User Manual

RINF

Document Information

Date:	Wed, 2025 Sep 17
Release Number:	2.0
Author(s):	European Union Agency for Railways
Reference:	

European Railway Agency, 120, rue Marc Lefrancq, F–59300 Valenciennes, France
Telephone: +33–(0)327–09 65 00 ▪ Fax: +33–(0)327–09 66 37
Website: http://www.era.europa.eu ▪ E-mail: servicedesk@era.europa.eu

Document History

Date	Author	Version	Comment	Modified Sections
Thu, 2014 Mar 6	European Dynamics	0.1	Document creation	All sections
Tue, 2015 Mar 10	European Dynamics	0.2	Updates for Release 1.1 – Iteration 2	All
Fri, 2017 Mar 24	Intrasoft International SA	0.3	Updates for Release 1.3.2	All
Wed, 2017 May 10	Intrasoft International SA	0.4	Updates for Release 1.3.3	All
Mon, 2019 Jul 1	Intrasoft International SA	0.5	Updates for Release 1.5	All
Mon, 2021 Aug 9	ERA	0.6	Updates for Release 1.6	2.4.4, 2.4.6, 2.5.4, 2.5.5
Fri, 2022 Dec 23	Universidad Politécnica de Madrid	0.7		2., 3., 4.
Mon, 2023 May 29	Universidad Politécnica de Madrid	0.8		5.
Thu, 2023, Nov 2	Universidad Politécnica de Madrid	0.9		
Wed, 2024, Feb 28	Universidad Politécnica de Madrid	1.0		
Mon, 2024, May 27	Universidad Politécnica de Madrid	1.1		Appendix A
Mon, 2024 Dec 16	Universidad Politécnica de Madrid	1.2	Updates on user roles and on ERA ontology v3.1.0	
Wed, 2025 Sep 17	European Union Agency for Railways	2.0	Updates on landing page, Data management and RU subscription types	2,7,8.2

Document Distribution List

Date	Name	Position	Organisation
Click here to enter a date.			
Click here to enter a date.			
Click here to enter a date.			

Document Review and Approval

	Date	Name	Signature
Author(s):	Mon, 2024 Dec 16	Universidad Politécnica de Madrid	
Reviewed by:	Click here to enter a date.		
Approved by:	Click here to enter a date.		

Table of Contents

1 Introduction

The present document is the User manual for the Register of Infrastructure (RINF) system. The Register of Infrastructure (RINF) software application is a web-based application facilitating at European level the access to the data of national registers of railway infrastructure. It has been developed on the basis of the specifications presented during the RINF meetings.

This platform enables any person to use a *Search Form* to retrieve information regarding the static rail network characteristics and capabilities of operational points and sections of lines, and to execute the *Route Compatibility Check (RCC)* where the objective is to check if a certain railway vehicle can travel the route between two operational points. Vehicle type information originated from the European Registry Authorized Type of Vehicle (ERATV). A *Map Explorer* is also available that displays Sections of Line(s) and Operational Point(s) over a map that can be zoomed to explore specific infrastructure elements.

The system also includes access to Data *stories*, which are a catalogue of SPARQL queries (together with their correspondence in natural language) that can be useful for different purposes to make use of the ERA Knowledge Graph (ERA KG). A *SPARQL endpoint* is capable of receiving and processing any query that is posed against ERA KG. Additionally, there is a link to the documentation of the current ERA *Ontology*.

For RINF System users that have specific roles such as IMs, NREs and RUs, there is the *Management* tab that in turn contains the *Data Assets Management* option to allow the user to load XML and RDF datasets and integrate them to the ERA KG, and the *Notifications Management* option to enable the subscription to system notifications when there are relevant updates to the data.

Section 2 of this User Manual contains the description of the RINF System Landing Page; Section 3 describes the functionalities of the *Search Form*; Section 4 describes the features of the *Route Compatibility Check* application; Section 5 describes the functionalities of the *Map Explorer*. Section 6 describes the use of the predefined queries in the *Data Stories* and the definition of queries in the SPARQL *Endpoint*; Section 7 describes the *Data Assets Management* option; finally, section 8 describes the *Notifications Management* option.

Following, we present the definitions, acronyms, and abbreviations used in this manual.

1.1 Definitions, acronyms, and abbreviations

- **ERA:** European Union Agency for Railways
- **ERA KG:** ERA Knowledge Graph
- **RINF**: Register of Infrastructure
- **NRE**: National Registry Entity, for setting up and maintenance of national register
- **IM**: Infrastructure Manager, in charge of directly submitting data to RINF application
- **RU**: Representative of a Railway Undertaking
- **MS**: Member State
- **OPs**: Operational Point(s)
- **PDF**: Portable Document Format
- **SoLs**: Sections of Line(s)
- **UI**: User Interface

1.2 User experience (system navigation and page composition)

The RINF system includes a web-based User interface and is accessible from any computer with an Internet browser and network accessibility. The system functionality is presented in a series of web pages which follow a standard template.

Each web page contains the following parts (see red outlines in Figure 1):

1. The Header contains the dropdown lists with the options that are available.
2. The Content section contains the actual content of the web page. The content of this section depends on the selection made from the header.
3. The footer contains links to information on the system's Terms of use, Privacy statement, and Copyright notice. The link to the User manual enables the User to download the User manual directly from the system.

Clicking on the ERA logo takes the User back to the Landing Page.

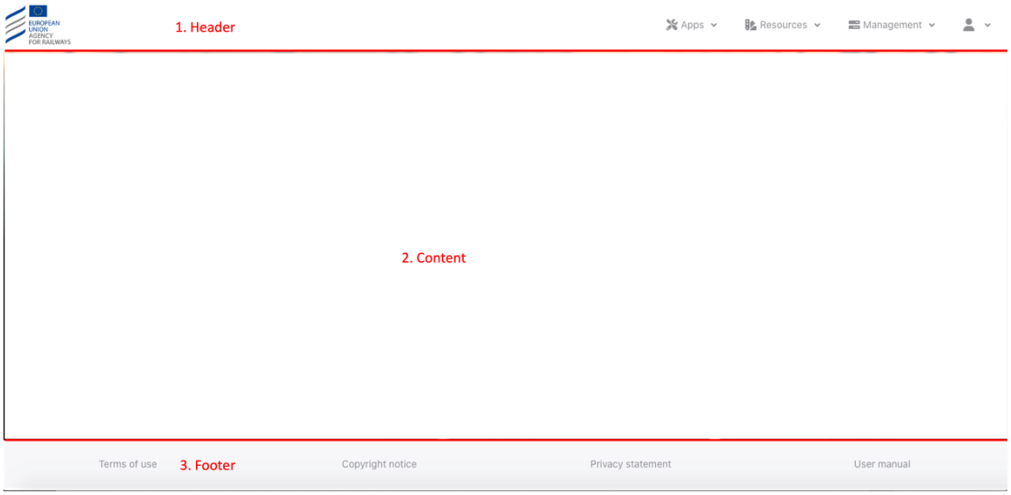


Figure 1. RINF web page structure

In this document, names for User interface tabs, views, menu selections, buttons, and text entry areas are highlighted like this.

2 Landing Page

The Landing Page can be seen in Figure 2.

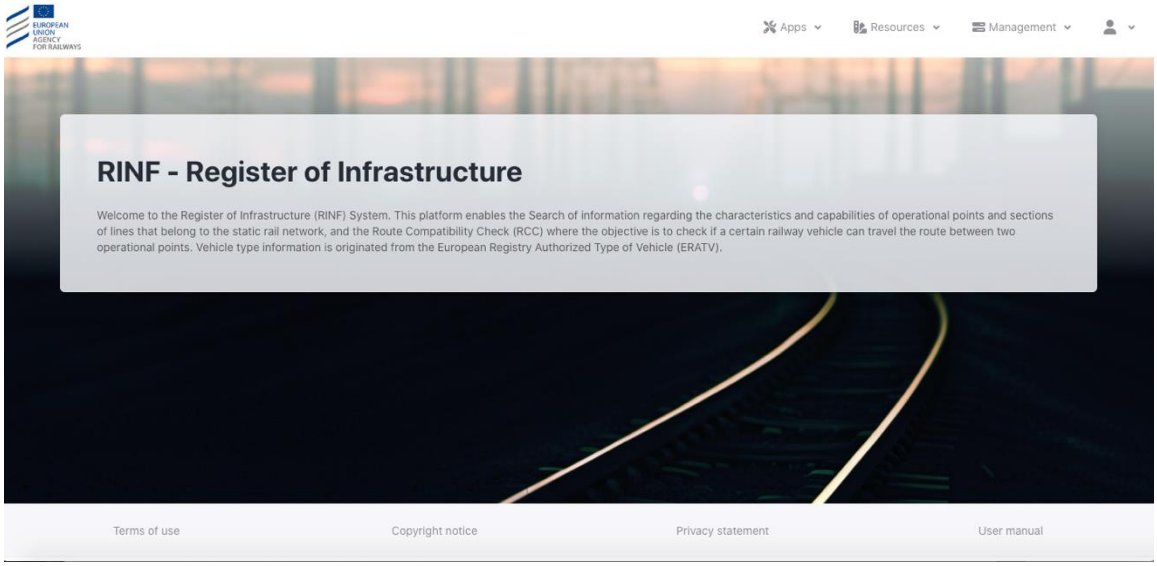


Figure 2. RINF Landing Page

The User has the following options in the RINF system:

- Header section dropdown lists:
  - Apps
    - Route Compatibility Check: Allows to check if a certain railway vehicle that belongs to a vehicle type, can travel the route between two operational points.
    - Map Explorer: Enables a view of the infrastructure elements on a map that can be zoomed. SoLs and OPs may be clicked to explore their parameters.
    - Search Form: Allows the User to search for the static rail network's characteristics and capabilities. Both OPs and SoLs may be searched.
    - Datasets Explorer: Most recently published datasets from European IMs
      For more tailored exploration, users can:
      - Use the Search page to filter results by specific criteria
      - Navigate the Map Explorer for spatial insights
      - Run SPARQL queries directly against the Endpoint for advanced data interrogation
  - Resources
    - Data Stories: Allows to navigate to a group of queries that are aligned with Use cases and their User stories.
    - Vocabulary: Allows to browse the documentation of the current ERA vocabulary.
    - Endpoint: Allows the User to specify a SPARQL query to the ERA KG.
  - Management
    - Data Assets Management and Notification Management (only for registered users)
  - (Users)
    - Before signing in
      - Sign In
      - Reset Password

        - Once the user has signed in
            - Choose user's Role
            - Sign Out
- Footer
    - User manual: Allows you to download the User manual.
    - Terms of use: Binding terms of use for any User of the system.
    - Privacy statement: Description of ERA's Data Protection policy.
    - Copyright notice: Link to ERA's Disclaimer and Copyright Notice.

2.1 Sign In

When the User clicks Sign In in the Users dropdown list , a screen will appear with all of the User's ERA accounts allowing the User to add a new account. There are three roles that may have been assigned to a User:

1. IM - Infrastructure Manager: Allowed to manage datasets that are owned by the specific company or organization.
2. RU - Railway Undertaker: Allowed to receive notifications when RINF parameters are updated.
3. NRE – National Representative: Acts as point of contact between the Agency and the infrastructure managers in the view of assisting and coordinating the infrastructure managers. They can represent infrastructure managers in their role of data providers to the RINF Application a User may have both roles as IM and NRE.

Once the User has signed in, the role must be chosen as shown in Figure 3 (in the case of the figure, the User has the three potential roles, but in other cases the User may have one or two roles). If the User has already signed in before, the latest selected role will be chosen by default.

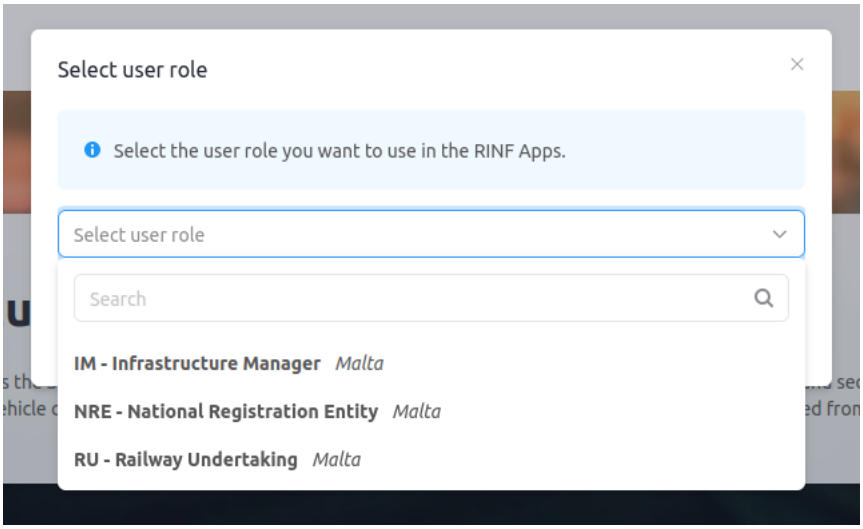


Figure 3. Choosing the User's role

3 Search for RINF Data

To search for RINF Data, the User clicks on the Search Form link available on the header section. Then, the Search form initial page is displayed. The User shall select one of the options in the Search object dropdown list (see Figure 4):

- Operational Point
- Section Of Line

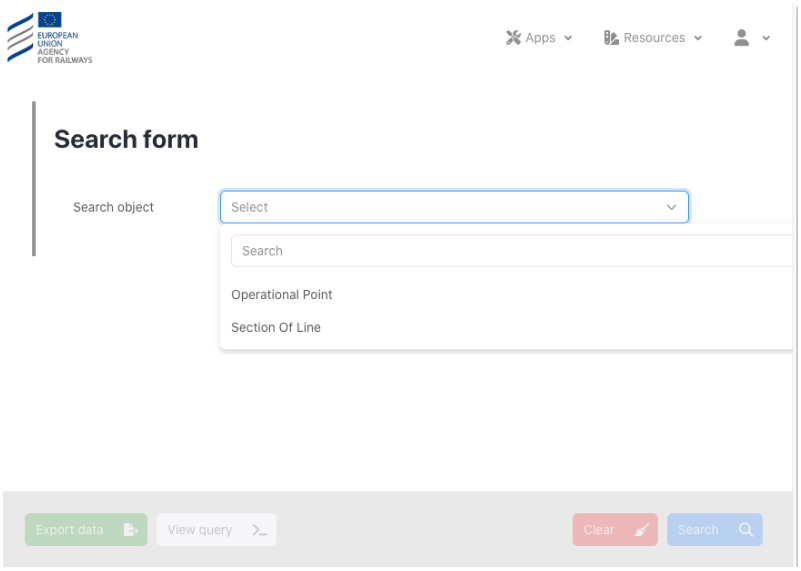


Figure 4: "Search RINF data" initial page

If no Member state is selected, then the search will be done for all member states. The User may directly select the member states for which the search will be conducted (see Figure 5).

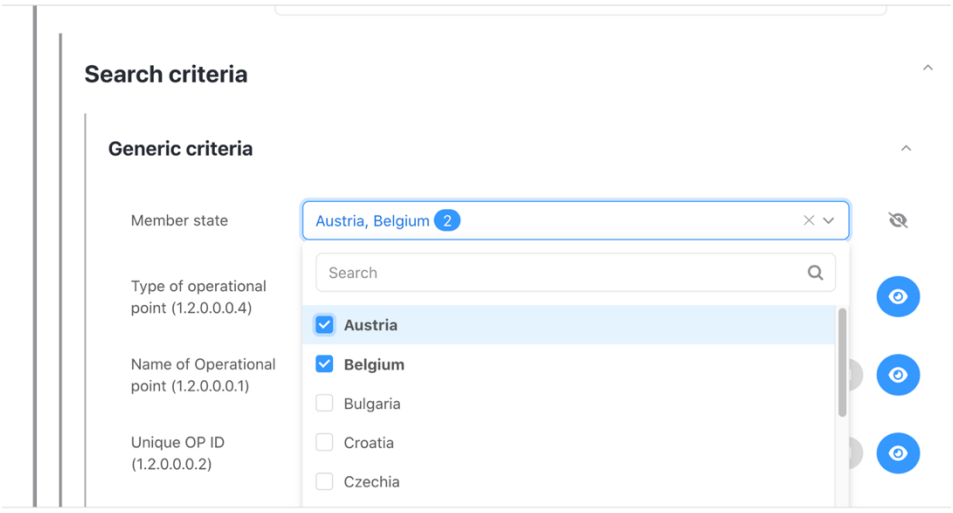


Figure 5: Searching by Member State

3.1 Search form and criteria visualization

As soon as the User selects either OPs or SoLs from the Search object dropdown list, the search form is updated according to the User's selection. The search fields are divided into sections that group related parameters, e.g., Track parameters, Tunnel parameters. The User can collapse or expand the sections of the search form

in order to save space on the page (see Figure 6). Even if a User selects to collapse a section, the values that the User has already inserted in the searching fields of this section will be maintained.

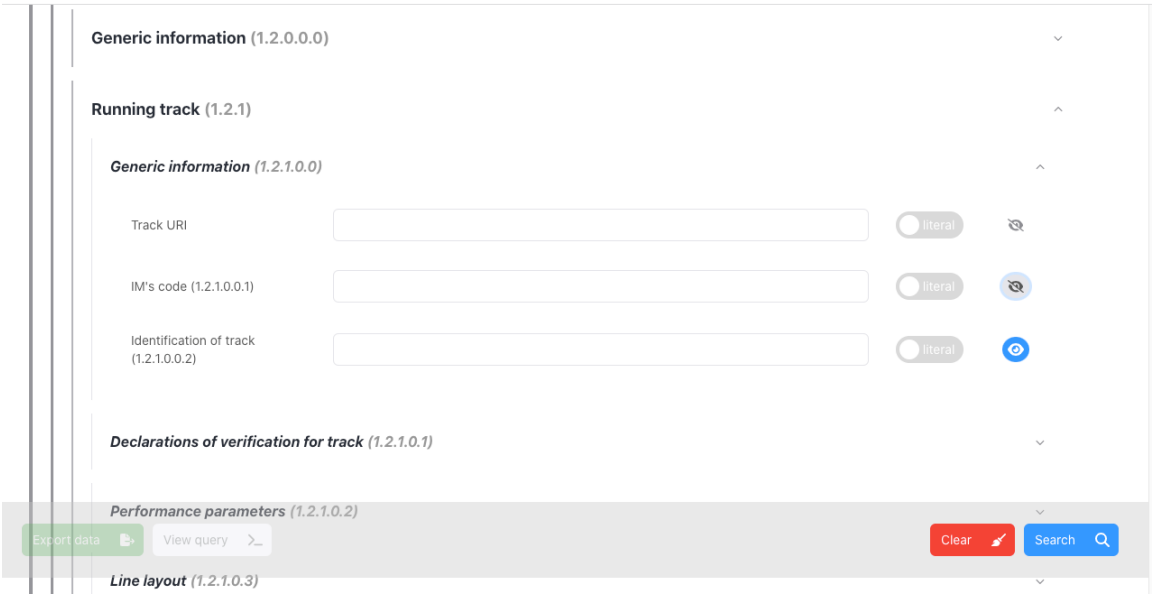


Figure 6. Expanding and collapsing sections of the search form

For each search criteria, the User may click the icon on the right side of the parameter to display the search criteria value. For example, in Figure 6, the Identification of track will be displayed in the results. The following attributes are displayed by default:

Operational Point

Generic criteria: Name of Operational point (1.2.0.0.0.1), Unique OP ID (1.2.0.0.0.2), Type of operational point (1.2.0.0.0.4), Geographical location of Operational Point (1.2.0.0.0.5).

Validity date: Validity start date, Validity end date

Section of Line

Generic criteria: Infrastructure manager's Code (1.1.0.0.0.1), National line ID (1.1.0.0.0.2).

Operational point at start of section of line: Name of start operational point (1.2.0.0.0.1), Geographical location of start operational point (1.2.0.0.0.5).

Operational point at end of section of line: Name of end operational point (1.2.0.0.0.1), Geographical location of operational point (1.2.0.0.0.5).

3.2 Search criteria types

There are the following types of search criteria:

3.2.1 Numeric

Numeric criteria that can be searched as exactly a value or between a range of values (see Figure 7). The User may select either option by clicking on the corresponding right-hand button (the default is exactly). For the exactly option, the User may type directly the value. If the User picks the between option, he can fill in both the from and to fields to search all OPs/SoLs for which the corresponding field has a value that

falls within the specified range of values. The border values are included. In essence, the search is defined as: *Greater than or equal than the "from" value AND less than or equal than the "to" value.*

Figure 7. Numeric criteria search by range of values

Additionally, the User can fill in only the from field, to search all OPs/SoLs for which the corresponding field has a value greater than or equal than the specified from value, or only the to field to search all OPs/SoLs for which the corresponding field has a value smaller than or equal with the specified to value.

3.2.2 Character string

Character string criteria that can be searched as a literal, i.e. as exactly a value that matches the string, or as regex (regular expression), i.e. a sequence of characters that specifies a search pattern in a text.

Regular expressions used in the Search form follow the SPARQL 1.1 Query Language specification that in turn indicates that the regex language is defined in XQuery 1.0 and XPath 2.0 Functions and Operators.

Part of the language to express regular expressions can be seen in the following table:

abc…	Letters
123…	Digits
\d	Any Digit
\D	Any Non-digit character
.	Any Character
\.	Period
[abc]	Only a, b, or c
[^abc]	Not a, b, nor c
[a-z]	Characters a to z
[0-9]	Numbers 0 to 9
\w	Any Alphanumeric character
\W	Any Non-alphanumeric character
{m}	m Repetitions
{m,n}	m to n Repetitions
*	Zero or more repetitions
+	One or more repetitions
?	Optional character
\s	Any Whitespace
\S	Any Non-whitespace character
^…$	Starts and ends
(…)	Capture Group
(a(bc))	Capture Sub-group
(.*)	Capture all
(abc|def)	Matches abc or def

3.2.2.1 Examples of regular expressions

1. A simple pattern that is used frequently is to search for a value that contains a certain character string. In this example the User searches for all OPs where the name contains the character string "Amsterdam"

(see Figure 8). In the search form, all regex are case insensitive, thus the search will match any combination of uppercase and lowercase characters.

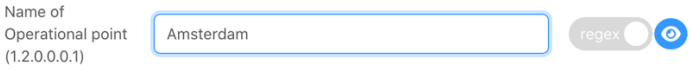


Figure 8. Searching by a regex pattern: the OP's name contains the character string Amsterdam.

Page 2 of the results of this search can be seen in Figure 9.

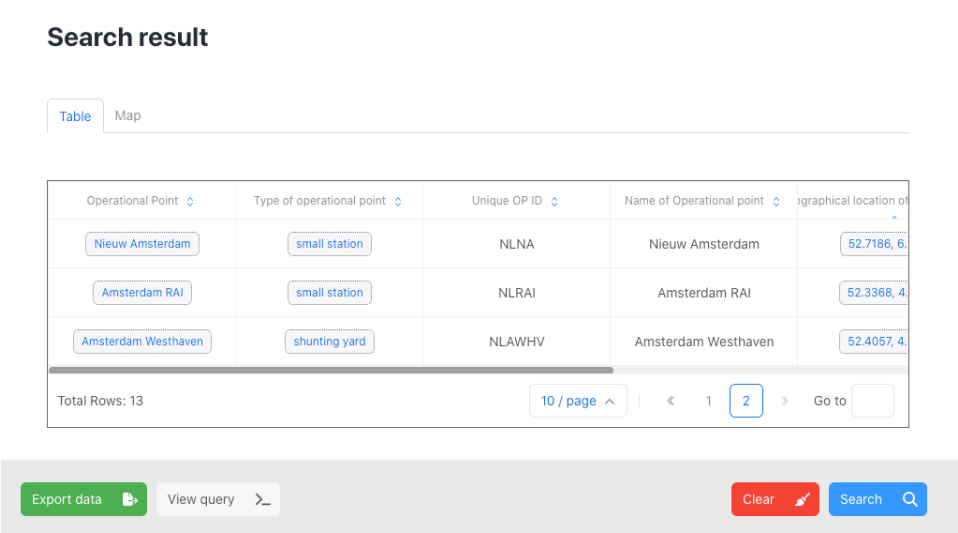


Figure 9. Excerpt of the result of the search by a regex pattern: the OP's name contains the character string Amsterdam.

2. The User searches for a value that starts with a certain character string. In this example the User searches for all OPs in the Netherlands where the Unique OP ID starts with "EU" (see Figure 10).

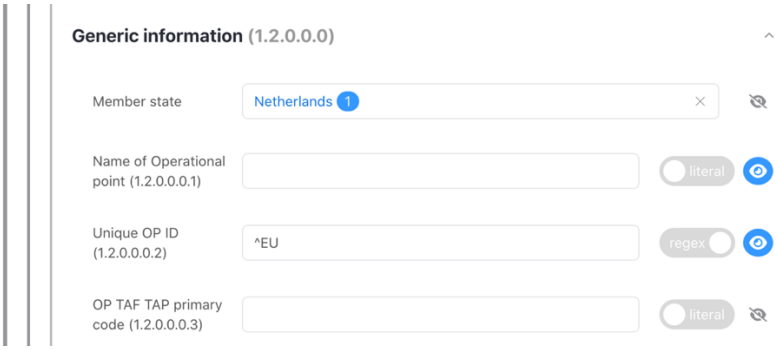


Figure 10. Searching by a regex pattern: the OP's ID starts with the character string "EU".

The results of this search are shown in Figure 11.

Figure 11. Excerpt of the result of the search by a regex pattern: OPs in the Netherlands where the Unique OP ID starts with the string "EU".

3. The User searches for a value that can contain one of two possible character strings, i.e. use of the logical OR (|) connector. In this example the User searches for all OPs in the Netherlands that contains either the character string “Centrum” or the character string “Centraal” (see Figure 12).

Figure 12. Searching by a regex pattern: the OP’s name contains either the character string “Centraal” or the character string “Centrum”.

The results of this search can be seen in Figure 13.

Operational Point	Type of operational point	Unique OP ID	Name of Operational point	ographical location of Operational P	Validity start date
Arnhem Centraal	passenger terminal	NLAH	Arnhem Centraal	51.9847, 5.9015	26/06/2023
Almere Centrum	station	NLALM	Almere Centrum	52.3751, 5.2181	06/04/2023
Amsterdam Centraal	passenger terminal	NLASD	Amsterdam Centraal	52.3791, 4.9008	10/08/2023
Barneveld Centrum	station	NLBNC	Barneveld Centrum	52.1399, 5.5903	06/04/2023
Betuweroute Centraal Uitwisselpunt Valburg	depot or workshop	NLBRCUP	Betuweroute Centraal Uitwisselpunt Valburg	51.898, 5.8018	06/10/2023
Den Haag Centraal	passenger terminal	NLGVC	Den Haag Centraal	52.0809, 4.3245	06/04/2023

Figure 13. Excerpt of the result of the search by a regex pattern: OPs in the Netherlands where the name contains either the character string “Centraal” or the character string “Centrum”.

4. The User searches for a value that contains certain classes of characters. In this example the User searches for OPs in Spain where the name contains information on its kilometre point (see Figure 14).

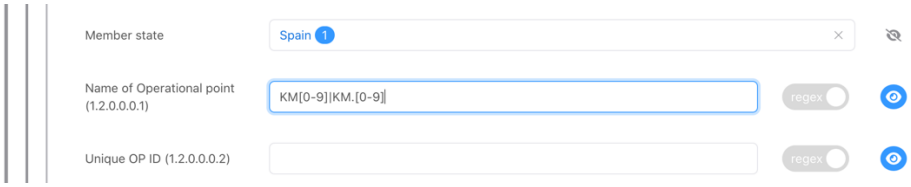


Figure 14. Searching by a regex pattern: the OP’s name contains numeric information on its Kilometer point.

The result of this search can be seen in Figure 15.

Operational Point	Type of operational point	Unique OP ID	Name of Operational point	Geographical location of Operational Point
TABOADELA AG.KM.446,1	station	ESA8248	TABOADELA AG.KM.446,1	42.2539, -7.81104
TABOADELA AG.KM.447,1	switch	ESA8249	TABOADELA AG.KM.447,1	42.2551, -7.82262
EL REGUERON AG KM 522,1	junction	ESA6133	EL REGUERON AG KM 522,1	37.9899, -1.05733
MURCIA DEL CARMEN AG. KM 526,8	junction	ESA6135	MURCIA DEL CARMEN AG. KM 526,8	37.9698, -1.10335
KM 141,417	junction	ESC0111	KM 141,417	37.1375, -4.29785
FUENCARRAL-AGUJA KM 4,5	junction	ESA1702	FUENCARRAL-AGUJA KM 4,5	40.512, -3.683
BENALMADENA - AGUJA KM 19,0	junction	ESA5450	BENALMADENA - AGUJA KM 19,0	36.6008, -4.5307
YELES AGUJA KM.34,397	junction	ESA6000	YELES AGUJA KM.34,397	40.1466, -3.7647

Figure 15. Excerpt of the result of the search by a regex pattern: OPs in Spain where the name contains information on its Kilometer point.

5. The user searches a pattern that matches any character. In this example the User searches for all OPs with tracks that have at least one platform (see Figure 16). In this case the User checks that the platform Id contains any character (the use of "." in a regex matches any character). The results of this search can be seen in Figure 17.
   Note that if no pattern is specified, the search returns a large set of answers where many of the tracks do not have any platform data (see Figure 18).

Figure 16. Searching by a regex pattern: all Ops with tracks that have at least one platform.

