KYKLOS 4.0

An Advanced Circular and Agile Manufacturing Ecosystem based on rapid reconfigurable manufacturing process and individualized consumer preferences

Deliverable D_._

Backend and Interoperability Layer Quick Guide

Editor(s):	Enrique Goyenechea
Responsible Partner:	ADSYS
Status-Version:	Draft - v0.1
Date:	27/05/2022
Distribution level (CO, PU):	PU

Project Number:	GA 872570
Project Title:	KYKLOS 4.0

Title of Deliverable:	Backend and Interoperability Layer Quick Guide
Due Date of Delivery to the EC:	N/A

Workpackage responsible for the Deliverable:	WP10 -
Editor(s):	Enrique Goyenechea
Contributor(s):	ADSYS
Reviewer(s):	N/A
Approved by:	N/A
Recommended/ mandatory readers:	All Partners interested in backend data visualization

Abstract:	Quick guide for data retrieval and general, practical information on how to interact with the Backend and the Interoperability Layer components
Keyword List:	Manual, guide, backend, interoperability, layer, help
Licensing information:	This work is licensed under Creative Commons Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0) http://creativecommons.org/licenses/by-sa/3.0/
Disclaimer	This document reflects only the author's views and the Commission is not responsible for any use that may be made of the information contained therein

Document Description

Document Revision History

Version	Date	Modifications Introduced	
		Modification Reason	Modified by
v0.1	27/05/2022	First draft version	ADSYS

Table of Contents

List of Figures

Terms and abbreviations

EC	European Commission
BE	Backend
IL	Interoperability Layer

1 Introduction

This document intends to be a quick reference to understand and interact with the Backend and Interoperability layer components, from a final user's perspective.

It comprehends the theoretical aspects and basic procedures to check the data and to connect with the sources, as well as retrieval procedures.

For more details, an alternative in form of several videos have been developed in order to make this guide easier. Furthermore, a contextual help is available within the platform, at the upper right corner of every page.

2 Context

The Backend and Interoperability Layer combination constitutes an important basis to the project in terms of data provision. They allow centralizing in a single point all incoming data from different components, offering them to partners interested in their consumption, abstracting them from their origin, and providing a reliable, standardized and authenticated source of information.

The BE is a means of storage, while the IL is in charge of retrieving these data from heterogeneous sources. They are quite overlapped.

In this context, an end user is the one who can have access to his data, to inspect how they have been collected and to query its values.

Currently [May, 2022] data are obtained from two partners and several Open Call Projects. Regarding the partners, data are obtained from ASTANDER and PINDOS.

- ASTANDER data consists of 37 signals, obtained via FTP. These signals map to a total of 210 values, which monitor a crane in a shipyard.
    - UPM provides an additional 35 signals.
- PINDOS data consists of 113 signals from various systems, obtained via MQTT. These signals monitor systems (pumps, water meters, conveyor belts, etc.) in a chicken processing plant.
    - UPM provides 112 additional signals.

3 Components Layout

The Interoperability Layer extracts information from external sources provided by partners, making use of different services to manage connectivity and data transformations. This information is then stored in a database. The backend API allows access to this information, both at UI level and directly by web services.

