Package 'umweltapir'

April 21, 2026

Title Access Umwelt.Info API

Version 0.1.0

Encoding UTF-8

Description Provides an R-based access to the datasets including their resources from the portal <https://umwelt.info>. The package allows for an easy integration of those datasets into your R-based workflows. The functionality of the package mirrors the web-based access as provided at <https://umwelt.info>. You can use the same queries and get the same datasets by accessing our API.

URL https://gitlab.opencode.de/umwelt-info/packages/-/tree/main/umweltapir?ref_type=heads

BugReports https://gitlab.opencode.de/umwelt-info/packages/-/work_items

License MIT + file LICENSE | Apache License 2.0

RoxygenNote 7.3.3

Imports httr2, jsonlite, tidyr

Suggests testthat (>= 3.0.0), dplyr

Config/testthat/edition 3

NeedsCompilation no

Author Johannes Vogel [aut, cre],
Maximilian Berthold [aut],
Luise Quoß [ctb],
Nationales Zentrum für Umwelt- und Naturschutzinformationen [cph]

Maintainer Johannes Vogel <johannes.vogel@uba.de>

Depends R (>= 4.1.0)

Repository CRAN

Date/Publication 2026-04-21 20:20:02 UTC

Contents

`download_resources` *Download resources from umwelt.info*

Description

Download all the resources attached to the datasets of a respective query:

Usage

```
unnest_and_filter(
  data,
  formats = c("CSV", "ZIP", "JSON", "JSON-LD", "GeoJSON", "TSV", "PDF",
    "Microsoft Excel Spreadsheet"),
  description_regex = NULL
)

preview_resources(data_preprocessed)

download_resources(data_preprocessed, base_dir = tempdir())
```

Arguments

`data`	An unnested and optionally filtered R dataframe which is written as output by one of the functions fetch_by_url, fetch_by_query or fetch_by_id
`formats`	A list of strings indicating accepted output formats of the recoures Possible values: run 'fetch_facet_values("resource_type")' to get a list of existing formats
`description_regex`	A string to filter only resources with a description containing the string
`data_preprocessed`	An unnested and optionally filtered R dataframe which is written as output by unnest_and_filter
`base_dir`	A directory where downloaded resources should be stored.

Value

No return value (resources are downloaded into the respective folder)

Examples

```
# To download the resources the workflow contains four steps. First you fetch the list of all
# datasets belonging to your query.
# The input link for the query can be generated in the interface of https://umwelt.info.
# See the tutorial
# https://umwelt.info/artikel/so-laden-sie-daten-bei-umweltinfo-mit-python-und-r-herunter
# for further details.
# In a second step the required columns are unnested and you can optionally filter for certain
```

```
# file formats (in the example here "CSV" and "ZIP") and create a subset of only those entries
# where the resource description contains the query (in this example "Ozon").
# Note that the unnesting is a prerequisite for preview_resources() and download_resources().
# Third, you create a preview of the resulting resources which would be downloaded.
# If you want to proceed, you can initiate the download in the fourth and final step.
if (interactive()) {
  url <-
 "https://md.umwelt.info/search/all?query=(Ozon)+AND+organisation%3A%2FLand%2FBayern%2Fopen.bydata"
  results <- fetch_by_url(url,
    columns = "resource_only"
  ) |>
   unnest_and_filter(formats =  c("Microsoft Excel Spreadsheet"), description_regex = "Ozon") |>
    preview_resources()
  results |> download_resources(base_dir = tempdir())
}
```

`fetch_data` *Fetch data from umwelt.info*

Description

These functions allow you to retrieve datasets from the umwelt.info metadata search API either by providing a search query or a complete URL.

Usage

```
fetch_by_query(query, language = "de", columns = NULL)

fetch_by_url(url, columns = NULL)

fetch_by_ids(ids)

fetch_facet_values(name = "type")
```

Arguments

`query`	A character vector containing the search query (e.g., "Ozon").
`language`	A string to determine the language of the search results. Possible values: de and en. Default: de (German).
`columns`	Either a vector of strings containing the selected columns or "resource_only" as a shortcut to select the columns "source", "id", "resources", "title" and "quality"
`url`	A character string containing the full API request URL.
`ids`	A list of character strings containing the ID(s) of datasets.
`name`	A character string containing the name of the facet for which all possible values will be returned (list). These can be used to create a new query. Possible values: type, topic, organisation, license, language and resource_type. Default value: type.

Value

A dataframe containing the dataset entries. Returns an empty dataframe if no results are found.

Examples

```
# Example 1: Fetching by a direct URL
if (interactive()) {
  api_url <- "https://md.umwelt.info/search/all?query=Luftqualität"
  result_list <- fetch_by_url(api_url)
}

# Example 2: Fetching by query string
# For background how to build a query see https://md.umwelt.info/swagger-ui/#/search/text_search
# If you want to know which facet values exist for a certain facet, you can use
# fetch_facet_values (see example 5).
if (interactive()) {
  result_list <- fetch_by_query("(Ozon) AND organisation:/Land/Bayern/open.bydata")
}

# Example 3: Select subset of columns and unnest columns (here the column "resources" is unnested
# into its subcolums "type", "url", "description", "direct_link" and "primary_content") and in a
# second step "type" is further unnested into "path" and "label"
if (interactive()) {
  result_list <- fetch_by_query("(Ozon) AND organisation:/Land/Bayern/open.bydata")
  colnames(result_list) # columns before unnesting
  result_list <- result_list |>
    tidyr::unnest(col = c("source")) |>
    (\(df) df[, c("source", "id", "resources", "title", "quality"), drop = FALSE])() |>
    tidyr::unnest(col = c("resources")) |>
    tidyr::unnest(col = c("type"))
  colnames(result_list) # columns after unnesting)
}

# Example 4: Fetching by a list of dataset IDs. This can e.g. be useful for downloading resources.
# After using preview_resources you can select a subset of from the preview list using
# fetch_by_ids() and forward it as input to download_resources().
ids <- c(
  "uvk-be/-sen-uvk-umwelt-luft-luftqualitaet-",
  "lanuk-nrw/-publikationen-publikation-bericht-ueber-die-luftqualitaet-im-jahre-2014",
  "metaver-hb/7F0A29F5-ECBC-476D-9C99-DC1A6A8043D0"
)
datasets <- fetch_by_ids(ids)
for (dataset in datasets) {
  print(dataset$title)
}

# Example 5: Fetching all possible values for the facet name organisation. This can be e.g. useful
# if you want to restrict the results to certain organisations when build your own query,
# so you know which organisations are available.
name <- "organisation"
organisations <- fetch_facet_values(name)
head(organisations)
```

```
# Example 7: Fetch multiple facets at the same time
# If you want to fetch more than one facet, the easiest way is to use fetch_by_query() for this.
if (interactive()) {result_list  <- fetch_by_query(
 query =
 "organisation:/Bund/Destatis OR organisation:'/Land/Statistische Ämter des Bundes und der Länder'")
}
```

Index