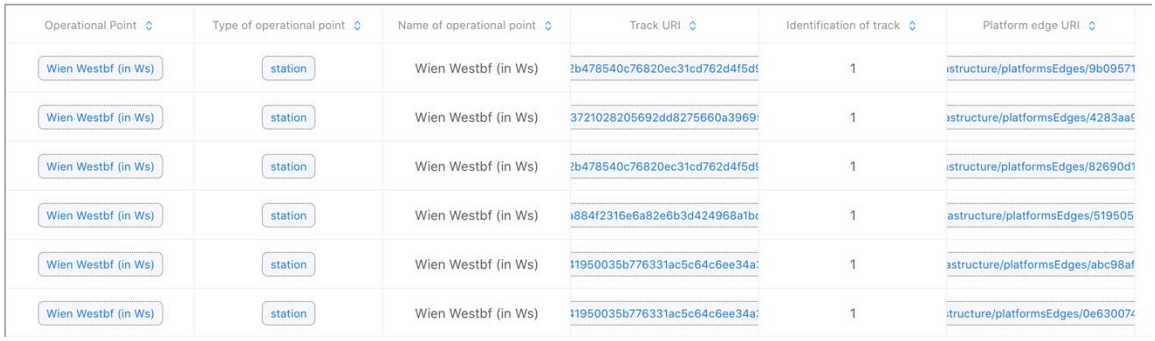


Figure 17. Excerpt of the results of a search by a regex pattern: all OPs with tracks that have at least one platform.

Figure 18. Searching for all OPs, tracks and platforms. Tracks with no platforms are also displayed.

3.2.3 Boolean

Boolean criteria, where the User may indicate if the search is for all the values that are True (see Figure 19) or all the ones that are False.

Figure 19. Boolean criteria search for those OPs where the parameter is True

In the specific case of Validity date, the User may specify values in Validity start date and Validity end date or choose the boolean criteria for Currently valid infrastructure elements, i.e. an OP. Currently valid set to True will search for all elements where the current date falls in the range of validity start and end dates. Setting this criterion to False will allow the User to search for elements that either have expired or are future plans. The Validity date criteria are shown in Figure 20.

Figure 20. Validity date criteria

3.2.4 List of predefined values

List of predefined values search where the User may check one or more of the predefined values in a dropdown list. He also may search the dropdown list by a character string and then check one or more of the values that are returned (see Figure 21).

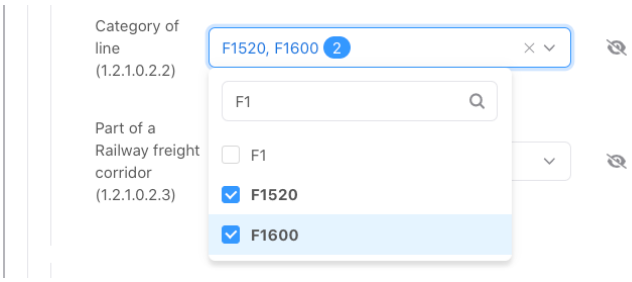


Figure 21. Search on a predefined set of values for Category of line

3.2.5 Location

Location criteria where the User clicks on the Map option (see Figure 22). Once the map is displayed the User may mark two locations (points) in the map that are the opposite corners of a bounding box that is displayed (see Figure 23). Once the map is closed, the values of the two pairs of coordinates are shown in the search form (lat0, long0) and (lat1, long1) as can be seen in Figure 24. The search will be executed for locations that fall within the bounding box.

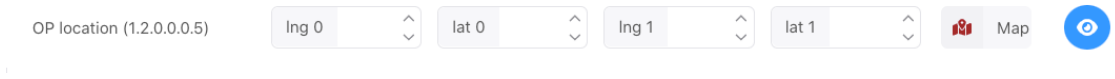


Figure 22. Search by OP location

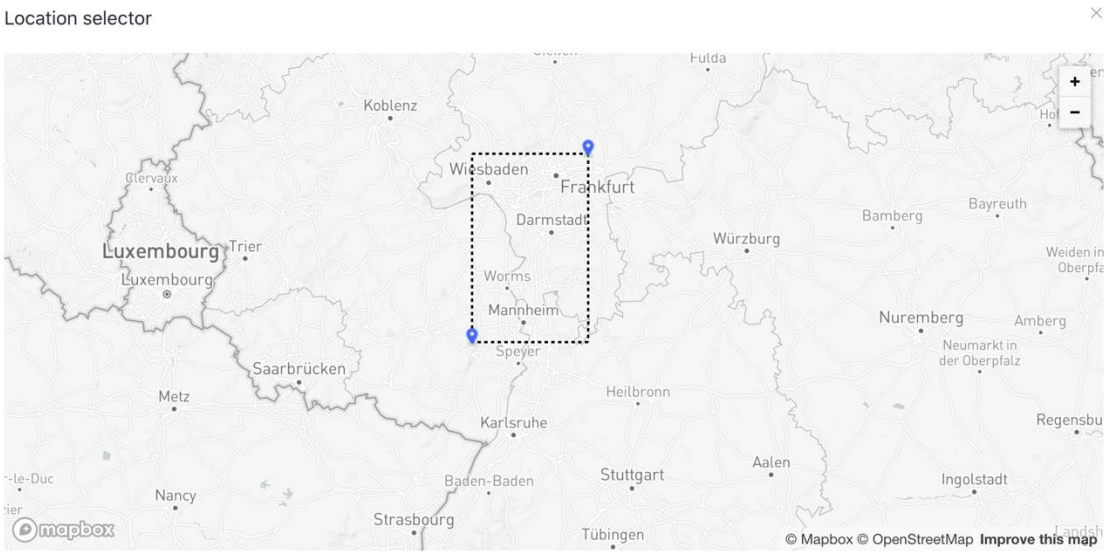


Figure 23. Bounding box for the search by location

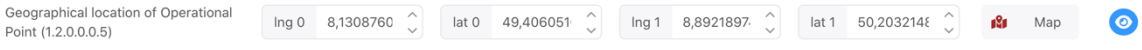


Figure 24. Coordinates of the bounding box for a search by location

3.2.6 Validity period

Validity period where the values of the validity start date and the validity end date may be introduced directly in the format dd/mm/yyyy or by clicking on the calendar icon. Additionally, the User may choose to search by Currently valid criteria, where the search for True will return those results where the current date falls within their validity period (see Figure 25).

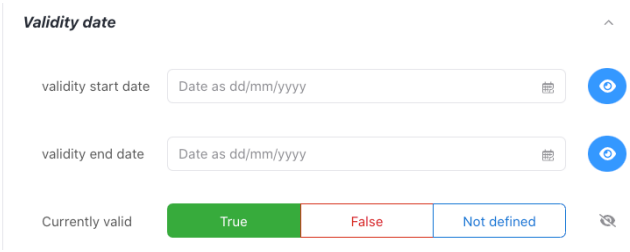


Figure 25. Searching the Currently valid elements

3.3 Search and results

To initiate the search process, the User may fill in any of the available search criteria and click on the Search button. The summary of the search criteria (fields selected and filters applied) appears above the space used for the search results (see Figure 26).

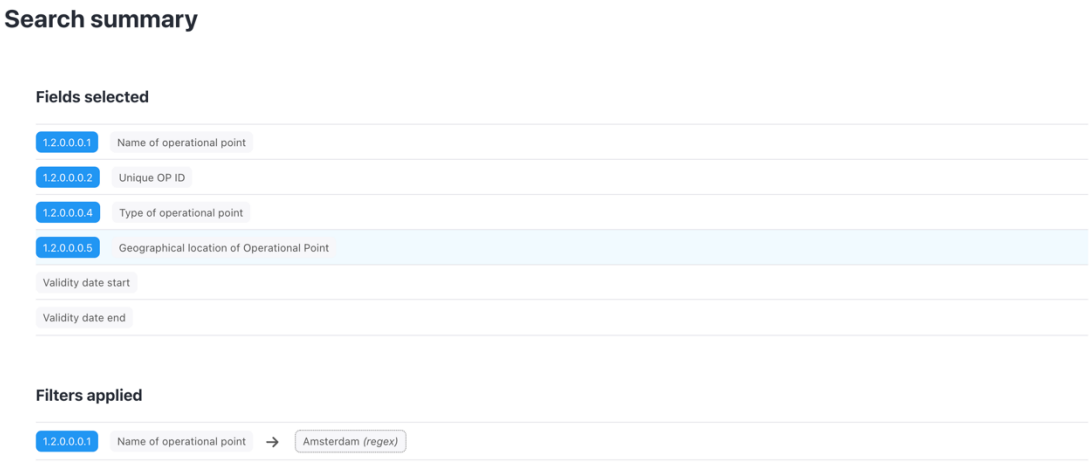


Figure 26. Search summary after selecting fields and specifying search criteria.

The User may also Clear all the search criteria values in the search form. Search results may be viewed as a Table or on a Map (see Figure 27).

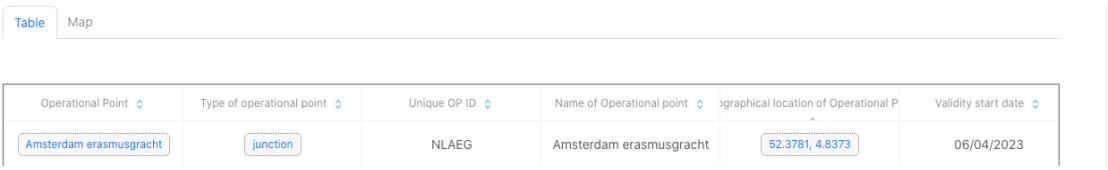


Figure 27. Search results can be displayed as a Table or on a Map.

3.3.1 Search summary

3.3.2 View search results as a Table

Once the Search button is clicked, the respective search results are displayed (by default) as a Table (see Figure 28).

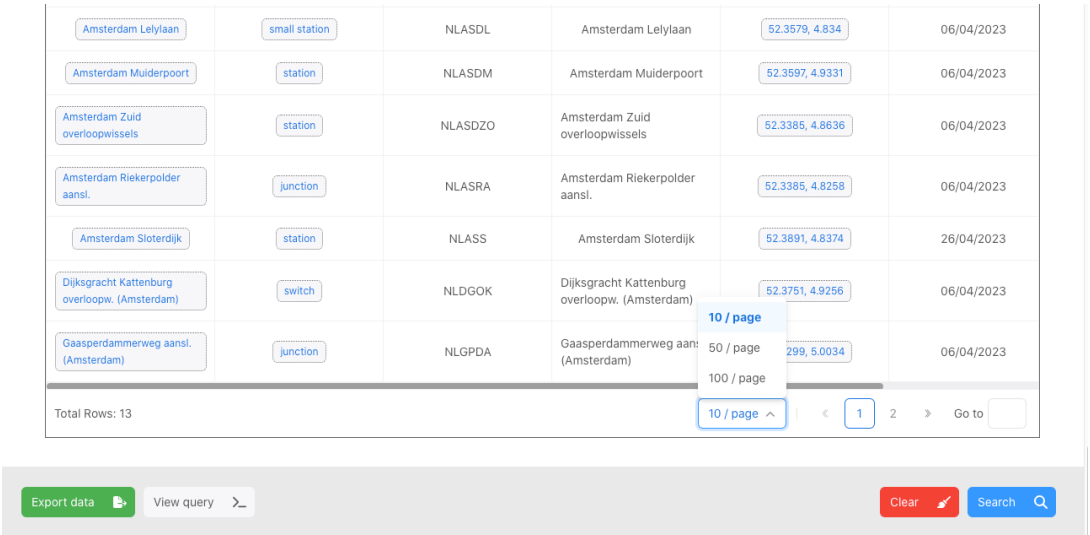


Figure 28. Excerpt of Page 1 of the Table of results on OPs where the name contains the character string “Amsterdam”

The search results are grouped into pages of 10 results. At the bottom of the list of results, the User may choose to display more results per page, by clicking beside the 50 / page or the 100 / page in the dropdown

list to change to 50 or 100 items per page. To go to a specific page, the User may click on the page number or Go to a specific page at the bottom right-hand side of the list of results (see Figure 28).

The results can be sorted in ascending or descending order by clicking on any of the available columns of the search results list (Figure 29).

Operational Point	Type of operational point	Unique OP ID	Name of Operational point	ographical location of Operational P	Validity start date
Amsterdam erasmusgracht	junction	NLAEG	Amsterdam erasmusgracht	52.3781, 4.8373	06/04/2023
Amsterdam Riekerpolder aansl.	junction	NLASRA	Amsterdam Riekerpolder aansl.	52.3385, 4.8258	06/04/2023
Gaasperdammerweg aansl. (Amsterdam)	junction	NLGPDA	Gaasperdammerweg aansl. (Amsterdam)	52.3299, 5.0034	06/04/2023
Amsterdam Centraal	passenger terminal	NLASD	Amsterdam Centraal	52.3791, 4.9008	10/08/2023
Amsterdam Lelylaan	small station	NLASDL	Amsterdam Lelylaan	52.3579, 4.834	06/04/2023

Figure 29. Sorting the results in ascending order by Type of operational point.

3.3.2.1 *View search result details*

In order to view the details of a search result, the User clicks on the "Label" link of the desired search result (see first column of Figure 29), Then a new window is displayed containing all the attributes and values for this element (see Figure 30).

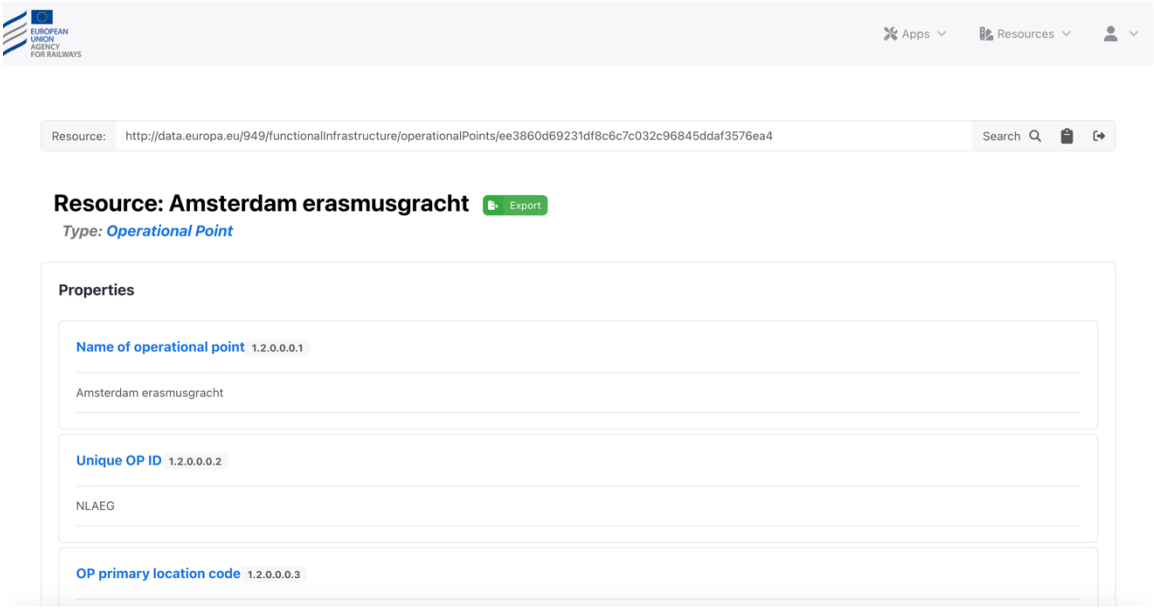


Figure 30. Detail of the Amsterdam erasmussgracht OP: Attributes and their Values

He may further click on some the values that are hyperlinks and obtain information on those elements (see Figure 31).

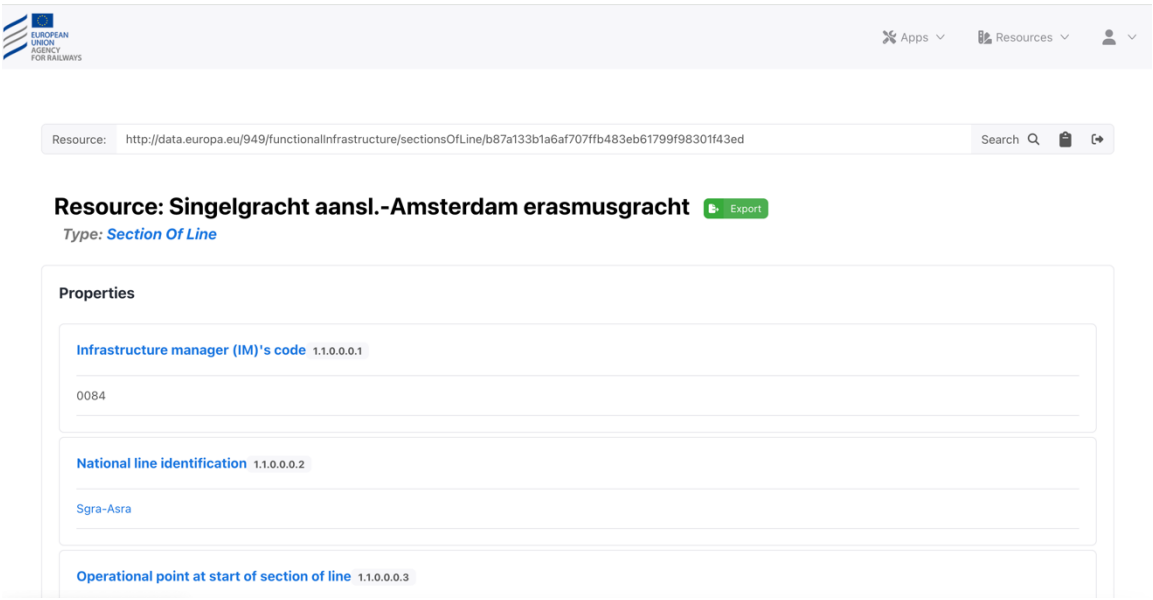


Figure 31. Information on the SoL Singelgracht aansl.-Amsterdam erasmusgracht where the Amsterdam erasmusgracht is the OP at the end of the SoL

The User may also click on any of the attribute names, obtaining detailed information on the attribute itself which is part of the ERA ontology (see Figure 32).

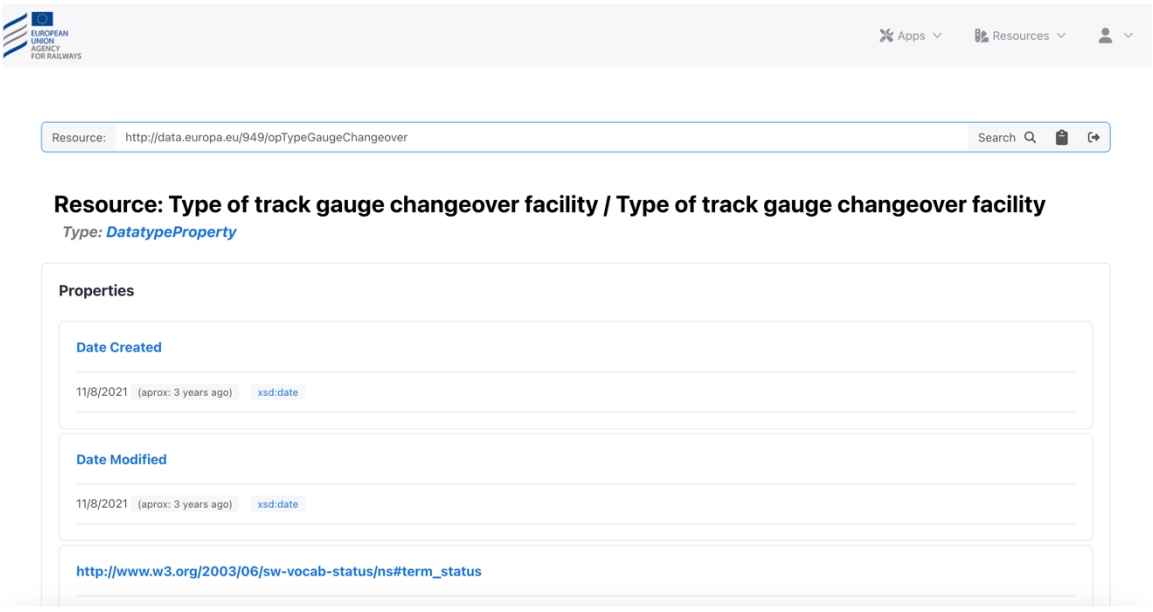


Figure 32. Information on the attribute Type of track gauge changeover facility

In the search results table, when the User clicks on labels for values that belong to a predefined list (taxonomies), another window with the details on the taxonomy concept will be displayed (see Figure 33).

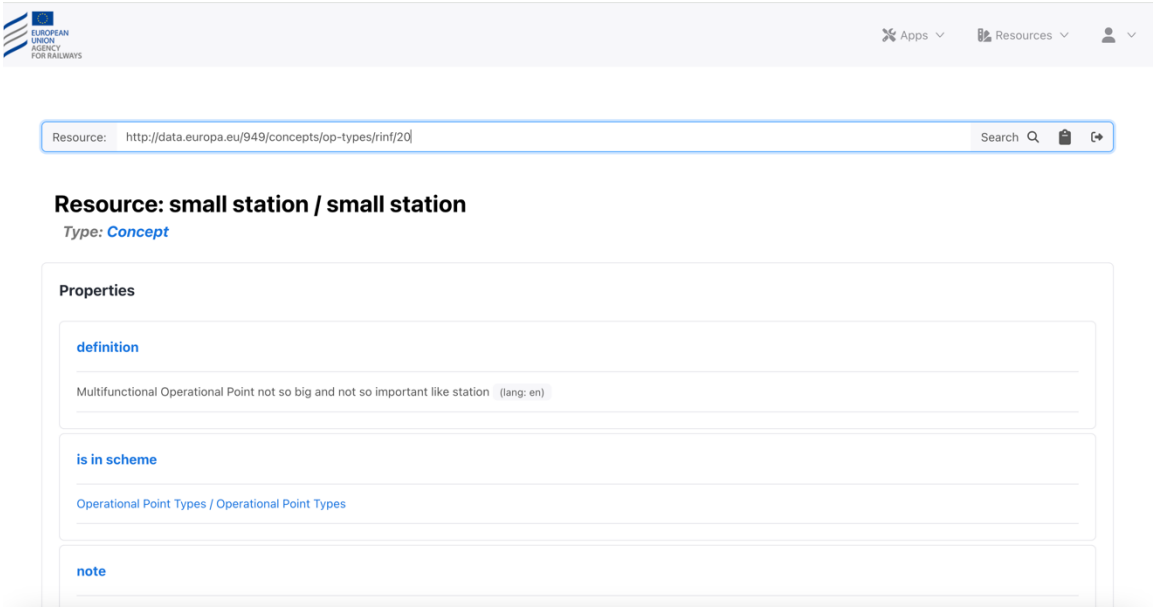


Figure 33. Detail of the concept small station that belongs to the taxonomy of Operational Point Types

3.3.3 View search results on a map

The User may view the search results on the map by selecting the Map tab. In order to navigate to a specific area of the map, the User can use the mouse for zooming or use the zoom in / zoom out buttons on the top right of the map.

Zooming in may be necessary to see all of the results, especially in cases that they are located nearby (see Figure 34).

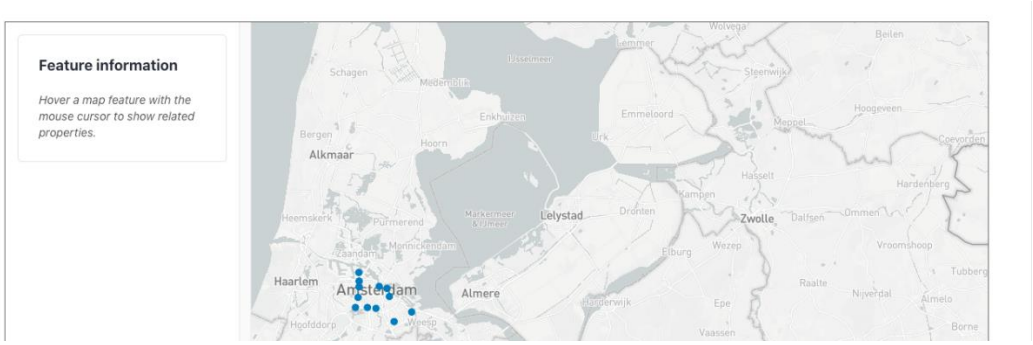


Figure 34. The map of results of a search on OPs that contain the character string "Amsterdam"

When doing a mouseover on an OP (blue dot) it is highlighted in pink and query result properties related to the OP will be displayed on the left of the map (see Figure 35).

Figure 35. Query result properties of an OP.

Similarly, SOLs Search results are displayed on a map as black lines. When doing a mouse over, the SOL is highlighted in blue and delimited by its start and end OPs and its query result properties are displayed on the left of the map (see Figure 36).

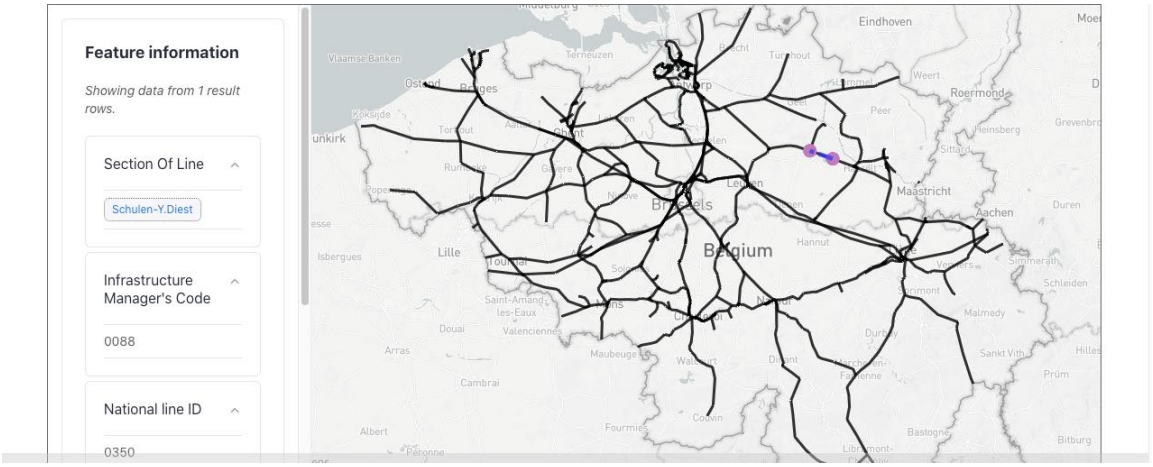


Figure 36. Query result properties of a SoL.

3.4 View query

The User may view the query that represents the search criteria that has been specified by clicking View query >_. The SPARQL query will be displayed (see Figure 37).

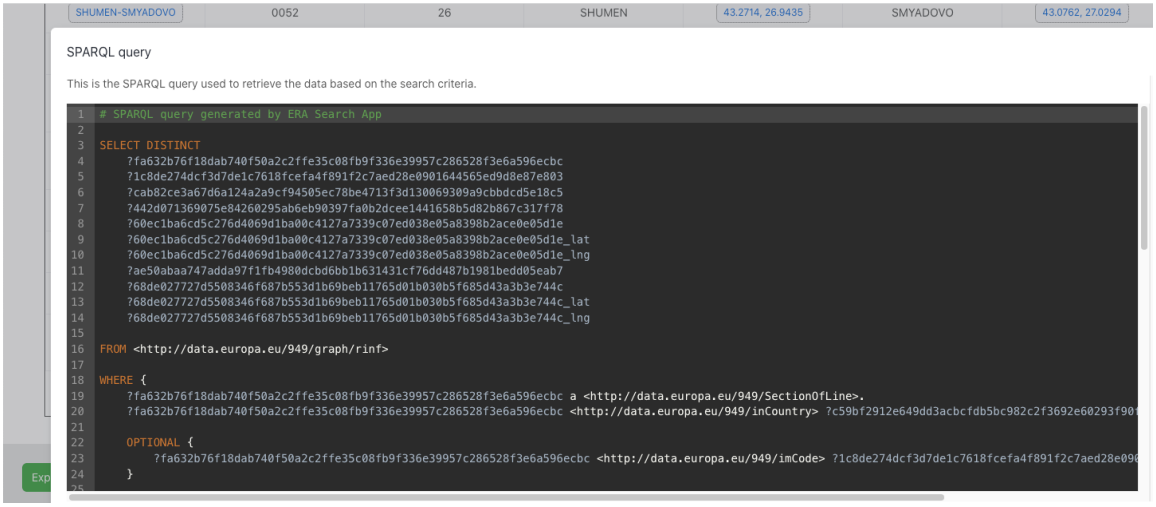


Figure 37. SPARQL query that corresponds to the search of OPs that contain the character string "Amsterdam".

3.5 Export search results

For the search that has been illustrated in Figure 9 , the User can click the Export data button on the results (Figure 28) The user can choose to export data as Tabular data, as Linked data or as Linked data full dump. Tabular data may be Excel XLSX or JSON whereas Linked data and Linked data full dump can be RDF/XML, N-Triples or JSON-LD RDF encoding formats (see Figure 38).

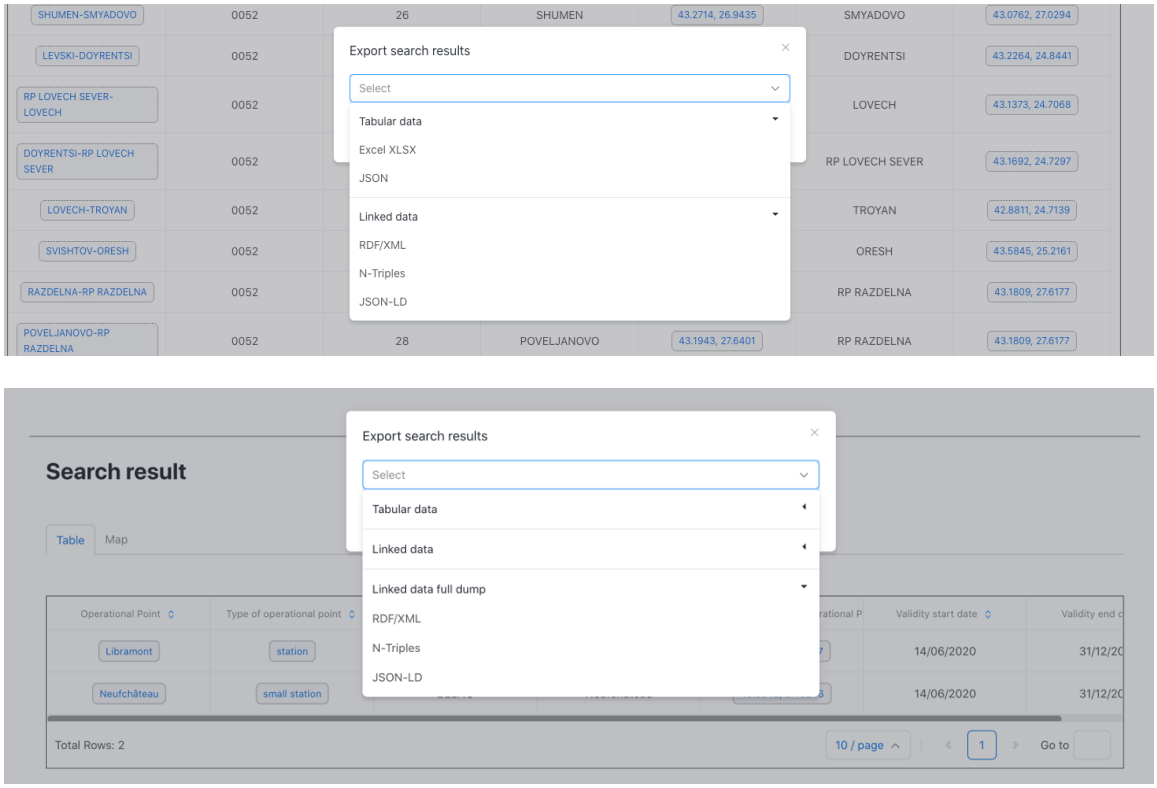


Figure 38. Options for exporting Search results.

