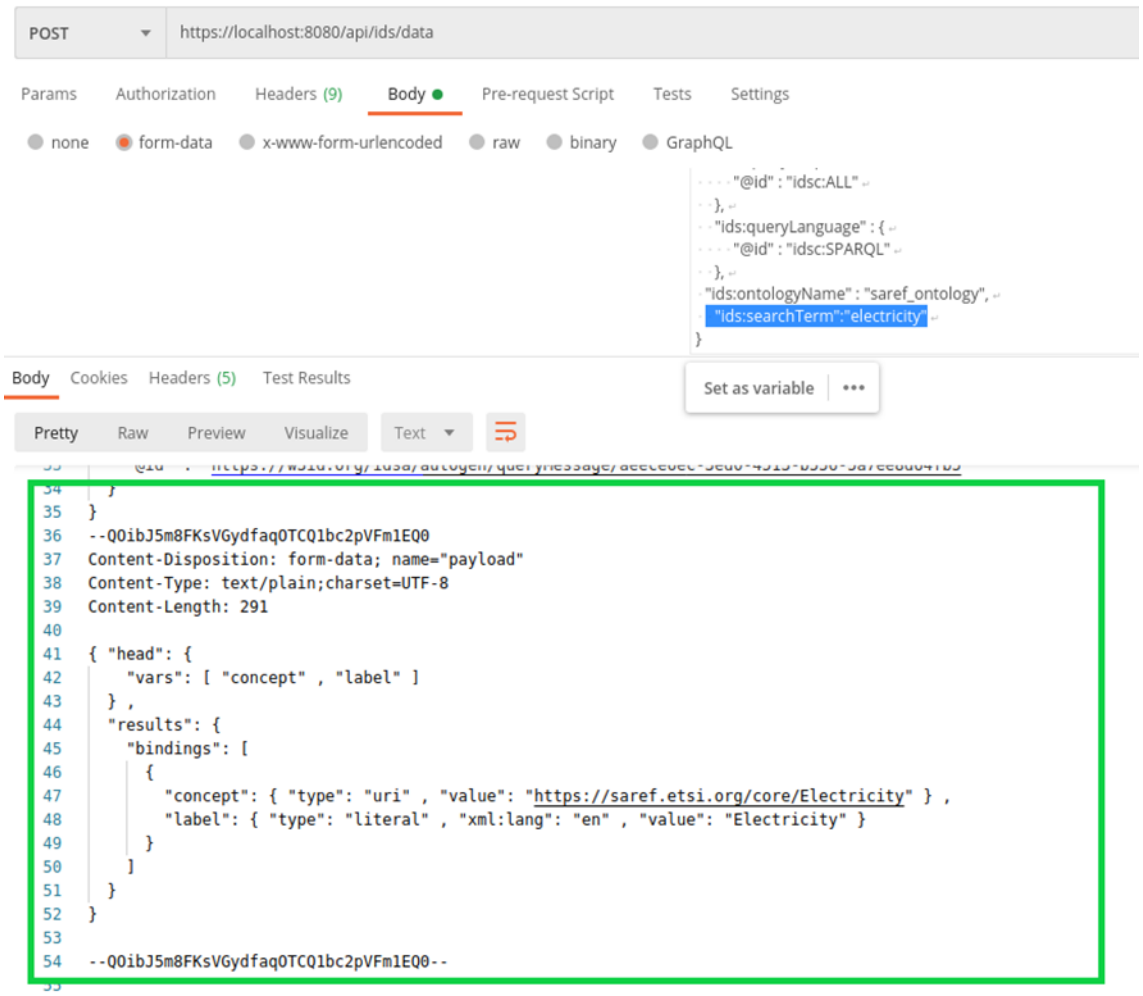


	KEY	VALUE
✓	header	{ ...
✓	payload	Prefix rdfs: <http://www.w3.org/2000/01/rdf-scher
	Key	Value

```
Content-Type: text/plain;charset=UTF-8
Content-Length: 474

{ "head": {
    "vars": [ "concept" , "label" ]
  } ,
  "results": {
    "bindings": [
      {
        "concept": { "type": "uri" , "value": "https://saref.etsi.org/core/EnergyUnit" } ,
        "label": { "type": "literal" , "xml:lang": "en" , "value": "Energy unit" }
      } ,
      {
        "concept": { "type": "uri" , "value": "https://saref.etsi.org/core/Energy" } ,
        "label": { "type": "literal" , "xml:lang": "en" , "value": "Energy" }
      }
```

