Package ‘FAIRmaterials’

June 27, 2024

Title Ontology Tools with Data FAIRification in Development

Version 0.4.2.1

Description Translates several CSV files with ontological terms and corresponding data into RDF triples. These RDF triples are stored in OWL and JSON-LD files, facilitating data accessibility, interoperability, and knowledge unification. The triples are also visualized in a graph saved as an SVG. The input CSVs must be formatted with a template from a public Google Sheet; see README or vignette for more information. This is a tool used by the SDLE Research Center at Case Western Reserve University to create and visualize material science ontologies, and it includes example ontologies to demonstrate its capabilities. This work was supported by the U.S. Department of Energy’s Office of Energy Efficiency and Renewable Energy (EERE) under Solar Energy Technologies Office (SETO) Agreement Numbers E-EE0009353 and DE-EE0009347, Department of Energy (National Nuclear Security Administration) under Award Number DE-NA0004104 and Contract number B647887, and U.S. National Science Foundation Award under Award Number 2133576.

License BSD_3_clause + file LICENSE

Encoding UTF-8

Imports dplyr, rdflib, jsonld, httr, utils, DiagrammeR, DiagrammeRsvg, stringr, xml2, tidyr, readr

RoxygenNote 7.3.1

Depends R (>= 3.1.1)

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Process Ontology CSV Files

Description
This function reads, processes, and combines ontology CSV files from a specified folder. These CSV files must be formatted with template from:

https://docs.google.com/spreadsheets/d/1k7gFRc1Ys1c-m65aWfFCxqk5UtrKZl9c3GyEFQvLSFU/edit?usp=drive_link

It can handle subdirectories by producing separate unmerged outputs for each subdirectory. The function generates an ontology tree plot, saves it as SVG, and serializes the RDF object to Turtle and JSON-LD formats in the specified folder. Additionally, it allows for the merging of multiple ontologies, with options to customize the merged RDF dataset’s metadata.

Usage

process_ontology_files(
  folder_path,
  add_external_onto_info = FALSE,
  include_graph_valuetype = TRUE,
  merge_base_uri = NULL,
  merge_title = NULL,
  merge_author = NULL,
  merge_version = NULL,
  merge_description = NULL
)

Arguments
folder_path Path to the folder containing ontology CSV files.
add_external_onto_info Logical indicating whether to update information in RDF.
include_graph_valuetype Logical indicating whether to include value types in the graph.
merge_base_uri  String containing base URI for the merged ontology. Defaults to NULL but is required for merging ontologies. Leave as default when only processing one ontology.

merge_title  String containing title for the merged ontology. Defaults to NULL will concatenate titles of all merged ontologies. Leave as default when only processing one ontology.

merge_author  One String containing authors for the merged ontology. Defaults to NULL will concatenate all unique Author names of merged ontologies. Leave as default when only processing one ontology.

merge_version  String containing version for the merged ontology. Defaults to NULL will leave version blank. Leave as default when only processing one ontology.

merge_description  String containing description for the merged ontology. Defaults to NULL will concatenate descriptions of all merged ontologies with relative titles as prefixes. Leave as default when only processing one ontology.

Details

This function processes ontology CSV files in a specified folder and its subdirectories. For each subdirectory, it creates separate unmerged outputs named with the folder name prefix. The merged output is generated in the main folder path.

If you have a single ontology, place all CSV files in the specified folder and run the function with default parameters. The function will generate outputs for this single ontology, including an ontology tree plot and serialized RDF formats.

For multiple ontologies across different subdirectories, the function will process each subdirectory separately. Each subdirectory's ontology will have its own set of outputs. Additionally, the function will merge the ontologies from all subdirectories. This merged output can have customized metadata such as title, authors, version, URL, and description. If no metadata is specified, the merging process concatenates titles, authors, and descriptions from all included ontologies. Note that specifying a base uri is required for merging ontologies.

Examples

# Create temporary directory
temp_dir <- tempdir()
XRay_test_folder <- file.path(temp_dir, "XRay")
dir.create(XRay_test_folder, recursive = TRUE)

# Copy CSV files from the package's extdata to the temporary directory
extdata_path <- system.file("extdata", "XRay", package = "FAIRmaterials")
file.copy(from = list.files(extdata_path, full.names = TRUE),
          to = XRay_test_folder, recursive = TRUE)

# Process the CSV files in temp the XRay folder
process_ontology_files(XRay_test_folder, add_external_onto_info = FALSE)

# Clean up
unlink(temp_dir, recursive = TRUE)
Index

process_ontology_files, 3