

Package: finna (via r-universe)

January 30, 2025

Title Access the 'Finna' API

Version 0.1.2

Date 2025-01-10

Maintainer Akewak Jeba <akjeba@utu.fi>

Description Provides functions to access and retrieve metadata from the 'Finna' API <<https://api.finna.fi/>>, which aggregates content from Finnish archives, libraries, and museums.

License BSD_2_clause + file LICENSE

Encoding UTF-8

LazyData true

Imports dplyr, tidyr, reshape2, glue, httr, xml2, jsonlite, ggplot2, readr, tibble, curl, progress, purrr

Suggests testthat (>= 3.0.0), rmarkdown, knitr

URL <https://ropengov.github.io/finna/>,
<https://CRAN.R-project.org/package=finna>

BugReports <https://github.com/rOpenGov/finna/issues>

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Config/testthat/edition 3

VignetteBuilder knitr

Depends R (>= 2.10)

Config/pak/sysreqs libicu-dev libxml2-dev libssl-dev libx11-dev

Repository <https://ropengov.r-universe.dev>

RemoteUrl <https://github.com/rOpenGov/finna>

RemoteRef HEAD

RemoteSha 587d7fcf59cbf1afff0cc918726dac283979fc43

Contents

analyze_metadata	2
analyze_trends_over_time	3
check_api_access	4
enrich_author_name	4
fetch_all_records	5
fetch_finna	6
fetch_viola_records	7
finna_cite	8
finna_interactive	8
full_fennica_records	9
get_finna_records	10
harvest_oai_pmh	11
load_offline_data	12
refine_metadata	13
save_for_offline	14
search_finna	15
search_publisher	16
timeline	17
top_plot	18

Index	20
--------------	-----------

analyze_metadata	<i>Analyze Refined Finna Metadata</i>
------------------	---------------------------------------

Description

Performs basic analysis on Finna metadata, summarizing the distribution of formats, years, and authors.

Usage

```
analyze_metadata(metadata)
```

Arguments

metadata A tibble containing refined Finna metadata.

Value

A list of tibbles with summaries of formats, years, and authors.

Examples

```
library(finna)
sibelius_data <- search_finna("sibelius")
refined_data <- refine_metadata(sibelius_data)
analyze_metadata(refined_data)
```

analyze_trends_over_time

Analyze Trends Over Time with Binned Years (Decades)

Description

This function analyzes how search results for a given query have trended over time, binned by decades. It plots the number of records found for each decade, allowing users to observe long-term trends.

Usage

```
analyze_trends_over_time(data, query = "Records Over Time")
```

Arguments

data	A tibble containing Finna search results with a Year column (as character or numeric).
query	A search query string (optional) to label the plot.

Value

A ggplot2 plot showing the trend of records over time.

Examples

```
finna_data <- search_finna("Sibelius")
trends <- analyze_trends_over_time(finna_data, "Sibelius")
print(trends)
```

check_api_access *Check Access to the Finna API*

Description

This function tests whether R can successfully connect to the Finna API by downloading the OpenAPI specification from <https://api.finna.fi/api/v1/?openapi>. It returns a logical value indicating the accessibility of the API.

Usage

```
check_api_access()
```

Value

A logical value:

- TRUE: The API is accessible.
- FALSE: The API is not accessible.

Examples

```
## Not run:  
# Check if the API is accessible  
access <- check_api_access()  
if (access) {  
  message("Finna API is accessible")  
} else {  
  message("Finna API is not accessible")  
}  
  
## End(Not run)
```

enrich_author_name *Enrich Author Name from 'Finna' API and Save Results*

Description

This function reads a CSV file from a URL containing Melinda IDs and author names. If the author name is missing (NA), it searches the 'Finna' API for the corresponding Melinda ID to retrieve and update the author name. The updated data is saved in a CSV file.

Usage

```
enrich_author_name(url, output_file = "updated_na_author_rows.csv")
```

Arguments

url	A character string specifying the URL of the CSV file with Melinda IDs and author names.
output_file	A character string specifying the output CSV file name.

Value

A tibble with updated author names. The file is saved to a temporary directory using `tempdir()`.

Examples

```
## Not run:
enrich_author_name(url = "https://example/na_author_rows.csv",
                  output_file = "updated_na_author_rows.csv")

## End(Not run)
```

fetch_all_records	<i>Fetch All Records from Finna API</i>
-------------------	---

Description

This function fetches records from the Finna API in chunks of 100,000, automatically paginating through the results until the maximum number of records is reached.