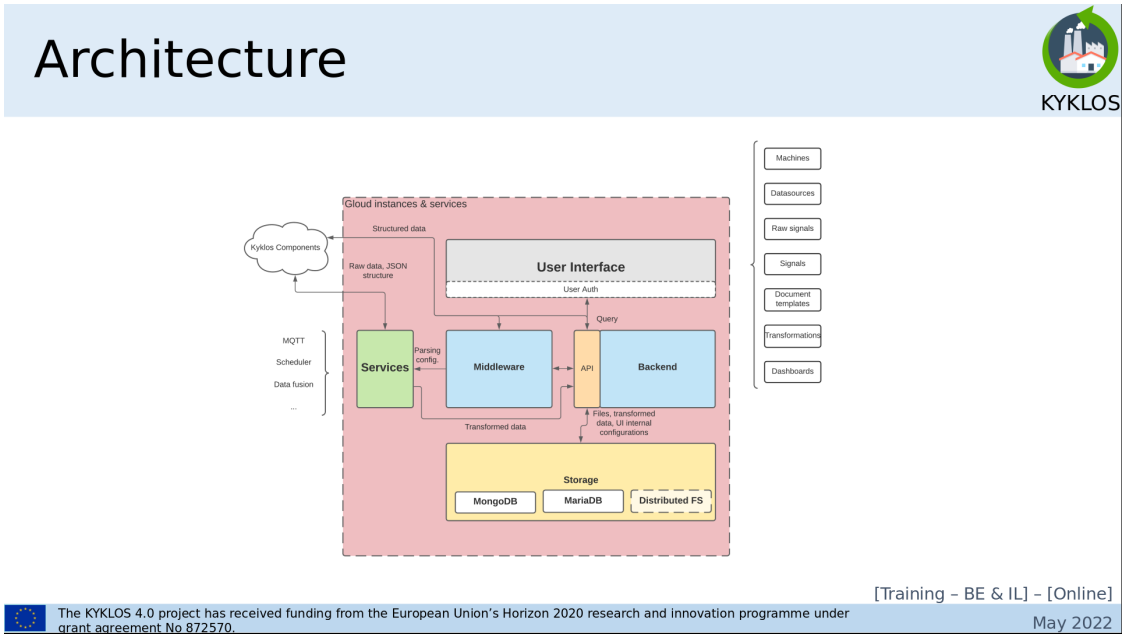


Figure 1: BE & IL layout and data flows

Entities allow to establish how to interact with the platform data. They are either purely conceptual abstractions (machines, signals) or identifiable with real, "physical" configurations (document templates, raw signals, etc.). We will talk about them in the configuration section.

4 User considerations

As aforementioned, users -end users- to this document are only capable of data inspection. A broader definition is presented here.

There are 4 roles available:

- Superadmin
  - There is only one, and it has all the available permissions.
- Admin
  - There can be more than one. It has configuration capabilities for data extraction and internal management of the platform [users, relationships between users, credentials, etc.].
- Operator
  - More than one can exist. It has data extraction and data reading configuration capabilities.
- Normal user

- This is the role provided to the project partners. It allows querying the data both from the UI and from the API.

The end user has no control over the interoperability layer, as these are configurations that have been managed by ADSYS to make life easier for the user and to speed up the integration process. However, you are fully capable of querying your data stored in the backend either graphically or textually via the API.

In any case user permissions can be modified individually if necessary.

As for information sharing, this is defined by the users to whom access is provided through the creation of the "Machines" entity and the selection of users with whom the values associated with that machine are shared. The relationships between users must be established by an administrator user.

5 Procedures

1.1 Data retrieval

1.1.1 UI

1. Log in https://kyklos-backend.kyklos40project.eu/login?site=kyklos
   - User has to log in at the URL with its credentials. These credentials have to be requested to ADSYS[1].