Figure 48: PLATOON IDS Vocabulary Provider - IDS Messages - Query Message - Header Fields- ontologyName - the result

- *ids:searchTerm*: If this field is present, it will run the search for the specified term within the specified ontology.



```
    "@id" : "idsc:ALL"
  },
  "ids:queryLanguage" : {
    "@id" : "idsc:SPARQL"
  },
 "ids:ontologyName" : "saref_ontology",
 "ids:searchTerm":"electricity"
}
```

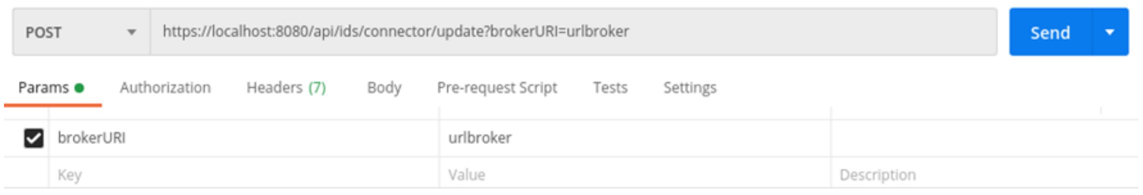
```
  }
}
--Q0ibJ5m8FKsVGydfaq0TCQ1bc2pVFm1EQ0
Content-Disposition: form-data; name="payload"
Content-Type: text/plain;charset=UTF-8
Content-Length: 291

{ "head": {
    "vars": [ "concept" , "label" ]
  } ,
  "results": {
    "bindings": [
      {
        "concept": { "type": "uri" , "value": "https://saref.etsi.org/core/Electricity" } ,
        "label": { "type": "literal" , "xml:lang": "en" , "value": "Electricity" }
      }
    ]
  }
}

--Q0ibJ5m8FKsVGydfaq0TCQ1bc2pVFm1EQ0--
```

Figure 49: PLATOON IDS Vocabulary Provider - IDS Messages - Query Message - Header Fields- searchTerm

5.1.4 ConnectorUpdateMessage

This message is used for registering the PLATOON IDS Vocabulary Provider at the PLATOON IDS Metadata Registry. This way, the rest of the connectors can look for the PLATOON IDS Vocabulary Provider and connect to it. It is a POST message where the input parameter is the URI of the IDS Metadata Registry.

URL to call:SERVER-URL:8080\api\ids\connector\update



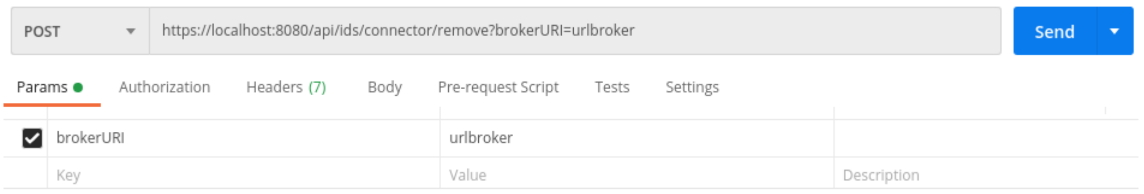
	Key	Value	Description
✓	brokerURI	urlbroker	

Figure 50: PLATOON IDS Vocabulary Provider - IDS Messages - ConnectorUpdateMessage

5.1.5 ConnectorUnavailableMessage

This message is used for unregistering PLATOON IDS Vocabulary Provider at the PLATOON IDS Metadata Registry. Similar to the previous message, it is a POST message where the input parameter is the URI of the Metadata Registry.

URL to call: SERVER-URL:8080\api\ids\connector\remove



https://localhost:8080/api/ids/connector/remove?brokerURI=urlbroker

✓ brokerURI	urlbroker	
Key	Value	Description

Figure 51: PLATOON IDS Vocabulary Provider - IDS Messages - ConnectorUnavailableMessage

5.2 Deployment of IDS Vocabulary Provider

The code of IDS Vocabulary Provider is hosted in the PLATOON Git repository: https://github.com/PLATOONProject/PLATOON_IDS-Vocabulary-Provider

The development environment of the IDS Vocabulary Provider consist on three main components:

- Vocol: an open-source Vocabulary Manager.
- Apache Jena Fuseki and TDB: A Sparql Server and database for storing and accessing the different vocabularies.
- IDS Vocabulary Provider: The core component that permits the communication between the conectors and the vocabulary provider.

