Package ‘xmlparsedata’

September 27, 2019

Title Parse Data of 'R' Code as an 'XML' Tree
Version 1.0.3
Author Gábor Csárdi
Maintainer Gábor Csárdi <csardi.gabor@gmail.com>
Description Convert the output of 'utils::getParseData()' to an 'XML' tree, that one can search via 'XPath', and easier to manipulate in general.
License MIT + file LICENSE
LazyData true
URL https://github.com/r-lib/xmlparsedata#readme
BugReports https://github.com/r-lib/xmlparsedata/issues
RoxygenNote 6.0.1
Suggests covr, testthat, xml2
Depends R (>= 3.0.0)
Encoding UTF-8
NeedsCompilation no
Repository CRAN
Date/Publication 2019-09-27 11:40:02 UTC

R topics documented:

xmlparsedata ......................................................... 2
xml_parse_data .................................................... 2
xml_parse_token_map .............................................. 3

Index 4
xml_parse_data

Parse Data of R Code as an 'XML' Tree

Description
Convert the output of `utils::getParseData()` to an 'XML' tree, that is searchable and easier to manipulate in general.

xml_parse_data
Convert R parse data to XML

Description
In recent R versions the parser can attach source code location information to the parsed expressions. This information is often useful for static analysis, e.g. code linting. It can be accessed via the `getParseData` function.

Usage
xml_parse_data(x, includeText = NA, pretty = FALSE)

Arguments
x
an expression returned from `parse`, or a function or other object with source reference information

includeText
logical; whether to include the text of parsed items in the result

pretty
Whether to pretty-indent the XML output. It has a small overhead which probably only matters for very large source files.

Details
xml_parse_data converts this information to an XML tree. The R parser’s token names are preserved in the XML as much as possible, but some of them are not valid XML tag names, so they are renamed, see the `xml_parse_token_map` vector for the mapping.

The top XML tag is `<exprlist>`, which is a list of expressions, each expression is an `<expr>` tag. Each tag has attributes that define the location: `line1`, `col1`, `line2`, `col2`. These are from the `getParseData` data frame column names.

See an example below. See also the README at https://github.com/r-lib/xmlparsedata#readme for examples on how to search the XML tree with the xml2 package and XPath expressions.

Value
An XML string representing the parse data. See details below.
xml_parse_token_map

See Also

xml_parse_token_map for the token names. https://github.com/r-lib/xmlparsedata#readme for more information and use cases.

Examples

code <- "function(a = 1, b = 2) \{\n  a + b\n}\n"
expr <- parse(text = code, keep.source = TRUE)

# The base R way:
geParserData(expr)
cat(xml_parse_data(expr, pretty = TRUE))

xml_parse_token_map  Map token names of the R parser to token names in xml_parse_data

Description

Some of the R token names are not valid XML tag names, so xml_parse_data needs to replace them to create a valid XML file.

Usage

xml_parse_token_map

Format

An object of class character of length 19.

See Also

xml_parse_data
Index

*Topic **datasets**
  * xml_parse_token_map, 3

getParseData, 2

parse, 2

xml_parse_data, 2, 3
xml_parse_token_map, 2, 3, 3
xmlparsedata, 2
xmlparsedata-package (xmlparsedata), 2