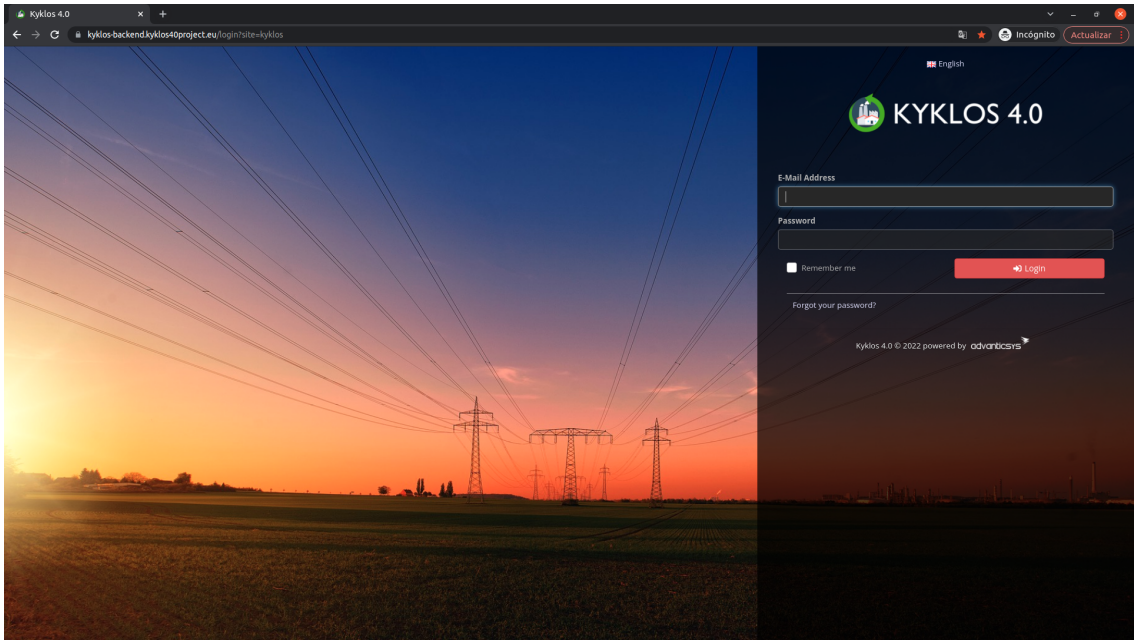


Figure 2: Log in page

1 enrique.goyenechea@advanticys.com – jheras@advanticsys.com

2. Access to the section data and analysis, subsection Historical data

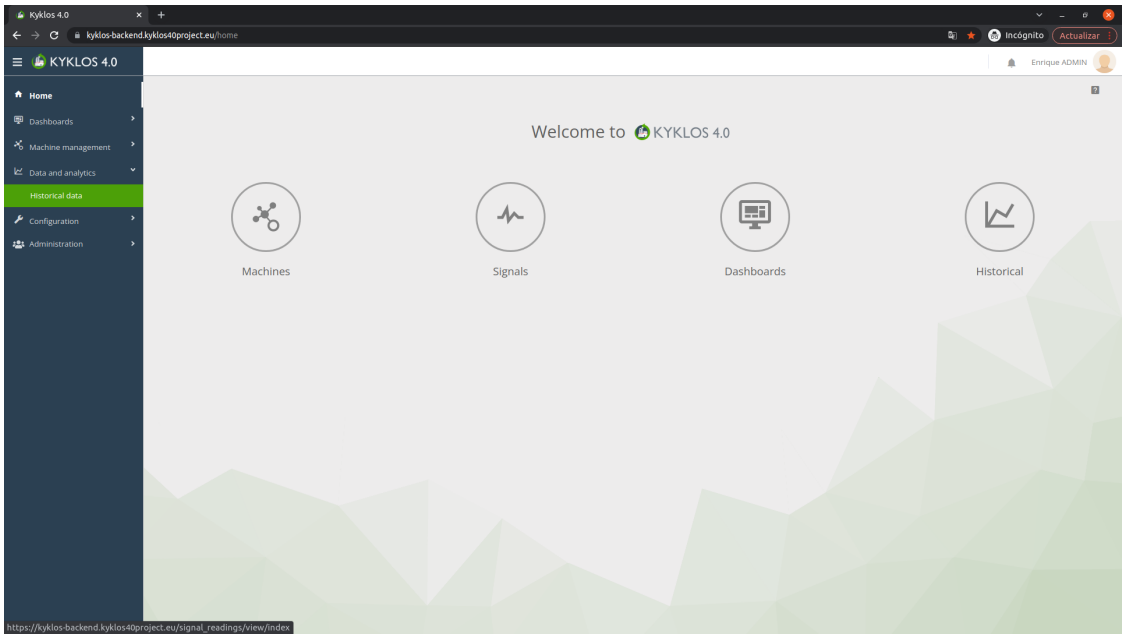


Figure 3: Menu selection, Historical data

This section has the following parts:

- Filters (1)
- Signal preview section (2)
- Area for displaying data in graphical format (3)
- Data display area in text format (4)

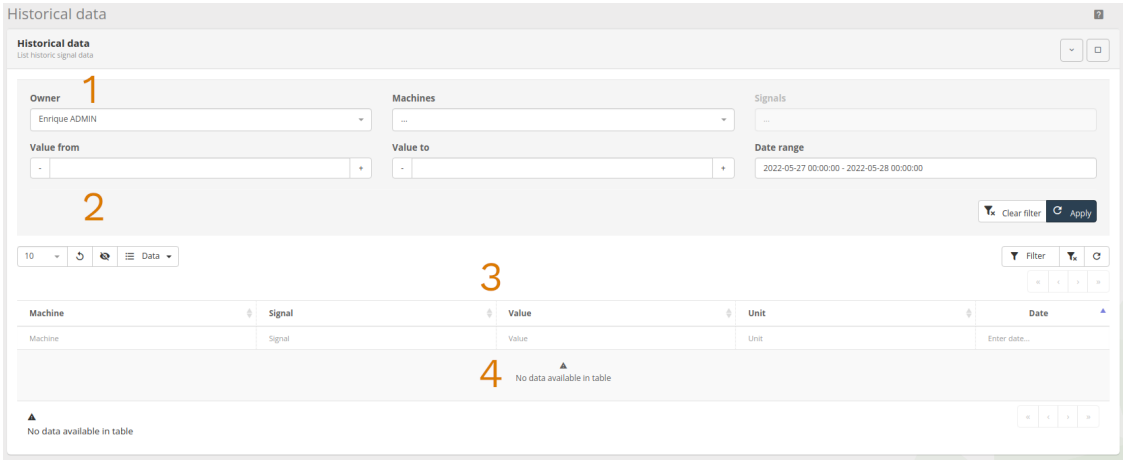


Figure 4: Historical data layout

Filters are cummulative and dynamic: The effect of the filters is summed, and their possible values depend on previous selections. Thus, we will have to select

the machine that groups our values, and select the signals we want to display. The signals will be added at the bottom, and will be assigned a color and a display format. It is important to determine which date we want to check -very recent ranges may have no data, and too large ranges may make the search useless due to excess of information-. In case of searching for a specific value, we can narrow it both above and below.

