Package 'W3CMarkupValidator'

February 6, 2023

Version 0.1-7

Title R Interface to W3C Markup Validation Services

Description R interface to a W3C Markup Validation service.
See <https://validator.w3.org/> for more information.

Depends R (>= 3.2.0)

Imports curl, utils, xml2

License GPL-2

NeedsCompilation no

Author Kurt Hornik [aut, cre] (<https://orcid.org/0000-0003-4198-9911>)

Maintainer Kurt Hornik <Kurt.Hornik@R-project.org>

Repository CRAN

Date/Publication 2023-02-06 19:35:46 UTC

R topics documented:

`inspect` *Inspect* R *objects*

Description

Display R objects in a convenient and informative way.

Usage

```
inspect(x, ...)
## S3 method for class 'w3c_markup_validate'
inspect(x, details = TRUE, ...)
## S3 method for class 'w3c_markup_validate_db'
inspect(x, details = TRUE, full = FALSE, ...)
```

Arguments

x	an R object for the generic; objects inheriting from the respective classes for the methods.
details	a logical recycled to length two indicating whether to display detailed information on errors and warnings, respectively, or a character vector with elements partially matching 'error' or 'warning'.
full	a logical indicating whether to provide information about validation results with no errors or warnings.
...	arguments to be passed to and from methods.

Details

`inspect()` is a generic function.

The methods for objects inheriting from `"w3c_markup_validate"` or `"w3c_markup_validate_db"` (single results of markup validation using `w3c_markup_validate`, or collections of such results) conveniently summarize the problems found by the validation service as collections of tables with columns giving the line, column and a description of the problem.

`w3c_markup_validate` *Validate Markup of Web Documents using W3C Markup Validation Services*

Description

Check the markup validity of web documents in HTML, XHTML, etc., using a W3C Markup Validation service.

Usage

```
w3c_markup_validate(baseurl = w3c_markup_validate_baseurl(),
                    uri = NULL, file = NULL, string = NULL,
                    opts = list())
```

Arguments

baseurl	a character string giving the URL of the W3C Markup Validation service to employ.
uri	a character string giving the URI to validate.
file	a character string giving the path of a file to validate.
string	a character string with the markup to validate.
opts	a named list or curlOptions object with options to use for accessing the validation service via getURL (in case uri is given) or postForm (in case file or string are given).

Details

Exactly one of uri, file or string must be given.

Validation is then performed by using the W3C Markup Validation service at the given URL, using the (still declared "experimental") SOAP 1.2 API of such a service (see https://validator.w3.org/docs/api.html for more information).

If a SOAP validation response could be obtained, w3c_markup_validate() returns the information in the response organized into an object of class "w3c_markup_validate", which is a list with the following elements:

valid a logical indicating the validity of the web document checked (TRUE iff there were no errors)

errorcount an integer giving the number of errors found.

errors a data frame with variables 'line', 'col', 'message', 'messageid', 'explanation' and 'source' with the obvious meanings, or NULL.

warningcount an integer giving the number of warnings found.

warnings a data frame with variables as for errors, or NULL.

This class has methods for print for compactly summarizing the results, an inspect method for inspecting details, and an as.data.frame method for collapsing the errors and warnings into a "flat" data frame useful for further analyses.

Note

The validation service provided by the W3C used by default for validation is a shared and free resource, and the W3C asks (see https://validator.w3.org/docs/api.html) for considerate use and possibly installing a local instance of the validation service: excessive use of the service will be blocked. In fact, it seems that since May 2015 W3C blocks access to the SOAP API, so one needs to use a different (local) validation service.

On Debian-based systems, a local instance can conveniently be installed via the system command apt-get install w3c-markup-validator and following the instructions for providing the validator as a web service.

One can use the environment variable W3C_MARKUP_VALIDATOR_BASEURL to specify the service to be employed by default. E.g., one can set this to "http://localhost/w3c-validator/check" for Debian-based systems as discussed above.

See Also

w3c_markup_validate_baseurl for getting and setting the URL of the validation service.

w3c_markup_validate_db for combining and analyzing collections of single validation results.

Examples

```
## Not much to show with this as it should validate ok
## (provided that the validation service is accessible):
tryCatch(w3c_markup_validate(uri = "https://CRAN.R-project.org"),
         error = identity)
```

---

w3c_markup_validate_baseurl

URL of W3C Markup Validation Service

Description

Get or set the URL of the W3C Markup Validation service to employ.

Usage

```
w3c_markup_validate_baseurl(new)
```

Arguments

new	a character string with the URL of the the W3C Markup Validation service to employ, or NULL indicating to use the W3C service as specified by the environment variable W3C_MARKUP_VALIDATOR_BASEURL, or if this is unset, at https://validator.w3.org/check.

Details

If no argument is given, the current URL is returned. Otherwise, the URL is set to the given one or (if new is NULL) the default one.

w3c_markup_validate_db *Collections of W3C Markup Validation Results*

Description

Create and manipulate collections of W3C markup validation results.

Usage

```
w3c_markup_validate_db(x, names = NULL)

w3c_markup_validate_files(files, baseurl = w3c_markup_validate_baseurl(),
                          opts = list())
w3c_markup_validate_uris(uris, baseurl = w3c_markup_validate_baseurl(),
                         opts = list())
```

Arguments

x	a list of w3c_markup_validate results.
names	a character vector of names for the elements in x, or NULL (default), indicating to use the names of x or to auto-generate names if these are NULL.
files	a character vector giving the names of files to validate.
uris	a character vector giving URIs to validate.
baseurl	a character string giving the URL of the W3C Markup Validation service to employ.
opts	see w3c_markup_validate.

Details

w3c_markup_validate_db() creates a db (data base) of w3c_markup_validate results as a list of these results with class "w3c_markup_validate_db". This class has methods for print and c for compactly summarizing and combining results, an inspect method for inspecting details, and an as.data.frame method for collapsing the errors and warnings into a "flat" data frame useful for further analyses.

w3c_markup_validate_files() and w3c_markup_validate_uris() validate the markup in the given files or URIs, with results combined into such results db objects. For files or URIs for which validation failed (which can happen for example when these contain characters invalid in SGML), the corresponding error condition objects are gathered into the "failures" attribute of the results db returned.

See Also

w3c_markup_validate_baseurl for getting and setting the URL of the validation service.

Examples

```
## Test files provided with this package:
dir <- system.file("examples", package = "W3CMarkupValidator")
files <- Sys.glob(file.path(dir, "*.html"))
if(!grepl("^http://validator.w3.org",
          w3c_markup_validate_baseurl())) {
    ## Validate.
    results <- w3c_markup_validate_files(files)
    results
    ## In case of failures, inspect the error messages:
    lapply(attr(results, "failures"), conditionMessage)
    ## Inspect validation results:
    inspect(results)
    inspect(results, full = TRUE)
    ## Turn results into a data frame:
    df <- as.data.frame(results)
    ## Tabulate error messages:
    table(substring(df$message, 1L, 60L))
    ## Inspect a particular set of error messages:
    df[df$message == "element \"font\" undefined", ]
    ## (Note that explanations are in HTML ...)
    ## Conveniently view the full records (modulo HTML markup):
    write.dcf(df)
}
```

Index