Package 'sasMap'

October 14, 2022

Title Static 'SAS' Code Analysis

Version 1.0.0

Maintainer Nic Crane <ncrane@mango-solutions.com>

Description A static code analysis tool for 'SAS' scripts. It is designed to load, count, extract, remove, and summarise components of 'SAS' code.

Depends R (>= 3.2.4)

Imports readr, stringr, stringi

Suggests testthat, markdown

URL https://github.com/MangoTheCat/sasMap

BugReports https://github.com/mangothecat/sasMap/issues

License MIT + file LICENSE

RoxygenNote 6.0.1

NeedsCompilation no

Author Nic Crane [aut, cre],
Ava Yang [aut],
Richard Pugh [aut],
Gregoire Gauriot [aut],
Jinjing Xie [aut],
Mango Solutions [cph]

Repository CRAN

Date/Publication 2017-08-18 09:00:25 UTC

R topics documented:

`countDataSteps` *Count number of data steps in a SAS file*

Description

Count number of data steps in a SAS file

Usage

```
countDataSteps(sasCode)
```

Arguments

`sasCode`	Character string containing SAS code

Examples

```
sasFile <- system.file('examples/SAScode/MainAnalysis.SAS', package='sasMap')
sasCode <- loadSAS(sasFile)
countDataSteps(sasCode)
```

`countLines` *Count lines in a SAS file*

Description

Count lines in a SAS file

Usage

```
countLines(sasCode)
```

Arguments

`sasCode`	Character string containing SAS code

Examples

```
sasFile <- system.file('examples/SAScode/MainAnalysis.SAS', package='sasMap')
sasCode <- loadSAS(sasFile)
countLines(sasCode)
```

`countProcSteps` *Count number of proc steps in a SAS file*

Description

Count number of proc steps in a SAS file

Usage

```
countProcSteps(sasCode)
```

Arguments

`sasCode`	Character string containing SAS code

Examples

```
sasFile <- system.file('examples/SAScode/MainAnalysis.SAS', package='sasMap')
sasCode <- loadSAS(sasFile)
countProcSteps(sasCode)
```

`countStatements` *Count statements in a SAS file*

Description

Count statements in a SAS file

Usage

```
countStatements(sasCode)
```

Arguments

`sasCode`	Character string containing SAS code

Examples

```
sasFile <- system.file('examples/SAScode/MainAnalysis.SAS', package='sasMap')
sasCode <- loadSAS(sasFile)
countStatements(sasCode)
```

extractMacroCalls *Extract macro calls from a string of SAS code*

Description

Extract macro calls from a string of SAS code

Usage

```
extractMacroCalls(sasCode, ignoreList = c("macro", "mend", "global", "let",
  "put", "if", "do", "end", "else"))
```

Arguments

sasCode	SAS code
ignoreList	Macro calls to ignore

Value

Vector of macro calls

Examples

```
sasFile <- system.file('examples/SAScode/Macros/Util1.SAS', package='sasMap')
sasCode <- loadSAS(sasFile)
extractMacroCalls(sasCode)
```

extractMacroDefs *Extract macro definitions from a string of SAS code*

Description

Extract macro definitions from a string of SAS code

Usage

```
extractMacroDefs(sasCode)
```

Arguments

sasCode	SAS code

Value

Vector of macro definitions

Examples

```
sasFile <- system.file('examples/SAScode/Macros/Util1.SAS', package='sasMap')
sasCode <- loadSAS(sasFile)
extractMacroDefs(sasCode)
```

---

extractProcs *Extract procs from a vector of SAS code*

Description

Extract procs from a vector of SAS code

Usage

```
extractProcs(sasCode)
```

Arguments

sasCode	SAS code

Examples

```
sasFile <- system.file('examples/SAScode/Macros/Util1.SAS', package='sasMap')
sasCode <- loadSAS(sasFile)
extractProcs(sasCode)
```

---

loadSAS *Load SAS Code in*

Description

Load SAS Code in

Usage

```
loadSAS(sasPath)
```

Arguments

sasPath	SAS file location

Value

Character string containing contents of the file

Examples

```
sasPath <- system.file('examples/SAScode/Macros/fun2.SAS', package='sasMap')
loadSAS(sasPath)
```

removeAllComments — *Remove single line comments from a character string containing SAS code*

Description

This version of this function removes all comments.

Usage

```
removeAllComments(sasCode)
```

Arguments

sasCode	Character string containing SAS code

Value

Character string of SAS code with comments removed

removeMultilineComments — *Remove multiline comments from SAS code*

Description

Remove multiline comments from SAS code

Usage

```
removeMultilineComments(sasCode)
```

Arguments

sasCode	Character string containing SAS code

Examples

```
sasFile <- system.file('examples/SAScode/Macros/Util2.SAS', package='sasMap')
sasCode <- loadSAS(sasFile)
removeMultilineComments(sasCode)
```

removeSingleLineComments

Remove single line comments from a vector of SAS statements

Description

Finds and removes all lines starting with "*"

Usage

```
removeSingleLineComments(sasVec)
```

Arguments

sasVec	Vector of SAS code

Examples

```
sasFile <- system.file('examples/SAScode/Macros/Util2.SAS', package='sasMap')
sasCode <- loadSAS(sasFile)
removeAllComments(sasCode)
```

removeWhitespaceCharacters

Remove whitespace characters from SAS code

Description

Remove whitespace characters from SAS code

Usage

```
removeWhitespaceCharacters(sasCode)
```

Arguments

sasCode	Character string containing SAS code

Examples

```
sasFile <- system.file('examples/SAScode/Macros/Util2.SAS', package='sasMap')
sasCode <- loadSAS(sasFile)
removeWhitespaceCharacters(sasCode)
```

splitIntoStatements *Split SAS code into statements*

Description

Split SAS code into statements

Usage

```
splitIntoStatements(sasCode)
```

Arguments

sasCode	Character string containing SAS code

Examples

```
sasPath <- system.file('examples/SAScode/Macros/fun2.SAS', package='sasMap')
sasCode <- loadSAS(sasPath)
splitIntoStatements(sasCode)
```

summariseSASScript *Summarise a SAS script*

Description

Summarise a SAS script

Usage

```
summariseSASScript(sasPath)
```

Arguments

sasPath	Path to SAS script

Examples

```
sasPath <- system.file('examples/SAScode/Macros/fun2.SAS', package='sasMap')
summariseSASScript(sasPath)
```

Index