

# Package ‘nadaverse’

December 11, 2025

**Type** Package

**Title** Browse Microdata Catalogs Using 'NADA' REST API

**Version** 0.1.0

**Date** 2025-11-18

**Description** Provides a unified, programmatic interface for searching, browsing, and retrieving metadata from various international organization data repositories that use the National Data Archive ('NADA') software, such as the World Bank, 'FAO', and the International Household Survey Network ('IHSN'). Functions allow users to discover available data collections, country codes, and access types, perform complex searches using keyword and spatial/temporal filters, and retrieve detailed study information, including file lists and variable-level data dictionaries. It simplifies access to microdata for researchers and policy analysts globally.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Imports** cli, httr2

**RoxygenNote** 7.3.2

**URL** <https://github.com/guturago/nadaverse>

**BugReports** <https://github.com/guturago/nadaverse/issues>

**Suggests** httpptest2, testthat (>= 3.0.0)

**Config/testthat.edition** 3

**NeedsCompilation** no

**Author** Gutama Girja Urago [aut, cre, cph] (ORCID:  
[<https://orcid.org/0000-0001-5588-2301>](https://orcid.org/0000-0001-5588-2301))

**Maintainer** Gutama Girja Urago <[girjagutama@gmail.com](mailto:girjagutama@gmail.com)>

**Repository** CRAN

**Date/Publication** 2025-12-11 13:50:07 UTC

## Contents

catalogs . . . . .	2
data_files . . . . .	4
search_catalog . . . . .	6
<b>Index</b>	<b>9</b>

---

### catalogs

*Small Helper Functions for Data Catalog Access*

---

#### Description

A suite of small **helper functions** designed to interact with and retrieve essential metadata from various international organization data repositories (catalogs). These functions standardize the process of obtaining lists of available **data access codes**, **collections**, **country codes**, and **latest entries** from specified sources.

#### Usage

```
catalogs(show = TRUE)

access_codes(catalog)

collections(catalog)

country_codes(catalog)

latest_entries(catalog, limit = NULL)

metadata(catalog, id)
```

#### Arguments

show	Logical. If TRUE, prints supported catalogs to console
catalog	A required character string specifying the name of the data catalog (e.g., "fao", "ilo", "wb") from which to retrieve metadata.
limit	A positive integer number, applicable only to latest_entries, indicating the <b>maximum number of results</b> (data entries) to return. If NULL or omitted, a default limit set by the API will be used.
id	A required <b>study identifier</b> . Accepts either the numeric <b>Study ID</b> (integer, e.g., 101) or the character <b>Study ID Number</b> (string, e.g., "ALB_2012_LSMS_v01_M_v01_A_PUF"). These values are typically returned in the search results from <a href="#">search_catalog()</a> , <a href="#">latest_entries()</a> or <a href="#">data_files()</a> .

## Details

All functions require a valid catalog name. The functions communicate with a backend API (implied by `base_url` and `get_response`) to fetch the requested data in a standardized format. The validity of the catalog is checked internally using `assert_catalog`.

## Value

A **data frame** containing the requested metadata, except for `metadata()`, which returns a **list**. The structure of the returned object varies by function:

- `access_codes`: Returns a data frame with columns related to data resource identifiers (e.g., `code`, `description`).
- `collections`: Returns a data frame detailing data groupings (e.g., `collection_id`, `name`).
- `country_codes`: Returns a data frame of standard country identifiers (e.g., `iso3c`, `country_name`).
- `latest_entries`: Returns a data frame of the **most recently** added datasets or entries, with columns reflecting their general metadata (e.g., `title`, `date_added`).
- `metadata`: Returns a **list** of the study metadata including detailed description, abstract, sampling methodology, and other study-specific details.

If the API call fails or no data is found, the function may return an empty data frame or raise an error.

## Supported Catalogs

The catalog argument must be one of the following short codes (case-insensitive) corresponding to the respective microdata repository. **The list is sorted alphabetically by code.**

- "df": **Data First** (<https://www.datafirst.uct.ac.za>)
- "erf": **Economic Research Forum** (<https://erfdataportal.com>)
- "fao": **Food and Agriculture Organization** (<https://microdata.fao.org>)
- "ihsn": **International Household Survey Network** (<https://catalog.ihsn.org>)
- "ilo": **International Labour Organization** (<https://webapps.ilo.org/surveyLib>)
- "india": **Government of India** (<https://microdata.gov.in>)
- "unhcr": **United Nations High Commissioner for Refugees** (<https://microdata.unhcr.org>)
- "wb": **The World Bank** (<https://microdata.worldbank.org>)

