

Package ‘normanR’

September 12, 2026

Title Client for the 'Norman Network Database System' REST API

Version 1.8.79

Description Offers a programmatic interface to the 'NORMAN Network' REST API, facilitating the systematic query, retrieval, and parsing of environmental chemistry data, emerging substance registries, and mass spectrometry workflows. All data structures are tailored to support downstream environmental data analysis and ecotoxicological modelling.

License GPL-3

Encoding UTF-8

Depends R (>= 4.1.0)

Imports httr2, jsonlite

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

URL <https://github.com/urniaz/normanR>

BugReports <https://github.com/urniaz/normanR/issues>

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Rafal Urniaz [aut, cre] (ORCID:
<<https://orcid.org/0000-0003-0192-2165>>)

Maintainer Rafal Urniaz <rafal.urniaz@cantab.net>

Repository CRAN

Date/Publication 2026-09-12 15:00:02 UTC

Contents

extract_norman_fields	2
fetch_norman	3
get_norman_data	4

get_norman_empodat_batch_data	5
norman_api_definitions	6
split_by_key	7
split_by_pattern	8
Index	9

extract_norman_fields	<i>Extract Specific Fields from Nested Norman List</i>
-----------------------	--

Description

This function iterates through a list of nested data (e.g., from the Norman API) and extracts values for specific field names provided by the user. It uses a recursive search to find the fields, meaning the user does not need to know the full path (e.g., just "City" instead of "Data source\$Organisation\$City").

Usage

```
extract_norman_fields(data_list, field_names)
```

Arguments

- data_list A list of lists (the standard output from JSON parsing).
- field_names A character vector of field names to extract (e.g., c("E-mail", "City")).

Value

A data.frame where columns are the requested fields and rows correspond to the items in the input list. Missing values are filled with NA.

Examples

```
## Not run:
fields <- c("id", "Name", "City", "Value", "Unit")
df <- extract_norman_fields(data_cas_1$Data, fields)
print(df)

## End(Not run)
```

fetch_norman

*Fetch Multiple Datasets from Norman API***Description**

This function interacts with the Norman Network Database System (NDS) API. It allows the user to pass a vector of values (e.g., multiple CAS numbers or NSIDs). It iterates through each value, performs the API request, and aggregates the results into a named list.

Usage

```
fetch_norman(
  module,
  parameter,
  values,
  page = NULL,
  format = "json",
  verbose = TRUE
)
```

Arguments

- | | |
|-----------|--|
| module | A character string specifying the database module. Allowed values: <ul style="list-style-type: none"> • "susdat" - Substance Database • "ecotox" - Ecotoxicology Database • "empodat" - EMPODAT Database • "passive" - Passive Sampling Database |
| parameter | A character string specifying the search parameter. The allowed parameters depend on the selected module: <ul style="list-style-type: none"> • For module = "susdat": "nsid", "casrn", "inchikey" • For module = "ecotox": "nsid", "casrn", "inchikey" • For module = "empodat": "nsid", "casrn", "inchikey", "country", "matrix", "id" • For module = "passive": "nsid", "casrn", "inchikey", "country", "matrix" <p>Parameter Descriptions:</p> <ul style="list-style-type: none"> • nsid: Norman SusDat ID (e.g., "NS00001027" or "1027") • casrn: CAS Registry Number (e.g., "1490-04-6") • inchikey: International Chemical Identifier Key (e.g., "NOOLISFMXDJSKH-UHFFFAOYSA-N") • country: Country Alpha-2 code (e.g., "SK") • matrix: Ecosystem/matrix ID (e.g., "3") • id: Empodat ID or range (e.g., "100" or "100:150") |
| values | A character or numeric value corresponding to the chosen parameter. |

page	(Optional) Integer or character. The page number for pagination. If NULL (default), the page segment is omitted from the URL.
format	A character string specifying the output format. Allowed values: "json", "xml". Defaults to "json".
verbose	Logical. If TRUE, prints progress messages to the console. Defaults to TRUE.

Value

- If format = "json": A list or data frame (parsed JSON).
- If format = "xml": A raw character string (XML content).

A named list where each element's name corresponds to the search value and the content is the data returned by the API. If a request fails, the element will contain an error message or NULL.

Examples

```
## Not run:
# Define multiple CAS numbers
cas_list <- c("3380-34-5", "1490-04-6")

# Fetch data for all values
results <- fetch_norman(
  module = "susdat",
  parameter = "casrn",
  values = cas_list
)

# Access specific result
triclosan_data <- results[["3380-34-5"]]

## End(Not run)
```

get_norman_data

Retrieve Data from Norman Network REST API

Description

This function interacts with the Norman Network Database System (NDS) API.

