Package ‘ojsr’

July 1, 2020

Type Package
Title Crawler and Scraper for Open Journal System (‘OJS’)
Version 0.1.1
Description Crawler for ‘OJS’ pages and scraper for meta-data from articles.
   You can crawl ‘OJS’ archives, issues, articles, galleys, and search results.
   You can scrap articles meta-data from their head tag in html, or from Open Archives Initiative (‘OAI’) records.
   Most of these functions rely on ‘OJS’ routing conventions
License GPL-3
Encoding UTF-8
LazyData true
Imports dplyr (>= 0.8.3), magrittr, oai, rvest, stringr, tidyr (>= 1.0), urltools, utils, xml2, purrr, rlang
Suggests knitr, rmarkdown, testthat, tidyverse
VignetteBuilder knitr
RoxygenNote 7.1.1
NeedsCompilation no
Author Gaston Becerra [aut, cre] (<https://orcid.org/0000-0001-9432-8848>)
Maintainer Gaston Becerra <gaston.becerra@gmail.com>
Repository CRAN
Date/Publication 2020-07-01 11:00:14 UTC

R topics documented:

get_articles_from_issue .................................................. 2
get_articles_from_search .............................................. 3
get_galleys_from_article ............................................. 3
get_html_meta_from_article ....................................... 4
get_issues_from_archive ............................................ 5
get_oai_meta_from_article ......................................... 6
get_articles_from_issue

Scrap an OJS issue and retrieves the articles’ url

Description
Takes a vector of OJS urls and scraps them to retrieve links to OJS articles.

Usage
get_articles_from_issue(input_url, verbose = FALSE)

Arguments
input_url Character vector.
verbose Logical.

Value
A long-format dataframe with the url you provided (input_url) and the articles url scrapped (output_url).

Examples
issues <- c(
  'https://revistas.ucn.cl/index.php/saludysociedad/issue/view/65',
)
articles <- ojsr::get_articles_from_issue(input_url = issues, verbose = TRUE)
get_articles_from_search

Description
Takes a vector of OJS urls, process them to create search result pages (including pagination) and scraps them to retrieve links to OJS articles

Usage
get_articles_from_search(input_url, search_criteria, verbose = FALSE)

Arguments
input_url Character vector.
search_criteria Character string
verbose Logical.

Value
A dataframe with the urls of the articles linked from the OJS issue page.

Examples

ejournals <- c(
  "https://revistapsicologia.uchile.cl/index.php/RDP/",
  "https://publicaciones.sociales.uba.ar/index.php/psicologiasocial/
)
criteria <- "actitudes"
search_result_pages <- ojsr::get_articles_from_search(input_url = journals,
  search_criteria = criteria, verbose = TRUE)

get_galleys_from_article

Description
Takes a vector of OJS urls and scraps them to retrieve links to OJS galleys
get_html_meta_from_article

Scrap metadata from the HTML of OJS articles

Description
Takes a vector of OJS urls and and scraps the metadata written in the html.

Usage
get_html_meta_from_article(input_url, verbose = FALSE)

Arguments
input_url Character vector.
verbose Logical.

Value
A long-format dataframe with the url you provided (input_url), the name of the metadata (meta_data_name), the content of the metadata (meta_data_content), the standard in which the content is annotated (meta_data_scheme), and the language in which the metadata was entered (meta_data_xmllang).
\texttt{get_issues_from_archive}

\textit{Scrap an OJS issues archive and retrieves the issues' url}

\textbf{Description}

Takes a vector of OJS urls and scraps their archive of issues to retrieve links to OJS issues.

\textbf{Usage}

\begin{verbatim}
get_issues_from_archive(input_url, verbose = FALSE)
\end{verbatim}

\textbf{Arguments}

\begin{itemize}
  \item \texttt{input_url} Character vector.
  \item \texttt{verbose} Logical.
\end{itemize}

\textbf{Value}

A long-format dataframe with the url you provided (input_url) and the url of issues found (output_url)

\textbf{Examples}

\begin{verbatim}
issues <- ojsr::get_issues_from_archive(input_url = journals, verbose = TRUE)
\end{verbatim}
get_oai_meta_from_article

Get OAI metadata from an OJS article url

Description

This functions access OAI records (within OJS) for any article for which you provided an url.

Usage

get_oai_meta_from_article(input_url, verbose = FALSE)

Arguments

input_url

Character vector.

verbose

Logical.

Details

Several limitations are in place. Please refer to vignette.

Value

A long-format dataframe with the url you provided (input_url), the name of the metadata (meta_data_name), and the content of the metadata (meta_data_content).

Examples

articles <- c(
  'https://publicaciones.sociales.uba.ar/index.php/psicologiasocial/article/view/2137', # article
  'https://dspace.palermo.edu/ojs/index.php/psicodebate/article/view/516/311' # xml galley
)
metadata_oai <- ojsr::get_oai_meta_from_article(input_url = articles, verbose = TRUE)

ojsr

ojsr: A package for scrapping OJS

Description

This package allows you scrap content (bibliographic metadata) from OJS front-pages and their OAI interfaces; This is useful when the OJS Rest API is not available (as in OJS installments prior to v3.1). It also includes function to parse OJS specific URL conventions.
parse_base_url

<table>
<thead>
<tr>
<th>Description</th>
<th>Parses urls against OJS routing conventions and retrieves the base url</th>
</tr>
</thead>
</table>

Takes a vector of urls and parses them according to OJS routing conventions, then retrieves OJS base url.

Usage

```r
parse_base_url(input_url)
```

Arguments

- `input_url` : Character vector.

Value

A vector of the same length of your input.

Examples

```r
mix_links <- c(
  'https://dspace.palermo.edu/ojs/index.php/psicodebate/issue/archive',
)
base_url <- ojsr::parse_base_url(input_url = mix_links)
```

parse_oai_url

<table>
<thead>
<tr>
<th>Description</th>
<th>Parses urls against OJS routing conventions and retrieves the OAI url</th>
</tr>
</thead>
</table>

Takes a vector of urls and parses them according to OJS routing conventions, then retrieves OAI entry url.

Usage

```r
parse_oai_url(input_url)
```

Arguments

- `input_url` : Character vector.
Value

A vector of the same length of your input.

Examples

```r
mix_links <- c(
  'https://dspace.palermo.edu/ojs/index.php/psicodebate/issue/archive',
)
oai_url <- ojsr::parse_oai_url(input_url = mix_links)
```
Index

get_articles_from_issue, 2
get_articles_from_search, 3
get_galleys_from_article, 3
get_html_meta_from_article, 4
get_issues_from_archive, 5
get_oai_meta_from_article, 6

ojsr, 6

parse_base_url, 7
parse_oai_url, 7