Usage

```
fetch_all_records(
  base_query = "*",
  base_filters = c("collection:\\"FEN\\""),
  sort = "main_date_str asc",
  limit_per_query = 1e+05,
  total_limit = Inf
)
```

Arguments

base_query	A string specifying the base query. Defaults to "*".
base_filters	A character vector of filters to apply to the query. Defaults to <code>c('collection:"FEN"')</code> .
sort	A string defining the sort order of the results. Default is "main_date_str asc".
limit_per_query	An integer specifying the number of records to fetch per query. Defaults to 100000.
total_limit	An integer specifying the maximum number of records to fetch. Defaults to Inf.

Value

A tibble containing all fetched records.

Examples

```
## Not run:
results <- fetch_all_records(
  base_query = "*",
  base_filters = c('collection:"FEN"'),
  sort = "main_date_str asc",
  limit_per_query = 100000,
  total_limit = Inf
)
print(results)

## End(Not run)
```

 fetch_finna

Fetch Finna Collection Data with Flexible Query

Description

This function retrieves data from the Finna API and formats it as a tidy tibble.

Usage

```
fetch_finna(
  query = NULL,
  limit = 0,
  facets = "building",
  lng = "fi",
  prettyPrint = TRUE
)
```

Arguments

query	The query string for filtering results. Defaults to NULL, which fetches data without a specific search term.
limit	Maximum number of results to fetch. Defaults to 0.
facets	Facet to retrieve, defaults to "building".
lng	Language for results, defaults to "fi".
prettyPrint	Logical, whether to pretty-print JSON responses.

Value

A tibble containing the fetched data with relevant fields.

Examples

```
## Not run:
  fetch_finna(query = "record_format:ead", limit = 0)
  fetch_finna() # Fetches data with no specific query

## End(Not run)
```

fetch_viola_records *Fetch Records by Year Ranges from Finna API (Including NA Dates)*

Description

This function fetches records from the Finna API in chunks divided by year ranges, handling missing date values.

Usage

```
fetch_viola_records(
  base_query = "*",
  base_filters = c("collection:\\"VIO\\""),
  year_ranges = list(c(0, as.numeric(format(Sys.Date(), "%Y")))),
  include_na = TRUE,
  limit_per_query = 1e+05,
  total_limit = Inf,
  delay_after_query = 5
)
```

Arguments

base_query	The base query string, defaults to "*".
base_filters	A character vector of filters for the search, e.g., c('collection:"VIO"').
year_ranges	A list of numeric vectors specifying year ranges, e.g., list(c(2000, 2005), c(2006, 2010)).
include_na	Whether to include records with missing main_date_str. Default is TRUE.
limit_per_query	Maximum number of records to fetch per query. Default is 100000.
total_limit	Maximum number of records to fetch overall. Default is Inf.
delay_after_query	Delay in seconds between queries. Default is 5.

Value

A tibble containing all fetched records.

finna_cite *Cite a Finna collection*

Description

Automatically generates a citation for a Finna collection result.

Usage

```
finna_cite(result, index, style = "citation")
```

Arguments

result	The Finna collection result as a tibble.
index	The index of the collection to cite (numeric).
style	The citation style to use (default: "citation"). See bibentry .

Value

A bibliographic entry (bibentry) printed in the specified style.

finna_interactive *Interactive Finna Search and Data Download*

Description

Provides an interactive interface to search, select, and download datasets from Finna API.

Usage

```
finna_interactive()
```

Value

A dataframe containing the selected dataset or downloaded data.

See Also

[search_finna\(\)](#), [fetch_finna\(\)](#), [finna_cite\(\)](#)

full_fennica_records *Fetch Fennica Collection Records by Year Ranges from Finna API (Including NA Dates)*

Description

This function fetches records from the Finna API in chunks divided by year ranges, handling missing date values for the Fennica collection.