5.2.1 Prerequisites

Prerequisite to run the IDS Vocabulary Provider:

- Docker
- Docker Compose
- Java
- Maven
- OpenSSL

5.2.2 Structure of IDS Vocabulary Provider

Normally there will exist one Vocabulary Provider for each ecosystem type. There will be a docker-compose file that will be in charge of creating all the corresponding images, the containers and running them. All the containers will be able to communicate internally.
Apart from the docker-compose file, there will be three dockerfiles, one for each of the mentioned components:

- **Vocol**: Will create the image for the vocol manager, downloading all the needed requirements; nodejs, npm etc. .and execute the application.
- **Fuseki**: Will create the image for the Fuseki server and launch the server.
- **IDS Vocabulary provider**: will create the image for the core component and launch the corresponding server, a Java environment running the Maven package of the code.

The IDS Vocabulary provider is a connector and so, it needs some certificates to implement a secure communication between the different components. Hence, it will communicate with a DAPS and the DAPS will give the connector a valid token each time a message is send. This component will communicate also with the Fuseki component to execute the requested queries and to obtain the results.

5.2.3 Creation of SSL Certificates

A valid X.509 certificate, signed by a trusted certification authority, is strongly recommended to avoid warnings about insecure HTTPS connections. The certificate needs to be of .crt format and must have the name server.crt. In case your certificate is of .pem format, it can be converted with the following commands, which require OpenSSL to be installed:

```
OpenSSL x509 -in mycert.pem -out server.crt
OpenSSL RSA -in mycert.pem -out server.key
mkdir cert
mv server.crt cert/
mv server.key cert/
```

5.2.4 Configuring the Docker-compose File

The docker-compose file is responsible of launching the three containers needed for the Vocabulary Provider to run properly and enable the communication of the three components as they are built in the same network.
Each of the components is exposed in one specific port. Then, if a certain port is occupied by another component of the ecosystem, it is possible to change this port through this file.
The changes of the port can be done in the Vocol component and in the IDS Vocabulary provider if needed, as these components are independent. To do that, edit the docker-compose file and change the corresponding port:

```
vocol-service:
  container_name: vocol_service_container
  build:
    context: .
    dockerfile: ./dockerfile-vocol
    network: host
  expose:
    - 3000
  ports:
    - "3000:3000"
```

Another crucial part of adapting the configuration is to provide the correct location of the X.509 certificate in the IDS messages service. Assuming the location of the certificate is "/home/vocolProject/cert," the corresponding configuration would be:

```
volumes:
- /home/vocolProject/cert:/etc/cert/
[…]
```

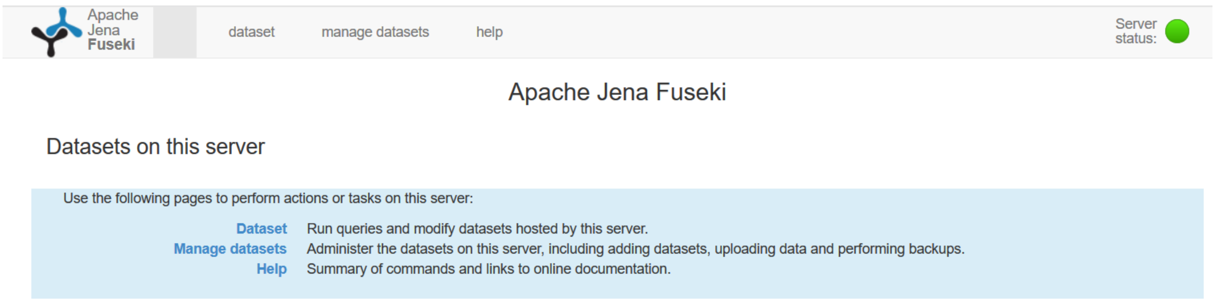
5.2.4.1 Run Application

To start up the IDS Vocabulary Provider, run the following command inside the directory of the docker-compose.yml file:

```
docker-compose up -d
```

This process can take several minutes to complete. You can test whether the IDS Vocabulary Provider has successfully started by opening the following urls:

- http://localhost:3000. This is the main page of the vocol service. The UI of the vocol will appear listing the existing ontologies.
- http://localhost:3030: If the fuseki server is running properly, you could see the main page for the fuseki manager. This page will only be used for manteinance purposes. The dot in the right (server status) side must be green.
- https://localhost:8080: If the IDS Messaging service is running this url will show an error message but it will be loaded.