## Author(s)

Gutama Girja Urago

## See Also

The main search function: [search\\_catalog](#)

## Examples

```
## Not run:
# --- Examples for Supported Catalogs ---

# 1. Data First (df): Get available access codes.
df_codes <- access_codes("df")

# 2. Economic Research Forum (erf): Get latest data entries (limited to 5).
erf_latest <- latest_entries("erf", limit = 5)

# 3. Food and Agriculture Organization (fao): Get available collections.
fao_collections <- collections("fao")

# 4. International Household Survey Network (ihsn): Get supported country codes.
ihsn_countries <- country_codes("ihsn")

# 5. International Labour Organization (ilo): Get available access codes.
ilo_codes <- access_codes("ilo")

# 6. Government of India (india): Get latest data entries (limited to 10).
india_latest <- latest_entries("india", limit = 10)

# 7. United Nations High Commissioner for Refugees (unhcr): Get available collections.
unhcr_collections <- collections("unhcr")

# 8. The World Bank (wb): Get supported country codes.
wb_countries <- country_codes("wb")

# Example for the metadata function (requires a study ID)
wb_study_metadata <- metadata("wb", id = 8098)
str(wb_study_metadata)

## End(Not run)
```

## Description

Retrieves information about the files included in a study, or the detailed **data dictionary** (variables) for the entire study or a specific data file.

## Usage

```
data_files(catalog, id)

data_dictionary(catalog, id, file_id = NULL)
```

## Arguments

catalog	A required character string specifying the name of the data catalog (e.g., "wb", "fao"). Valid codes can be found in the documentation for <code>catalogs()</code> .
id	A required study identifier. Accepts either the numeric <b>Study ID</b> (integer, e.g., 101) or the character <b>Study ID Number</b> (string, e.g., "ALB_2012_LSMS_v01_M_v01_A_PUF"). These values are typically returned in the search results from <code>search_catalog()</code> , <code>latest_entries()</code> or <code>data_files()</code> .
file_id	An optional character identifier, applicable only to <code>data_dictionary()</code> . This is the ID of a specific data file within the study, typically found in the <code>file_id</code> column returned by <code>data_files()</code> . If NULL (default), <code>data_dictionary()</code> attempts to fetch variables for the entire study.

## Details

`data_files()` returns the list of files available for a study, along with metadata like file name, size, and ID.

`data_dictionary()` retrieves the variable-level metadata, including variable names, labels, and definitions. If `file_id` is provided, it retrieves the dictionary for that specific file; otherwise, it attempts to fetch the dictionary for the entire study. The function automatically detects whether the provided study identifier (`id`) is numeric or character.

## Value

The return value depends on the function called:

- `data_files()`: A **data frame** detailing the files associated with the study. Typical columns include `file_name`, `dfile_id`, `file_type`, and `file_size`.
- `data_dictionary()`: A **data frame** containing the variable-level metadata (the data dictionary). Typical columns include `name`, `label`, and `var_id`.

If the API returns no files or variables, a warning message is issued.

## Author(s)

Gutama Girja Urago

## See Also

[search\\_catalog](#), [latest\\_entries](#)

## Examples

```
## Not run:  
# Example 1: Get the list of files for a World Bank study (using idno)  
study_idno <- "ALB_2012_LSMS_v01_M_v01_A_PUF"  
files_wb <- data_files(catalog = "wb", id = study_idno)  
print(files_wb)  
  
# Example 2: Get the data dictionary for the entire study (using idno)
```

```

dictionary_all <- data_dictionary(catalog = "wb", id = study_idno)
head(dictionary_all)

# Example 3: Get the data dictionary for a specific file
# First, retrieve the files to find a file_id (dfile_id)
file_id_to_use <- files_wb$file_id[1] # Use the ID of the first file
dictionary_file <- data_dictionary(
  catalog = "wb",
  id = study_idno,
  file_id = file_id_to_use
)
head(dictionary_file)

## End(Not run)

```

**search\_catalog***Search Catalogs***Description**

Performs a comprehensive search in the specified catalog's API endpoint, utilizing a full range of available searching, filtering, and sorting parameters.

**Usage**

```

search_catalog(
  catalog,
  keyword = NULL,
  from = NULL,
  to = NULL,
  country = NULL,
  inc_iso = NULL,
  collection = NULL,
  created = NULL,
  dtype = NULL,
  sort_by = NULL,
  sort_order = NULL,
  ps = NULL,
  page = NULL,
  rows = TRUE
)

```

**Arguments**

<b>catalog</b>	A required character string specifying the name of the data catalog (e.g., "fao", "wb"). Valid codes can be found in the documentation for <code>access_codes()</code> .
<b>keyword</b>	A character string used to search data titles, descriptions, and keywords (e.g., "lsms").

