# Package 'scraEP'

July 3, 2018

**Type** Package

**Title** Scrape European Parliament Careers

**Version** 1.1

**Date** 2018-07-01

**Author** Julien Boelaert <jubo.stats@gmail.com>

**Maintainer** Julien Boelaert <jubo.stats@gmail.com>

**Description** A utility to webscrape the in-house careers of members of the European parliament, from its website <http://www.europarl.europa.eu>.

**License** GPL (>= 3)

**Imports** XML, RCurl, data.table

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2018-07-03 15:30:03 UTC

## R topics documented:



---

`scraEP-package` *Scrape the careers of all members of European parliament.*

---

### Description

Fetch all in-house career information of MEPs from the European parliament's website, and put them in a data frame.

### Details

| | |
|---|---|
| Package: | scraEP |
| Type: | Package |
| Version: | 1.1 |
| Date: | 2018-07-01 |
| License: | GPL (>=3) |

Function `scraEP` downloads and extracts career information from the EP's website.

Function `xscrape` is a general tool to extract information from html pages into data frames using XPath queries.

## Author(s)

Julien Boelaert <jubo.stats@gmail.com>

---

`scraEP` *Extract all (in-house) career information of MEPs from the European parliament's website.*

---

## Description

This function downloads all career information from the European parliament's website <http://www.europarl.europa.eu> or from local html files, and extracts it into a data frame.

## Usage

```
scraEP(local.html= NA, save.html= NA, max.try= 20)
```

## Arguments

| | |
|---|---|
| `local.html` | a character string, indicating the local directory from which the MEP's html files are imported (existing files will be overwritten). If `NA` (default), the career information is downloaded from the EP's website. |
| `save.html` | a character string, indicating the local directory in which the downloaded html files should be saved. If `NA` (default), the downloaded files are not saved to local filesystem. |
| `max.try` | numeric: maximum number of tries to download html files from the EP's website. |

## Details

Downloading all the pages can take a long time, be sure to store the result in an object.

## Value

A data.frame, where each row is a career step recorded on the EP's website.

## Author(s)

Julien Boelaert <jubo.stats@gmail.com>

---

`wiki` *Wikipedia page for R.*

---

## Description

Toy example to extract data using `xscrape`.

## Usage

```
data(wiki)
```

## Format

Object `wiki` is a character vector.

## Details

Object `wiki` is a raw webpage, containing the source of the 'R (programming language)' article on English wikipedia (<https://en.wikipedia.org/wiki/R_(programming_language)>, retrieved on 15/11/2017).

## Author(s)

Julien Boelaert <jubo.stats@gmail.com>

---

`xscrape` *Extract information from webpages to a data.frame, using XPath queries.*

---

## Description

This function transforms an html page (or list of pages) into a data.frame, extracting nodes specified by their XPath.

## Usage

```
xscrape(pages, col.xpath, row.xpath= "/html", collapse= " | ", encoding= "UTF-8")
```

## Arguments

| | |
|---|---|
| pages | an object of class XMLDocument (as returned by function htmlParse), or list of such objects. Alternatively, a character vector containing the URLs of webpages to be parsed. These are the webpages that information is to be extracted from. |
| col.xpath | a character vector of XPath queries. Each element of this vector will be a column in the resulting data.frame. If the vector is named, these names are given to the result's. |
| row.xpath | (optional) a character string, containing an XPath query. This functions as an intermediary node: if specified, each result of this XPath query (on each page) becomes a row in the resulting data.frame. If not specified (default), the intermediary nodes are whole html pages, so that each page becomes a row in the result. |
| collapse | (optional) a character string, containing the separator that will be used in case a col.xpath query yields multiple results. |
| encoding | (optional) a character string, containing the encoding parameter that will be used by htmlParse if pages is a vector of URLs. |

## Details

If a col.xpath query designs a full node, only its text is extracted. If it designs an attribute (eg ends with '/@href' for weblinks), only the attribute's value is extracted.

If a col.xpath query matches no elements in a page, returned value is NA. If it matches multiple elements, they are concatenated into a single character string, separated by collapse.

## Value

A data.frame, where each row corresponds to an intermediary node (either a full page or an XML node within a page, specified by row.xpath), and each column corresponds to the text of an col.xpath query.

## Author(s)

Julien Boelaert <jubo.stats@gmail.com>

## Examples

```
## Extract all external links and their titles from a wikipedia page
data(wiki)
wiki.parse <- XML::htmlParse(wiki)
links <- xscrape(wiki.parse,
                 row.xpath= "//a[starts-with(./@href, 'http')]",
                 col.xpath= c(title= ".", link= "./@href"))

## Not run:
## Convert results from a search for 'R' on duckduckgo.com
## First download the search page
duck <- XML::htmlParse("http://duckduckgo.com/html/?q=R")
## Then run xscrape on the dowloaded and parsed page
```

```
results <- xscrape(duck,
                   row.xpath= "//div[contains(@class, 'result__body')]",
                   col.xpath= c(title= "./h2",
                                snippet= ".//*[@class='result__snippet']",
                                url= ".//a[@class='result__url']/@href"))

## End(Not run)

## Not run:
## Convert results from a search for 'R' and 'Julia' on duckduckgo.com
## Directly provide the URLs to xscrape
results <- xscrape(c("http://duckduckgo.com/html/?q=R",
                     "http://duckduckgo.com/html/?q=julia"),
                   row.xpath= "//div[contains(@class, 'result__body')]",
                   col.xpath= c(title= "./h2",
                                snippet= ".//*[@class='result__snippet']",
                                url= ".//a[@class='result__url']/@href"))

## End(Not run)
```

# Index