Apache Jena Fuseki

Datasets on this server

Use the following pages to perform actions or tasks on this server:

Dataset	Run queries and modify datasets hosted by this server.
Manage datasets	Administer the datasets on this server, including adding datasets, uploading data and performing backups.
Help	Summary of commands and links to online documentation.

Figure 52: IDS Vocabulary Provider - Successfully started test

To enter to a running container, you could use:
Docker exec -it ids_messages_container /bin/bash : to get a bash shell in the container
Exit to get out

To see the logs of the container:
Go to the directory containing the docker files and:
Docker-compose logs

To interact with a container:
Docker attach vocol_service_container :

The latter could be useful, for example when uploading new ontologies if it is needed to enter some user or password.

5.2.4.2 Update

Containers can be either hot updated or restarted to apply the changes. To hot update a container, run the following command:
docker-compose up -d—no-deps—build <container name>

Alternatively, one can restart the entire service by running:
docker-compose down : To stop all the containers
docker-compose up –d
If you want to get all the images create again, you could use:
Docker-compose build : To build again the images
docker-compose up –d

6 Conclusion

This deliverable reports the design of the PLATOON marketplace module that implements a common endpoint to access the data and energy services. The PLATOON Marketplace is a place to search and discover data assets and services provided by the Marketplace participants fostering the interaction between data/service consumers and providers.
The user interface of the PLATOON Marketplace will reflect all the metadata of the datasets and services registered in the PLATOON IDS Metadata Registry. Upon a settled agreement between the two parties, the data flow will start between the data/service consumer and provider through the corresponding IDS connectors as per explained in deliverable D3.4. The Clearing House will contain the log of the settled agreement.
In addition, the developed IDS Vocabulary Provider enhances the capabilities of the PLATOON Marketplace regarding interoperability. It allows the data/data analytics tools users/consumers to easily understand the data and services published in the Marketplace, which facilitates the implementation and integration of these assets.
Finally, the IDS Identity Provider adopted by these PLATOON Marketplace components will create, maintain, monitor, and validate the participants' identity information. This deliverable aimed to reflect this scenario with examples.

The second version of this deliverable (D3.9) due in M30 will demonstrate the interaction of all the PLATOON Marketplace components through a real-life large scale pilot.

References

[1] “International Data Spaces Association,” [Online]. Available: https://internationaldataspaces.org/download/20861/.

[2] P. D.-I. B. Otto, S. Steinbuß, A. Teuscher and D.-I. S. Lohmann, Reference Architecture Model, vol. 3.0, International Data Spaces Association, April 2019.

[3] S. Steinbuss and S. Bader, Specification: IDS Clearing House, International Data Spaces Association, 2020.

[4] “International Data Spaces Jive,” [Online]. Available: Industrialdataspace.jiveon.com/docs/DOC-3232.

Appendix A: IDS Vocabulary Provider – IDS messages

DescriptionRequestMessage

Header:

```
{
"@context" : {
"ids" : "https://w3id.org/idsa/core/",
"idsc" : "https://w3id.org/idsa/code/"
},
"@type" : "ids:DescriptionRequestMessage",
"@id" : "https://w3id.org/idsa/autogen/descriptionRequestMessage/cc5afba5-db62-4c68-
9858-6fd27c9f521b",
"ids:senderAgent" : {
"@id" : "https://w3id.org/idsa/autogen/baseConnector/7b934432-a85e-41c5-9f65-
669219dde4ea"
},
"ids:issuerConnector" : {
"@id" : "https://w3id.org/idsa/autogen/baseConnector/7b934432-a85e-41c5-9f65-
669219dde4ea"
},
"ids:issued" : {
"@value" : "2020-10-13T13:55:54.345+02:00",
"@type" : "http://www.w3.org/2001/XMLSchema#dateTimeStamp"
},
"ids:modelVersion" : "4.0.0",
"ids:securityToken" : {
"@type" : "ids:DynamicAttributeToken",
"@id" : "https://w3id.org/idsa/autogen/dynamicAttributeToken/21b0ba17-dfb3-42f2-b7d0-
ece4debfa4af",
"ids:tokenValue" : "...",
"ids:tokenFormat" : {
"@id" : "idsc:JWT"
}
},
"ids:recipientConnector" : [ {
"@id" : "https://localhost:8080/api/ids/data"
} ]
}
```