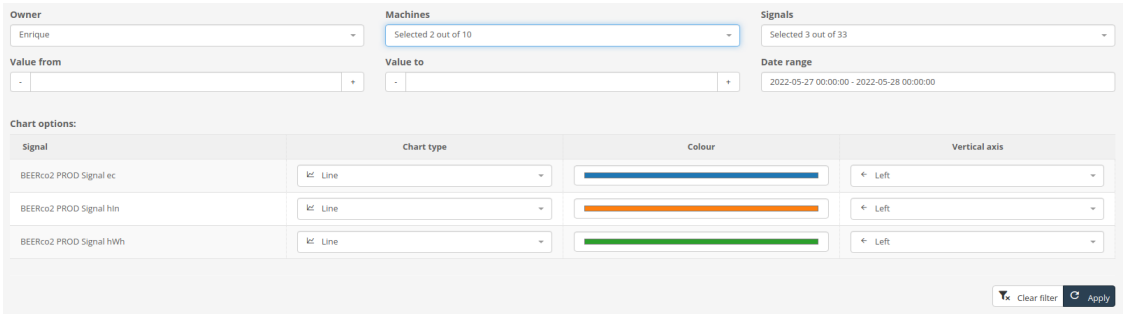


Figure 5: Filter configuration and signal selection

Once the search is launched, two loading indicators are displayed. The upper one is that of the graph, which will take longer. It may be that the values or the combination of signals cause the graph cannot be displayed. At the bottom, a paginated table with the query results will be displayed. This table allows you to export the data and filter it.

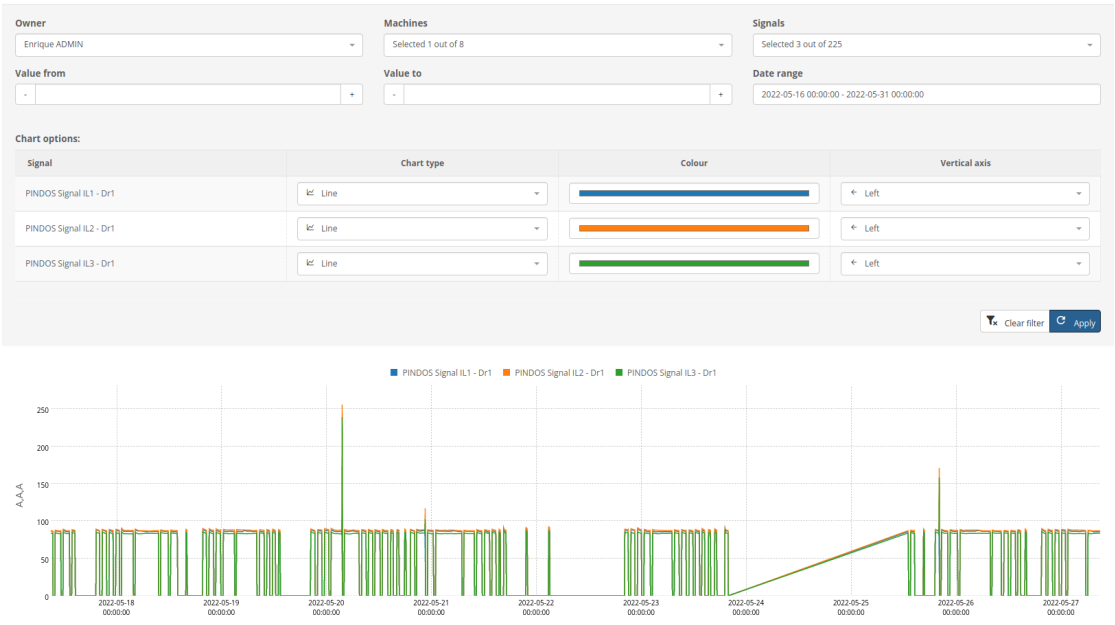
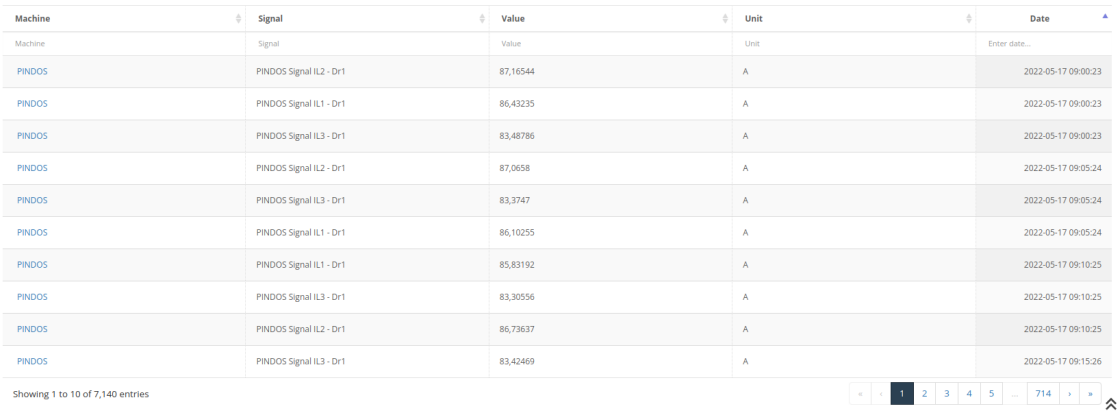