If the Linked data option is chosen, only the columns that are viewed in the results table will be exported. The Linked data full dump option exports all the properties of the OPs or SoLs that appear in the result.

For example, when the User chooses the Excel XLSX option, the results are exported as an Excel spreadsheet where each column corresponds to a column in the results table (see Figure 39).

I17

	A	B	C	D	E	F	G
1	**Operational Point**	**Type of operational point**	**Unique OP ID**	**Name of Operational point**	**Geographical location of Operational Point**	**Validity start date**	**Validity end date**
2	Amsterdam erasmusgracht	junction	NLAEG	Amsterdam erasmusgracht	52.3781, 4.8373	06/04/2023	31/12/2023
3	Amsterdam Riekerpolder aansl.	junction	NLASRA	Amsterdam Riekerpolder aansl.	52.3385, 4.8258	06/04/2023	31/12/2023
4	Gaasperdammerweg aansl. (Amsterdam)	junction	NLGPDA	Gaasperdammerweg aansl. (Amsterdam)	52.3299, 5.0034	06/04/2023	31/12/2023
5	Amsterdam Centraal	passenger terminal	NLASD	Amsterdam Centraal	52.3791, 4.9008	19/06/2023	31/12/2023
6	Amsterdam Lelylaan	small station	NLASDL	Amsterdam Lelylaan	52.3579, 4.834	06/04/2023	31/12/2023
7	Nieuw Amsterdam	small station	NLNA	Nieuw Amsterdam	52.7186, 6.8487	06/04/2023	31/12/2023
8	Amsterdam RAI	small station	NLRAI	Amsterdam RAI	52.3368, 4.8907	06/04/2023	31/12/2023
9	Dijksgracht Kattenburg overloopw. (Amsterdam)	switch	NLDGOK	Dijksgracht Kattenburg overloopw. (Amsterdam)	52.3751, 4.9256	06/04/2023	31/12/2023
10	Amsterdam Westhaven	shunting yard	NLAWHV	Amsterdam Westhaven	52.4057, 4.8365	26/04/2023	31/12/2023
11	Amsterdam Bijlmer	station	NLASB	Amsterdam Bijlmer	52.3114, 4.948	03/07/2023	31/12/2023
12	Amsterdam Muiderpoort	station	NLASDM	Amsterdam Muiderpoort	52.3597, 4.9331	06/04/2023	31/12/2023
13	Amsterdam Zuid overloopwissels	station	NLASDZO	Amsterdam Zuid overloopwissels	52.3385, 4.8636	06/04/2023	31/12/2023
14	Amsterdam Sloterdijk	station	NLASS	Amsterdam Sloterdijk	52.3891, 4.8374	26/04/2023	31/12/2023
15							

Figure 39. Search results in Excel XLSX format.

If the User chooses JSON, the results are presented in this format where the file includes the heading and the data as shown in Figure 40.

```
{
  "head": [
    {
      "key": "era:OperationalPoint",
      "label": "Operational Point",
      "uri": "http://data.europa.eu/949/OperationalPoint"
    },
    {
      "key": "era:OperationalPoint>era:opType",
      "label": "Type of operational point",
      "rinf_index": "1.2.0.0.0.4",
      "uri": "http://data.europa.eu/949/opType"
    },
    {
      "key": "era:OperationalPoint>era:uopid",
      "label": "Unique OP ID",
      "rinf_index": "1.2.0.0.0.2",
      "uri": "http://data.europa.eu/949/uopid"
    },
    {
      "key": "era:OperationalPoint>era:opName",
      "label": "Name of operational point",
      "rinf_index": "1.2.0.0.0.1",
      "uri": "http://data.europa.eu/949/opName"
    },
    {
      "key": "era:OperationalPoint>wgs:location",
      "label": "Geographical location of Operational Point",
      "rinf_index": "1.2.0.0.0.5",
      "uri": "http://www.w3.org/2003/01/geo/wgs84_pos#location"
    },
    {
      "key": "era:OperationalPoint>era:validityStartDate",
      "label": "Validity start date",
      "uri": "http://data.europa.eu/949/validityStartDate"
    },
    {
      "key": "era:OperationalPoint>era:validityEndDate",
      "label": "Validity end date",
      "uri": "http://data.europa.eu/949/validityEndDate"
    }
  ],
  "data": [
```


Figure 40. Excerpt of the Export results in Plain JSON format.

Similarly, if the User chooses Linked data full dump, and then the N-Triples option, the results are presented in this format as shown in Figure 41.


```
<http://data.europa.eu/949/functionalInfrastructure/tracks/b15cbac87b0d48887c009553feec532f40449a67>
<http://data.europa.eu/949/canonicalURI>    <http://data.europa.eu/949/functionalInfrastructure/tracks/BELRB_Libramont%20-%20CDS%20224%20-%20Section%204302> .
<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/d3c29cc19b3a17bb827387a0c926c656908711e1>
<http://data.europa.eu/949/notYetAvailable> <http://data.europa.eu/949/platformHeight> .
<http://data.europa.eu/949/functionalInfrastructure/tracks/b571c66c507921ac2cdeda9970fa199a51e2a2dc>
<http://data.europa.eu/949/tenClassification>   <http://data.europa.eu/949/concepts/ten-classifications/rinf/40> .
<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/62b9db5eb049dc08a65830d2cdb036d57100e054>
<http://www.w3.org/2000/01/rdf-schema#label>    "Libramont - Voie V - Section 4295, Libramont - Quai V / VI L 4295" .
<http://data.europa.eu/949/functionalInfrastructure/operationalPoints/7eb53d9877df4f4b21adb708c10d320784941115>
<http://www.opengis.net/ont/geosparql#hasGeometry>  <http://data.europa.eu/949/locations/%2B5.4524560/49.8542330> .
<http://data.europa.eu/949/functionalInfrastructure/tracks/07e63ca3bafcfd27447293e0e7139b1ee4056085>
<http://data.europa.eu/949/hashSource>  "BELRB/10/2020-06-14_2078-12-31/L165(1) Libramont - SAS 055 - Section 20247/2018-09-30_2078-12-31" .
<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/62b9db5eb049dc08a65830d2cdb036d57100e054>
<http://data.europa.eu/949/notYetAvailable> <http://data.europa.eu/949/assistanceStartingTrain> .
<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/653f7ead71a60ef587315c199cefa2dffdfc9799>
<http://data.europa.eu/949/notYetAvailable> <http://data.europa.eu/949/platformHeight> .
<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/62b9db5eb049dc08a65830d2cdb036d57100e054>
<http://data.europa.eu/949/notYetAvailable> <http://data.europa.eu/949/areaBoardingAid> .
<http://data.europa.eu/949/functionalInfrastructure/tracks/67897ba803536a4c48f68641082abad1c6405782>
<http://data.europa.eu/949/tenClassification>   <http://data.europa.eu/949/concepts/ten-classifications/rinf/40> .
<http://data.europa.eu/949/functionalInfrastructure/tracks/a06f33cf694b2498e5c4c9d7bb0099041327582d>
<http://data.europa.eu/949/notYetAvailable> <http://data.europa.eu/949/verificationINF> .
<http://data.europa.eu/949/functionalInfrastructure/tracks/41c4e0485b0722ceb234401352621e0e0f69bf17>
<http://data.europa.eu/949/notYetAvailable> <http://data.europa.eu/949/lineCategory> .
<http://data.europa.eu/949/functionalInfrastructure/tracks/41c4e0485b0722ceb234401352621e0e0f69bf17>
<http://data.europa.eu/949/validityEndDate> "2078-12-31"^^<http://www.w3.org/2001/XMLSchema#date> .
<http://data.europa.eu/949/functionalInfrastructure/tracks/619c3162cd1fc394d48b712f20f493ceba4af72d>
<http://data.europa.eu/949/canonicalURI>    <http://data.europa.eu/949/functionalInfrastructure/tracks/BELRB_Libramont%20-%20Voie%20207%20-%20Section%204297> .
<http://data.europa.eu/949/functionalInfrastructure/tracks/1a506943888c2039d0c79c083371b0c1195cfba6>
<http://data.europa.eu/949/validityStartDate>   "2013-06-19"^^<http://www.w3.org/2001/XMLSchema#date> .
<http://data.europa.eu/949/functionalInfrastructure/tracks/07e63ca3bafcfd27447293e0e7139b1ee4056085>
<http://www.w3.org/2000/01/rdf-schema#label>    "L165(1) Libramont - SAS 055 - Section 20247" .
```


Figure 41. Excerpt of the Export results (Linked data full dump) in N-Triples format.

4 Route Compatibility Check

When the User selects the option Route Compatibility Check (RCC) in the page header, the RCC check application is triggered where the objective is to check if a certain railway vehicle (read as a locomotive unit, passenger car, wagon, etc.) of an authorized Vehicle type, can travel the route From: one OP To: another OP. Each route is composed of SoLs (tracks) with different technical parameters. Each track is between two operations points.

The initial window is structured in three areas: (1) Header, (2) Input area for the From: and To: OPs and for the Vehicle type, (3) Area for RCC results, and (4) Map area where the route that is being checked is shown with OPs and SoLs (see Figure 42).

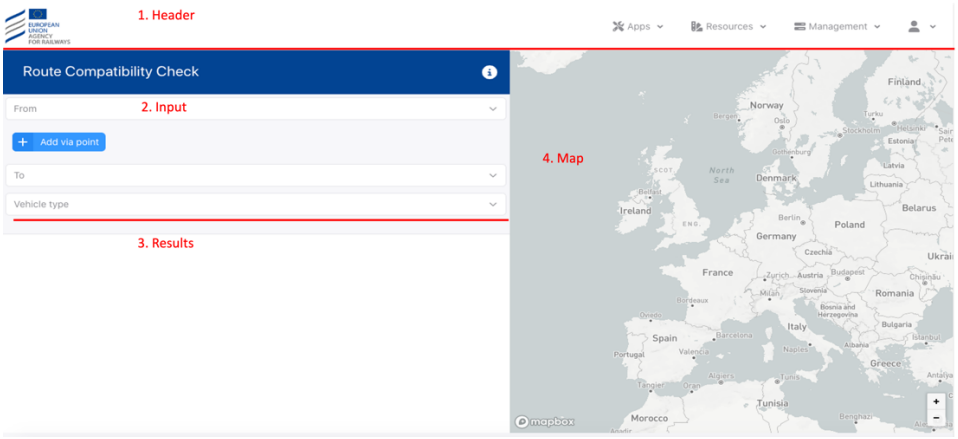


Figure 42. RCC application page structure

The User may see the result of the route that is calculated between two OPs. The From: OP and the To: OP are introduced (with the autocompletion feature), and the OPs and SoLs can be seen in the map, and also in the "Results" part of the window. Note that no information is given on compatibility as no Vehicle Type has been introduced (see Figure 43).

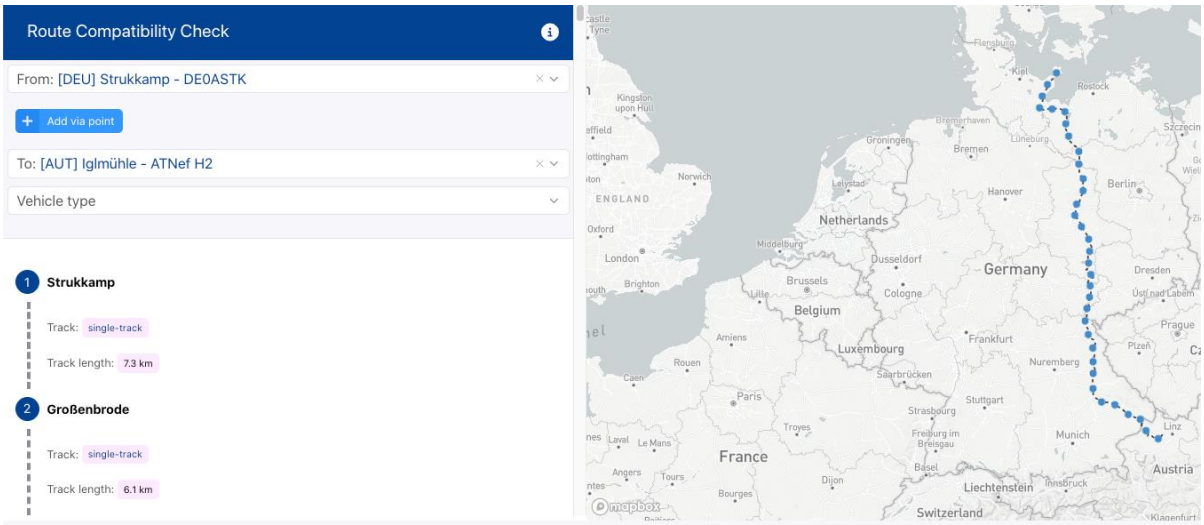


Figure 43. A route between two OPs

When the User clicks on any OP in the results, details of the OP are presented using the same interface as in the search application. Similarly, details of a track between two OPs are displayed when the User clicks on the track label in the results.

When the User introduces the Vehicle type, the RCC results are displayed. A summary is given for the set of parameter compatibility checks (see Figure 44). The Vehicle type name may be clicked, and the vehicle type details will be displayed in another window (see Figure 45).

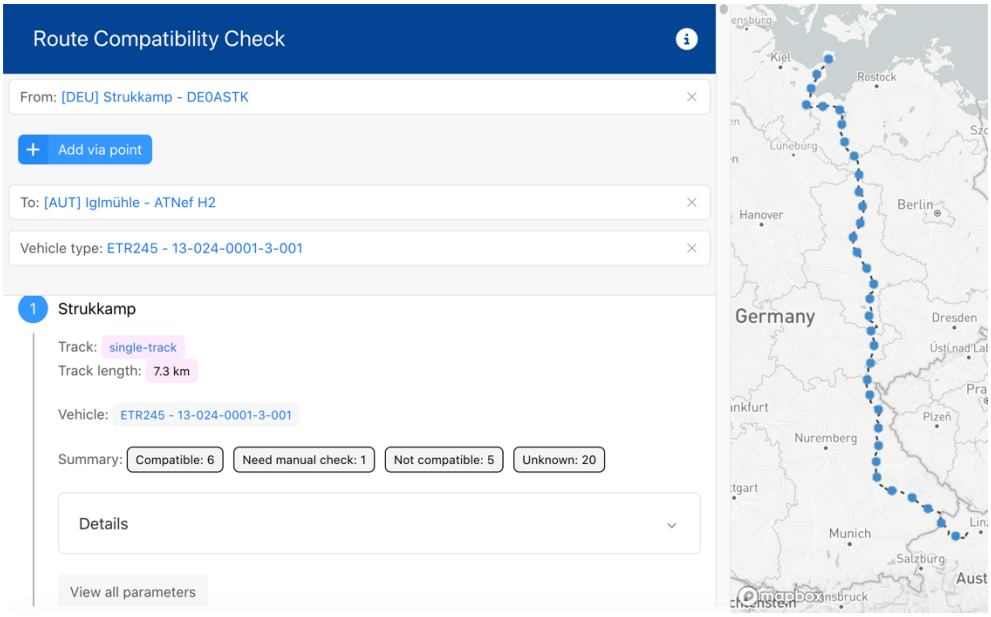


Figure 44. RCC check summary

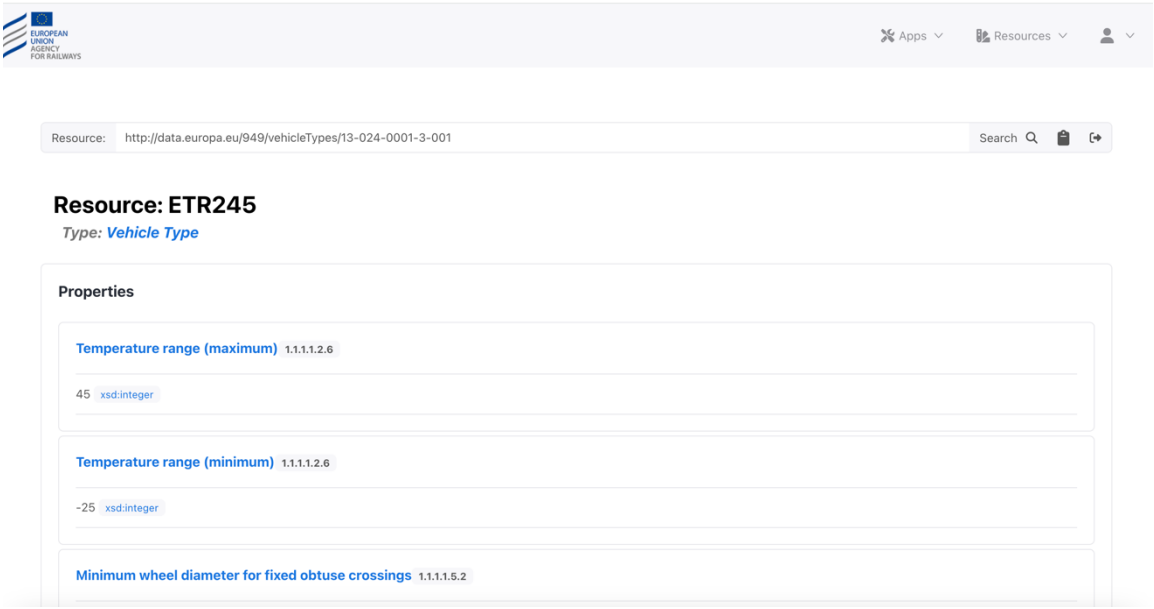


Figure 45. Detailed information on a vehicle type

When the User clicks Details, the values for each parameter are displayed. The summary gives the number of parameters that are Compatible, the number of Not compatible parameters, the number of parameters that

additionally Need manual check by the User, and the number of parameters where their values are Unknown and could not be checked (see Figure 46).

The following colour code indicates the parameter compatibility:

- Green: The parameter values are compatible
- Red: The parameter values are incompatible
- Gray: There is not enough data for the compatibility check

The first column is the name of the parameter used in the check (Check name), the second column contains the values of the parameter for the Track, and the third column contains the values of the parameter for the Vehicle type.

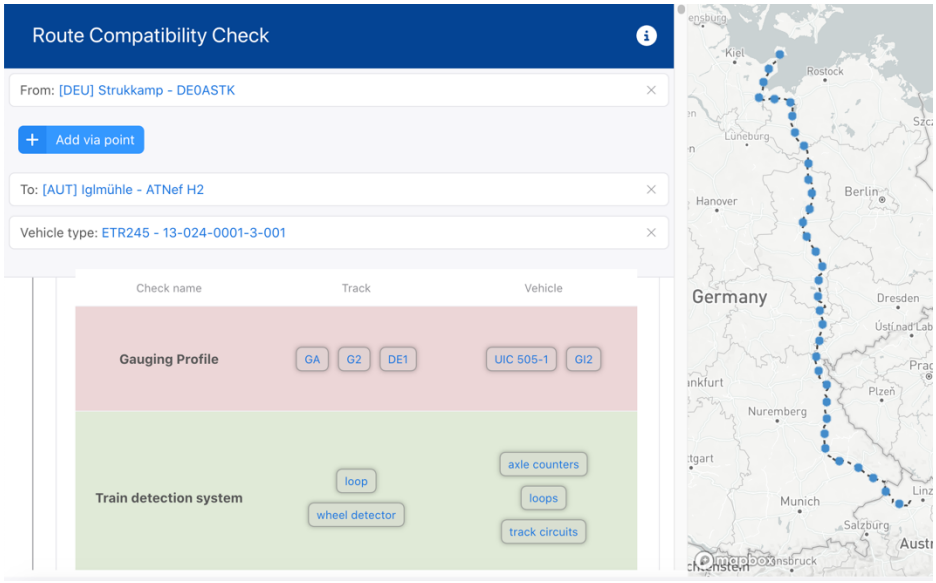


Figure 46. Details of parameter compatibility results: Gauging Profile parameter values are not compatible whereas there are Train detection system parameter values that are compatible

Another example of the compatibility results is given in Figure 47.

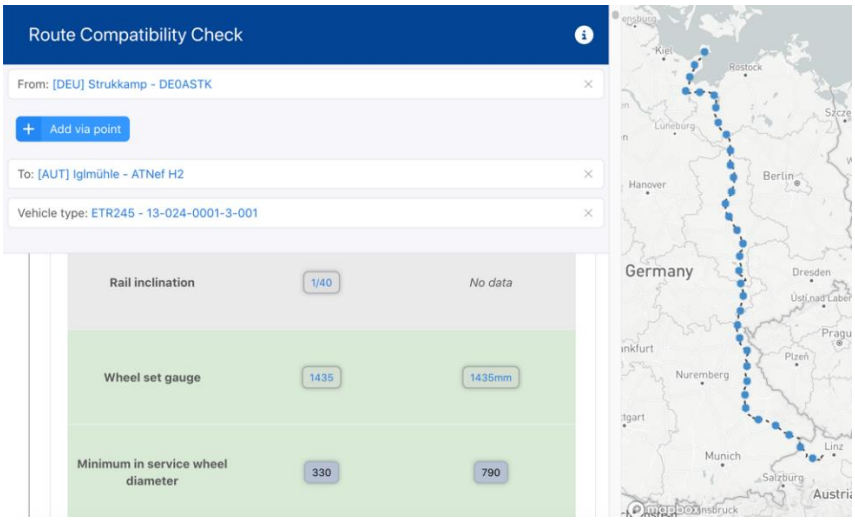


Figure 47. Details of parameter compatibility results: Wheel set gauge and Minimum in service wheel diameter parameter values are compatible whereas there is not enough data to check the compatibility of the Rail inclination

When the parameter values belong to a predefined set (taxonomy) they can be clicked, and details will be given on the value (see Figure 48).

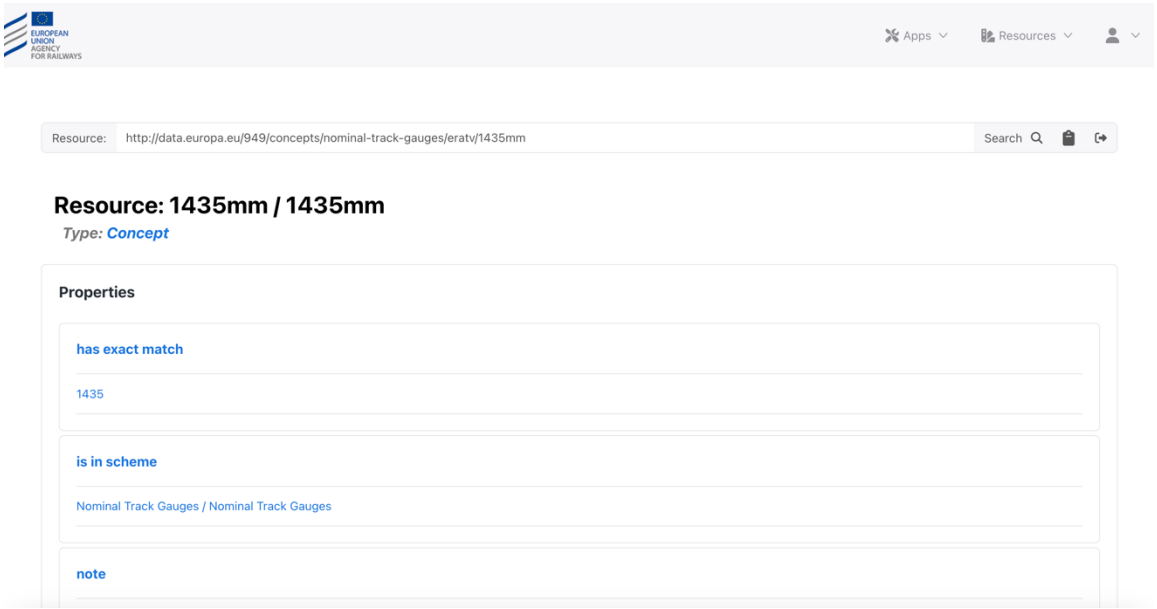


Figure 48. Details on the value 1435mm of parameter Wheel set gauge for the Vehicle type. This value matches exactly (skos:exactMatch) to the track's value, 1435. Thus, the values are compatible as can be seen in Figure 47

For each track, when the User clicks on View all parameters, a modal window is shown with the list of all parameters and their values (see Figure 49).

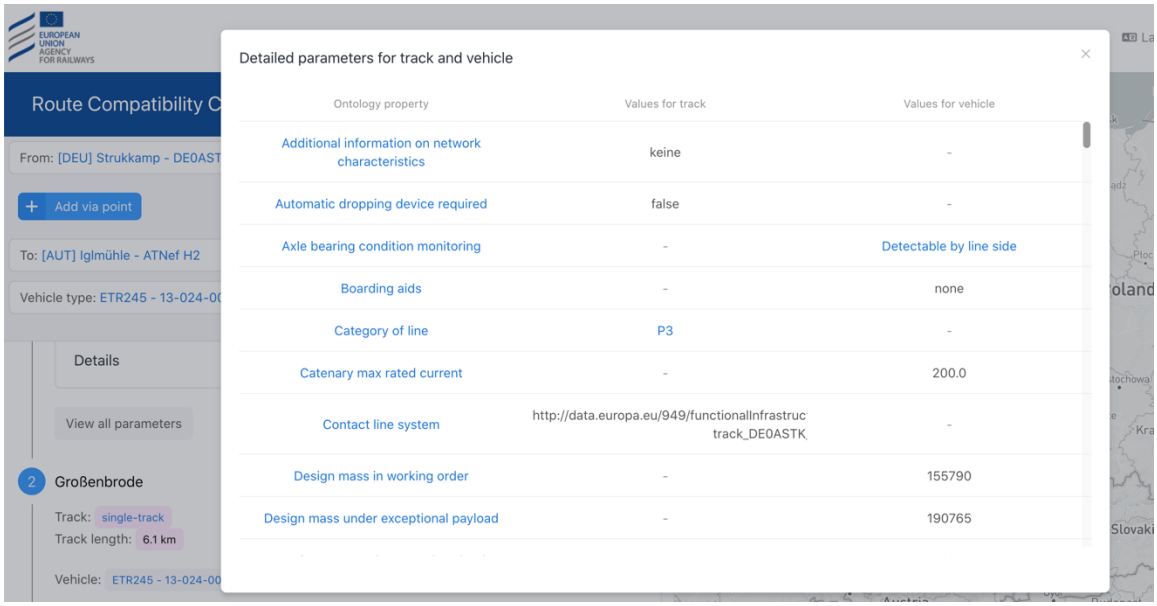


Figure 49. Detailed parameter values for track and vehicle

5 Map Explorer

When the User clicks on Map Explorer under the Apps menu, a map will be displayed with all the elements of OPs and SoLs in the RINF infrastructure (see Figure 52). The User has options for the Infrastructure and Base map, and a Map area can also be selected.

Options for Infrastructure can be seen in Figure 50.

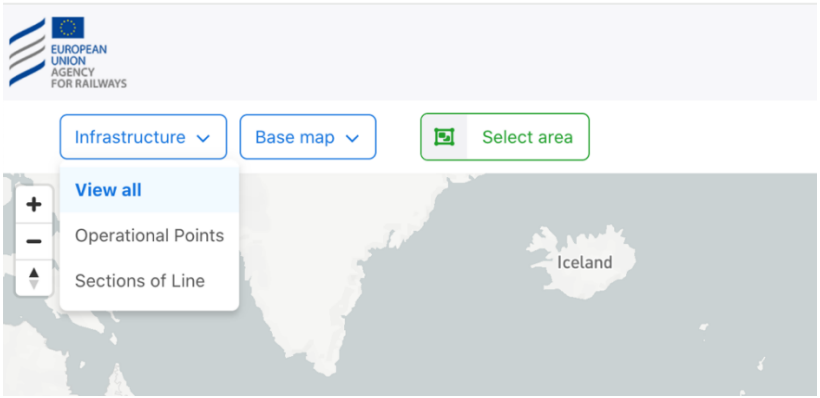


Figure 50. OPs or SoLs can be viewed on the map

Options for Base map can be seen in Figure 51. The Light map is the default. A Dark map, Streets, Satellite and Navigation views may be chosen.

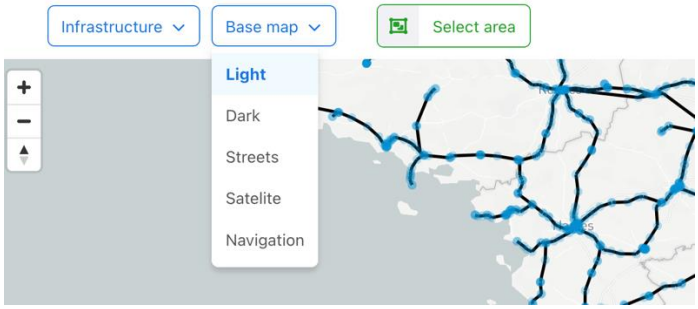


Figure 51. Options for Base map

In this example, only data from Switzerland has been loaded to the System.