Payload:
Empty

ResponseMessage

```
{
  "@context" : {
    "ids" : "https://w3id.org/idsa/core/",
    "idsc" : "https://w3id.org/idsa/code/"
  },
  "@type" : "ids:ResponseMessage",
  "@id" : "https://w3id.org/idsa/autogen/responseMessage/ec957493-8e8e-4416-9b7b-
d464c0899434",
  "ids:modelVersion" : "4.0.0",
  "ids:issued" : {
    "@value" : "2021-12-21T15:04:25.706Z",
    "@type" : "http://www.w3.org/2001/XMLSchema#dateTimeStamp"
  },
  "ids:correlationMessage" : {
    "@id" : "https://w3id.org/idsa/autogen/queryMessage/aeece6ec-3ed0-4513-b556-
5a7ee8d64fb5"
  },
  "ids:issuerConnector" : {
    "@id" : "https://ids_vocabulary_provider.com/"
  },
  "ids:senderAgent" : {
    "@id" : "https://w3id.org/idsa/autogen/baseConnector/7b934432-a85e-41c5-9f65-
669219dde4ea"
  },
  "ids:securityToken" : {
    "@type" : "ids:DynamicAttributeToken",
    "@id" : "https://w3id.org/idsa/autogen/dynamicAttributeToken/54578635-0868-40fa-933d-
82140d01f320",
    "ids:tokenValue" :
"eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJMRk9IbnR0cGFjbTJpS0VhSkw
tclNpSWlFTFlMbEkxRmxmQlBfZ3UzMmh3In0.eyJleHAiOjE2NDAwOTkxMjUsIm5iZiI6
MTY0MDA5OTA2NSwiaWF0IjoxNjQwMDk5MDY1LCJqdGkiOiJhYTljYjYwNS05NDdk
LTQ4ZmItYWI1Yy02MjQzMDVhMDgyMzYiLCJpc3MiOiJodHRwczovL2RhZWtpbi5kYX
BzLnRlY25hbGlhLmNvbTo4NDQzL2F1dGgvcmVhbG1zL0RhZWtpbiIsImF1ZCI6Imh0dH
BzOi8vdzNpZC5vcmcvaWRzYS9jb2RlL0lEU19DT05ORUNUT1JTX0FMTCIsInN1YiI6Ik
YzOjY0OjIwOkUzOjJCOkE3OjVBOkMyOkMzOjA5OkFFOkQyOjA4OkZBOkE4OkIwOj
YyOjIxOjdCOjgwOmtleWlkOkIyOkEyOkNFOjBGOjRGOkQxOjVGOkUyOjVFOkEwOkQ
zOkYyOjQyOjAxOjc1Ojk4OjBBOkRBOjg0OjAyIiwic2NvcGUiOiJpZHNjOklEU19DT05O
RUNUT1JfQVRUUklCVVRFU19BTEwiLCJzZWN1cml0eVByb2ZpbGUiOiJpZHNjOkJBU
0VfQ09OTkVDVE9SX1NFQ1VSSVRZX1BST0ZJTEUiLCJAdHlwZSI6ImlkczpEYXRQY
Xlsb2FkIiwicmVmZXJyaW5nQ29ubmVjdG9yIjoiaHR0cHM6Ly92b2NhYnVsYXJ5LmRhZ
Wtpbi50ZWNuYWxpYS5jb20iLCJAY29udGV4dCI6Imh0dHBzOi8vdzNpZC5vcmcvaWRz
YS9jb250ZXh0cy9jb250ZXh0Lmpzb25sZCJ9.BDzp_yE1pHXO2sWlwwEvkzdgL9hMKFC
YE-it_U2FJ3AFggqZiShOHX0adEc8qoK4o-
_MbO_1AweDPvvNQvldU50cdt9Zd6_UkyjORK272WGqdrDj7pFQmZJueH3XAAkECnO
WuF-DgyqxA6DWLhl93_-VLgDwfLWtRUbXIljw7Wc3fZrZZ-8JX_-
NHufpUjtNFYusDrqLppVVuDttoK2h3YUqCnsI6Tjg9rSwlmWsUon4ib8hcF93_E6FA7XC
WkAed1nZubPEA3YDzvRaw03g_Sep-ajrKFcg-
_DKCJK4eBQw7wdm9GXHH9jZRY0kjgZA7YmwjCaPMFd50HUMllNUzg",
   "ids:tokenFormat" : {
     "@id" : "https://w3id.org/idsa/code/JWT"
   }
  }
}
```