Figure 6: Chart configuration and results display



Machine	Signal	Value	Unit	Date
Machine	Signal	Value	Unit	Enter date...
PINDOS	PINDOS Signal IL2 - Dr1	87,16544	A	2022-05-17 09:00:23
PINDOS	PINDOS Signal IL1 - Dr1	86,43235	A	2022-05-17 09:00:23
PINDOS	PINDOS Signal IL3 - Dr1	83,48786	A	2022-05-17 09:00:23
PINDOS	PINDOS Signal IL2 - Dr1	87,0658	A	2022-05-17 09:05:24
PINDOS	PINDOS Signal IL3 - Dr1	83,3747	A	2022-05-17 09:05:24
PINDOS	PINDOS Signal IL1 - Dr1	86,10255	A	2022-05-17 09:05:24
PINDOS	PINDOS Signal IL1 - Dr1	85,83192	A	2022-05-17 09:10:25
PINDOS	PINDOS Signal IL3 - Dr1	83,30556	A	2022-05-17 09:10:25
PINDOS	PINDOS Signal IL2 - Dr1	86,73637	A	2022-05-17 09:10:25
PINDOS	PINDOS Signal IL3 - Dr1	83,42469	A	2022-05-17 09:15:26

Showing 1 to 10 of 7,140 entries « ‹ 1 2 3 4 5 ... 714 › »

Figure 7: Data results table

1.1.2 API

In order to facilitate access to data from the API, a number of Postman collections have been provided. Since the API allows management of all backend and interoperability platform entities, there are a multitude of endpoints or URLs to query.

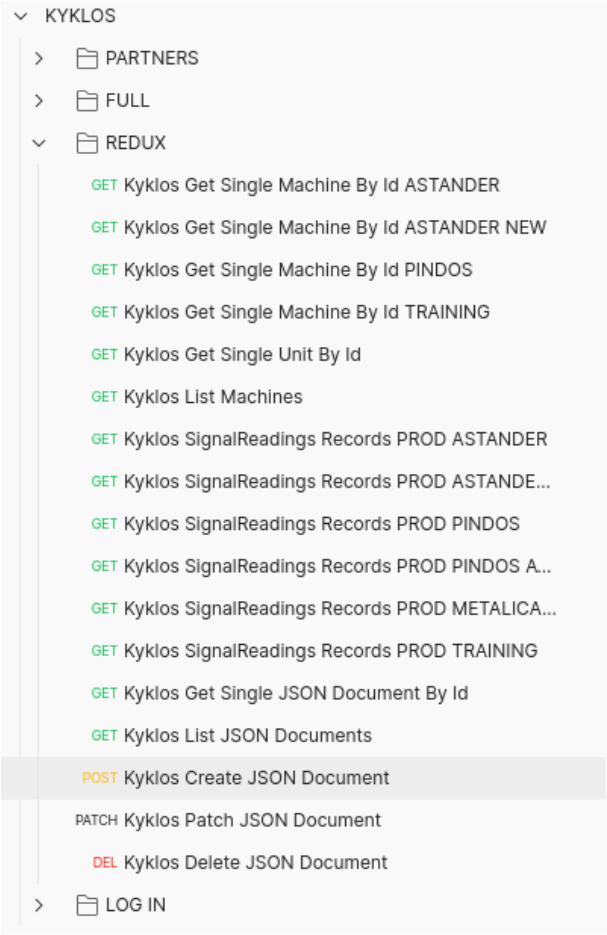


Figure 8: API endpoints

These have been reduced to the following for data signal retrieval:

1.1.2.1 Log in

Allows logging in with the user's credentials. It is IMPORTANT to note that it is not possible to be logged into the API and the UI at the same time. Failure to do so will lead to error.

1.1.2.2 List Machines

Allows to list the machines to which a user has access, either by owning them or by being shared by him.