Usage

```
full_fennica_records(  
  base_query = "*",  
  base_filters = c("collection:\"FEN\""),  
  year_ranges = list(c(0, as.numeric(format(Sys.Date(), "%Y")))),  
  include_na = TRUE,  
  limit_per_query = 1e+05,  
  total_limit = Inf,  
  delay_after_query = 5  
)
```

Arguments

base_query	The base query string, defaults to "*".
base_filters	A character vector of filters for the search, e.g., c('collection:"FEN"').
year_ranges	A list of numeric vectors specifying year ranges, e.g., list(c(2000, 2005), c(2006, 2010)).
include_na	Whether to include records with missing main_date_str. Default is TRUE.
limit_per_query	Maximum number of records to fetch per query. Default is 100000.
total_limit	Maximum number of records to fetch overall. Default is Inf.
delay_after_query	Delay in seconds between queries. Default is 5.

Value

A tibble containing all fetched records.

get_finna_records *Get Finna Records by IDs with Extended Options*

Description

This function retrieves multiple Finna records based on a vector of record IDs. You can specify which fields to return, the language, and the pagination options.

Usage

```
get_finna_records(  
  ids,  
  field = NULL,  
  prettyPrint = FALSE,  
  lng = "fi",  
  page = 1,  
  limit = 100  
)
```

Arguments

ids	A vector of record IDs to retrieve.
field	A vector of fields to return. Defaults to NULL, which returns all default fields.
prettyPrint	Logical; whether to pretty-print the response. Defaults to FALSE.
lng	Language for returned translated strings. Defaults to "fi".
page	The page number to retrieve. Defaults to 1.
limit	The number of records to return per page. Defaults to 20.

Value

A tibble containing the retrieved records data with provenance information.

Examples

```
records <- get_finna_records("fikka.3405646", field = "title", prettyPrint = TRUE, lng = "en-gb")  
print(records)
```

harvest_oai_pmh	<i>Harvest Metadata from an OAI-PMH Server</i>
-----------------	--

Description

This function harvests metadata records from an OAI-PMH-compliant server in batches, using a custom User-Agent string to identify the service and returns them in a tibble format.

Usage

```
harvest_oai_pmh(  
  base_url,  
  metadata_prefix,  
  set = NULL,  
  verbose = TRUE,  
  user_agent = "FinnaHarvester/1.0",  
  output_file = NULL,  
  record_limit = NULL  
)
```

Arguments

<code>base_url</code>	A string. The base URL of the OAI-PMH server.
<code>metadata_prefix</code>	A string. The metadata format to request (e.g., "oai_dc", "marc21").
<code>set</code>	A string. Optional. A set specifier to limit the harvested records (e.g., "non_dedup").
<code>verbose</code>	A logical. Whether to display progress messages. Default is TRUE.
<code>user_agent</code>	A string. A custom User-Agent string to identify the service. Default is "Finna-Harvester/1.0".
<code>output_file</code>	output file to be saved as a csv file.
<code>record_limit</code>	limits the number of records that the user wants to fetch

Value

A tibble with the harvested records containing selected metadata fields.

Examples

```
## Not run:  
  
# Example for oai_dc (Dublin Core)  
records_oai_dc <- harvest_oai_pmh(  
  base_url = "https://api.finna.fi/OAI/Server",  
  metadata_prefix = "oai_dc",  
  user_agent = "MyCustomHarvester/1.0"  
)
```

```
# Example for marc21 (MARC 21)
records_marc21 <- harvest_oai_pmh(
  base_url = "https://api.finna.fi/OAI/Server",
  metadata_prefix = "marc21",
  user_agent = "MyCustomHarvester/1.0"
)

# Example for oai_vufind_json (VuFind JSON)
records_oai_vufind_json <- harvest_oai_pmh(
  base_url = "https://api.finna.fi/OAI/Server",
  metadata_prefix = "oai_vufind_json",
  user_agent = "MyCustomHarvester/1.0"
)

# Example for oai_ead (Encoded Archival Description)
records_oai_ead <- harvest_oai_pmh(
  base_url = "https://api.finna.fi/OAI/Server",
  metadata_prefix = "oai_ead",
  user_agent = "MyCustomHarvester/1.0"
)

# Example for oai_ead3 (Encoded Archival Description version 3)
records_oai_ead3 <- harvest_oai_pmh(
  base_url = "https://api.finna.fi/OAI/Server",
  metadata_prefix = "oai_ead3",
  user_agent = "MyCustomHarvester/1.0"
)

# Example for oai_forward (Forward metadata format)
records_oai_forward <- harvest_oai_pmh(
  base_url = "https://api.finna.fi/OAI/Server",
  metadata_prefix = "oai_forward",
  user_agent = "MyCustomHarvester/1.0"
)

# Example for oai_lido (Lightweight Information Describing Objects)
records_oai_lido <- harvest_oai_pmh(
  base_url = "https://api.finna.fi/OAI/Server",
  metadata_prefix = "oai_lido",
  user_agent = "MyCustomHarvester/1.0"
)

# Example for oai_qdc (Qualified Dublin Core)
records_oai_qdc <- harvest_oai_pmh(
  base_url = "https://api.finna.fi/OAI/Server",
  metadata_prefix = "oai_qdc",
  user_agent = "MyCustomHarvester/1.0"
)

## End(Not run)
```