QueryMessage

Header:

```
{
"@context" : {
"ids" : "https://w3id.org/idsa/core/",
"idsc" : "https://w3id.org/idsa/code/"
},
"@type" : "ids:QueryMessage",
"@id" : "https://w3id.org/idsa/autogen/queryMessage/aeece6ec-3ed0-4513-b556-
5a7ee8d64fb5",
"ids:securityToken" : {
"@type" : "ids:DynamicAttributeToken",
"@id" : "https://w3id.org/idsa/autogen/dynamicAttributeToken/5129decb-1967-402e-b2fc-
0f9695c9253e",
"ids:tokenFormat" : {
"@id" : "idsc:JWT"
},
"ids:tokenValue" :
"eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6ImRlZmF1bHQifQ.eyJpZHMtYXR0c
mlidXRlcyI6sIm5iZiI6MTU4NjE3NTY0MCwiZXhwIjoxNTg2MTc5MjQwfQ.Puo5rFxDIW
CDgEFbW9ms-VKhtJeE_imm0LTIVuTXXR-0NKkmoqC4IEbB6YQbsG0t3HEYpA-
k2oPdDSYW1ScMu5mbbjQBlL5JEH1eUrHAjmUhnIt-oQ4rlu2vDFpWH-mcIfOMbKdw"
},
"ids:senderAgent" : {
"@id" : "http://example.org"
},
"ids:issuerConnector" : {
"@id" : "https://broker.ids.isst.fraunhofer.de/"
},
"ids:modelVersion" : "4.0.0",
"ids:issued" : {
"@value" : "2021-10-06T14:20:47.512+02:00",
"@type" : "http://www.w3.org/2001/XMLSchema#dateTimeStamp"
},
"ids:queryScope" : {
"@id" : "idsc:ALL"
},
"ids:queryLanguage" : {
```

```
"@id" : "idsc:SPARQL"
},
"ids:ontologyName":" saref4ener_ontology"
}
```

Payload:

```
SELECT ?s ?p1 ?p2 ?o WHERE { GRAPH ?g {
<https://w3id.org/idsa/core/AccessTokenResponseMessage> ?p1 ?o . OPTIONAL { ?s ?p2
<https://w3id.org/idsa/core/AccessTokenResponseMessage> .} } }
```

<u>QueryMessage (search for a term)</u>

```
{
"@context" : {
"ids" : "https://w3id.org/idsa/core/",
"idsc" : "https://w3id.org/idsa/code/"
},
"@type" : "ids:QueryMessage",
"@id" : "https://w3id.org/idsa/autogen/queryMessage/aeece6ec-3ed0-4513-b556-
5a7ee8d64fb5",
"ids:securityToken" : {
"@type" : "ids:DynamicAttributeToken",
"@id" : "https://w3id.org/idsa/autogen/dynamicAttributeToken/5129decb-1967-402e-b2fc-
0f9695c9253e",
"ids:tokenFormat" : {
"@id" : "idsc:JWT"
},
"ids:tokenValue" :
"eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6ImRlZmF1bHQifQ.eyJpZHMtYXR0c
mlidXRlcyI6sIm5iZiI6MTU4NjE3NTY0MCwiZXhwIjoxNTg2MTc5MjQwfQ.Puo5rFxDIW
CDgEFbW9ms-VKhtJeE_imm0LTIVuTXXR-0NKkmoqC4IEbB6YQbsG0t3HEYpA-
k2oPdDSYW1ScMu5mbbjQBlL5JEH1eUrHAjmUhnIt-oQ4rlu2vDFpWH-mcIfOMbKdw"
},
"ids:senderAgent" : {
"@id" : "http://example.org"
},
"ids:issuerConnector" : {
"@id" : "https://broker.ids.isst.fraunhofer.de/"
},
"ids:modelVersion" : "4.0.0",
"ids:issued" : {
"@value" : "2021-10-06T14:20:47.512+02:00",
"@type" : "http://www.w3.org/2001/XMLSchema#dateTimeStamp"
},
"ids:queryScope" : {
"@id" : "idsc:ALL"
},
"ids:queryLanguage" : {
"@id" : "idsc:SPARQL"
},
"ids:ontologyName":"saref4ener_ontology",
"ids:searchTerm":"energy"
```

}
Payload
empty.