Usage

```
get_norman_data(module, parameter, value, page = NULL, format = "json")
```

Arguments

module	A character string specifying the database module. Allowed values: • "susdat" - Substance Database • "ecotox" - Ecotoxicology Database • "empodat" - EMPODAT Database • "passive" - Passive Sampling Database
parameter	A character string specifying the search parameter. The allowed parameters depend on the selected module: • For module = "susdat": "nsid", "casrn", "inchikey" • For module = "ecotox": "nsid", "casrn", "inchikey" • For module = "empodat": "nsid", "casrn", "inchikey", "country", "matrix", "id" • For module = "passive": "nsid", "casrn", "inchikey", "country", "matrix" Parameter Descriptions: • nsid: Norman SusDat ID (e.g., "NS00001027" or "1027") • casrn: CAS Registry Number (e.g., "1490-04-6") • inchikey: International Chemical Identifier Key (e.g., "NOOLISFMXDJSKH-UHFFFAOYSA-N") • country: Country Alpha-2 code (e.g., "SK") • matrix: Ecosystem/matrix ID (e.g., "3") • id: Empodat ID or range (e.g., "100" or "100:150")
value	A character or numeric value corresponding to the chosen parameter.
page	(Optional) Integer or character. The page number for pagination. If NULL (default), the page segment is omitted from the URL.
format	A character string specifying the output format. Allowed values: "json", "xml". Defaults to "json".

Value

- If format = "json": A list or data frame (parsed JSON).
- If format = "xml": A raw character string (XML content).

get_norman_empodat_batch_data

Batch Retrieve and Paginate Norman Empodat Database

Description

This function extends get_norman_data by supporting multiple search values and automatic pagination for the 'empodat' module.

Usage

```
get_norman_empodat_batch_data(
  parameter,
  values,
  saveToDir = NULL,
  dropMemory = FALSE
)
```

Arguments

parameter	A character string (e.g., "nsid", "casrn").
values	A character vector of values to search for (e.g., c("3380-34-5", "50-00-0")).
saveToDir	= when NULL files are not saved, when directory the files are saved as "values.json" file name
dropMemory	when true, it clears memory buffer after every loop, data will not be accessible from memory (json\$Data is NULL)

Value

A consolidated list containing all records retrieved across all values and pages.

norman_api_definitions

Norman API Definitions (Dictionary)

Description

This function provides the norman API definitions. It assigns a list structure that acts as a dictionary, allowing for auto-completion (Intellisense) in RStudio when accessing modules and parameters.

Usage

```
norman_api_definitions(path = NULL)
```

Arguments

path	A character string specifying the path to the JSON file. If NULL, it attempts to locate 'api_definitions.json' within the package's 'extdata' directory.
------	--

Value

A named list (dictionary) containing two main sections:

- **modules:** A list where names are module names (e.g., susdat) and values are character vectors of allowed parameters.
- **parameters:** A list where names are parameter keys and values are their English descriptions.
- **patterns:** A collection of common regular expression patterns used for splitting or filtering data

Examples

```
## Not run:
# 1. Load the definitions
defs <- load_api_definitions()

# 2. Access available modules (RStudio will trigger auto-complete after $)
# defs$modules$Substance

# 3. Access parameter descriptions
# defs$parameters$`Country Alpha-2 code`

## End(Not run)
```

split_by_key

*Split JSON Files Based on Unique Key Values***Description**

This function scans a source directory for JSON files, loads them, and splits each file into multiple smaller JSON files based on the unique values of a specified key. Each unique value results in a separate file.

Usage

```
split_by_key(input_dir, output_dir, split_key, data_path = NULL)
```

Arguments

input_dir	Character. Path to the directory containing source JSON files.
output_dir	Character. Path to the directory where split files will be saved.
split_key	Character. The name of the key/field used to isolate unique values (e.g., "Name of country" or "Sample matrix").
data_path	Character. If the JSON has a nested structure (like Norman API), provide the name of the element containing the list of records (e.g., "Data"). Defaults to NULL (assumes the JSON is a top-level list).

Details

The function creates the output directory if it does not exist. Filenames are generated using the pattern: original_name.json.

Value

Logical. Returns TRUE if the operation completes successfully.

split_by_pattern

Split JSON Files Based on Partial Key Values (Pattern Matching)

Description

This function scans a source directory for JSON files, loads them, and splits each file into multiple smaller JSON files based on the partial key values (pattern matching) of a specified key. Each unique value results in a separate file.

Usage

```
split_by_pattern(  
  input_dir,  
  output_dir,  
  split_key,  
  pattern = NULL,  
  data_path = NULL  
)
```

Arguments

input_dir	Character. Path to the folder containing source JSON files.
output_dir	Character. Path to the folder where output files will be saved.
split_key	Character. The JSON key to inspect (e.g., "Sampling date").
pattern	Character (Regex). A pattern to extract. If NULL, splits by full value.
data_path	Character. The element in the JSON containing the records list.

Details

The function creates the output directory if it does not exist. Filenames are generated using the pattern: original_name.json.

Value

Logical. Returns TRUE if the operation completes successfully.

Index

`extract_norman_fields`, [2](#)

`fetch_norman`, [3](#)

`get_norman_data`, [4](#)

`get_norman_empodat_batch_data`, [5](#)

`norman_api_definitions`, [6](#)

`split_by_key`, [7](#)

`split_by_pattern`, [8](#)