1.1.2.3 Single Machine

Allows access to the details of a particular machine. This is useful to know the ids of the signals to be retrieved.

1.1.2.4 Signal Readings

Allows you to retrieve the values read from the specified signals, filter them by date and sort them. You have to use the identifiers obtained in the previous endpoints to determine the signals. Note there are pre-configured endpoints for specific partner signals.

The result is values for each signal independently specified with details such as unit, etc.

Figure 9: Signal readings results

Filters, sorts and paging

The format of the filters for these queries is an array composed as follows. It is suggested to use a previously formed request changing values:

- Basic structure

?page=1**&length**=10**&filter=[**{"scope":"**comp_signal_id**","type":"object-id","operator":"in","value":["6188f8d1f796675f2931ece2","61894e73a696a53e4103d3c2","61894e9a3bd2dc45c0309972","618955fba696a53e4103d3c3","61895629a15d1114f912d092","6189565d6f14e052104f3f73","6189572ca15d1114f912d093","6189576b6f14e052104f3f75","6189578b96db441b3f7b7502","618957be32fd5c059c7dd9a3","618957dc96db441b3f7b7503","618d150bfb099503d07f8c8a","618d18b94ef2e62fdd17ce46","618d3c29fb099503d07f8d55","618d3c6aa73af145294f12f3","618d3c9bfb099503d07f8d56","618d3cc354ef02535004a23a","618d3ce7fb099503d07f8d57","618d3d155c2d32157b434782","618d3d35ba7f19144a228d23","618d3dd354ef02535004a23b","618d3e2c7ec30a4cf2014a5b","618d3e575c2d32157b434783","618d3ecf54ef02535004a23c","618d3ee8e191ea48057c1a35","618d3f01fb099503d07f8d58","618d3f1ba73af145294f1302","618d3f4254ef02535004a23d","618d3f5b5c2d32157b434784","618d3f76fb099503d07f8d59","618d3fbf7ec30a4cf2014a75","618d3fdce191ea48057c1a43","618d3ff7a73af145294f1310","618d402f7ec30a4cf2014a76","618d404754ef02535004a24b","618d4082e191ea48057c1a44","618d409768387a67545e6760","618d40a9a73af145294f1312","618d40c55c2d32157b434785","618d40f9fb099503d07f8d6d","618d410f54ef02535004a259","618d412668387a67545e6761","618d415e54ef02535004a25a","618d4193fb099503d07f8d7b","618d41aaab30e8765c08dfe3","618d41d54ef2e62fdd17cf74","618d41e9a73af145294f131f","618d423d7ec30a4cf2014a90","618d4253ba7f19144a228d59","618d4276ab30e8765c08dff7","618d42d3536f70692e5839b2","618d42ee536f70692e5839b3","618d4309fb099503d07f8d83","618d4344fb099503d07f8d84","618d43633f6027149c2c0f5f","618d4381ba7f19144a228d5a","618d439f54ef02535004a268","618d4425536f70692e5839b8","618d443754ef02535004a275","618d4452536f70692e5839b9","618d44683938a421336e5aa2","618d4490ba7f19144a228d6e","618d44a4e191ea48057c1a66","618d44bb3938a421336e5aaa","618d44d3ba7f19144a228d6f","618d44ecfb099503d07f8d92","618d4507536f70692e5839be","618d453c3f6027149c2c0f60","618d456068387a67545e6783","618d4579e191ea48057c1a74","618d458d54ef02535004a283","618d45ef3f6027149c2c0f68","618d460c3f6027149c2c0f69","618d464754ef02535004a284","618d4b40ba7f19144a228d90","618d4f4e536f70692e5839f2","618d4fade191ea48057c1a8d","618d4fcd68387a67545e67e3","618d4ff7ba7f19144a228d9f","618d51aaa73af145294f138f","618d51ccfb099503d07f8ded","618d51e43938a421336e5b7b","618d522c68387a67545e6812","618d52593938a421336e5b7c","618d5287536f70692e583a07","618d52e1ba7f19144a228db4","618d5307a73af145294f139d","618d534f3f6027149c2c1001","618d538a68387a67545e6824","618d53b7536f70692e583a15","618d53fdba7f19144a228db6","618d542e54ef02535004a2e8","618d5456ba7f19144a228db7","618d54834ef2e62fdd17cff6","618d54adba7f19144a228db8","618d54c84ef2e62fdd17cff7","618d551454ef02535004a2f5","618d554368387a67545e6825","618d5563a73af145294f139e","618d55e83f6027149c2c1013","618d560854ef02535004a2f6","618d5644fb099503d07f8e3f"]},{"scope":"**date**","type":"date-range","operator":">= <","value":"2021-08-11 00:00:00 - 2021-11-13 00:00:00"}**]&sort=[**{"scope":"**date**","value":"asc"}**]**