<u>ConnectorUpdateMessage</u>
Header

```
{
  "@context" : {
    "ids" : "https://w3id.org/idsa/core/",
    "idsc" : "https://w3id.org/idsa/code/"
  },
  "@type" : "ids:ConnectorUpdateMessage",
  "@id" : "https://w3id.org/idsa/autogen/connectorUpdateMessage/1d2dca75-5d27-4c4c-
b698-113241df1869",
  "ids:modelVersion" : "4.0.0",
  "ids:securityToken" : {
    "@type" : "ids:DynamicAttributeToken",
    "@id" : "https://w3id.org/idsa/autogen/dynamicAttributeToken/5b456eba-9527-4c32-9b7a-
1fc46a82a77b",
    "ids:tokenFormat" : {
      "@id" : "idsc:JWT"
    },
    "ids:tokenValue" :
"eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6ImRlZmF1bHQifQ.eyJpZHMtYXR0c
mlidXRlcyI6eyJzZWN1cml0eV9wcm9maWxlIjp7ImF1ZGl0X2xvZ2dpbmciOjJ9LCJpZHNf
bWVtYmVyc2hpcCI6dHJ1ZSwiaWRzLXVyaSI6Imh0dHA6Ly9zb21lLXVyaSIsInRyYW5z
cG9ydENlcnRzU2hhMjU2IjoiYmFjYjg3OTU3NTczMGJiMDgzZjI4M2ZkNWI2N2E4Y2I4
OTY5NDRkMWJlMjhjN2IzMjExN2NmYzc1N2M4MWU5NiJ9LCJzY29wZXMiOlsiaWRz
X2Nvbm5lY3RvciJdLCJhdWQiOiJJRFNfQ29ubmVjdG9yIiwiaXNzIjoiaHR0cHM6Ly9kYX
BzLmFpc2VjLmZyYXVuaG9mZXIuZGUiLCJzdWIiOiJDPURFLE89RnJhdW5ob2ZlcixPV
T1JU1NULENOPTQ5ZmE5ODE1LTk1NTUtNDAzOC04YTlhLTRlMzZkZTM3YmY0NSI
sIm5iZiI6MTU4NjE3NTY0MCwiZXhwIjoxNTg2MTc5MjQwfQ.Puo5rFxDIWCDgEFbW9
ms-
VKhtJeE_imm0LTIVuDXw3Gk3tLRy2uDz0lQ1yvBafmHzR29Cx3GDjhl6_LDvFfROFdQN
iKeHq3I2gg6zzTYUEyf1FCBmmZg0Njj_0b_v6w_Atb9qTi0wu4IWieWvEeB32b4W4s2wf
w7s9GVJ7Gxbb3EpzzsorWbYDcwOPjRjHxJnLIqHBWg_JUdwxQtSh871mVYtzutwxCSthJ
0A2u5XB6CNYOpyMGrTXXR-0NKkmoqC4IEbB6YQbsG0t3HEYpA-
k2oPdDSYW1ScMu5mbbjQBlL5JEH1eUrHAjmUhnIt-oQ4rlu2vDFpWH-mcIfOMbKdw"
  },
  "ids:senderAgent" : {
    "@id" : "http://ids_vocabulary_provider.com"
  },
  "ids:issued" : {
    "@value" : "2021-12-21T04:00:00.000Z",
    "@type" : "http://www.w3.org/2001/XMLSchema#dateTimeStamp"
  },
  "ids:issuerConnector" : {
    "@id" : "https://broker.ids.isst.fraunhofer.de/"
  }
}
```

Payload:

```
{
  "@context" : {
    "ids" : "https://w3id.org/idsa/core/",
    "idsc" : "https://w3id.org/idsa/code/"
  },
  "@type" : "ids:BaseConnector",
  "@id" : "https://w3id.org/idsa/autogen/baseConnector/7b934432-a85e-41c5-9f65-
669219dde4ea",
  "ids:version" : "0.0.1-SNAPSHOT",
  "ids:description" : [ {
    "@value" : "IDS Connector for hosting the messages for the Vocabulary Provider",
    "@type" : "http://www.w3.org/2001/XMLSchema#string"
  } ],
  "ids:hasDefaultEndpoint" : {
    "@type" : "ids:ConnectorEndpoint",
    "@id" : "https://w3id.org/idsa/autogen/connectorEndpoint/711719ed-05fe-40c6-9137-
62c7599d2367",
    "ids:accessURL" : {
      "@id" : "https://vocabulary.daekin.tecnalia.com:8080/api/ids/data"
    }
  },
  "ids:securityProfile" : {
    "@id" : "https://w3id.org/idsa/code/BASE_SECURITY_PROFILE"
  },
  "ids:maintainer" : {
    "@id" : "https://www.tecnalia.com/"
  },
  "ids:curator" : {
    "@id" : "https://www.tecnalia.com/"
  },
  "ids:inboundModelVersion" : [ "4.0.0" ],
  "ids:outboundModelVersion" : "4.0.0",
  "ids:title" : [ {
    "@value" : "IDS Vocabulary Provider",
    "@type" : "http://www.w3.org/2001/XMLSchema#string"
  } ]
}
```