<code>from</code>	An integer indicating the <b>start year</b> for the data collection's coverage period (e.g., 2000).
<code>to</code>	An integer indicating the <b>end year</b> for the data collection's coverage period (e.g., 2010).
<code>country</code>	A character vector. Provide one or more <b>country names</b> or <b>ISO 3 codes</b> (case-insensitive). For valid codes, see <code>country_codes()</code> . Multiple values should be passed as a vector, e.g., <code>c("afg", "Indonesia", "bra")</code> .
<code>inc_iso</code>	A logical value. If TRUE, the results data frame will include the ISO3 country codes; otherwise, it will contain only country names. <b>Default:</b> NULL.
<code>collection</code>	A character vector. Filters results by the data collection repository ID, which is returned in the <code>repo_id</code> column by <code>collections()</code> . Multiple IDs can be searched by passing a vector.
<code>created</code>	A character string used to filter results by the <b>date of creation</b> or update within the catalog. Use the date format YYYY-MM-DD. <ul style="list-style-type: none"> <li>• Single date: "2020/04/01" (returns records created on or after this date).</li> <li>• Date range: "2020/04/01-2020/04/20" (returns records within the range).</li> </ul>
<code>dtype</code>	A character vector. Filters results by one or more data access types. Valid values include: "open", "direct", "public", "licensed", "enclave", "remote", and "other". See <code>access_codes()</code> for a list of available types by catalog. Example: <code>c("open", "licensed")</code> .
<code>sort_by</code>	A character string used to specify the column by which to sort the results. Valid values are: "rank", "title", "nation" (for country), or "year". Note that "country" is automatically mapped to the API field "nation".
<code>sort_order</code>	A character string indicating the sort direction. Must be either "asc" (ascending) or "desc" (descending).
<code>ps</code>	An integer indicating the number of records to display <b>per page</b> of results. <b>Default:</b> 15 records.
<code>page</code>	An integer specifying the <b>page number</b> of the search results to return.
<code>rows</code>	A logical value. If TRUE, the function returns only a data frame containing the list of returned studies; otherwise, a list containing detailed search metadata (e.g., total records found, total pages) instead of the data records themselves. <b>Default:</b> TRUE.

## Details

This function constructs a complex API query based on the provided arguments (such as keywords, temporal range, geography, and access types) and returns the matching data entries. The function automatically handles URL encoding and JSON parsing.

All parameters correspond directly to the search options available on the NADA (National Data Archive) platform used by organizations like the World Bank and FAO.

## Value

If `rows` = TRUE (default), returns a **data frame** where each row is a data entry matching the search criteria. If `rows` = FALSE, returns a **list** containing search metadata, including the total number of records found and the search parameters used.

**Author(s)**

Gutama Girja Urago

**See Also**

[access\\_codes](#), [collections](#), [country\\_codes](#), [latest\\_entries](#)

**Examples**

```
## Not run:  
# Example 1: Basic search for a keyword in the World Bank catalog  
wb_search <- search_catalog(  
  catalog = "wb",  
  keyword = "LSMS",  
  ps = 5, # 5 records per page  
  page = 1  
)  
head(wb_search)  
  
# Example 2: Search by country and year range  
fao_search <- search_catalog(  
  catalog = "fao",  
  country = c("Kenya", "UGA"),  
  from = 2010,  
  to = 2020,  
  sort_by = "year",  
  sort_order = "desc"  
)  
  
# Example 3: Filter by access type and get search information  
ilo_info <- search_catalog(  
  catalog = "ilo",  
  keyword = "labor",  
  dtype = "public",  
  rows = FALSE  
)  
print(ilo_info$found) # Check total number of records found  
  
# Example 4: Include ISO codes in results  
ihsn_results <- search_catalog(  
  catalog = "ihsn",  
  inc_iso = TRUE  
)  
head(ihsn_results)  
  
## End(Not run)
```

# Index

access\_codes, 8  
access\_codes (catalogs), 2  
  
catalogs, 2  
collections, 8  
collections (catalogs), 2  
country\_codes, 8  
country\_codes (catalogs), 2  
  
data\_dictionary (data\_files), 4  
data\_files, 4  
data\_files(), 2  
  
latest\_entries, 5, 8  
latest\_entries (catalogs), 2  
latest\_entries(), 2  
  
metadata (catalogs), 2  
  
search\_catalog, 3, 5, 6  
search\_catalog(), 2