Package 'tinytest2JUnit'

March 12, 2024

Type Package

Title Convert 'tinytest' Output to JUnit XML

Version 1.0.3

Maintainer Anne-Katrin Hess <anne-katrin.hess@openanalytics.eu>

Description Unit testing is a solid component of automated CI/CD pipelines. 'tinytest' - a light-weight, zero-dependency alternative to 'testthat' was developed. To be able to integrate 'tinytests' results into common CI/CD
systems the 'tinytests'-
object needs to be converted to JUnit XML format. 'tinytest2JUnit' enables this conversion while keeping the zero-dependency nature.

Suggests tinytest

URL https://github.com/openanalytics/tinytest2JUnit

BugReports https://github.com/openanalytics/tinytest2JUnit/issues

RoxygenNote 7.2.3

Encoding UTF-8

NeedsCompilation no

Author Anne-Katrin Hess [aut, cre],
Lennart Tuijnder [aut]

Repository CRAN

Date/Publication 2024-03-12 07:30:03 UTC

R topics documented:

`constructTestcaseTag` *Construct JUnit* `</testcase>` *tag*

Description

Construct JUnit `</testcase>` tag based on a single `tinytest` result.

Usage

```
constructTestcaseTag(tinytest)
```

Arguments

`tinytest`	a `tinytest`-object representing an individual test case.

Value

`XMLtag`: with tag-name = `tinytest` and contains the test result per test.

`constructTestsuitesTag` *Construct the JUnit* `</testsuites>` *tag*

Description

Convert the `tinytests`-object containing test across possibly multiple files into a JUnit `</testsuites>` tag.

Usage

```
constructTestsuitesTag(testResults)
```

Arguments

`testResults`	`tinytests`-object to convert into a JUnit XML object. Usually the result of calling `tinytest::test_package()` or `tinytest::run_test_dir()`.

Details

Reference for JUnit XML format: https://llg.cubic.org/docs/junit/

Value

XMLtag: with tag-name = </testsuites>. This is the root of the JUnit XML document.

---

constructTestsuiteTag *Construct JUnit </testsuite> tag*

Description

Construct the </testsuite> tag of a tinytest, given all the tinytest results from a single test file.

Usage

```
constructTestsuiteTag(testResultsSingleFile)
```

Arguments

testResultsSingleFile	tinytesta-object with all test results of a specified test file.

Value

XMLtag: with tag-name = </testsuite> that contains all the test results per test file.

---

escapeXmlText *Escape xml text*

Description

Escape the characters '<' and & in a character vector meant to be xml-text.

Usage

```
escapeXmlText(x)
```

Arguments

x	a character vector meant to be xml-text.

Value

The same character vector x but xml text escaped.

format.XMLtag *Format method for XMLtag class*

Description

Format S3 method for the XMLtag-class

Usage

```
## S3 method for class 'XMLtag'
format(x, level = 0, ...)
```

Arguments

x	an XMLtag-object
level	print depth level. For each level 2 spaces are added to the left. The content of a tag is automatically indented with 1 level.
...	to ignore

Value

character(1) vector of the formatted XML tag.

print.XMLtag *Print method for XMLtag class.*

Description

Print method for XMLtag class.

Usage

```
## S3 method for class 'XMLtag'
print(x, ...)
```

Arguments

x	a XMLtag-object
...	to be ignored

Value

invisibly the string that was printed to stdout.

tag *XML tag*

Description

Create a list object that roughly mimics the behaviour of a simplistic XML tag element. Supported are XML tag-name, tag-attributes and tag-content.

Usage

```
tag(name, attributes = list(), content = list())
```

Arguments

name	character(1) specifying the name of the tag.
attributes	named-list being the XML attributes. Names = attribute names, Values = attribute value.
content	unnamed-list being the content XML-tag. Each element is placed next to each other in the tag.

Value

a XMLtag-object.

writeJUnit *Write the results of a* tinytests*-object into JUnit xml report.*

Description

Write the tinytests-object to a JUnit XML reporting file.

Usage

```
writeJUnit(tinytests, file, overwrite = TRUE)
```

Arguments

tinytests	tinytests-object to convert to JUnit xml.
file	character(1): Full file path to the .xml file to write the JUnit xml to. Example: "/home/user/documents/results.xml".
overwrite	logical(1): should the file be overwritten if it already exist? By default TRUE.

Value

invisible: TRUE.

Side-effects

Side effects are registered as a tests in the JUnit output and have been given a status "SIDE-EFFECT". The call and diff is also returned in the standard-output of the testcase tag.

They are however not considered as failures and would thus not stop a pipeline.

Errors

In case of overwrite = FALSE and the file already exists an error is thrown.

See Also

The JUnit XML report format: https://llg.cubic.org/docs/junit/

Examples

```
# Run tests with `tinytest`
dirWithTests <- system.file("example_tests/multiple_files",package = "tinytest2JUnit")
testresults <- tinytest::run_test_dir(dirWithTests, verbose = FALSE)
# temporary output file to save JUnit XML to
tmpFile <- tempfile(fileext = ".xml")
writeJUnit(tinytests = testresults, file = tmpFile)
```

Index