ConnectorUnavailableMessage
Header:

```
{
  "@context" : {
    "ids" : "https://w3id.org/idsa/core/",
    "idsc" : "https://w3id.org/idsa/code/"
  },
  "@type" : "ids:ConnectorUnavailableMessage",
  "@id" : "https://w3id.org/idsa/autogen/connectorUnavailableMessage/63d6a13f-abf9-42f7-
ab26-335b7927606e",
  "ids:affectedConnector" : {
    "@id" : "https://broker.ids.isst.fraunhofer.de/"
  },
  "ids:senderAgent" : {
    "@id" : "http://ids_vocabulary_provider.com"
  },
  "ids:securityToken" : {
    "@type" : "ids:DynamicAttributeToken",
    "@id" : "https://w3id.org/idsa/autogen/dynamicAttributeToken/e80bc6c0-ddfe-483b-8c07-
0bac59d99f51",
    "ids:tokenFormat" : {
      "@id" : "idsc:JWT"
    },
    "ids:tokenValue" :
"eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6ImRlZmF1bHQifQ.eyJpZHMtYXR0c
mlidXRlcyI6eyJzZWN1cml0eV9wcm9maWxlIjp7ImF1ZGl0X2xvZ2dpbmciOjJ9LCJpZHNf
bWVtYmVyc2hpcCI6dHJ1ZSwiaWRzLXVyaSI6Imh0dHA6Ly9zb21lLXVyaSIsInRyYW5z
cG9ydENlcnRzU2hhMjU2IjoiYmFjYjg3OTU3NTczMGJiMDgzZjI4M2ZkNWI2N2E4Y2I4
OTY5NDRkMWJlMjhjN2IzMjExN2NmYzc1N2M4MWU5NiJ9LCJzY29wZXMiOlsiaWRz
X2Nvbm5lY3RvciJdLCJhdWQiOiJJRFNfQ29ubmVjdG9yIiwiaXNzIjoiaHR0cHM6Ly9kYX
BzLmFpc2VjLmZyYXVuaG9mZXIuZGUiLCJzdWIiOiJDPURFLE89RnJhdW5ob2ZlcixPV
T1JU1NULENOPTQ5ZmE5ODE1LTk1NTUtNDAzOC04YTlhLTRlMzZkZTM3YmY0NSI
sIm5iZiI6MTU4NjE3NTY0MCwiZXhwIjoxNTg2MTc5MjQwfQ.Puo5rFxDIWCDgEFbW9
ms-
VKhtJeE_imm0LTIVuDXw3Gk3tLRy2uDz0lQ1yvBafmHzR29Cx3GDjhl6_LDvFfROFdQN
iKeHq3I2gg6zzTYUEyf1FCBmmZg0Njj_0b_v6w_Atb9qTi0wu4IWieWvEeB32b4W4s2wf
w7s9GVJ7Gxbb3EpzzsorWbYDcwOPjRjHxJnLIqHBWg_JUdwxQtSh871mVYtzutwxCSthJ
0A2u5XB6CNYOpyMGrTXXR-0NKkmoqC4IEbB6YQbsG0t3HEYpA-
k2oPdDSYW1ScMu5mbbjQBlL5JEH1eUrHAjmUhnIt-oQ4rlu2vDFpWH-mcIfOMbKdw"
  },
  "ids:issued" : {
    "@value" : "2021-12-21T17:00:00.000Z",
    "@type" : "http://www.w3.org/2001/XMLSchema#dateTimeStamp"
  },
  "ids:modelVersion" : "4.0.0",
  "ids:issuerConnector" : {
    "@id" : "https://broker.ids.isst.fraunhofer.de/"
  }
}
```

Payload:
Empty