

Package ‘semverutils’

February 22, 2020

Title Semantic Version Utilities

Version 0.1.0

Description Semantic Versions allow for standardized management versions.
This package implements semantic versioning handling in R. using R6 to
create a mutable object that can handle deciphering and checking versions.

License MIT + file LICENSE

URL <https://github.com/ajwtech/semverutils>

BugReports <https://github.com/ajwtech/semverutils/issues>

Encoding UTF-8

LazyData true

Imports R6, foreach

Suggests roxygen2, testthat (>= 2.1.0)

RoxygenNote 7.0.2

NeedsCompilation no

Author Adam Wheeler [aut, cre, cph]

Maintainer Adam Wheeler <ajwtech@gmail.com>

Repository CRAN

Date/Publication 2020-02-22 10:00:02 UTC

R topics documented:

semVer	2
Index	4

semVer *semVer is an R6 class to create an object for working with Semantic versioning*

Description

semVer is an R6 class to create an object for working with Semantic versioning

semVer is an R6 class to create an object for working with Semantic versioning

Public fields

SemVerVersion version of semantic versioning rules to be followed currently only version 2.0.0 is supported.

Methods

Public methods:

- [semVer\\$higherThanAll\(\)](#)
- [semVer\\$new\(\)](#)
- [semVer\\$clone\(\)](#)

Method `higherThanAll()`: `higherThanAll()` test if the vector of versions passed in are all lower than this object

Usage:

```
semVer$higherThanAll(versions)
```

Arguments:

`versions` character vector of semantically formatted version numbers

Returns: atomic boolean

Examples:

```
sem <- semVer$new("3.0.0")
sem$higherThanAll(c("2.0.2", "0.1.0"))
```

Method `new()`: initializes a new `semVer` object with the version supplied

Usage:

```
semVer$new(version = "", SemVerVersion = "2.0.0")
```

Arguments:

`version` character of semantically formatted version numbers

`SemVerVersion` defaults to "2.0.0" no other options currently available.

Returns: `semVer` object holding version data

Examples:

```
sem <- semVer$new("3.0.0")
```

Method clone(): The objects of this class are cloneable with this method.

Usage:

```
semVer$clone(deep = FALSE)
```

Arguments:

deep Whether to make a deep clone.

Examples

```
## -----  
## Method `semVer$higherThanAll`  
## -----  
  
sem <- semVer$new("3.0.0")  
sem$higherThanAll(c("2.0.2", "0.1.0"))  
  
## -----  
## Method `semVer$new`  
## -----  
  
sem <- semVer$new("3.0.0")
```

Index

semVer, [2](#)