Figure 52. Excerpt of the map of Europe with data from Switzerland.

When this map is zoomed in, the OPs (dots) and SoLs (lines) can be distinguished. A mouse over gives the User the label of the element (see Figure 53).

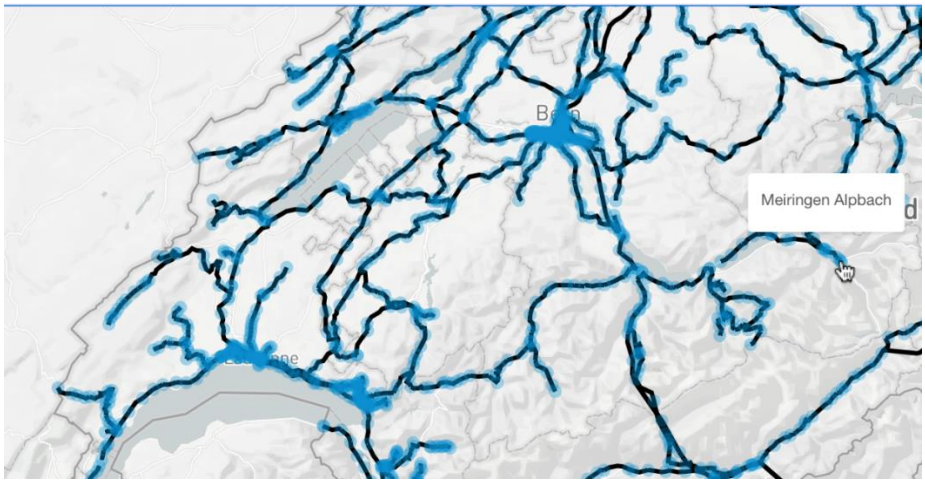


Figure 53. Zoomed in map where an operational point's name is displayed.

When the User clicks on this element, a dialog box is displayed where the User may choose among several OPs that are close to the OP that was selected (see Figure 54).

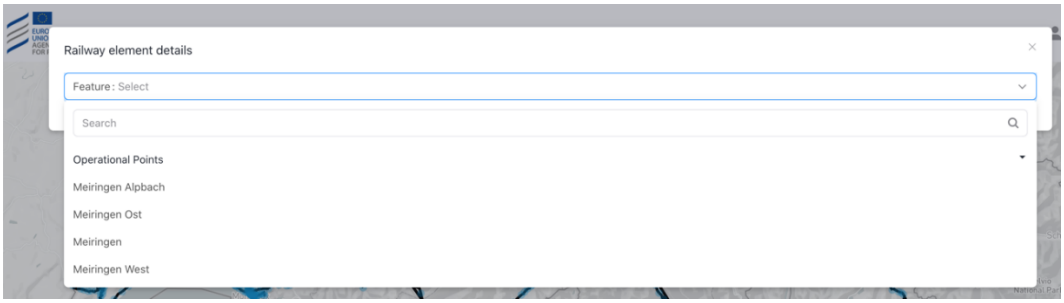


Figure 54. The User may search for an OP or choose among those that are close to the one selected on the map.

When an OP is selected, its generic properties are displayed (see Figure 55).

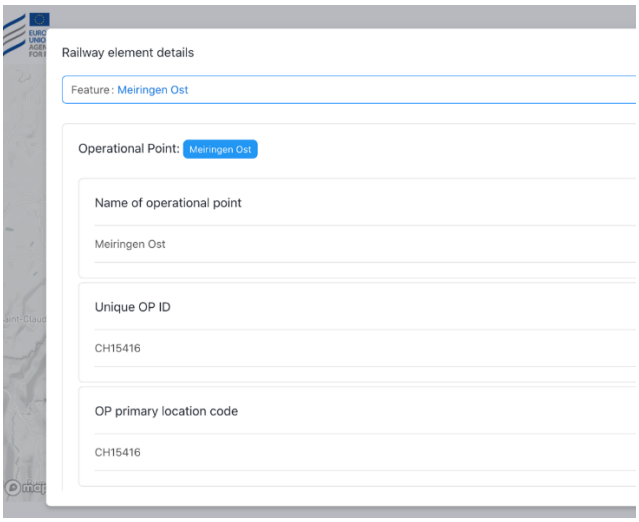


Figure 55. Generic properties of the OP “Meiringen Ost”.

The User may also click on a SoL and its generic properties will be displayed (see Figure 56).

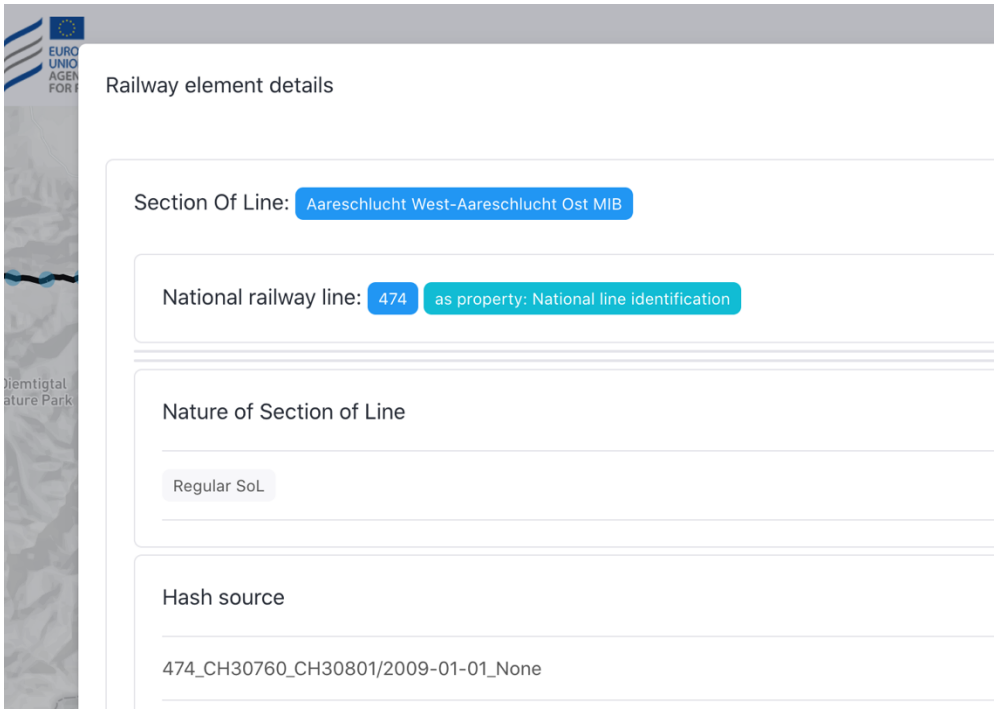


Figure 56. Generic properties of the SoL "Aareschlucht West-Aareschlucht Ost MIB".

6 Data stories and SPARQL endpoint

6.1 Data stories

Data stories are a catalogue of natural language queries and their corresponding SPARQL queries. They represent the User stories that have been defined for each of the RINF System's Use cases.

When this option is chosen, a window is presented with the queries grouped in the following categories. These categories can be expanded or collapsed by clicking on their title.

- Queries used to retrieve total number of RINF elements per member state and their total length
- Queries for specific core parameters
  These queries apply to a specific core parameter (load capability) for all tracks of a member state or for all tracks of a specific member state. Note that these queries can be replicated for other core parameters and other member states.
- Query focused on the presence of different types of parameters
  This query determines which core parameters are applicable to which types of ontology elements according to the current KG.
  Taking into account the result of this query, the following queries retrieve the values of the applicable parameters to each type of RINF element, across all member states or for a specific member state. Note that these queries can be replicated for other member states.
  - Queries for Tracks
  - Queries for Sections of Lines
  - Queries for Tunnels
  - Queries for Sidings
  - Queries for Operational Points
  - Queries for Platforms
  - Queries for Contact Line Systems
  - Queries for Train Detection Systems
- Queries that have been collected from ERA stakeholders
- Queries related to the TEN-T data space
  Most of these queries are restricted to the SoL (and its related tracks) that starts in a specific OP and ends in another specific OP. These queries can be replicated for other SoLs by changing the names of the start and end OPs.

All of the queries have been grouped together in SPARQL notebooks providing a unified view of specific requirements: (1) parameter completeness, (2) other queries collected from ERA stakeholders, and (3) TEN-T data space.

Each query has a Description in natural language and a button View query (see Figure 57).

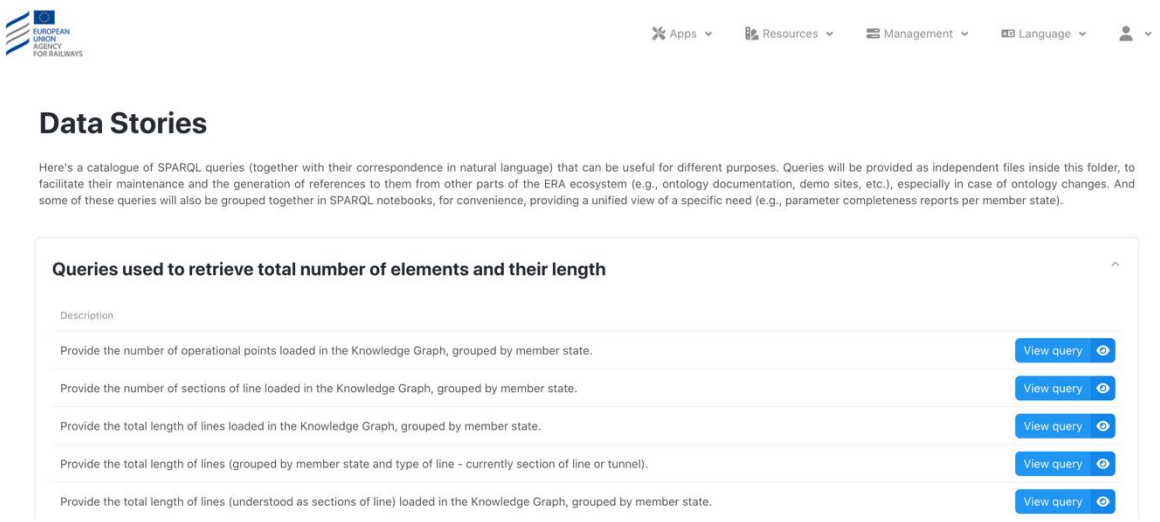


Figure 57. The Data stories query interface

For each query, when the user clicks the View query button, a window is shown with the SPARQL query and two buttons: Save query and Execute query (see Figure 58).

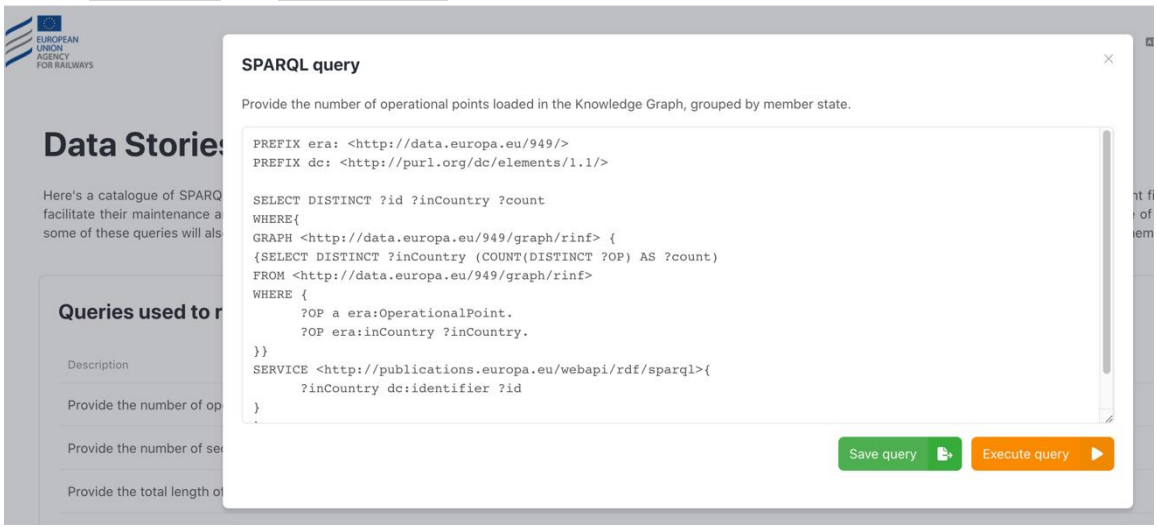


Figure 58. Data stories. Saving or executing a query

Save query downloads the query in SPARQL format (`query.sparql`). When the User clicks Execute query, the query is executed in the SPARQL endpoint, results are shown in Table format, and query results can be downloaded as a `.csv` file by clicking the button ⬇ (see Figure 59).

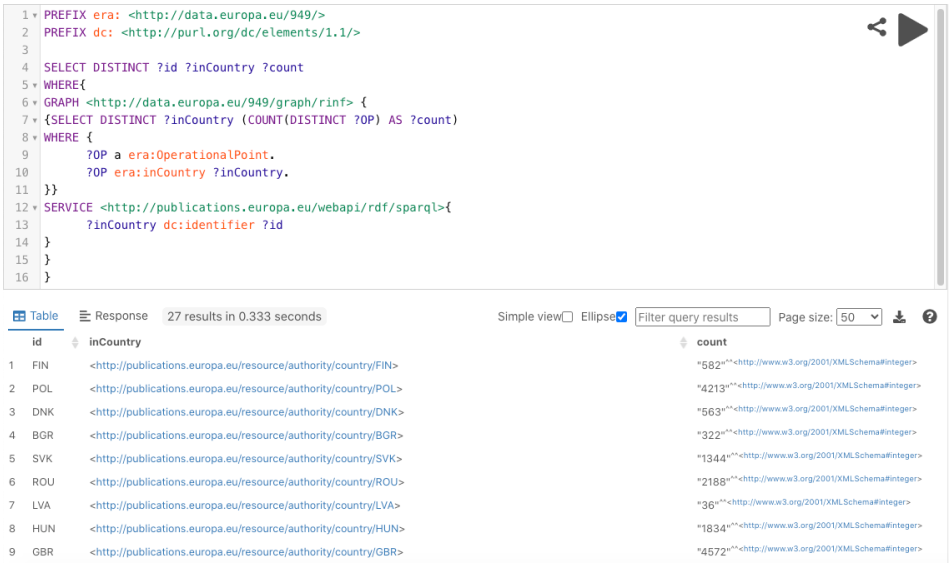


Figure 59. Execution of a Data story query in the SPARQL endpoint

6.2 SPARQL endpoint

SPARQL is a query language for RDF data, which is used to represent information in the form of subject-predicate-object triples. SPARQL allows you to search for and retrieve specific information from RDF datasets.

A SPARQL query consists of several components: prefixes, variables, and triple patterns.

Prefixes

Prefixes are used to define namespaces for the predicates used in the query. They can be declared at the beginning of the query using the PREFIX keyword, followed by the prefix and its corresponding namespace URI. For example:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX era: <http://data.europa.eu/949/>
```

This declares two prefixes, `foaf:` and `era:`, which correspond to the FOAF and ERA namespaces, respectively.

Variables

Variables are used to specify which pieces of data you want to retrieve from the dataset. They are denoted by a question mark (?) followed by a variable name, such as `?opName` or `?uopid`.

Triple patterns

Triple patterns are used to specify the conditions that must be met for the query to return results. They consist of a subject, predicate, and object, separated by spaces and enclosed in angle brackets. For example:

```
<http://data.europa.eu/949/functionalInfrastructure/operationalPoints/ATAa>
era:uopid ?uopid
```

This triple pattern specifies that the subject is the resource with the URI `http://data.europa.eu/949/functionalInfrastructure/operationalPoints/ATAa`, the predicate is `era:uopid`, and the object is a variable `?uopid`.

Example query

Here's an example SPARQL query that retrieves the uopid (operational point unique identifier) and name of all the operational points in an RDF dataset:

```
PREFIX era: <http://data.europa.eu/949/>
SELECT ?uopid ?opName
WHERE {
    ?op a era:OperationalPoint;
    era:uopid ?uopid;
    era:opName ?opName.
}
```

This query declares a prefix for the ERA namespace and specifies that we want to retrieve the values of the variable `?uopid` and `?opName` for all resources that are instances of the `era:OperationalPoint` class.

The `WHERE` clause specifies the conditions that must be met for the query to return results. In this case, the User is looking for resources that have the type `era:OperationalPoint` and a value for the `era:uopid` and `era:opName` property.

Named graphs for resources in the triple store

All the resources published in the triple store MUST be stored in their corresponding named graph, so as to facilitate their management.

The named graph for each type of resources is as follows:

- ERA ontology: `http://data.europa.eu/949/graph/ontology`
- SKOS concept schemes: `http://data.europa.eu/949/graph/skos`
- Data transformed from RINF XML files: `http://data.europa.eu/949/graph/rinf`
- Data transformed from the ERATV database: `http://data.europa.eu/949/graph/eratv`
- Data about infrastructure managers: `http://data.europa.eu/949/graph/im`

Using Graphs

In SPARQL, graphs can be used to organize and segment RDF data into named sets of triples that can be queried independently. Graphs are identified by URIs and can be queried using the GRAPH keyword in a SPARQL query.

To use graphs in SPARQL, follow these steps:

1. Define the namespace prefixes that you will be using in the query using the PREFIX keyword.
2. Use the `SELECT` keyword to specify the variables that you want to retrieve in the query.
3. Use the `GRAPH` keyword to specify the named graph that you want to query. You can specify the graph either in the `FROM` clause, the `FROM NAMED` clause, or directly in the `WHERE` clause using the `GRAPH` keyword.
4. Use the `WHERE` clause to specify the conditions that must be met for the query to return results.

Here's an example SPARQL query that uses a named graph to retrieve `era:OperationalPoint` triples with their `era:opName` property values:

```
PREFIX era: <http://data.europa.eu/949/>

SELECT ?opName
WHERE {
    GRAPH <http://data.europa.eu/949/graph/rinf> {
        ?op a era:OperationalPoint;
        era:opName ?opName.
   }
}
```

In this example, we define a namespace prefix using the `PREFIX` keyword: `era:` for the ERA vocabulary namespace.

The `SELECT` keyword is used to specify the `?opName` variable that we want to retrieve in the query.

The `WHERE` clause is used to specify the conditions that must be met for the query to return results. In this case, we are looking for resources that have the type `era:OperationalPoint` and an `era:opName` property in the named graph with the URI `http://data.europa.eu/949/graph/rinf`.

Note that the `GRAPH` keyword can also be used in combination with the `OPTIONAL` keyword to retrieve data from a named graph only if it is available, or to join data from multiple named graphs in the same query.

In order to execute this query, you click the Endpoint option in the Resources dropdown list. A window will be displayed where you can paste the query (see Figure 60).

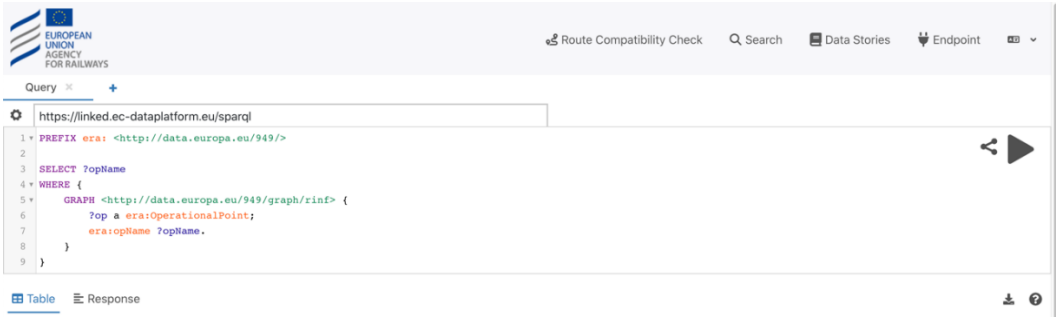


Figure 60. SPARQL endpoint interface

Once the user clicks on the Execute button ▶, the answers to the query will be displayed (see Figure 61).

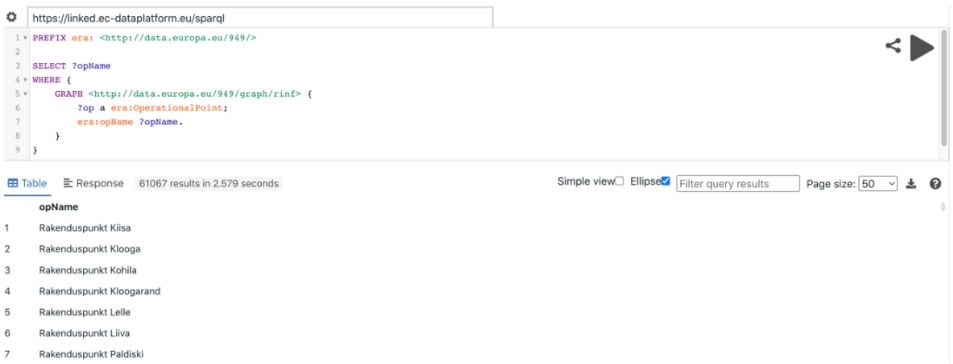


Figure 61. Excerpt of the example query answers

7 Data Assets Management

In case the User has the IM or the NRE roles, once the User chooses the role, the Management option will appear in the heading as can be seen in Figure 62. There are two types of data assets:

1. Datasets: A RINF Dataset is any XML or RDF file that the User wishes to Delete, Download, Upload or Publish into the ERA KG.

   Users may also upload compressed files that are large (.zip) that may contain only one file.

2. Documents: Reference documents are those files that need to be uploaded in order to be accessible through a URL. They are the values of some of the RINF parameters, e.g. document regarding rules or restrictions, braking information document.

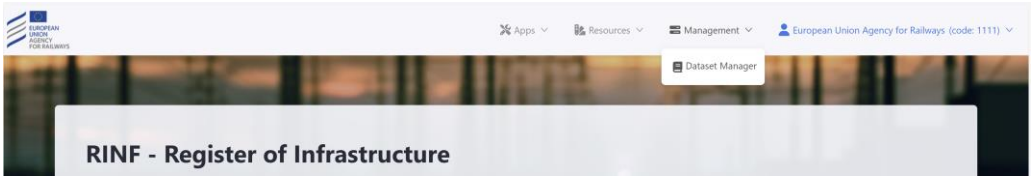


Figure 62. The Management option.

When the User clicks on the Dataset Manager option, a window with two tabs is presented (see Figure 63): (1) Datasets, and (2) Reference documents.

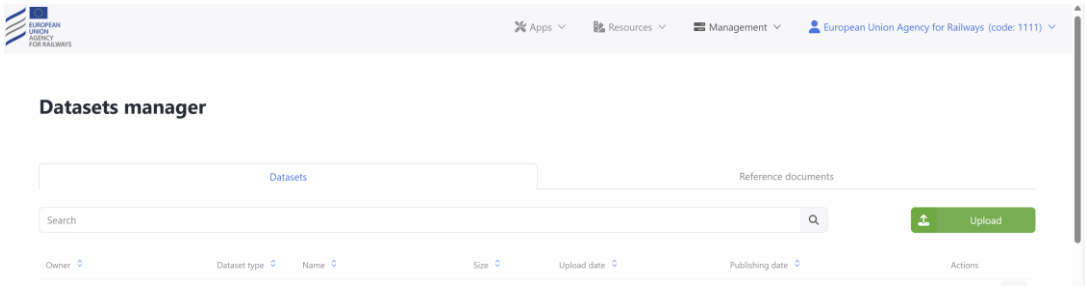


Figure 63. The Datasets manager page.

7.1 IM representation by NREs

In order to allow Users with the NRE role to manage datasets, they must decide which IM they are representing at each time, among those for which the NRE is responsible. This can be done from the Select User Role option in the Login menu option, as shown in Figure 64.

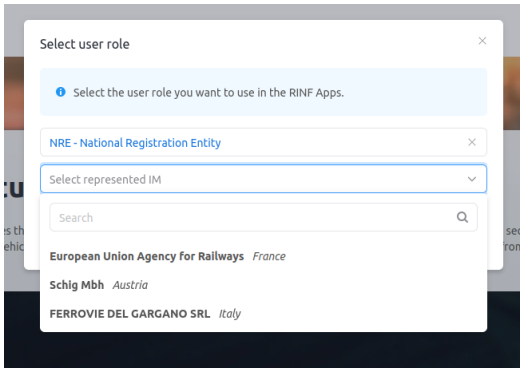


Figure 64. A User with an NRE role selects an IM to represent, from those available for that NRE.

Once the IM to be represented is selected, the top-left part of the main menu displays the NRE and IM that has been selected, as show in . At any point in time, the NRE may decide to represent another IM, using the same menu option.

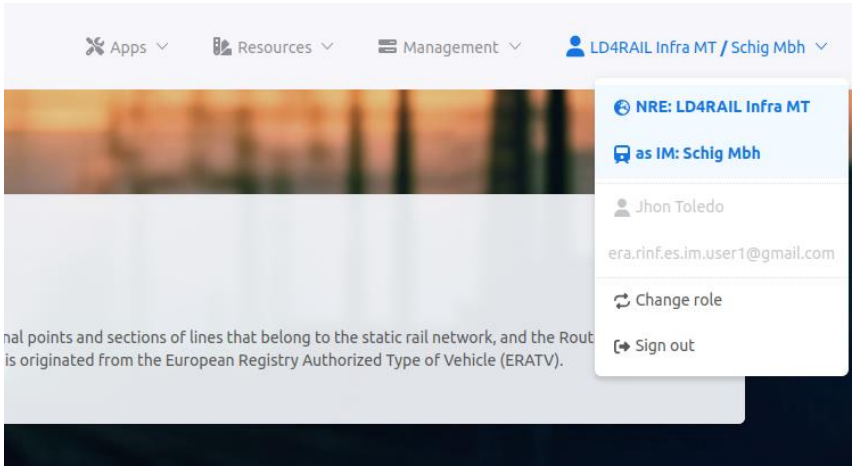


Figure 65. Menu after a User with an NRE role selects an IM to represent.

7.2 Datasets and Pipeline

The columns in the Datasets table shown in Figure 63 are as follows:

- Owner: The full name of the User who has uploaded the file.
- Dataset type: The type of the dataset file. Values can be:
  - XML_FULL which is a complete Dataset file that contains all of the RINF elements managed by the IM in XML format
  - RDF_FULL which is a complete Dataset file that contains all of the RINF elements managed by the IM or the NRE. These datasets in RDF format.
  - RDF_PARTIAL_INSERT which is a Dataset file in RDF format that contains only those elements that need to be inserted.
  - RDF_PARTIAL_UPDATE which is a Dataset file in `zip` format that contains in turn two RDF files for deleting and inserting triples as a “bundle”.
  - RDF_PARTIAL_DELETE which is a Dataset file in RDF format that contains only those elements that need to be deleted.
- Name: The name of the uploaded file.
- Size: The size of the file in MBs.
- Upload date: The date and time when the Dataset file was uploaded.
- Publishing date: The date when the Dataset file was published into the ERA KG.
- Actions: Actions that the user may carry out for the corresponding Dataset file:
  - Download the file
  - Rename the file
  - Delete the file
  - Actions related to the Dataset pipeline. Options for pipeline success or failure are shown: or . When the user clicks on the icon, a screen is shown with the Pipeline and its steps. The icon indicates success or failure in the corresponding step.

Figure 66 illustrates the flow for the Dataset file related to the different tasks in the pipeline.

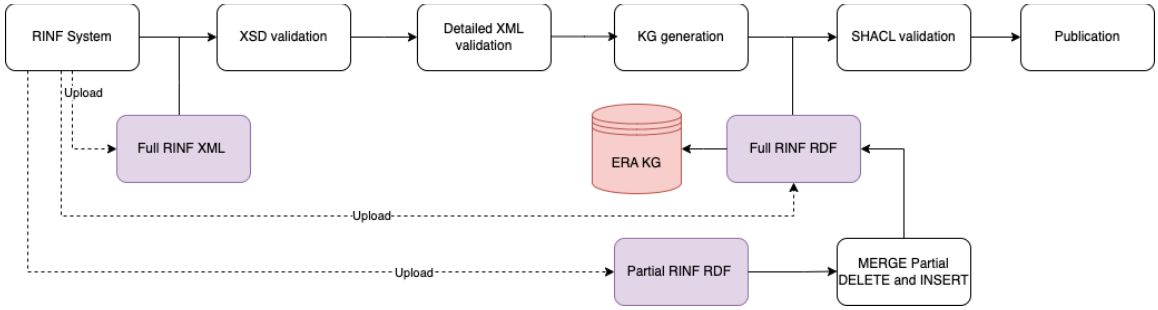


Figure 66. Flow for Dataset management

A user may upload Full RINF XML, Full RINF RDF or Partial RDF Dataset files. To generate Full and Partial RDF files, the User may follow the Guide for tooling and annotation in RDF in Appendix A. These RDF files are following v3.0. of the ERA Ontology.

When the user uploads a file, the following steps are executed:

1. For a Full RINF XML Dataset file, the validation against its XMLSchema (XSD) is executed and the Dataset file may pass or not the *XSD validation*.
    a. If the Dataset file passes the validation, the *Detailed XML Validation* is executed (step 2.).
    b. If the Dataset file does not pass the validation, error messages are output to the user.
2. For a Full RINF XML Dataset file, a component that implements a detailed validation of all parameters and business rules is executed, and the Dataset file may pass or not the *Detailed XML Validation*. If the Dataset file does not pass the validation an error report is output to the user.
3. For a Full RINF XML dataset file the *KG Generation* step is executed and a Full RINF RDF will be generated.
4. For Partial RINF Dataset files the *Merge PARTIAL DELETE and INSERT* process will be executed and a new Full RINF RDF is generated.
5. The Full RINF RDF is then validated against the SHACL rules (*SHACL Validation*).
    a. If the Dataset file passes the validation checks, then the User may execute the *Publication* into the ERA KG step.
    b. If the Dataset file does not pass the *SHACL Validation,* then the errors are output to the User. In this case (or if there are errors in the previous steps), the pipeline (*Publication*) may not be continued.

7.2.1 Upload

When the User clicks this option, a box will be displayed where the User chooses the type of Dataset file to be uploaded (see Figure 67). Options are Full XML Dataset, Full RDF Dataset, Partial RDF Dataset (insert), Partial RDF Dataset (update), and Partial RDF Dataset (delete). In this example the Full XML Dataset option has been chosen for an XML file. Note that compressed files `(.zip` extension), may be uploaded.

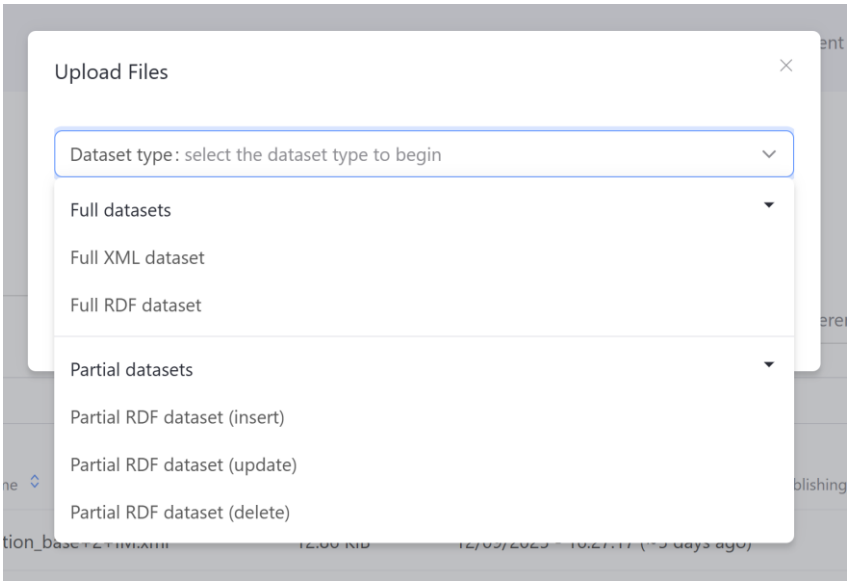


Figure 67. A box where the User may choose the type of Dataset file to be uploaded.

7.2.1.1 *Upload Full XML dataset*

First, a file with errors with respect to its XMLSchema will be uploaded (See Figure 68)

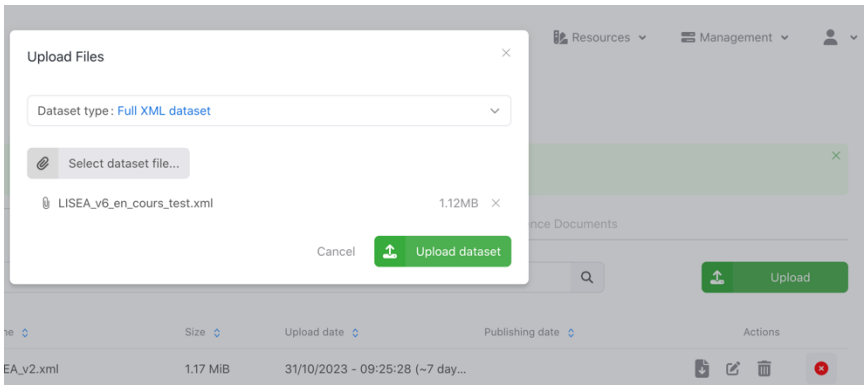


Figure 68. The dataset file has been chosen.

Thus, once the file is Uploaded the dataset file will be added to the list (see Figure 69).

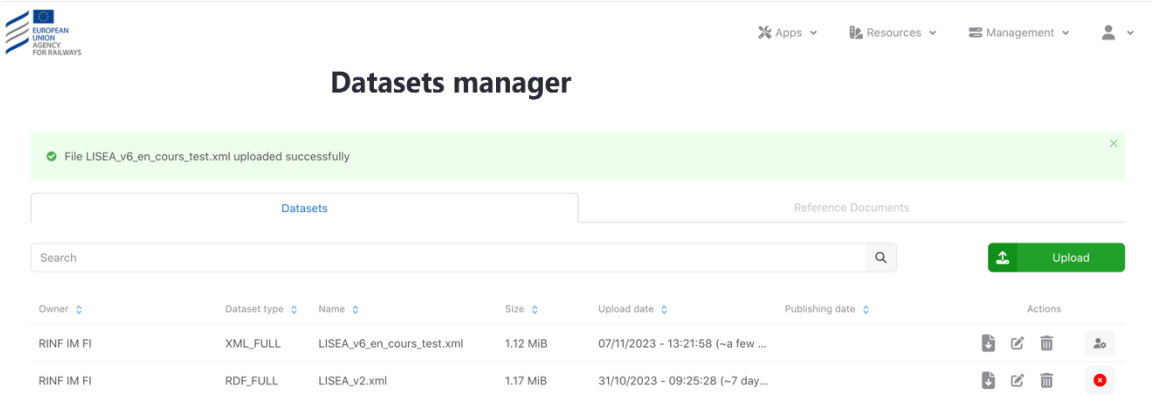


Figure 69. The dataset file has been uploaded successfully.

When the user clicks the Actions icon , the Dataset processing pipeline is presented (see Figure 70).

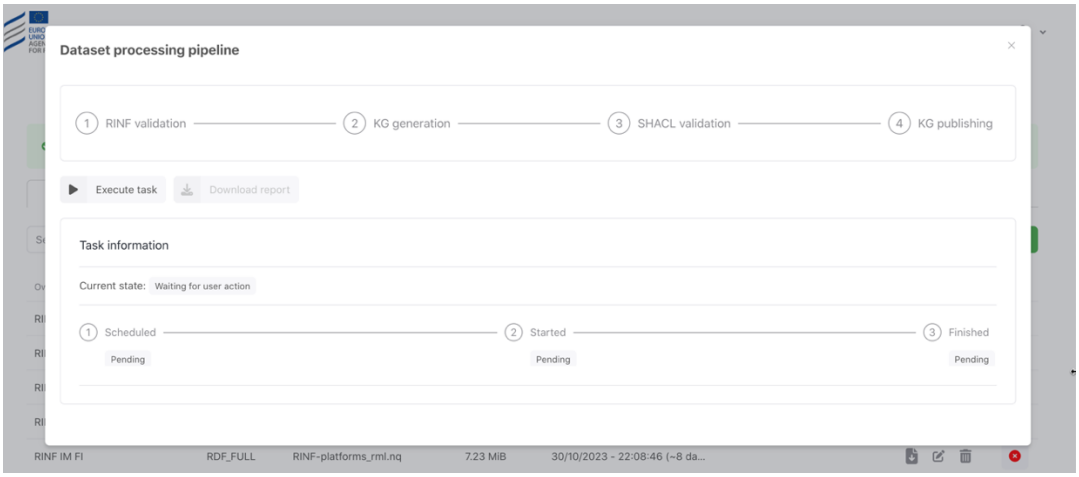


Figure 70. The Dataset processing pipeline.

All of the pipeline steps are in grey because none has been executed yet. You can see that the first pipeline step is the RINF validation, this corresponds to the validation against the XSD followed by a detailed validation of all parameters and business rules. When the user clicks Execute task, it will highlight the first step. It will also show its Current state, that it is Scheduled for running and it will highlight in its execution that it is Scheduled (see Figure 71).

Figure 71. The RINF validation step is Scheduled for running.

In this case the XSD validation step fails, and the user may download the error report (see Figure 72). The error report states:

```
XSD validation failed.
Element 'SOLLength': This element is not expected. Expected is ( SOLOPStart )., line 7
```

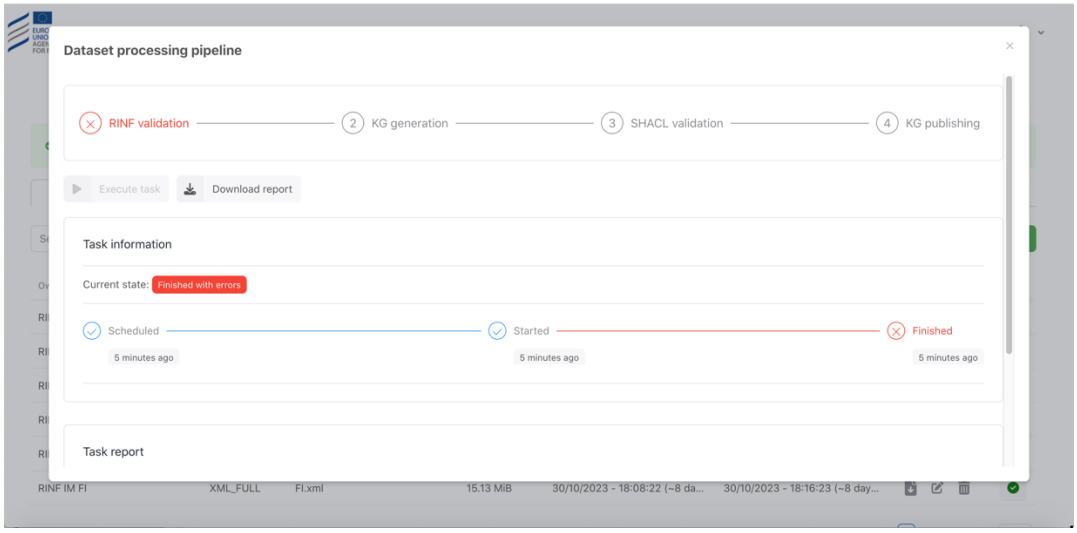


Figure 72. The RINF validation has Finished with errors due to failure in the XSD validation.

Now, suppose that the user fixes those errors and tries to Upload the file again. Now the file will finish successfully this step of the pipeline (see Figure 73).

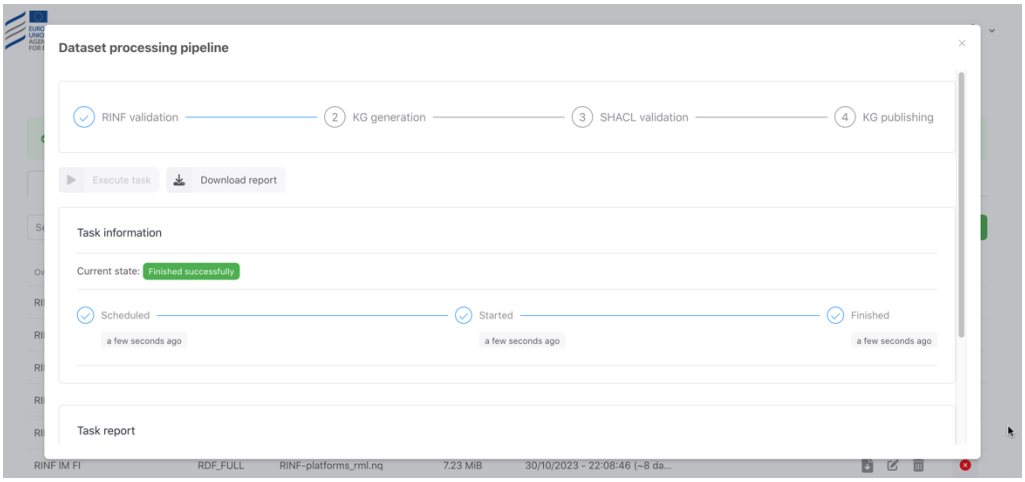


Figure 73. The RINF validation has Finished successfully.

In the case, that the XML detailed validation does not pass, an error report will be generated with all the error messages and their corresponding Line number in the file.

Now we will upload a `.zip` files. Note that the when the file is uploaded it is decompressed (see Figure 74).

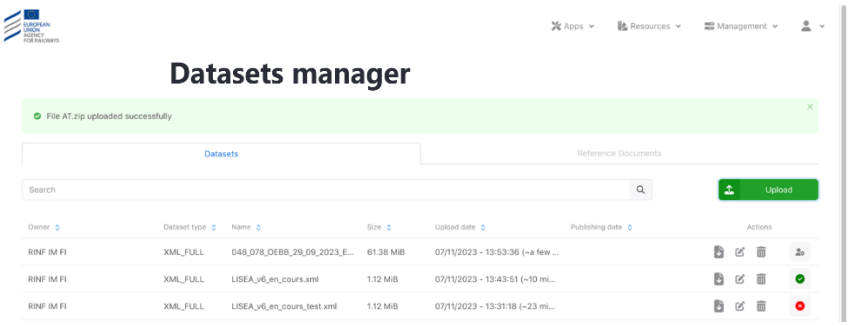


Figure 74. The AT.zip file has been uploaded.

Once the user clicks the Pipeline (Actions) icon, and Executes the RINF Validation task, it will first show that the task is Running, and it will highlight in its execution that it has Started (see Figure 75).

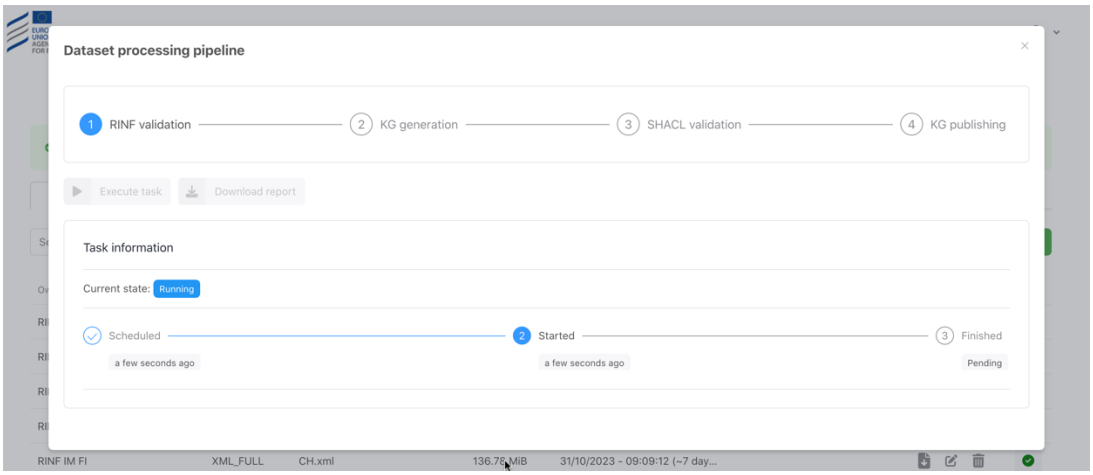


Figure 75. The RINF validation task is Running.

The RINF validation has Finished with errors due to the failure of the detailed XML validation on parameters and business rules (see Figure 76).

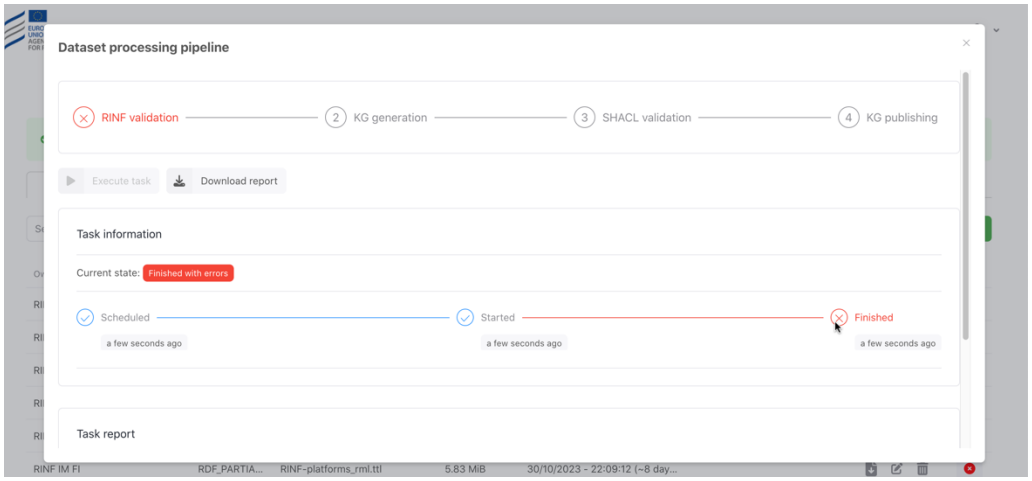


Figure 76. The validation has Finished with errors due to failure in the detailed XML validation.

An excerpt of the error report follows:

```
RINF index: 1.1.1.3.7.11.1
Rule name: SOLTrackParameter, ID=CTD_MinAxleLoadByVehicleCat
Error line: 681926
Error message: Applicable ('Y') when parameter 1.1.1.3.7.1.1(CTD_DetectionSystem) the selected option
is 'track circuit' or 'wheel detector'
Line for 1.1.1.3.7.1.1 : 681912
---------------
RINF index: 1.1.1.3.7.15.1
Rule name: SOLTrackParameter, ID=CTD_TSIMaxImpedanceWheelset
Error line: 681930
Error message: Applicable ('Y') when parameter 1.1.1.3.7.1.1(CTD_DetectionSystem) the selected option
is 'track circuit'
Line for 1.1.1.3.7.1.1 : 681912
---------------
RINF index: 1.1.1.3.7.17
```

```
Rule name: SOLTrackParameter, ID=CTD_MaxSandOutput
Error line: 681932
Error message: Applicable ('Y') when parameter 1.1.1.3.7.1.1(CTD_DetectionSystem) the selected option
is 'track circuit'
Line for 1.1.1.3.7.1.1 : 681912
```

These three error messages correspond to violations to the business rules where the parameter is mandatory ("Applicable") when another parameter, in this case 1.1.1.3.7.1.1 (CTD_DetectionSystem) has a certain value: for the first rule, values must be "track circuit" or "wheel detector" whereas for the second and third rules, the value must be "track circuit".

In the first error message, the business rule for parameter 1.1.1.3.7.11.1 is "Applicable ('Y') only when for parameter 1.1.1.3.7.1.1 the selected option is 'track circuit' or 'wheel detector'". The corresponding XML line is:

```
<SOLTrackParameter ID="CTD_MinAxleLoadByVehicleCat" IsApplicable="N" OptionalValue="" Set="track circuit" Value=""/>
```

The XML line for CTD_DetectionSystem is:

```
<SOLTrackParameter ID="CTD_TCCheck" IsApplicable="Y" OptionalValue="Direct current Track circuit" Set="track circuit" Value="10"/>
```

It corresponds to a type "track circuit," so the business rule is not satisfied. Similarly, the other two error messages are output due to the same reason: the value for CTD_DetectionSystem corresponds to "track circuit "and the parameter has been defined with `IsApplicable="N"`.

If the User now tries to execute the KG generation task, a confirmation message will be displayed (See Figure 77).

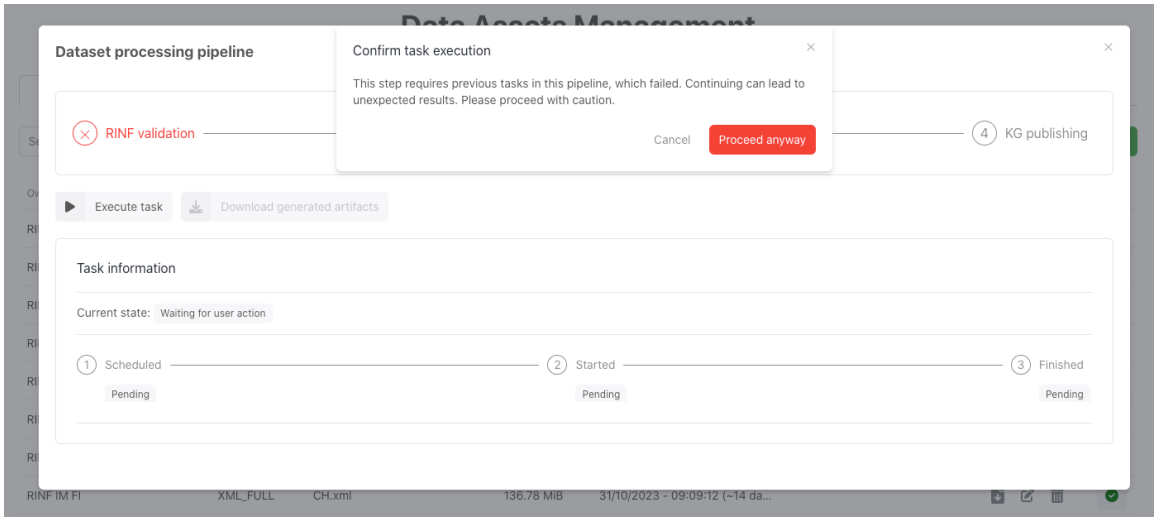


Figure 77. The User receives a confirmation message when the previous task has not finished successfully.

The User may Proceed anyway with the execution or decide to Cancel.

Now, you can see that another file has been uploaded, `2023_10_17 FUC RINF.xml`. The RINF validation step of the pipeline has been executed and the task has finished successfully (see Figure 78).

In the list of files, the first row which corresponds to this file has a green check mark to indicate that the task triggered by the User has executed successfully (see

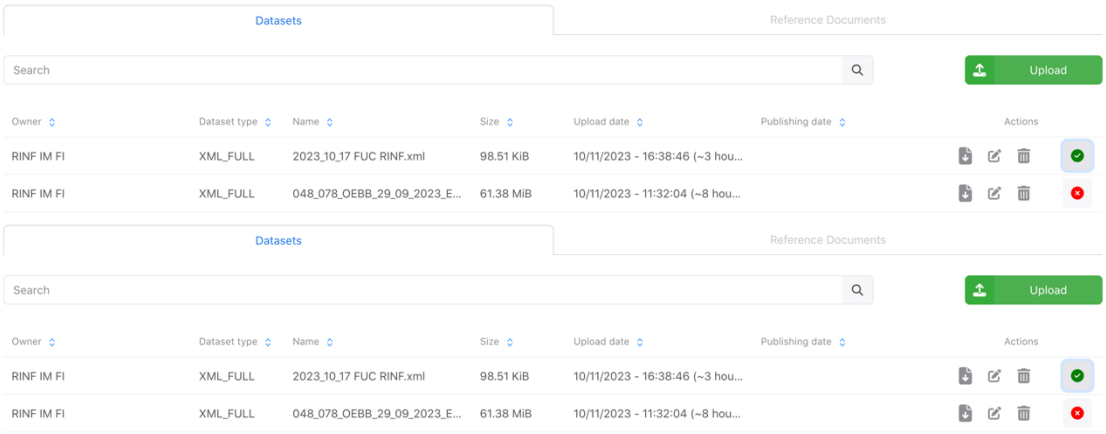


Figure 79).

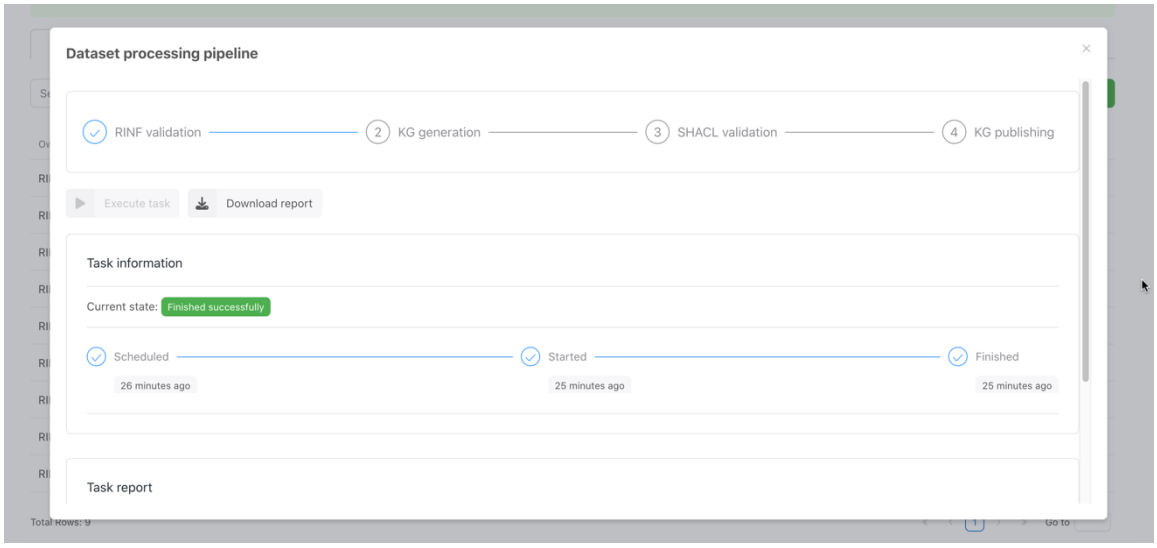


Figure 78. The RINF validation has Finished successfully.

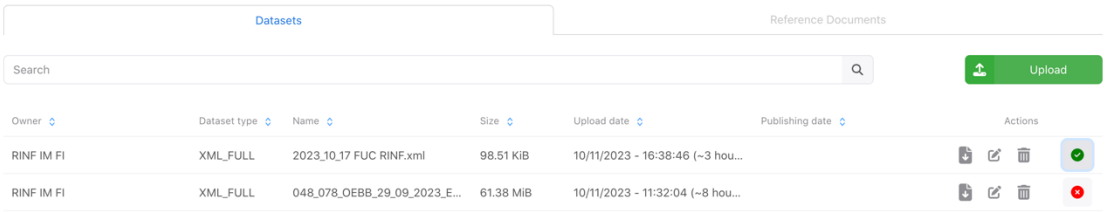


Figure 79. The pipeline for the file in the first row has a green checkmark.

Now the user will execute the KG Generation pipeline by clicking Execute task. Once the task has Finished, the Dataset processing pipeline is updated as can be seen in Figure 80.

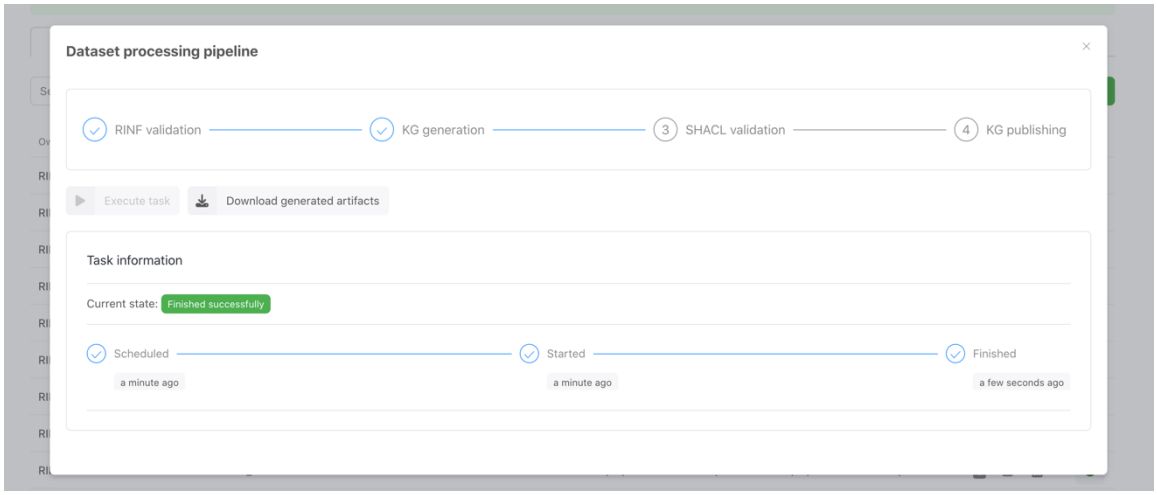


Figure 80. The KG generation task has Finished successfully.

Next, the User executes the SHACL validation step. For this, he clicks the task in the pipeline and then clicks Execute task. In this case this task has Finished with errors (see Figure 81).

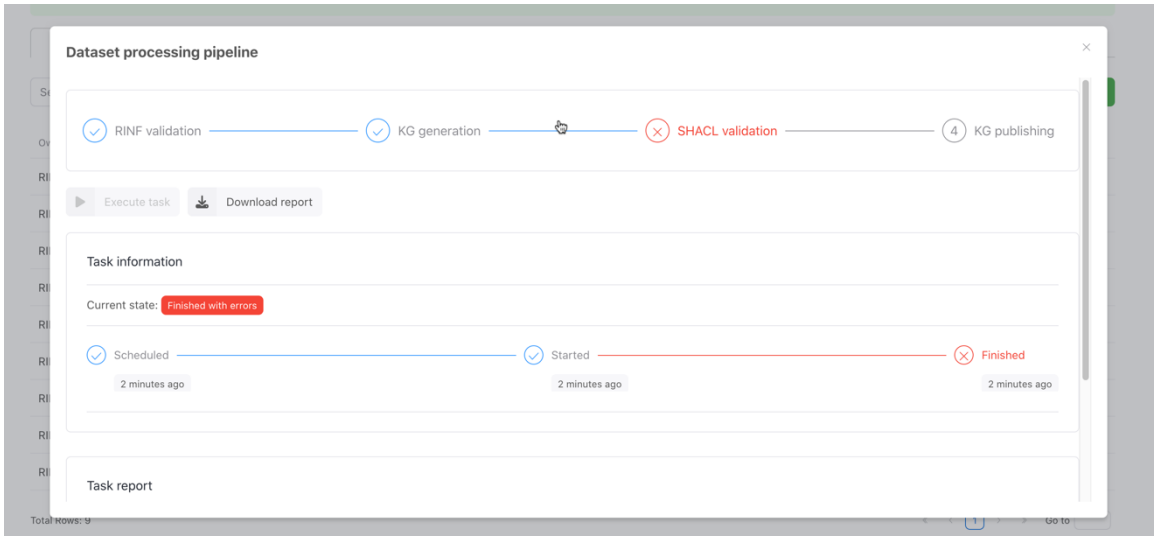


Figure 81. The SHACL validation has Finished with errors.

The User may download the error report that indicates for each SHACL shape, i.e. each validation rule, if it passed the validation. For example, the following is the result of the validation of the Contact line system type that states that values must belong to a predefined list (SKOS). In this case the validation has passed:

```
# contactLineSystemTypeSKOS
Validation Report
Conforms: True
```

An example of an error follows:

```
# maximumTrainCurrent
Validation Report
Conforms: False
Results (1):
Constraint Violation in DatatypeConstraintComponent
(http://www.w3.org/ns/shacl#DatatypeConstraintComponent):
        Severity: sh:Violation
        Source Shape: <http://data.europa.eu/949/shapes/maximumTrainCurrent>
        Focus Node:
```

```
<http://data.europa.eu/949/functionalInfrastructure/contactLineSystems/5
66000-1_FR9900002157_Voie%202%20LGV%20SEA_FR9900003614_ElectrifiedOCL>
Value Node: Literal("5300.4" = None,
datatype=<http://www.w3.org/2001/XMLSchema#integer>)
Result Path: <http://data.europa.eu/949/maxTrainCurrent>
Message: maxTrainCurrent (1.1.1.2.2.2): The contact line system defines
the maximum allowable train current . This error is due to having a value
that is not an integer or having an integer that does not follow the
pattern [NNNN].
```

The error is due to a value `5300.4` that must be provided as an integer for this property, for the specific Contact Line System with URI

```
<http://data.europa.eu/949/functionalInfrastructure/contactLineSystems/566000-
1_FR9900002157_Voie%202%20LGV%20SEA_FR9900003614_ElectrifiedOCL>
```

If the user tries now to continue to the next step, to Publish, a confirmation message as the one shown in Figure 77 will be displayed. If the User decides to Proceed anyway, the generated Member State's KG will be merged with the current RINF KG and will be available to all the Apps and resources.

7.2.1.2 Upload Full RDF Dataset

Users may upload Dataset files in RDF. The file must be encoded as format N-triples (with extension `.nt`).

The Dataset file `full-rdf-test.nt` will be uploaded (See Figure 82).

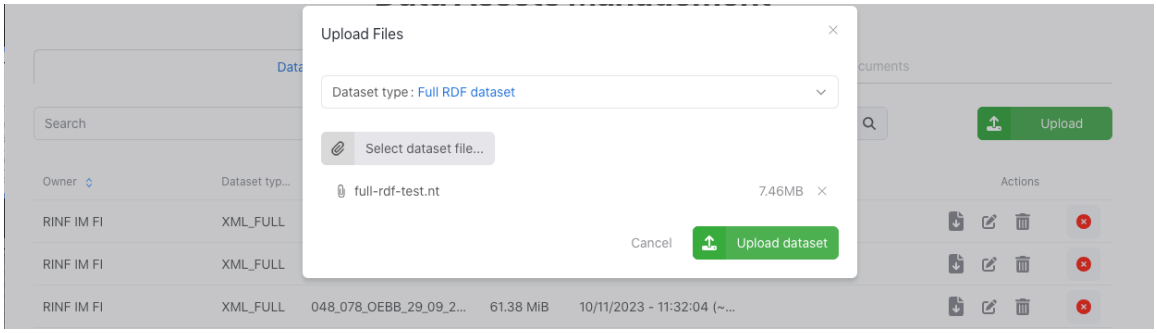


Figure 82. A Full RDF Dataset file will be uploaded.

Once the Dataset file is uploaded it is added to the list of files. When the Actions pipeline icon is clicked, the Dataset processing pipeline is displayed with steps SHACL validation and KG publishing (see Figure 83).

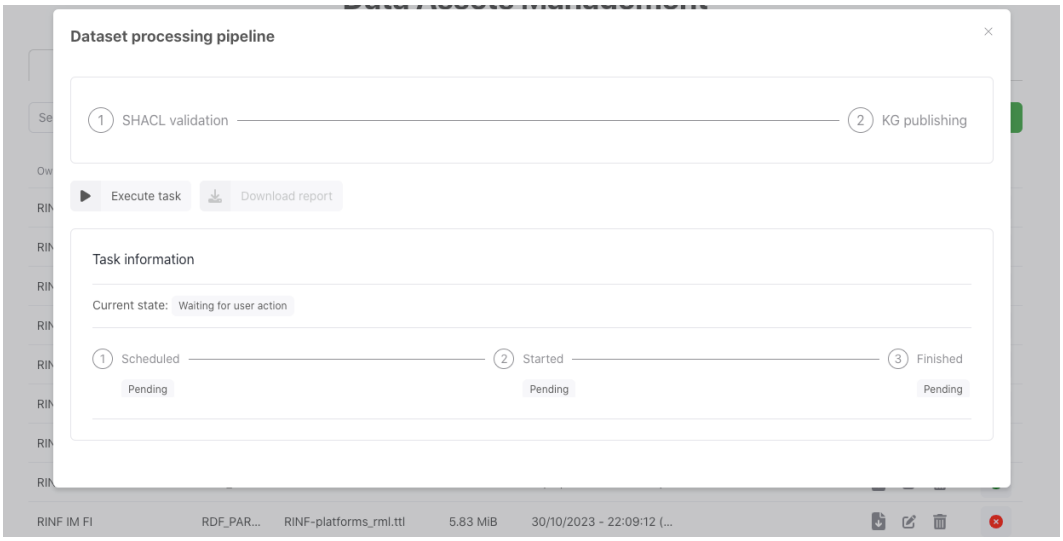


Figure 83. The Dataset processing pipeline for a Full RDF Dataset file.

The User may now execute the SHACL validation step and in case it passes without errors, the Dataset file may be published (KG publishing).

7.2.1.3 Upload Partial RDF Datasets

This functionality is meant for a User who wishes to do a small set of insertions, deletions, or updates (deletions and corresponding insertions) to the Dataset, e.g., updating the value of a parameter. In case of very large insert, delete or update files, we recommend going for a Full RDF dataset option instead.

When the User clicks the Upload button, and then the Partial Datasets option, he may choose among Partial RDF dataset (insert), Partial RDF dataset (delete), and Partial RDF dataset (update) (see Figure 84).

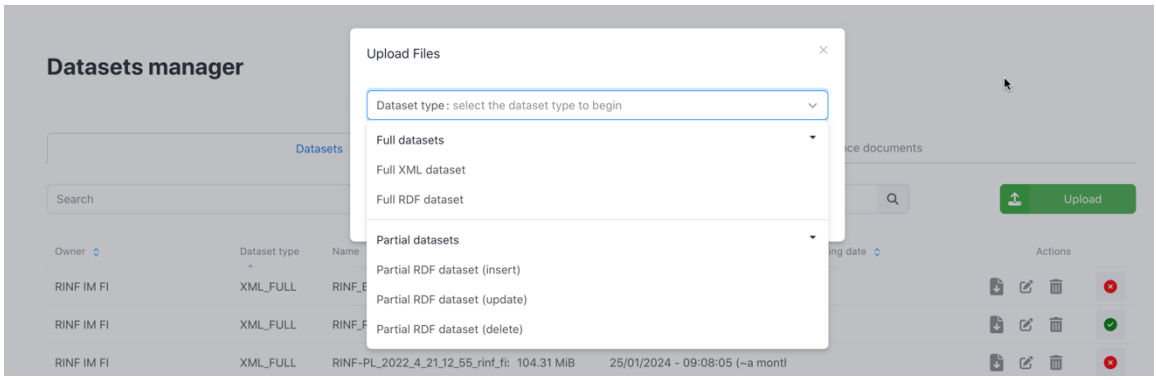


Figure 84. The Partial datasets options.

The User is able to use this functionality only if the corresponding Full XML dataset or Full RDF dataset has been already published in the KG. The reason for this is that the pipeline for Partial RDF Dataset executes insertions and deletions in the corresponding KG as seen in Figure 85.

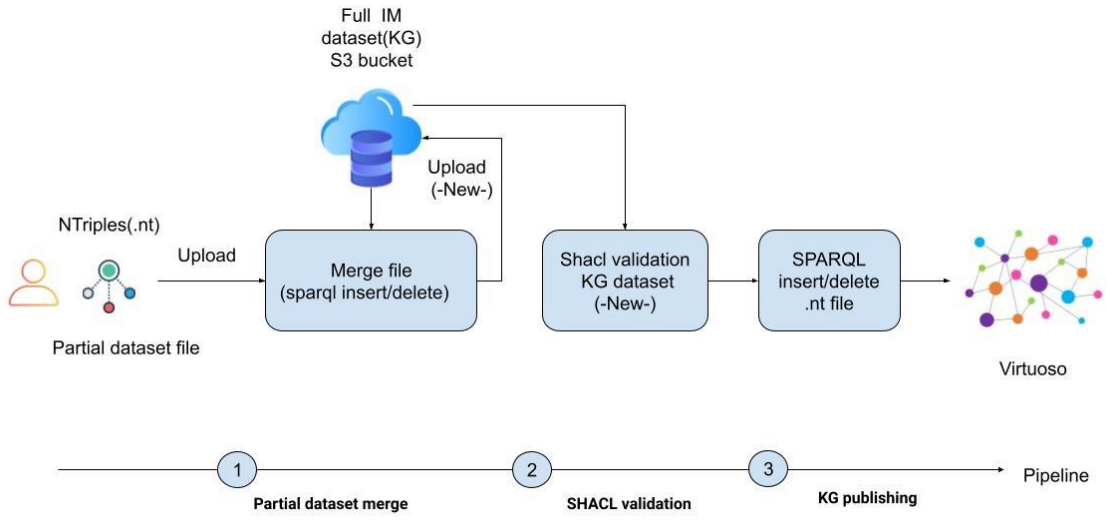


Figure 85. Architecture and pipeline for Partial RDF datasets.

The User uploads the file. It is merged into the corresponding KG through SPARQL inserts and deletes. The next step in the pipeline is the SHACL validation. The final step, the KG publication, takes place again through SPARQL inserts and deletes against the KG that is already published in the triple store.

The requirements for each type of partial dataset are as follows:

- Partial RDF dataset (insert): The file is in N-Triples (`nt`) format.
- Partial RDF dataset (update): The dataset is a `zip` compressed file that includes two files in N-Triples (`nt`) format: (1) `INSERT.nt` for the new triple set to be inserted, and (2) `DELETE.nt` for the old triple set to be removed.
- Partial RDF dataset (delete): The file is in N-Triples (`nt`) format.

In the following example, the User wishes to update the name of an OP. For this he will create two files for the deletion and the insertion of this parameter value.

The Partial RDF delete dataset can be seen in Figure 86 and the corresponding Partial RDF insert dataset is shown in Figure 87.

```
<http://data.europa.eu/949/functionalInfrastructure/operationalPoints/dcae55ea5e7836721e1ca3fb95871e6
27de017e1> <http://data.europa.eu/949/opName> "Wien Westbf (in Ws)".
```

Figure 86. Partial RDF delete dataset: the triple with the "old" value.

```
<http://data.europa.eu/949/functionalInfrastructure/operationalPoints/dcae55ea5e7836721e1ca3fb95871e6
27de017e1> <http://data.europa.eu/949/opName> "Wien Westbf (in Ws) NEW".
```

Figure 87. A partial RDF insert dataset: the triple with the "new" value.

The user will upload these two files in the compressed format using the Partial RDF file (update) option. When the User clicks this option a dialog box will appear where the user may select the Dataset file to upload (see Figure 88).

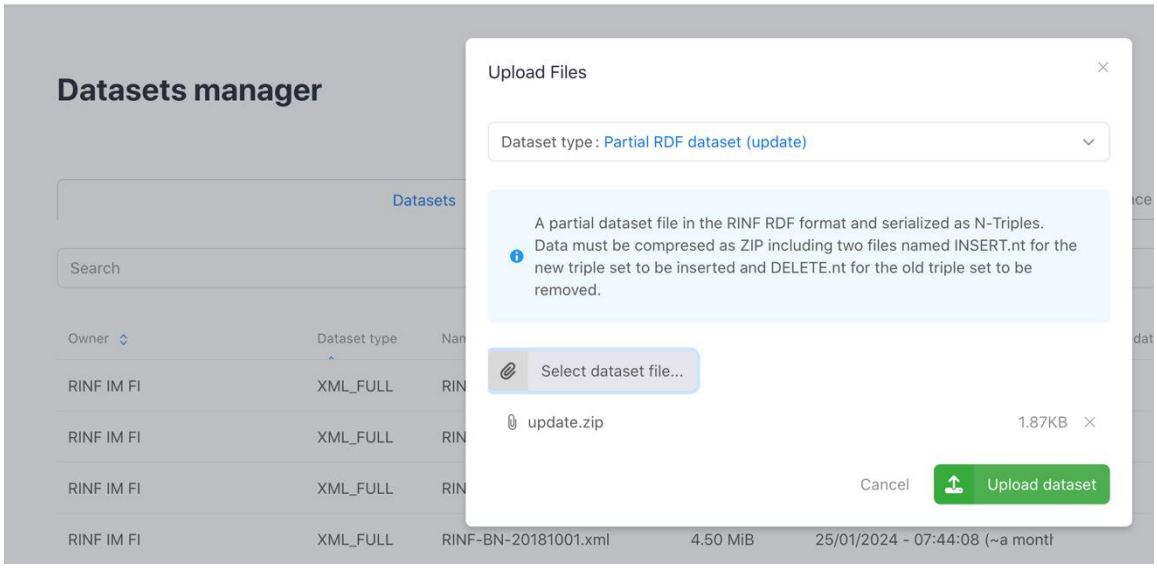


Figure 88. Selecting a Partial RDF dataset (update) to upload. The `update.zip` file is selected.

The pipeline can be seen in Figure 89.

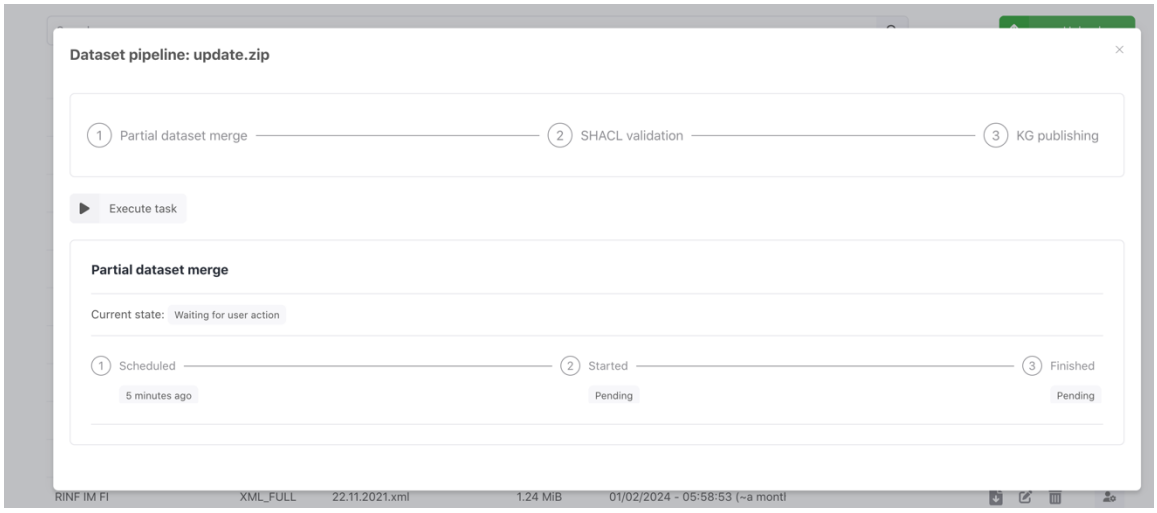


Figure 89. The pipeline for Partial RDF dataset (update).

Once the User has executed all of the steps in the pipeline, the KG will include the updated triple (the triple with the "old" value has been deleted and the triple with the "new" value has been inserted). In this example the pipeline has Finished successfully as shown in Figure 90.

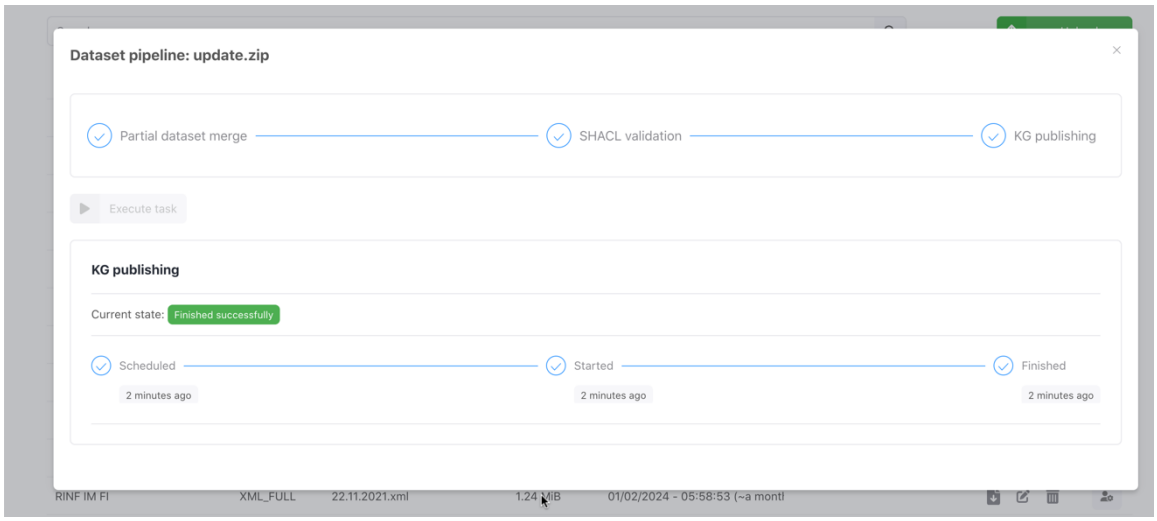


Figure 90. The pipeline for Partial RDF dataset (update) has Finished successfully.

In this example several points need to be taken into account:

- An error message will be displayed if the User has not published previously a Full RDF Dataset or a Full XML Dataset. This is the case for all the Partial Datasets options.
- If the User would have uploaded separately the Partial RDF dataset (delete) file, the SHACL validation would have Finished with errors because the OP name is a mandatory property.

  If the User would have uploaded separately the Partial RDF dataset (insert) file, the SHACL validation would have Finished with errors because there can be at most one OP name. The execution of the pipeline can be seen in

- Figure 91.

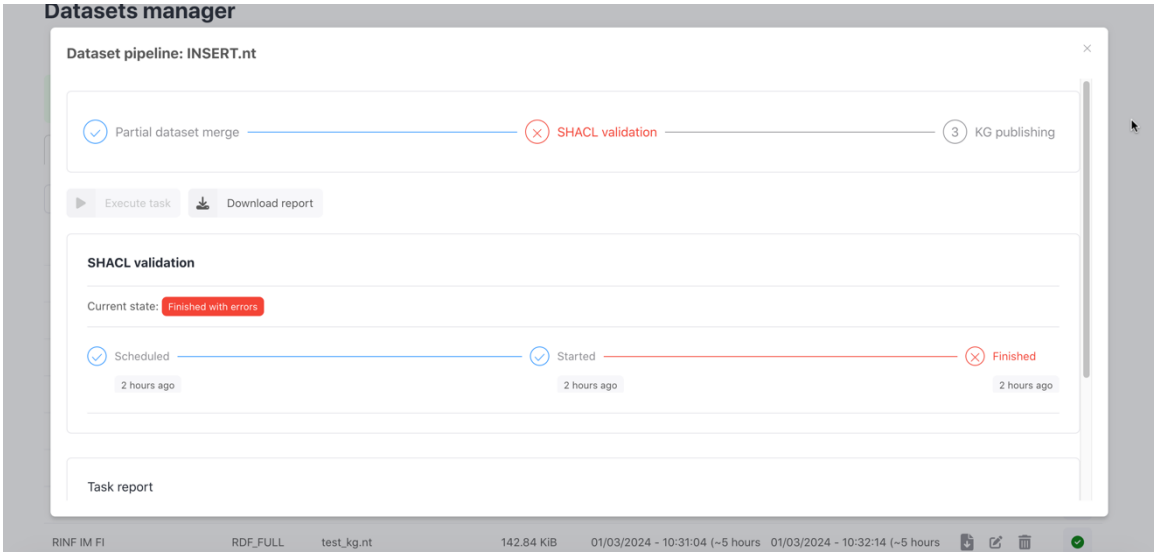


Figure 91. Execution of the Partial RDF dataset (insert) where the SHACL validation has Finished with errors.

In the next example there is a triple for a property that until now has been “not yet available” but now has a value. The User will delete the “not yet available” triple and insert the triple with a value for this property.

The RDF partial delete file can be seen in Figure 92 and the corresponding RDF dataset insert file is shown in Figure 93.

```
<http://data.europa.eu/949/functionalInfrastructure/operationalPoints/dcae55ea5e7836721e1ca3fb95871e6
27de017e1> <http://data.europa.eu/949/notYetAvailable>
<http://data.europa.eu/949/opTypeGaugeChangeover>.
```

Figure 92. A partial RDF delete dataset for a "not yet available" value.