- Signal IDs
  - Format

{"**scope**":"comp_signal_id","**type**":"object-id","**operator**":"in","**value**":[<DOUBLE_QUOTED_SIGNAL_IDS>]}

  - Example

{"scope":"comp_signal_id","type":"object-id","operator":"in","value":["6188f8d1f796675f2931ece2","61894e73a696a53e4103d3c2","61894e9a3bd2dc45c0309972"]}

- Date range
  - Format

    {"**scope**":"date","**type**":"date-range","**operator**":"<OPERATOR>”,”**value**”:”<DATES>”

  - Example

    {"scope":"date","type":"date-range","operator":">= <","value":"2021-08-11 00:00:00 - 2021-11-13 00:00:00"}

- Paging
  - Format
    - page=<PAGE_NUM>
    - length=<RECORDS_PER_PAGE>

1.2 JSON file retrieval

It is possible to store files in JSON format at the backend, firstly associated to their creator user and secondly to a machine. This means that access to the files is controlled according to ownership.

The access to these items is contemplated through the API, exclusively. There are 5 endpoints, or URLs to be called. A Postman collection is available.

- List files
- Single file
- Create file
- Update file
- Delete file

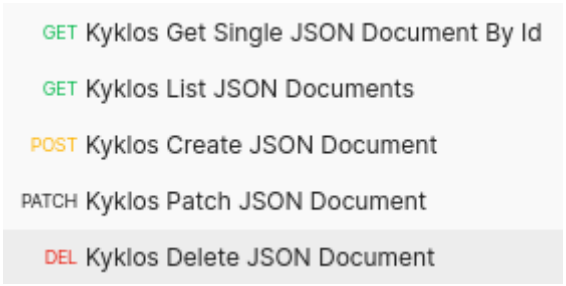


Figure 10: Log in page

The fields to be observed for these items are the following:

- Name
- Description
- Associated machine ID - The valid ID of a machine to which the document is to be associated. Machine: Conceptual grouping of signals. For example: PINDOS, ASTANDER
- [Creator User ID] - The ID of the current logged-in user is taken by default.
- Content - JSON content as-is

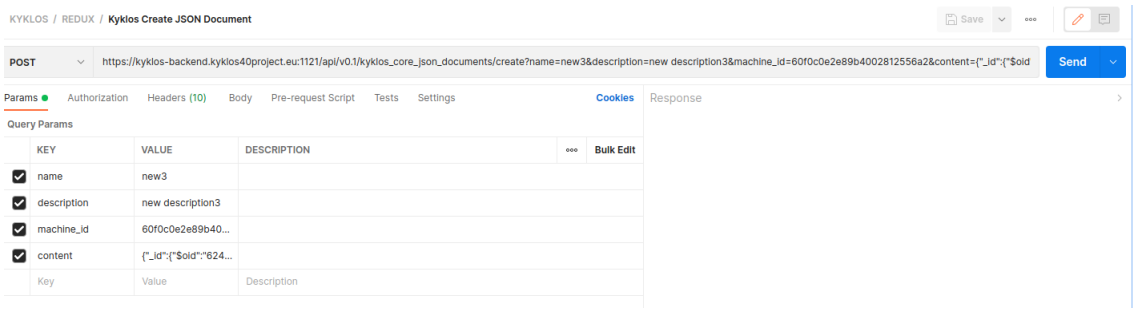


Figure 11: Entities scheme

Retrieval filters support wildcards (%), and can be accumulated in the List endpoint. Name, description and content can be updated by means of the PATCH endpoint. Single document deletion is possible by using DELETE endpoint. All individual endpoints require a valid document ID to success, specified in the URL.