load_offline_data *Load 'Finna' Search Results from Offline File*

Description

This function loads previously saved 'Finna' search results from a local .rds file for offline access.

Usage

```
load_offline_data(file_name = "offline_search_results")
```

Arguments

`file_name` A string representing the name of the file to load. The function automatically appends ".rds" if not already included.

Value

A tibble or data frame containing the loaded search results.

Examples

```
## Not run:
search_results <- search_finna("sibelius")
save_for_offline(search_results, "sibelius_search_results")
offline_data <- load_offline_data("sibelius_search_results")
print(offline_data)

## End(Not run)
```

refine_metadata *Refine Finna Metadata*

Description

The `refine_metadata` function cleans and standardizes Finna metadata by:

- **Validating Required Fields:** Checks for the presence of key metadata fields and returns NULL if any are missing.
- **Handling Missing Values:** Replaces NA values in critical fields with descriptive placeholder text (e.g., "Unknown Title").
- **Selecting Relevant Fields:** Keeps only the following fields for streamlined analysis:
 - Title: The title of the resource.
 - Author: The creator or author of the resource.
 - Year: The publication or release year.
 - Language: The language of the resource.

- **Formats:** The format(s) of the resource (e.g., Book, Audio).
- **Subjects:** The subject keywords or classifications.
- **Library:** The owning library or institution.
- **Series:** The series or collection the resource belongs to.

Usage

```
refine_metadata(data)
```

Arguments

`data` A tibble containing raw Finna metadata.

Value

A tibble with selected, cleaned metadata fields, or NULL if required fields are missing.

Examples

```
library(finna)
sibelius_data <- search_finna("sibelius")
refine_metadata(sibelius_data)
```

`save_for_offline` *Save 'Finna' Search Results for Offline Access*

Description

This function saves 'Finna' search results and metadata locally to a file in .rds format, allowing users to access and analyze the data offline without an internet connection.

Usage

```
save_for_offline(data, file_name = "offline_search_results")
```

Arguments

`data` A tibble or data frame containing the 'Finna' search results.

`file_name` A string representing the name of the file to save. The function automatically appends ".rds" to the name if not already included.

Value

No return value. Called for its side effects of saving the data to a file.

Examples

```
## Not run:
search_results <- search_finna("sibelius")
save_for_offline(search_results, "sibelius_search_results")

## End(Not run)
```

search_finna

Finna Index Search with Total Limit Option

Description

This function retrieves records from the Finna index with an option to limit the total number of records returned. The function paginates through the results, fetching records until the specified total limit is reached.

Usage

```
search_finna(
  query = NULL,
  type = "AllFields",
  fields = NULL,
  filters = NULL,
  facets = NULL,
  facetFilters = NULL,
  sort = "relevance,id asc",
  limit = 100,
  lng = "fi",
  prettyPrint = FALSE
)
```

Arguments

query	description
type	A string specifying the type of search. Options include "AllFields", "Title", "Author", "Subject". Defaults to "AllFields".
fields	A vector of fields to be returned in the search results. Defaults to NULL, which returns a standard set of fields.
filters	A vector of filter queries to refine the search. Defaults to NULL.
facets	A vector specifying which facets to return in the results. Defaults to NULL.
facetFilters	A vector of regular expressions to filter facets. Defaults to NULL.
sort	A string defining the sort order of the results. Options include: <ul style="list-style-type: none"> "relevance,id asc" (default) "main_date_str desc" (Year, newest first)

	<ul style="list-style-type: none"> • "main_date_str asc" (Year, oldest first) • "last_indexed desc" (Last modified) • "first_indexed desc" (Last added) • "callnumber,id asc" (Classmark) • "author,id asc" (Author) • "title,id asc" (Title)
limit	An integer specifying the total number of records to return across multiple pages.
lng	A string for the language of returned translated strings. Options are "fi" - Finnish, "en-gb" - English, "sv" - Swedish, "se" - Sami. Defaults to "fi" - Finnish.
prettyPrint	A logical value indicating whether to pretty-print the JSON response. Useful for debugging. Defaults to FALSE.