```
<http://data.europa.eu/949/functionalInfrastructure/operationalPoints/dcae55ea5e7836721e1ca3fb95871e6
27de017e1> < <http://data.europa.eu/949/opTypeGaugeChangeover> "type
abc"^^<http://www.w3.org/2001/XMLSchema#string>.
```

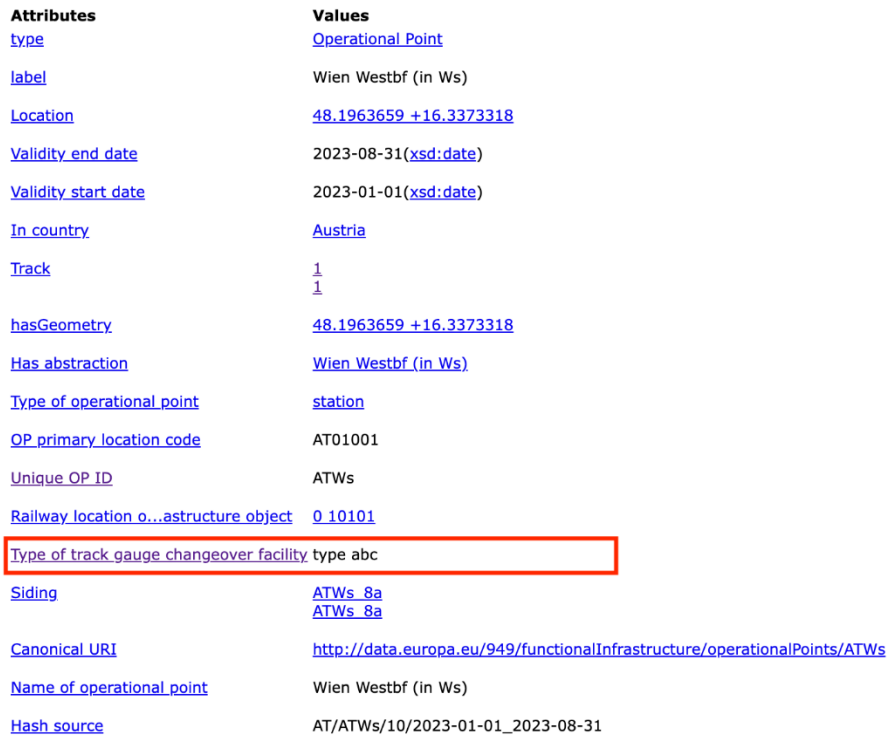
Figure 93. An RDF partial insert dataset with the value for the property.

Then the current value for the property Type of track gauge changeover facility will be Not provided.

Attributes	Values
type	Operational Point
label	Wien Westbf (in Ws)
Location	48.1963659 +16.3373318
Validity end date	2023-08-31(xsd:date)
Validity start date	2023-01-01(xsd:date)
In country	Austria
Track	1 1
hasGeometry	48.1963659 +16.3373318
Has abstraction	Wien Westbf (in Ws)
Type of operational point	station
OP primary location code	AT01001
Unique OP ID	ATWs
Railway location o...astructure object	0 10101
Siding	ATWs_8a ATWs_8a
Canonical URI	http://data.europa.eu/949/functionalInfrastructure/operationalPoints/ATWs
Not provided	Type of track gauge changeover facility
Name of operational point	Wien Westbf (in Ws)
Hash source	AT/ATWs/10/2023-01-01_2023-08-31

Figure 94. The OP has not provided a value for Type of track gauge changeover facility (note that this is an old user interface and will be replaced in the next version of the user manual).

The Partial RDF dataset (update) option will be used to delete the "not yet available" triple and insert the property value. The new published KG can be seen in Figure 95.



Attributes	Values
type	Operational Point
label	Wien Westbf (in Ws)
Location	48.1963659 +16.3373318
Validity end date	2023-08-31(xsd:date)
Validity start date	2023-01-01(xsd:date)
In country	Austria
Track	1 1
hasGeometry	48.1963659 +16.3373318
Has abstraction	Wien Westbf (in Ws)
Type of operational point	station
OP primary location code	AT01001
Unique OP ID	ATWs
Railway location o...astructure object	0 10101
Type of track gauge changeover facility	type abc
Siding	ATWs_8a ATWs_8a
Canonical URI	http://data.europa.eu/949/functionalInfrastructure/operationalPoints/ATWs
Name of operational point	Wien Westbf (in Ws)
Hash source	AT/ATWs/10/2023-01-01_2023-08-31

Figure 95. The OP with a value for Type of track gauge changeover facility (note that this is an old user interface and will be replaced in the next version of the user manual).

7.3 Documents

When the User clicks on the Reference Documents tab, a table with all the reference documents that the User has uploaded will appear (see Figure 96).

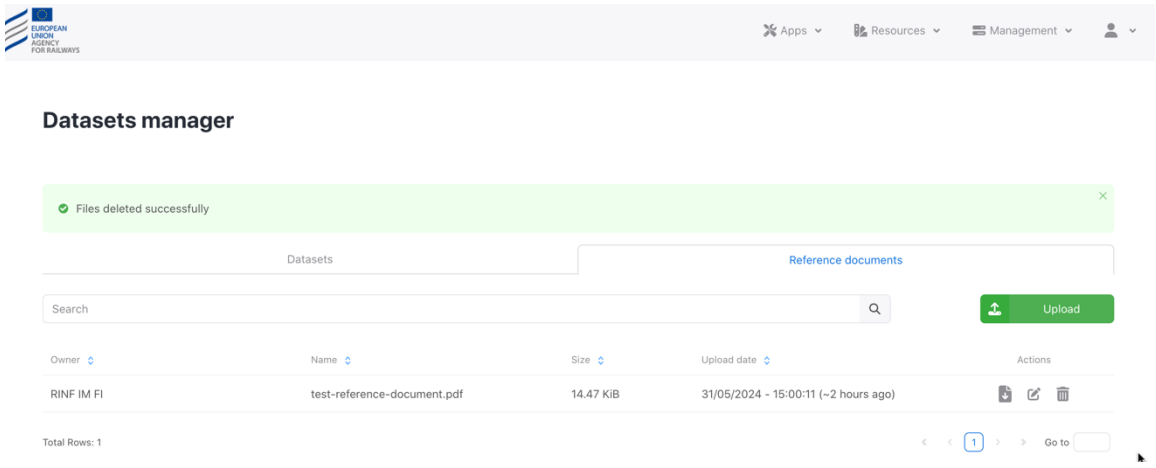


Figure 96. The Documents table.

The columns in the Documents table are as follows:

- Owner: The full name of the User who has uploaded the document.

- Name: The name of the document that has been uploaded.
- Size: The size of the document in MBs.
- Date Uploaded: The date when the document was uploaded.
- Actions. The User can Download, Rename or Delete the file.

Please note that the Name of the document that is being uploaded must be identical (including special characters) to the name provided as a value of the corresponding parameter, i.e. 1.1.1.1.2.4.4 (IPP_StructureCheckDocRef), 1.1.1.1.3.1.3 1.2.1.0.3.6 (ILL_GaugeCheckDocRef).

8 Notifications management

A User in the role of RU can choose the Notifications management option that appears in the Management dropdown list of the Landing Page.

Once the option is clicked, a Notifications manager window is displayed (see Figure 97)

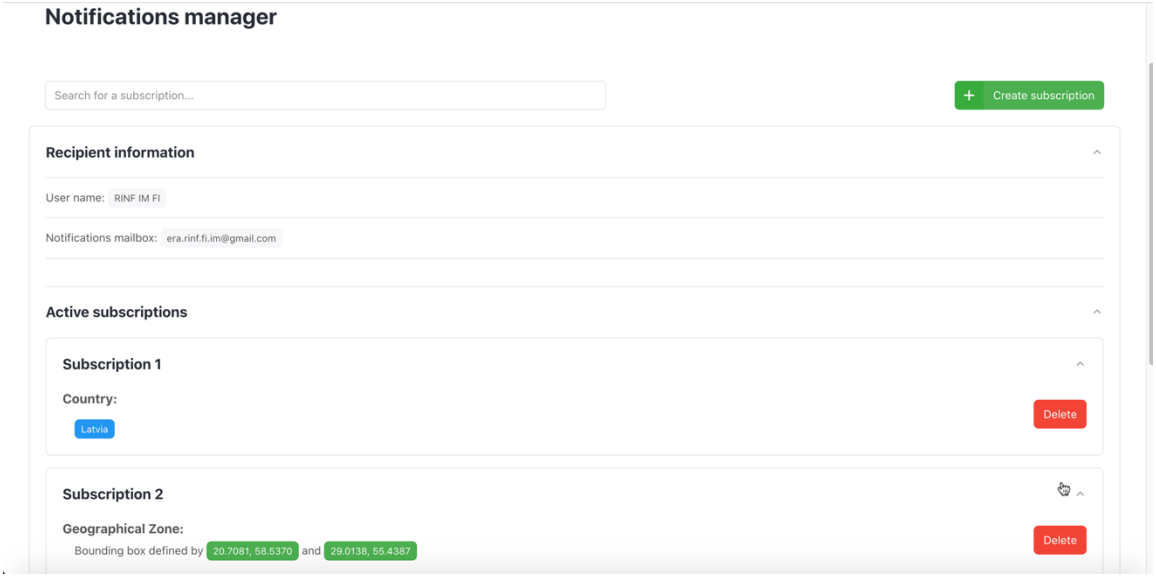


Figure 97. The Notifications Manager Page.

The Notifications management option provides all the functionalities that a User may need to manage subscriptions to notifications. Notifications are sent to subscribed users when changes are made to the knowledge graph (KG) that affect the RINF elements for which they created the subscription. The System will send notifications when the User is subscribed to changes on specific Countries, Operational Points, Sections of Lines or Vehicle Types. In Figure 97 the User has two active subscriptions: a Country subscription, Latvia, and a Geographical Zone subscription defined by the bounding box coordinates.

The following functionalities are included:

- A User wants to Search for a subscription
- A User wants to Create subscription
- A User wants Delete a subscription

Note that the Recipients information includes the User name and the Notifications mailbox where messages will be sent.

8.1 Search for a subscription

The User introduces free text in the box under the main title, and the Active subscriptions will be filtered as shown in Figure 98.

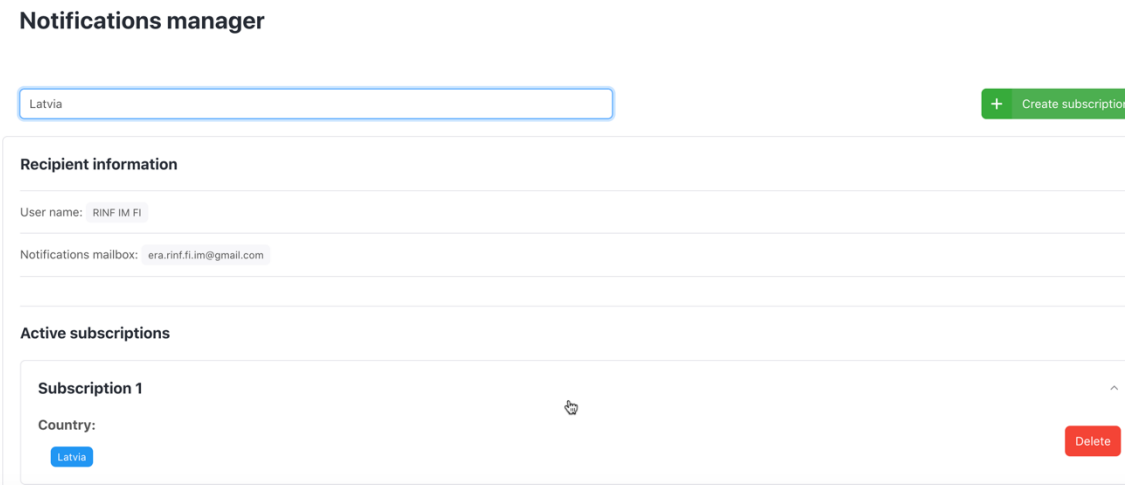


Figure 98. Search for subscriptions that contain the word "Netherlands".

8.2 Create subscription

To create a new subscription the User clicks Create subscription button. Six types of subscriptions will be displayed in a dropdown list (see Figure 99)

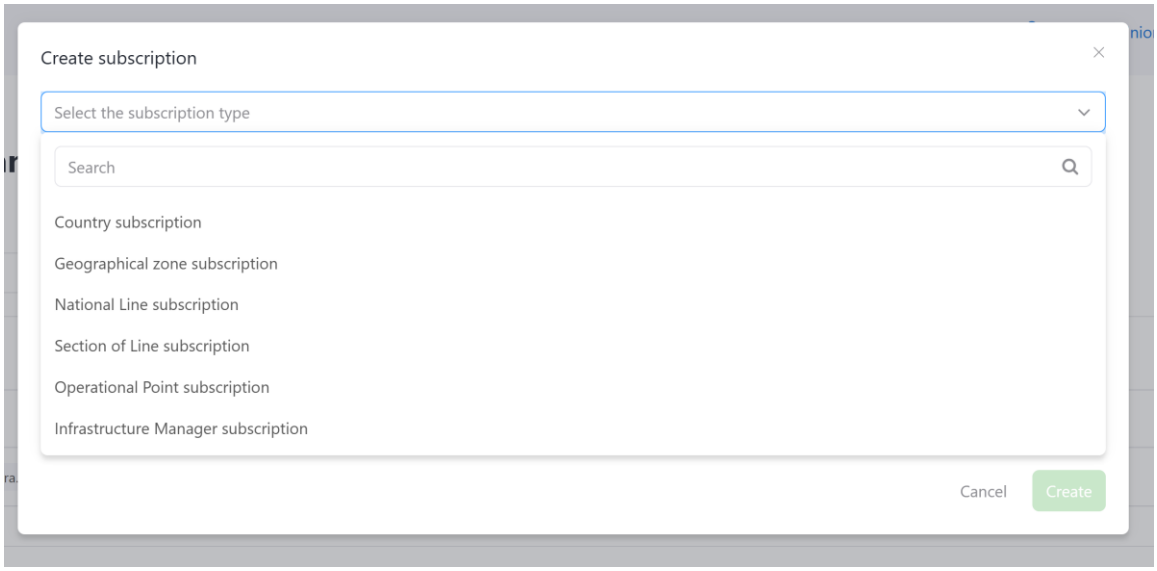


Figure 99. Subscription types

When the user selects the Geographical zone subscription type,

When the User selects Country subscription a box will be displayed where the country may be selected (see Figure 100).

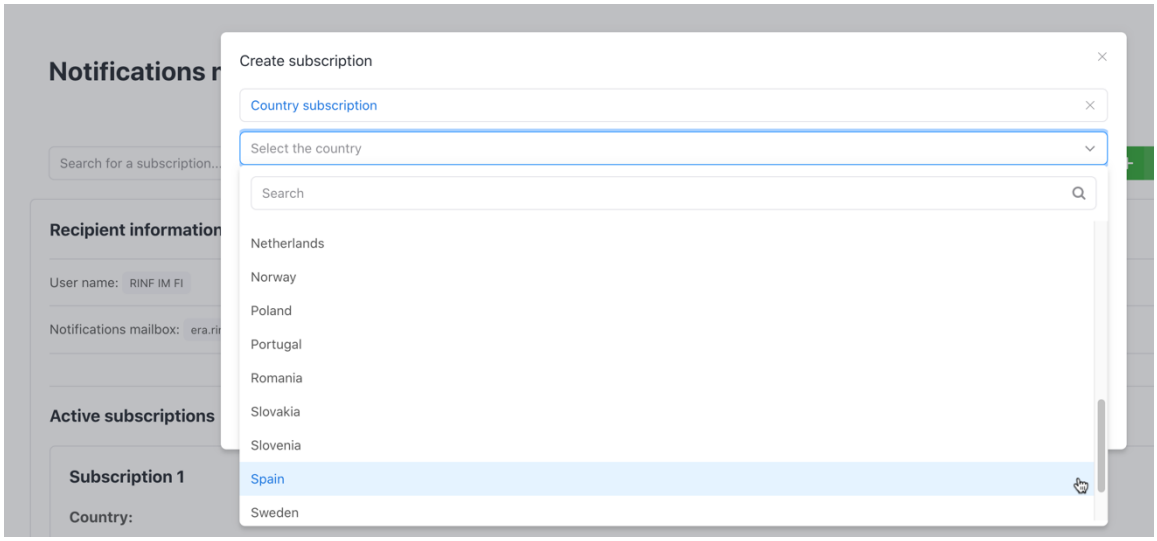


Figure 100. Creation of a Country subscription. The country Spain is selected.

Once the user clicks Create, the subscription is created and displayed as the last subscription in the Active subscriptions list (see Figure 101).

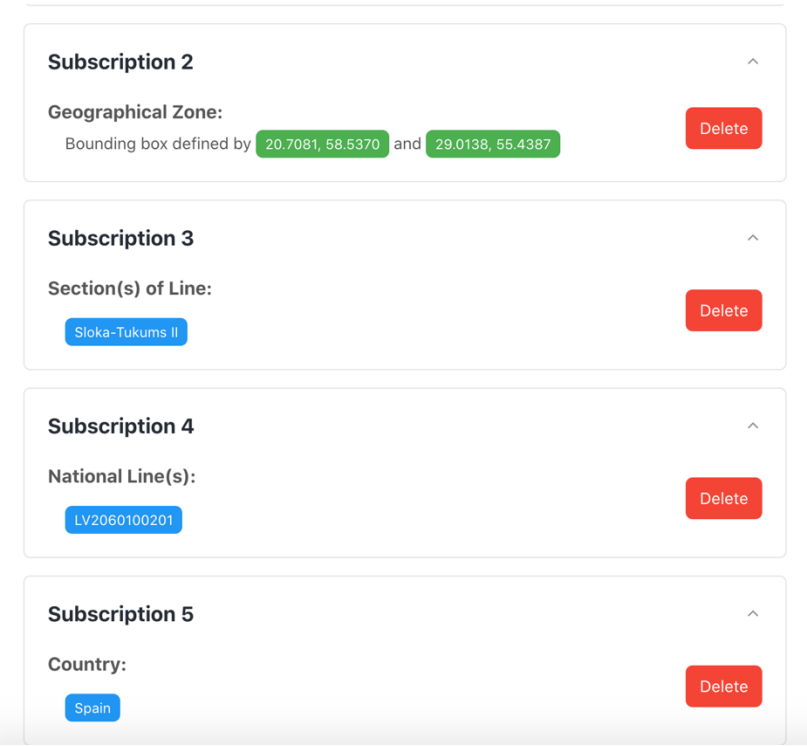


Figure 101. The Active subscriptions list includes a Country subscription that has been created.

If the Geographical Zone subscription type is selected, a map is displayed where the User can define a bounding box (see Figure 102).

When the user clicks Create, the subscription is created, and the coordinates of the bounding box are displayed (see Figure 103).

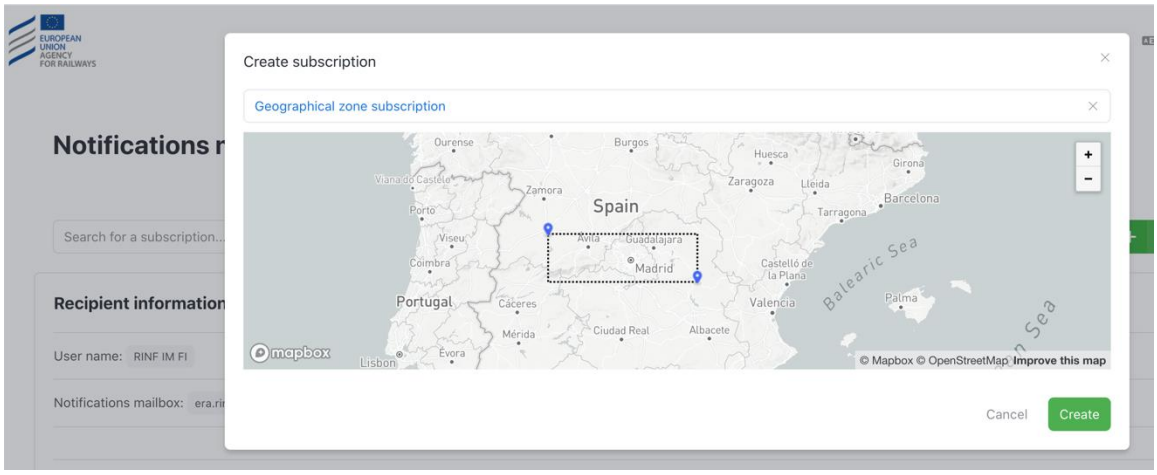


Figure 102. The definition of a bounding box for a Geographical Zone subscription

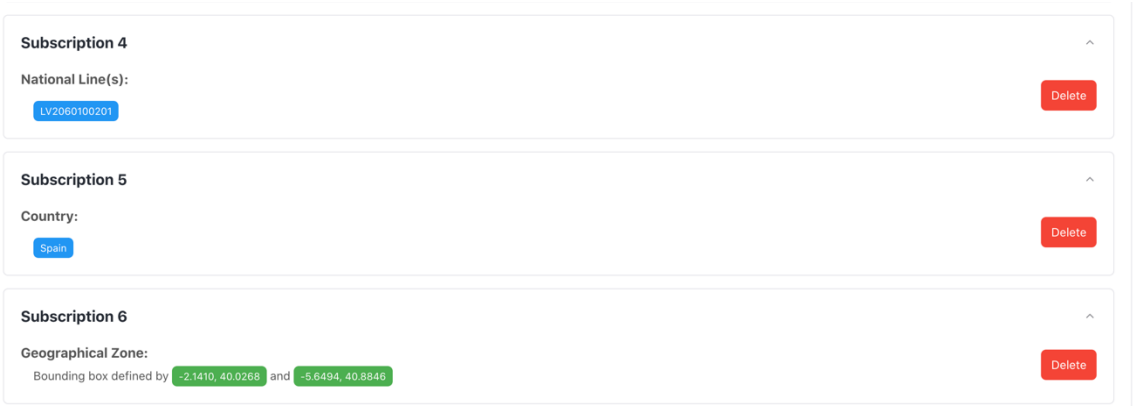


Figure 103. A subscription to a Geographical Zone has been created

If the User selects the Operational Point subscription type, a dialog box will be displayed where the user can search for an Operational Point that he wishes to subscribe to (see Figure 104).

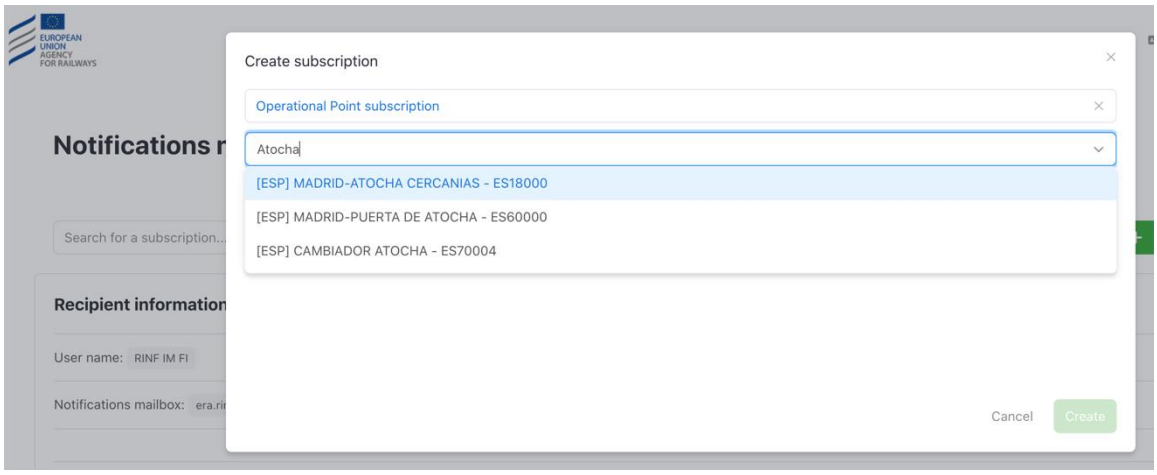


Figure 104. Creation of an Operational Point subscription. A search is done by the name Atocha.

Once the operational point is selected, it is added to the list of Selected: operational points (see Figure 105).

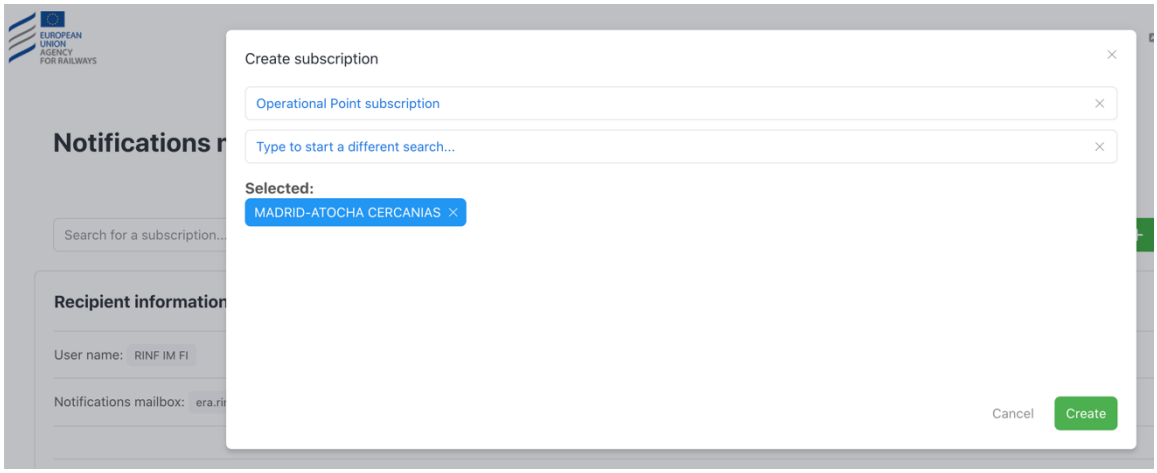


Figure 105. The User has selected an operational point for subscription

When the user clicks Create, the subscription is added at the end of the Active subscriptions list (see Figure 106).

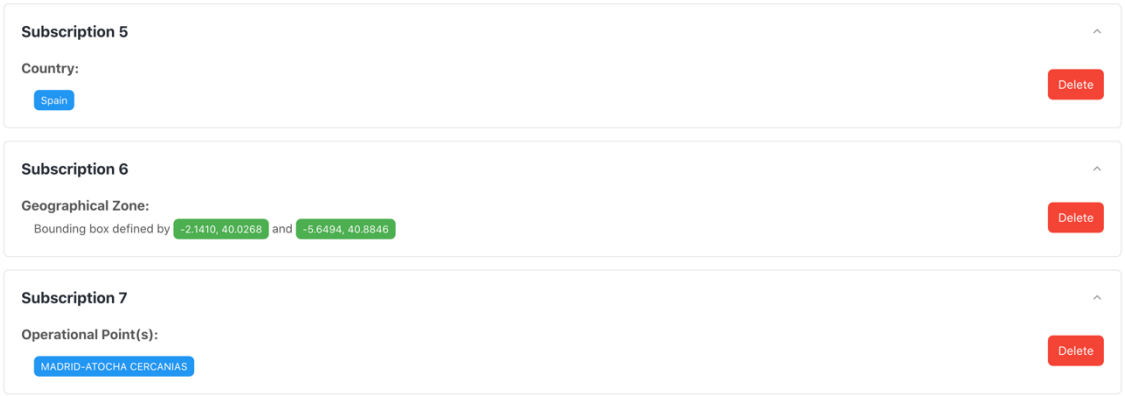


Figure 106. A subscription to an operational point has been created

Similarly, subscriptions can be created for a Section of Line or a National Line.

For the Operational Point subscription type, a dialog box will be displayed where the user can search for an Infrastructure Manager (IM) that he wishes to subscribe to (see Figure 107).

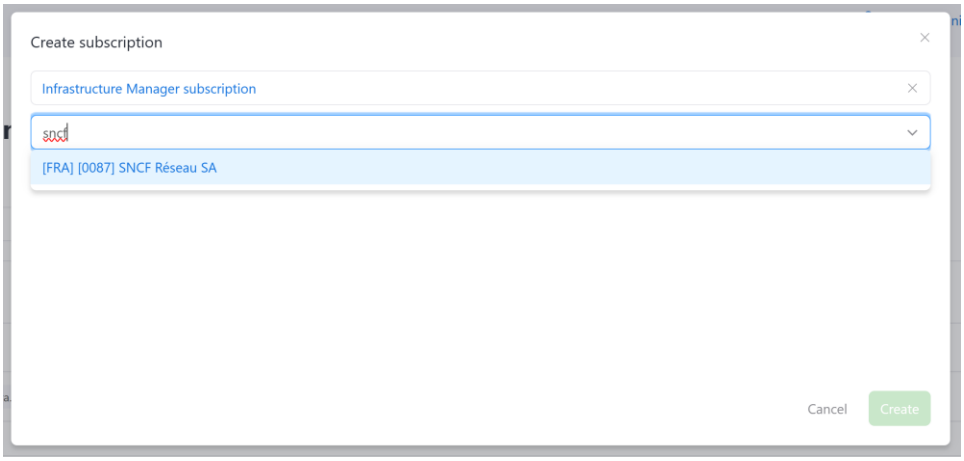


Figure 107. The User has selected an Infrastructure Manager for subscription

8.3 Delete

To delete a subscription, the User clicks the Delete button that is next to the subscription. The subscription will be deleted and will not appear anymore in the Active subscriptions list.

9 References

- Consolidated text: Commission Implementing Regulation (EU) 2019/777 of 16 May 2019 on the common specifications for the register of railway infrastructure and repealing Implementing Decision 2014/880/EU (Text with EEA relevance).

  (https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A02019R0777-20230928)

Appendix A. Guidance for annotation in RDF using the ERA ontology

A.1 Introduction

This document presents a guidance and a suggestion of tools for the development of Total or Partial RDF dataset files that can be uploaded in the Dataset Manager option of the RINF Landing Page as the following types:

- Full RDF dataset which is a complete Dataset file in RDF format that contains all of the RINF elements managed by the IM or the NRE.
- Partial RDF dataset (insert) which is a Dataset file in RDF format that contains only those elements that need to be inserted.
- Partial RDF dataset (delete) which is a Dataset file in RDF format that contains only those elements that need to be deleted.
- Partial RDF dataset (update) which is a Dataset file in zip format that contains in turn two RDF files for deleting and inserting triples as a "bundle".

The development of an RDF dataset file involves the annotation of RINF data with the ERA vocabulary concepts (https://data-interop.era.europa.eu/era-vocabulary/). The annotation consists in tagging or labeling the data using the classes and properties defined in the Vocabulary, i.e. the enrichment of data with explicit, machine-readable domain knowledge (*semantification*). Annotating data is essentially assigning vocabulary terms to specific data instances.

In this document the terms Ontology and Vocabulary will be used interchangeably.

Section A.2 of this Appendix briefly describes the *semantification* concept. Section A.3 explains RINF data annotation with several use cases, and Section A.4 describes some of the tools that can be used for data annotation.

A.2 Semantification and triples

Semantification is the process of describing data by relating them to the terms (concepts, attributes, and relationships) defined in one or more ontologies, i.e. stating to which terms they belong to, and the relationships that they maintain.

The ontology-based description of the data, otherwise known as *semantically annotated data* is encoded in the ontology language, e.g., OWL, RDF, and is expressed in *triples* or statements, much like the ontologies themselves.

Data that has been semantified by an ontology is typically stored and made accessible in a *knowledge graph*. A Knowledge Graph represents a collection of interlinked entities, each representing a separate piece of information. Knowledge graphs put data in context via linking and semantic metadata and this way provide a framework for data integration, unification, analytics and sharing.

A *Semantic Annotation* is a statement that relates data to terms and relationships in the ontology.

A statement can be seen as a *triple* composed of a Subject, Predicate and Object. A graphical shorthand notation for triples is depicted in Figure 108.

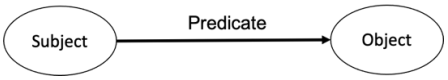


Figure 108. Shorthand graphical notation for triples

Semantic annotations relating individuals to the concepts in the ERA vocabulary may be “ATWs is an Operational Point”, “10101_ATWs_ATOw is a Section of line”, “Section of line 10101_ATWs_ATOw has as OP start, ATWs”. This example can be seen in Figure 109.

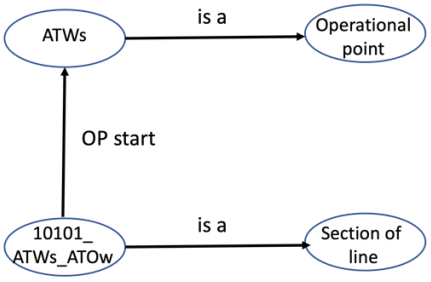


Figure 109. An example of semantic annotations

A.3 Use cases for the annotation of data

In this section, several approaches for annotating RINF data will be presented. Some of the system’s functionalities such as the Export in the Search Form or the KG generated in the pipeline of a Full XML dataset can be used in this process.

Note that RDF code may also be generated from scratch using one of the tools presented in Section A.4 or using a text editor.

A.3.1 Update of the value of a parameter

This Use Case can be addressed using two different approaches. For each one, an example will be provided.

Approach 1. Update the KG that was generated in the Dataset Management pipeline

We will use data provided by an IM. There is a SOL with the following information:

- SOL line identification: "566000-1"
- SOL OP start:"FR9900002157"
- SOL OP end: "FR9900003614"
- SOL length: "1.533"
- Validity start date: "2017-01-01"
- Validity end date: "2382-01-01"

Suppose that the IM wants to modify the SOL length parameter to "1.633".

The user can download the KG generated in the pipeline in the KG generation step, as seen in Figure 110. This file is in “Turtle” format.

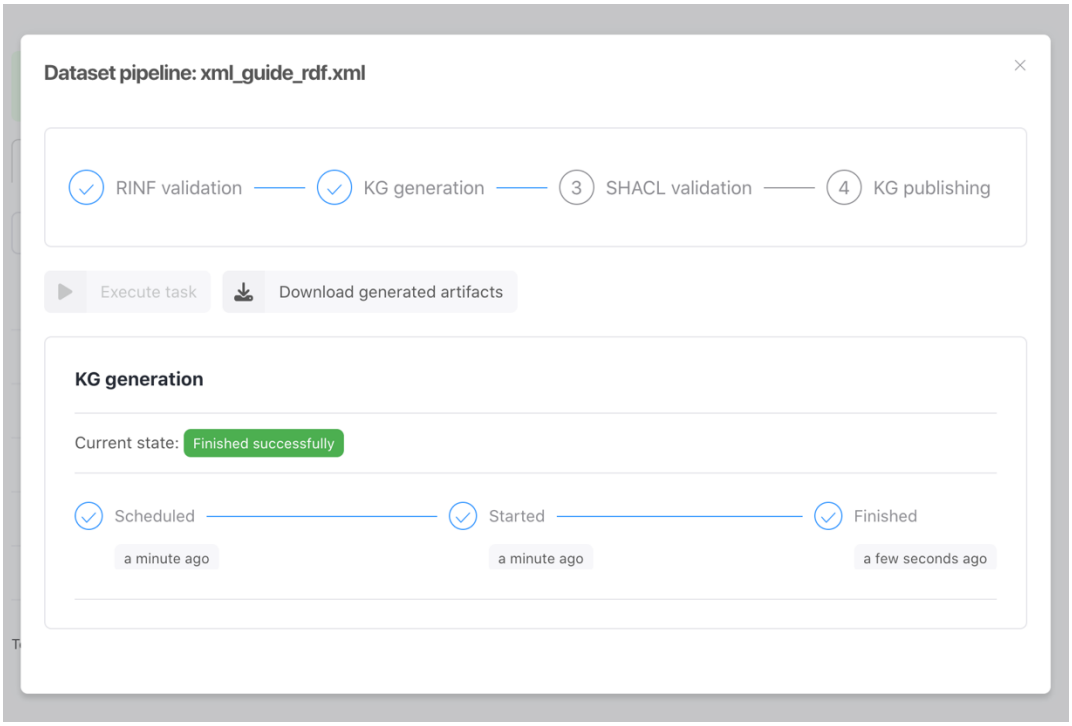


Figure 110. An example of a Dataset pipeline

Note that in order to ensure that the KG for this Use Case is valid, the RINF validation step must have Finished successfully.

In order to search the triples that need to be modified, the User will need to generate the unique Hash ID. The input string for the SoL has the following structure (Hash key structures are presented in Appendix B, Section B.4.1:

```
{SOLineIdentification}_{SOLOPStart}_{SOLOPEnd}/{ValidityDateStart}_{ValidityDateEnd}
```

Thus, for this example the input string follows:

"566000-1_FR9900002157_FR9900003614/2017-01-01_2382-01-01"

This input string is the value of the property Hash source that can be viewed through the Search Form or through a query using the Endpoint.

The `hashcode.py` script that follows is then executed:

```
import hashlib
def sha1_encrypt(input_string):
    # Create a new SHA-1 hash object
    sha1_hash = hashlib.sha1()
    # Update the hash object with the input string
    sha1_hash.update(input_string.encode('utf-8'))
    # Get the hexadecimal representation of the digest
    encrypted_string = sha1_hash.hexdigest()
    return encrypted_string
input_string = "566000-1_FR9900002157_FR9900003614/2017-01-01_2382-01-01"
```

```
encrypted_string = sha1_encrypt(input_string)
print("Original string:", input_string)
print("SHA-1 encrypted string:", encrypted_string)
```

The encrypted id is “2fbbc34d4f63f7ac9276fe43475fcb333dcbceec”. This id can be found in the KG file. The triple in the RDF file that the user will update can be seen in the following block:

```
<http://data.europa.eu/949/functionalInfrastructure/sectionsOfLine/2fbbc34d4f63f7ac9276fe43475fcb333dcbceec>
<http://data.europa.eu/949/length>
"1533"^^<http://www.w3.org/2001/XMLSchema#double>.
```

The new block is now:

```
<http://data.europa.eu/949/functionalInfrastructure/sectionsOfLine/2fbbc34d4f63f7ac9276fe43475fcb333dcbceec>
<http://data.europa.eu/949/length>
"1633"^^<http://www.w3.org/2001/XMLSchema#double>.
```

The User can now upload the updated KG file as a Full RDF dataset and execute the corresponding pipeline.

Approach 2. Use the Export option in the Search Form

The User can search for the element with the parameter value that needs to be updated. In this case it is a track within a SoL. It has the following information:

- SOL start OP id: “IT01077”
- SOL end OP id: “IT01933”
- Identification of track: “013-090.BC-BC02”
- Rail inclination (SKOS taxonomy): http://data.europa.eu/949/concepts/rail-inclinations/rinf/20 (Preferred label is “1/20”)

Suppose that the IM wants to modify the Rail inclination parameter to "1/10", which means that now its value must be “http://data.europa.eu/949/concepts/rail-inclinations/rinf/10”.

Figure 111 shows the filters that have been applied in the Search Form for this SoL.

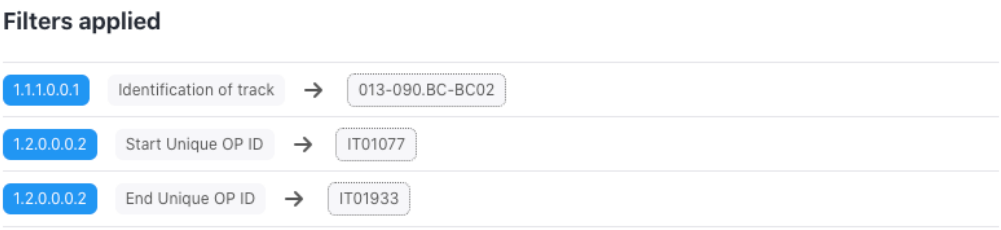


Figure 111. Filters applied in the Search Form for a specific SoL

The result can be seen in Figure 112.

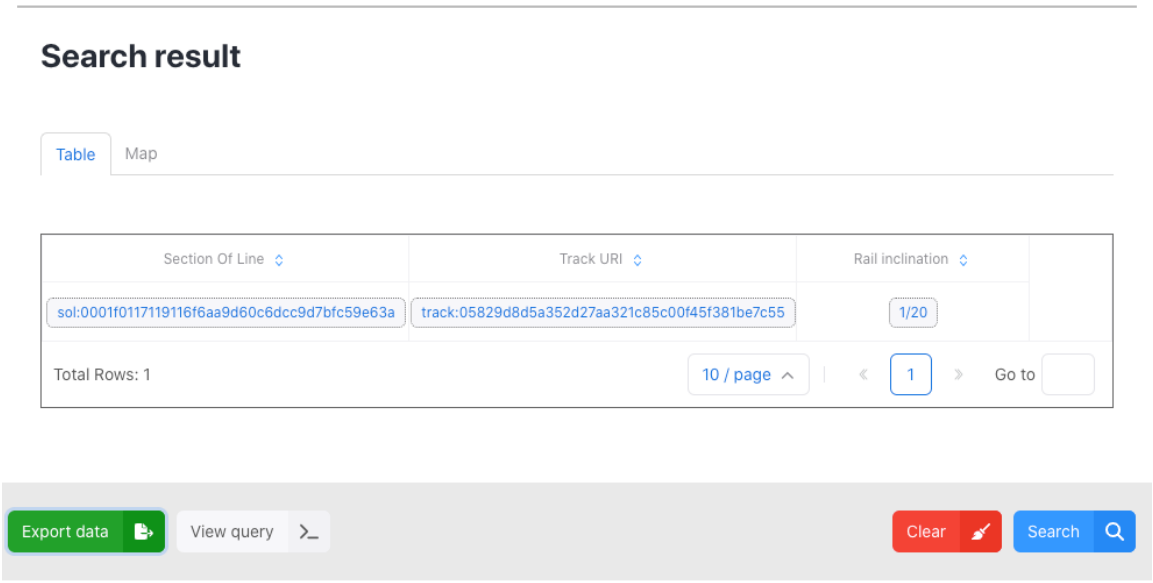


Figure 112. Search result of the SoL that will be updated

The User selects the Full Linked data dump N-triples option in the drop-down menu as seen in Figure 113.

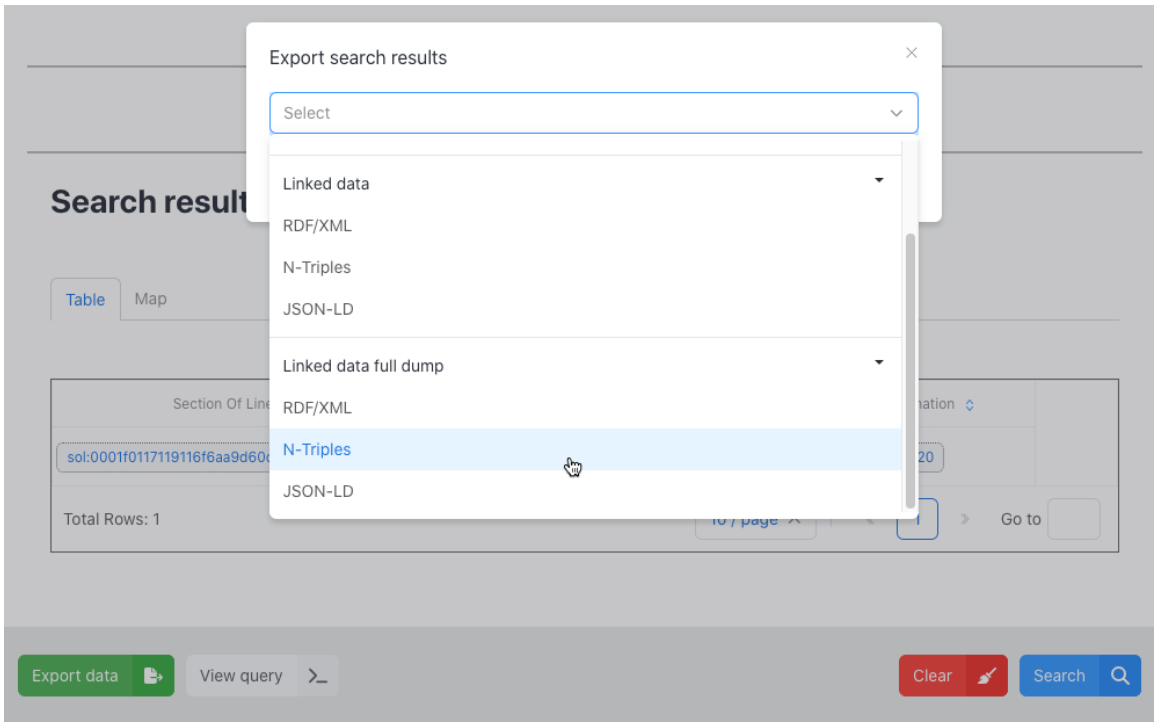


Figure 113. The User exports the SoL in N-triples format

The SoL with all its elements will be exported in. nt format. In order to update the parameter in the specific track, the triples for this track should be searched using its Hash Id. In the search form, the User can right click on the Track URI and copy the Hash id:

"05829d8d5a352d27aa321c85c00f45f381be7c55".

The triple in the RDF file that the user will delete can be seen in the following block:

```
<http://data.europa.eu/949/functionalInfrastructure/tracks/05829d8d5a352d27aa321c85c00f45f381be7c55>
```

```
<http://data.europa.eu/949/railInclination>
<http://data.europa.eu/949/concepts/rail-inclinations/rinf/20>.
```

The triple in the RDF file that the user will insert can be seen in the following block:

```
<http://data.europa.eu/949/functionalInfrastructure/tracks/05829d8d5a352d27aa321c85c00f45f381be7c55>
<http://data.europa.eu/949/railInclination>
<http://data.europa.eu/949/concepts/rail-inclinations/rinf/10>.
```

The User can now either:

- Upload the first block as a Partial RDF dataset (delete) file and execute the pipeline. Following, upload the second block with the updated value as a Partial RDF dataset (insert) file and execute the pipeline.
- Create the `DELETE.nt` and `INSERT.nt` files containing respectively the first and second blocks, compressing these in one .zip file, uploading it as a Partial RDF dataset (update), and executing the pipeline.

Note that both options require the Full RDF dataset (KG) to be already published.

A.3.2 Add the value of a parameter that had not been provided (Not provided)

This Use case addresses the requirement of parameter values that have not been provided yet but that eventually their value is determined.

Approach 1. Use the Export option in the Search Form

The User will provide a value for the parameter `gauging profile` that until now was `Not provided`. The SoL and track are the same than in the previous example:

- SOL start OP id: “IT01077”
- SOL end OP id: “IT01933”
- Identification of track: “013-090.BC-BC02”
- Gauging profile (SKOS taxonomy): “Not provided”

The triple in the RDF file that the user will delete can be seen in the following block:

```
<http://data.europa.eu/949/functionalInfrastructure/tracks/05829d8d5a352d27aa321c85c00f45f381be7c55>
<http://data.europa.eu/949/notYetAvailable>
<http://data.europa.eu/949/gaugingProfile>.
```

The triple in the RDF file that the user will insert can be seen in the following block:

```
<http://data.europa.eu/949/functionalInfrastructure/tracks/05829d8d5a352d27aa321c85c00f45f381be7c55>
<http://data.europa.eu/949/gaugingProfile>
<http://data.europa.eu/949/concepts/gaugings/rinf/70>.
```

As in the previous example, the User can then either upload these two files as partial datasets for deletion and insertion, or also as a partial dataset for update.

A.3.3 Delete an infrastructure element

This Use Case addresses the deletion of any infrastructure element, e.g., OP, platform edge, tunnel. The User has to consider that all the triples where the element is a subject or an object must be deleted in order for the KG to remain consistent.

Approach 1. Use the Export option in the Search Form

The User wants to delete a certain platform edge, "1 (Dtz)", within the Station with name "Delft". The information is as follows:

- Name of OP: "Delft"
- Name of track:"661"
- Identification of the platform edge: "1 (Dtz)"

In the search form the filters applied can be seen in Figure 114.

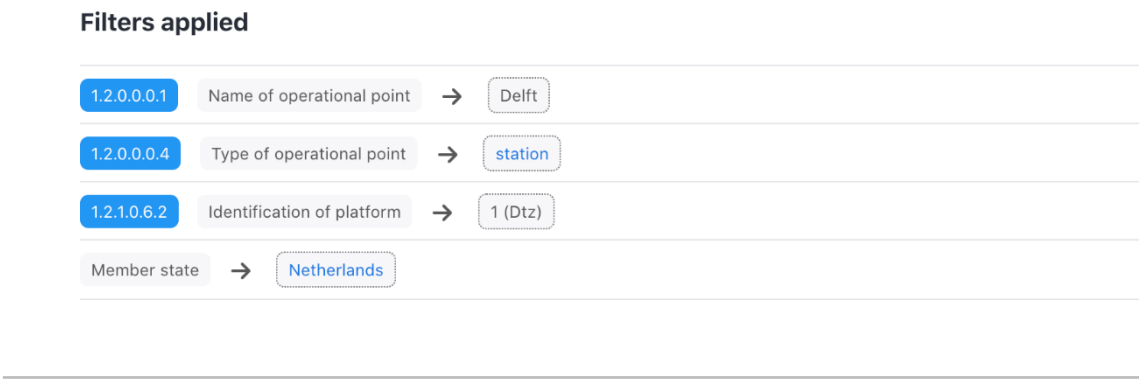


Figure 114. Filters applied in the Search form for a specific platform edge.

The search results can be seen in Figure 115.

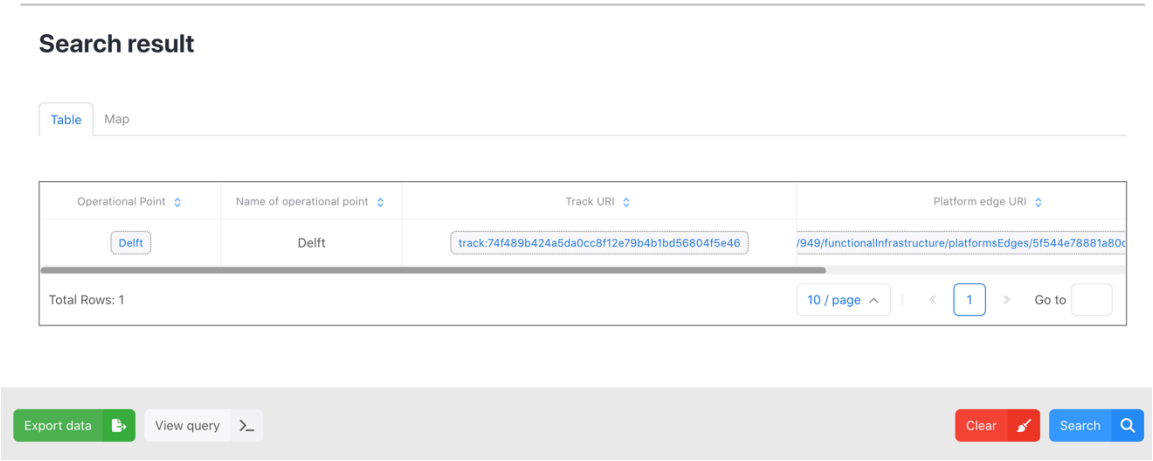


Figure 115. Search result of the platform edge to be deleted.

The OP with all its elements will be exported in. nt format. In order to delete the specific platform, the triples for this track should be searched using its Hash Id. The Hash id of the platform is:

"5f544e78881a80d8cb988232b46c12b6d1b6514d".

Then, a new file must be generated with all of the triples where this id is either in the subject or the object of the triple.

This set of triples contains information on the platform´s type (PlatformEdge), its label, platform Id and IM code:

```
<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/5f544e78881a80d8cb988232b46c12b6d1b6514d>    <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
      <http://data.europa.eu/949/PlatformEdge>.

<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/5f544e78881a80d8cb988232b46c12b6d1b6514d>    http://www.w3.org/2000/01/rdf-schema#label "661, 1 (Dtz)".

<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/5f544e78881a80d8cb988232b46c12b6d1b6514d>    <http://data.europa.eu/949/platformId> "1 (Dtz)".

<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/5f544e78881a80d8cb988232b46c12b6d1b6514d>    <http://data.europa.eu/949/imCode>   "0084".
```

The track that the platform edge belongs to:

```
<http://data.europa.eu/949/functionalInfrastructure/tracks/74f489b424a5da0cc8f12e79b4b1bd56804f5e46>    <http://data.europa.eu/949/platformEdge>
      <http://data.europa.eu/949/functionalInfrastructure/platformsEdges/5f544e78881a80d8cb988232b46c12b6d1b6514d>.
```

The parameter values:

```
<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/5f544e78881a80d8cb988232b46c12b6d1b6514d >    <http://data.europa.eu/949/tenClassification>

<http://data.europa.eu/949/concepts/ten-classifications/rinf/10>.

<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/5f544e78881a80d8cb988232b46c12b6d1b6514d>    <http://data.europa.eu/949/tenClassification>
      <http://data.europa.eu/949/concepts/ten-classifications/rinf/30>.

<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/5f544e78881a80d8cb988232b46c12b6d1b6514d>    <http://data.europa.eu/949/platformHeight>
      <http://data.europa.eu/949/concepts/platform-heights/rinf/40>.

http://data.europa.eu/949/functionalInfrastructure/platformsEdges/5f544e78881a80d8cb988232b46c12b6d1b6514d>    <http://data.europa.eu/949/length>
      "276.0"^^<http://www.w3.org/2001/XMLSchema#double>.

<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/5f544e78881a80d8cb988232b46c12b6d1b6514d>    <http://data.europa.eu/949/areaBoardingAid>
      "0"^^<http://www.w3.org/2001/XMLSchema#integer>.

<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/5f544e78881a80d8cb988232b46c12b6d1b6514d>    <http://data.europa.eu/949/assistanceStartingTrain>
      "false"^^<http://www.w3.org/2001/XMLSchema#boolean>.
```

The validity dates:

```
<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/5f544e78881a80d8cb988232b46c12b6d1b6514d >    <http://data.europa.eu/949/validityStartDate>      "2023-04-06"^^<http://www.w3.org/2001/XMLSchema#date>.

<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/5f544e78881a80d8cb988232b46c12b6d1b6514d>    <http://data.europa.eu/949/validityEndDate> "2024-12-31"^^<http://www.w3.org/2001/XMLSchema#date>.
```

Its canonical URI and hash source:

```
<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/5f544e78881a80d8cb988232b46c12b6d1b6514d>     <http://data.europa.eu/949/canonicalURI>
       <http://data.europa.eu/949/functionalInfrastructure/platformsEdges/NLDT_661_1Dtz>.

<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/5f544e78881a80d8cb988232b46c12b6d1b6514d>     http://data.europa.eu/949/hashSource        "NLDT/2023-04-06_2024-12-31/661/2023-04-06_2024-12-31/1 (Dtz)/2023-04-06_2024-12-31".
```

The User can then upload this set of triples as a Partial RDF dataset (delete) and execute the pipeline.

A.3.4 Insert an infrastructure element

This Use Case addresses the addition of any infrastructure element, e.g., OP, platform edge, tunnel, track. The User has to take into account the Vocabulary concepts, i.e. all the relationships (object properties), and attributes (datatype properties) where this infrastructure object "class" participates, i.e. where this class is part of the "Domain" or the "Range" of these properties. Note that the User must be careful to add all the relationships and attributes that are mandatory in order for the KG to remain consistent.

Also, these types of insertions may have a cascading effect where other related infrastructure elements must be added together with all of their properties.

Approach 1. Use the Export option in the Search Form

The user may export an instance of an infrastructure object that is of the same type of the object to be added and generate the set of triples as in the example described in the previous Section 0. Then this file can be used as a template to create the set of triples for the new infrastructure object.

Suppose the User wants to add a new platform edge to the same track than the previous example. The information is as follows:

- Name of OP: "Delft"
- Name of track: "661"
- Identification of the platform edge: "1 (Dtz) new"

First, a new Hash URI needs to be created for this platform edge. The input string for the platform edge has the following structure:

> {UOPID}/OPValidityDateStart}_{OPValidityDateEnd}/{OPTrackIdentification}/{Track_ValidityDateStart}_{Track_ValidityDateEnd}/{OPTrackPlatformIdentification}/{PlatfValidityDateStart}_{PlatfValidityDateEnd}

Thus, for this example the input string follows:

> "NLDT/2023-04-06_2024-12-31/661/2023-04-06_2024-12-31/1 (Dtz) new/2023-05-01_2025-12-31"

The `hashcode.py` script presented in Section A.3.1 is then executed:

The encrypted id is "`fea48fc003e909f094ca1d3889a1d3739b83e17c`"

Now the new set of triples can be created with this hash id and the corresponding properties.

The following set of triples contains information on the platform´s type (`PlatformEdge`), its label, platform Id and IM code:

```
<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/
fea48fc003e909f094ca1d3889a1d3739b83e17c>   <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
       <http://data.europa.eu/949/PlatformEdge>.

<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/
fea48fc003e909f094ca1d3889a1d3739b83e17c>   <http://www.w3.org/2000/01/rdf-schema#label>   "661,   1
(Dtz) new".

<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/
fea48fc003e909f094ca1d3889a1d3739b83e17c>   <http://data.europa.eu/949/platformId> "1 (Dtz) new".

<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/
fea48fc003e909f094ca1d3889a1d3739b83e17c>   <http://data.europa.eu/949/imCode>   "0084".
```

The track that the platform edge belongs to:

```
<http://data.europa.eu/949/functionalInfrastructure/tracks/74f489b424a5da0cc8f12e79b4b1bd56804f5e4
6>      <http://data.europa.eu/949/platformEdge>
       <http://data.europa.eu/949/functionalInfrastructure/platformsEdges/
fea48fc003e909f094ca1d3889a1d3739b83e17c>.
```

The parameter values:

```
<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/
fea48fc003e909f094ca1d3889a1d3739b83e17c>   <http://data.europa.eu/949/tenClassification>

<http://data.europa.eu/949/concepts/ten-classifications/rinf/10>.

<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/
fea48fc003e909f094ca1d3889a1d3739b83e17c>   <http://data.europa.eu/949/tenClassification>
       <http://data.europa.eu/949/concepts/ten-classifications/rinf/30>.

<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/
fea48fc003e909f094ca1d3889a1d3739b83e17c>   <http://data.europa.eu/949/platformHeight>
       <http://data.europa.eu/949/concepts/platform-heights/rinf/40>.

http://data.europa.eu/949/functionalInfrastructure/platformsEdges/
fea48fc003e909f094ca1d3889a1d3739b83e17c>   <http://data.europa.eu/949/length>
       "302.0"^^<http://www.w3.org/2001/XMLSchema#double>.

<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/
fea48fc003e909f094ca1d3889a1d3739b83e17c>   <http://data.europa.eu/949/areaBoardingAid>
       "0"^^<http://www.w3.org/2001/XMLSchema#integer>.

<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/
fea48fc003e909f094ca1d3889a1d3739b83e17c>   <http://data.europa.eu/949/assistanceStartingTrain>
       "true"^^<http://www.w3.org/2001/XMLSchema#boolean>.
```

The validity dates:

```
<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/
fea48fc003e909f094ca1d3889a1d3739b83e17c>   <http://data.europa.eu/949/validityStartDate> "2023-05-
01"^^<http://www.w3.org/2001/XMLSchema#date>.

<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/
fea48fc003e909f094ca1d3889a1d3739b83e17c>   <http://data.europa.eu/949/validityEndDate> "2025-12-
31"^^<http://www.w3.org/2001/XMLSchema#date>.
```

Its canonical URI and hash source:

```
<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/
fea48fc003e909f094ca1d3889a1d3739b83e17c>   <http://data.europa.eu/949/canonicalURI>
       <http://data.europa.eu/949/functionalInfrastructure/platformsEdges/NLDT_661_1Dtznew>.

<http://data.europa.eu/949/functionalInfrastructure/platformsEdges/
fea48fc003e909f094ca1d3889a1d3739b83e17c>   http://data.europa.eu/949/hashSource     "NLDT/2023-04-
06_2024-12-31/661/2023-04-06_2024-12-31/1 (Dtz) new/2023-05-01_2025-12-31".
```

The User can then upload this set of triples as a Partial RDF dataset (insert) file and execute the pipeline.

Approach 2. Use tools that allow the annotation of data

The User wants to insert a new OP and can use any of the tools described in Section A.4 to accomplish this task.

According to the Vocabulary, the relationships (object properties) where an operational point participates are the following:

- track
- opType
- siding
- localRulesOrRestrictionsDoc
- opEnd
- opStart

Note that the track and siding properties may imply adding new infrastructure elements and all of their properties need to be added also, i.e. a cascading insertion.

The opEnd and opStart properties will relate the OP to one or more existing or new SoLs. Again, new SoLs must be added together with all of their properties.

The opType property is a SKOS taxonomy and its URI is:

http://data.europa.eu/949/concepts/op-types/rinf/10

The localRulesOrRestrictionsDoc property points to an element that belongs to the class Document, and the instance and all of its properties must be added.

In all cases hash URIs must be generated for all objects using the code hashcode.py script that is included in Section A.3.1.

The datatype properties of an OP follow:

- uOPId
- opName
- tafTAPCode
- opTypeGaugeChangeover
- localRulesOrRestrictions

The mandatory properties must be added taking in consideration its datatype.

Once, the annotation is done, the instances in the tool may be exported in .nt format and uploaded in the system as a Partial RDF dataset (insert) file.

A.4 Tooling for data annotation

Several tools can facilitate the annotation of RINF data with the ERA Vocabulary:

1. Protégé: An ontology editor and knowledge acquisition system that supports ontology development and semantic annotations. The User can create or locate the Individuals and annotate them using the Property assertions view. Properties are selected from the ontology and values are provided for the properties that describe the individual.
2. TopBraid Composer: An ontology editor with features for ontology development, data integration, and semantic annotations. It works through projects where the ontology and data reside. The User navigates to the Instances View, locates the data instance, and then annotations can be added by selecting properties from the ontology and adding the values that describe the data instance.
3. PoolParty. A semantic technology platform that includes features for ontology management, text mining, and data annotation. The User opens the project where the ontology and data are stored. The Annotator tool is used to annotate data with concepts from the ontology. An interface is provided where ontology concepts are selected and values are added for the data.

Appendix B. Hash URIs and Canonical URIs

SoLs and OPs and their elements, tracks, platforms, sidings and tunnels may include validity dates in order to specify future plans. Each of these has their own set of parameters. Thus, a resource (URI) must be created for each although the element physically the same one.

These URIs are **Hash URIs** where the last portion of the URI is a Hash value. The input to the Hash function is the element's unique key. The Keys for each type of infrastructure element are described in the section "Hash keys" of this documentation. They have been generated using the function sha1.

There is also the **Canonical URI** for each element, that represents its basic URI without considering its validity dates. Each infrastructure element's resource has a pointer to its canonical URI.

B.1 Assumptions

1. Validity dates may indicate a period (validity start date and validity end date) or only the start date (validity start date).
2. There are elements that do not have validity date.
3. The infrastructure's topology considers only the element's canonical URIs.
4. The hierarchies of elements are as follow:
   - OperationalPoint
     - OPTrack
       - OPTrackTunnel
       - OPTrackPlatform
     - OPSiding
       - SidingTunnel
   - SectionOfLine
     - SOLTrack
       - SOLTunnel

B.2 URIs general structure

B.2.1 Canonical URI

http://data.europa.eu/949/functionalInfrastructure/<class-infrastructure-element>/<id-infrastructure-element>

Example: http://data.europa.eu/949/functionalInfrastructure/tracks/DE0HDSU_auf Anfrage_181996 where:

- <class-infrastructure-element> is "tracks"
- <id-infrastructure-element> is "DE0HDSU_auf Anfrage_181996" which concatenates the OP unique id and the Track's id

B.2.2 Hash URI

http://data.europa.eu/949/functionalInfrastructure/<class-infrastructure-element>/<hash-value>

Example: http://data.europa.eu/949/functionalInfrastructure/tracks/bcb6271b0bd5714058cb37179069777722058501 where:

- <class-infrastructure-element> is "tracks"
- <hash-value> is "bcb6271b0bd5714058cb37179069777722058501" which was generated using the OP id, OP type, the track id, and the validity dates

B.3 Canonical URIs

OperationalPoint

http://data.europa.eu/949/functionalInfrastructure/operationalPoints/{UOPID}

Border points URI:

http://data.europa.eu/949/functionalInfrastructure/operationalPoints/{country}/{EUXXXXX}

OPTrack

http://data.europa.eu/949/functionalInfrastructure/tracks/{UOPID}_{OPTrackIdentification}

OPTrackTunnel

http://data.europa.eu/949/functionalInfrastructure/tunnels/{UOPID}_{OPTrackIdentification}_{OPTrackTunnelIdentification}

OPTrackPlatform

http://data.europa.eu/949/functionalInfrastructure/platformsEdges/{UOPID}_{OPTrackIdentification}_{OPTrackPlatformIdentification}

OPSiding

http://data.europa.eu/949/functionalInfrastructure/sidings/{UOPID}_{OPSidingIdentification}

OPSidingTunnel

http://data.europa.eu/949/functionalInfrastructure/tunnels/{UOPID}_{OPSidingIdentification}_{OPSidingTunnelIdentification}

SectionOfLine

http://data.europa.eu/949/functionalInfrastructure/sectionsOfLine/{SOLLineIdentification}_{SOLOPStart}_{SOLOPEnd}

SOLTrack

http://data.europa.eu/949/functionalInfrastructure/tracks/{SOLLineIdentification}_{SOLOPStart}_{SOLTrackIdentification}_{SOLOPEnd}

SOLTunnel

http://data.europa.eu/949/functionalInfrastructure/tunnels/{SOLTunnelIdentification}_{SOLTunnelStartLongitude}_{SOLTunnelStartLatitude}_{SOLTunnelEndLongitude}_{SOLTunnelEndLatitude}

B.4 Hash URIs

In order to generate a Hash URI a hash key needs to be input to the hash function sha1. Hash keys should be formed as indicated in Section B.4.1.

A Python script that implements the sha1 function follows:

```
import hashlib
def sha1_encrypt(input_string):
    # Create a new SHA-1 hash object
    sha1_hash = hashlib.sha1()
    # Update the hash object with the input string
    sha1_hash.update(input_string.encode('utf-8'))
    # Get the hexadecimal representation of the digest
    encrypted_string = sha1_hash.hexdigest()
    return encrypted_string
input_string = <hash key>
encrypted_string = sha1_encrypt(input_string)
print("Original string:", input_string)
```

B.4.1 Hash keys

In general, for all the infrastructure elements there exist three cases for input keys: (1) With validity dates, (2) With no validity dates, (3) With only validity start date, (4) With only validity end date. We describe the keys for all these cases for OperationalPoint, but it applies equally to all other elements.

OperationalPoint

(1) URI with validity dates:

{country}/{UOPID}/{OPType}/{ValidityDateStart}_{ValidityDateEnd}

(2) URI with no validity dates:

{country}/{UOPID}/{OPType}/None_None

(3) URI with only validity start date

{country}/{UOPID}/{OPType}/{ValidityDateStart}_None

(4) URI with only validity start date

{country}/{UOPID}/{OPType}/None_{ValidityDateEnd}

Border points URI

{country}/{EUXXXXX}/{OPType}/{ValidityDateStart}_{ValidityDateEnd}

OPTrack

{UOPID}/{OPType}/{OPValidityDateStart}_{OPValidityDateEnd}/{OPTrackIdentification}/{Track_ValidityDateStart}_{Track_ValidityDateEnd}

OPTrackTunnel

{UOPID}/{OPType}/{OPValidityDateStart}_{OPValidityDateEnd}/{OPTrackIdentification}/{TrackValidityDateStart}_{TrackValidityDateEnd}/{OPTrackTunnelIdentification}/{TunnelValidityDateStart}_{TunnelValidityDateEnd}

OPTrackPlatform

{UOPID}/{OPType}/{OPValidityDateStart}_{OPValidityDateEnd}/{OPTrackIdentification}/{TrackValidityDateStart}_{TrackValidityDateEnd}/{OPTrackPlatformIdentification}/{PlatfValidityDateStart}_{PlatfValidityDateEnd}

OPSiding

{UOPID}/{OPType}/{OPValidityDateStart}_{OPValidityDateEnd}/{OPSidingIdentification}/{SidingValidityDateStart}_{SidingValidityDateEnd}

OPSidingTunnel

{UOPID}/{OPType}/{OPValidityDateStart}_{OPValidityDateEnd}/{OPSidingIdentification}/{SidingValidityDateStart}_{SidingValidityDateEnd}/{OPSidingTunnelIdentification}/{TunnelValidityDateStart}_{TunnelValidityDateEnd}

SectionOfLine

{SOLineIdentification}/{SOLOPStart}/{SOLOPEnd}/{ValidityDateStart}_{ValidityDateEnd}

SOLTrack

{SOLineIdentification}/{SOLOPStart}/{SOLOPEnd}/{SOLValidityDateStart}_{SOLValidityDateEnd}/{SOLTrackIdentification}/{TrackValidityDateStart}_{TrackValidityDateEnd}

SOLTrackTunnel

{SOLineIdentification}/{SOLValidityDateStart}_{SOLValidityDateEnd}/{SOLTrackIdentification}/{TrackValidityDateStart}_{TrackValidityDateEnd}/{SOLTunnelIdentification}/{SOLTunnelStartLongitude}_{SOLTunnelStartLatitude}_{SOLTunnelEnd_Longitude}_{SOLTunnelEnd_Latitude}