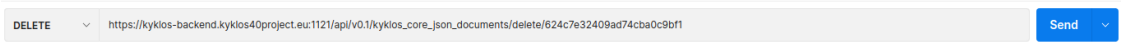


Figure 12: JSON Document action specification

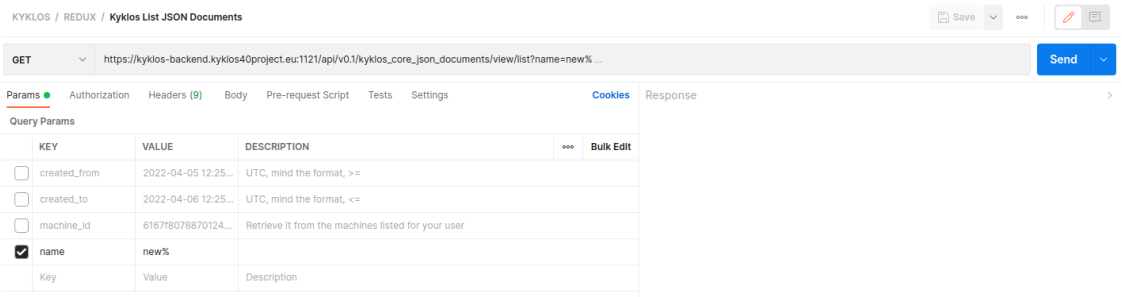


Figure 13: JSON Document filtering

6 Set up

The following settings are not within the reach of the basic user owned by partners. In any case, a brief description is given below. They consist of the complete configuration of the data extraction, specifying:

- The connection method: where the data is, what protocol is to be used and the necessary configurations for both connectivity and security.
- The data format: What structure the input data has got.
- The source of the data: The inconmin
- The entity that will group the data, in order to control access to the data and classify them conceptually.
- The signals to be mapped and their type: Which attributes are to be retrieved, and what to expect from them: Their ranges, their type, their position in the input structure, etc.
- The transformations on the previous values to be displayed to the user: Mathematical operations on them, decimal positions to be displayed, units, etc.

Entities are used to manage these configurations.

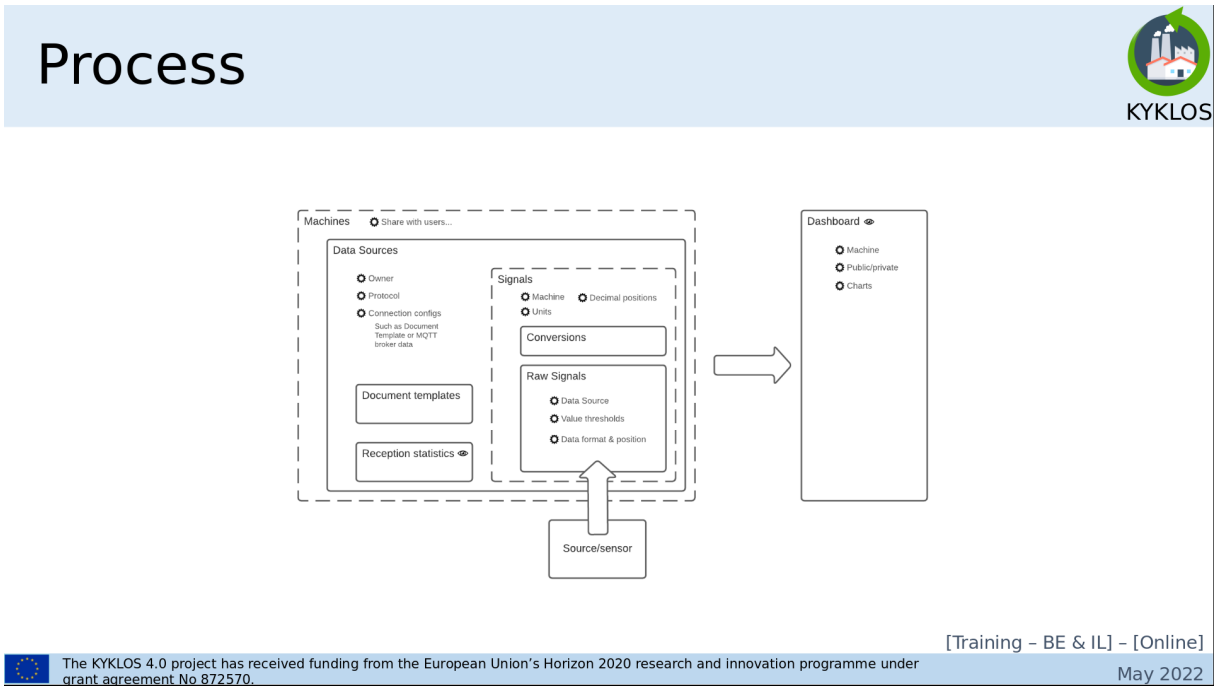


Figure 14: Entities scheme

The order of definition of these entities -the same as showed above- is not trivial, since, in spite of defining the same process, on the same data and under the same conditions, they form an architecture in which some entities depend on others for their definition.

7 Conclusions

We've briefly exposed within this document the interaction ways with both BE & IL components, adding context and concepts which can serve as a helping asset during the data fetch process. Hence, the target of audience of this document is the final user, or a technical user which want to check the status of the data.

The information in this document can help to understand how data are gathered and how to check these data, in order to perform the validation of a correct performance by the final users in a WP10 context, or as a general how-to guide.

In case of further assistance needed, contact at enrique.goyenechea@advanticys.com