Value

A tibble containing the search results with relevant fields extracted and provenance information.

Examples

```
search_results <- search_finna("sibelius", sort = "main_date_str desc", limit = 100)
print(search_results)
```

search_publisher	<i>Finna Publisher Search</i>
------------------	-------------------------------

Description

This function retrieves only the publisher information from the Finna index based on the search query.

Usage

```
search_publisher(
  query = NULL,
  limit = 100,
  lng = "fi",
  filters = NULL,
  prettyPrint = FALSE
)
```

Arguments

query	A string specifying the search query.
limit	An integer specifying the total number of records to return.
lng	A string for the language of returned translated strings. Defaults to "fi".
filters	A vector of filter queries to refine the search. Defaults to NULL.
prettyPrint	A logical value indicating whether to pretty-print the JSON response. Defaults to FALSE.

Value

A tibble containing the record IDs and their respective publishers.

Examples

```
publishers <- search_publisher("sibelius", limit = 10)
print(publishers)
```

timeline	<i>Retrieve Timeline</i>
----------	--------------------------

Description

Timeline data for selected variable (possibly across various groups).

Usage

```
timeline(
  x,
  field = "titlecount",
  group = NULL,
  nmin = 0,
  mode = "absolute",
  time.window = 10,
  time.field = "Year",
  plot.type = NULL
)
```

Arguments

x	data frame
field	Numeric field to summarize in the timeline. The number of entries (title count) per decade is used by default. If this argument is used, the sum of entries per decade for this field is given.
group	Optional. Name for a data field that indicates groups to compare.
nmin	Include only entries with at least nmin absolute frequency
mode	"absolute" or "relative"
time.window	Time window for the timeline in years. Default: 10 (publication decade).
time.field	Specify the field to be used for time. By default: "Year", or if time.window is 10, then "publication_decade"
plot.type	generates a plot with options like "lineplot" or "barplot".

Value

data.frame

Author(s)

Leo Lahti <leo.lahti@iki.fi>

References

See citation("bibliographica")

Examples

```
## Not run: timeline(df, "gatherings", plot.type = "lineplot")
```

top_plot

Plot Top Entries

Description

Visualizes the top entries for a given field in a data frame. Count and percentage statistics is also shown as needed.

Usage

```
top_plot(
  x,
  field = NULL,
  ntop = NULL,
  highlight = NULL,
  max.char = Inf,
  show.rest = FALSE,
  show.percentage = FALSE,
  log10 = FALSE
)
```

Arguments

x	Data frame, vector or factor
field	Field to show
ntop	Number of top entries to show
highlight	Entries from the 'field' to be highlighted
max.char	Max number of characters in strings. Longer strings will be cut and only max.char first characters are shown. No cutting by default
show.rest	Show the count of leave-out samples (not in top-N) as an additional bar.
show.percentage	Show the proportion of each category with respect to the total sample count.
log10	Show the counts on log10 scale (default FALSE)

Value

ggplot object

Author(s)

Leo Lahti <leo.lahti@iki.fi>

References

See citation("bibliographica")

Examples

```
## Not run: p <- top_plot(x, field, 50)
```

Index

* utilities

[timeline](#), [17](#)

[top_plot](#), [18](#)

[analyze_metadata](#), [2](#)

[analyze_trends_over_time](#), [3](#)

[bibentry](#), [8](#)

[check_api_access](#), [4](#)

[enrich_author_name](#), [4](#)

[fetch_all_records](#), [5](#)

[fetch_finna](#), [6](#)

[fetch_finna\(\)](#), [8](#)

[fetch_viola_records](#), [7](#)

[finna_cite](#), [8](#)

[finna_cite\(\)](#), [8](#)

[finna_interactive](#), [8](#)

[full_fennica_records](#), [9](#)

[get_finna_records](#), [10](#)

[harvest_oai_pmh](#), [11](#)

[load_offline_data](#), [12](#)

[refine_metadata](#), [13](#)

[save_for_offline](#), [14](#)

[search_finna](#), [15](#)

[search_finna\(\)](#), [8](#)

[search_publisher](#), [16](#)

[timeline](#), [17](#)

[top_plot](#), [18](#)