Package 'jsonStrings'

May 25, 2021

Type Package

Title Manipulation of JSON Strings

Version 1.0.0

Maintainer Stéphane Laurent <laurent_step@outlook.fr>

Description Fast manipulation of JSON strings. Allows to extract or delete an element in a JSON string, merge two JSON strings, and more.

License GPL (>= 2)

Imports Rcpp (>= 1.0.0), methods

LinkingTo Rcpp

RoxygenNote 7.1.1

Encoding UTF-8

SystemRequirements C++11

URL https://github.com/stla/jsonStrings

BugReports https://github.com/stla/jsonStrings/issues

NeedsCompilation yes

Author Stéphane Laurent [aut, cre],
Niels Lohmann [cph] ('nlohmann/json' C++ library)

Repository CRAN

Date/Publication 2021-05-25 11:50:05 UTC

R topics documented:

`as.character.jsonString` *JSON string to character string*

Description

Convert a JSON string to a character string.

Usage

```
## S3 method for class 'jsonString'
as.character(x, pretty = FALSE, ...)
```

Arguments

`x`	a JSON string
`pretty`	logical value, whether to pretty-format the string
`...`	ignored

Value

A character string.

`jsonAddProperty` *Add new property*

Description

Add a new property to a JSON string.

Usage

```
jsonAddProperty(jsonstring, key, value)
```

Arguments

jsonstring	a JSON string representing an object
key	a character string, the key of the new property
value	a JSON string, the value of the new property

Value

A JSON string.

Examples

```
jsonAddProperty("{\"a\":[1,2,3],\"b\":\"hello\"}", "c", "[1,2]")
```

---

jsonAt *Access an element in a JSON string*

Description

Extract an element in a JSON string by giving a path of keys or indices.

Usage

```
jsonAt(jsonstring, path)
```

Arguments

jsonstring	a JSON string representing an array or an object
path	path to the element to extract; either a character vector of keys, an integer vector of indices, or a list made of keys and indices

Value

A JSON string.

Examples

```
jstring <- "[1,[\"a\",99],{\"x\":[2,3,4],\"y\":42}]"
jsonAt(jstring, 1)
jsonAt(jstring, list(2, "x"))
jsonAt(jstring, list(2, "z"))
```

jsonErase *Erase property or element*

Description

Erase an object property or an array element from a JSON string.

Usage

```
jsonErase(jsonstring, at)
```

Arguments

jsonstring	a JSON string representing an object or an array
at	either a character string, the key of the property to be erased, or an integer, the index of the array element to be erased

Value

A JSON string.

Examples

```
jsonErase("{\"a\":[1,2,3],\"b\":\"hello\"}", "a")
```

jsonHasKey *Does key exist?*

Description

Checks whether a key is present in a JSON string.

Usage

```
jsonHasKey(jsonstring, key)
```

Arguments

jsonstring	a JSON string
key	character string

Value

TRUE if the given key is present in the JSON string, FALSE otherwise.

Examples

```
jsonHasKey("{\"a\":[1,2,3],\"b\":\"hello\"}", "b")
jsonHasKey("[1,2,3]", "a")
```

---

jsonIs *Check type of JSON string*

Description

Check the type of a JSON string.

Usage

```
jsonIs(jsonstring, type)
```

Arguments

jsonstring	a JSON string
type	the type to be checked, one of "array", "object", "string", "number", "integer", "null", "boolean"

Value

A logical value.

Examples

```
jsonIs("[1,2]", "array")
```

---

jsonMerge *Merge JSON strings*

Description

Merge two JSON strings.

Usage

```
jsonMerge(jsonstring1, jsonstring2)
```

Arguments

jsonstring1	a JSON string
jsonstring2	a JSON string

Value

A JSON string.

Examples

```
jstring1 <- jsonString("{\"a\":[1,2,3],\"b\":\"hello\"}")
jstring2 <- jsonString("{\"a\":[4,5,6],\"c\":\"goodbye\"}")
jsonMerge(jstring1, jstring2)
```

`jsonPatch` *Patch a JSON string*

Description

Apply a JSON patch to a JSON string.

Usage

```
jsonPatch(jsondoc, jsonpatch)
```

Arguments

`jsondoc`	a JSON string representing an object or an array
`jsonpatch`	a JSON patch, a JSON string representing an array (see the link in details)

Details

See jsonpatch.com.

Value

A JSON string.

Examples

```
jsondoc <- jsonString("{\"a\":[1,2,3],\"b\":\"hello\"}")
jsonpatch <- jsonString("[
  {\"op\": \"remove\", \"path\": \"/a\"},
  {\"op\": \"replace\", \"path\": \"/b\", \"value\": null}
]")
jsonPatch(jsondoc, jsonpatch)
```

jsonPush *Append an element*

Description

Append an element to a JSON string.

Usage

```
jsonPush(jsonstring1, jsonstring2)
```

Arguments

jsonstring1	JSON string representing an array
jsonstring2	JSON string

Value

A JSON string.

Examples

```
jstring1 <- jsonString("[1,2,3]")
jstring2 <- jsonString("{\"a\":\"hello\",\"b\":\"goodbye\"}")
jsonPush(jstring1, jstring2)
```

jsonSize *Number of elements*

Description

Number of elements in a JSON string.

Usage

```
jsonSize(jsonstring)
```

Arguments

jsonstring	a JSON string

Value

An integer.

Examples

```
jsonSize("{\"a\":[1,2,3],\"b\":\"hello\"}")
```

jsonString *JSON string*

Description

Create a JSON string.

Usage

```
jsonString(string)
```

Arguments

string	a character string

Value

A JSON string (external pointer).

Examples

```
jsonString("[1,[\"a\",99],{\"x\":[2,3,4],\"y\":42}]")
```

jsonType *Type of JSON string*

Description

The type of a JSON string.

Usage

```
jsonType(jsonstring)
```

Arguments

jsonstring	a JSON string

Value

A character string indicating the type of the JSON string.

Examples

```
jsonType("[1,2]")
```

jsonUpdate *Update object*

Description

Update a JSON string.

Usage

```
jsonUpdate(jsonstring1, jsonstring2)
```

Arguments

jsonstring1	a JSON string representing an object
jsonstring2	a JSON string representing an object

Value

A JSON string.

Examples

```
jstring1 <- jsonString("{\"a\":[1,2,3],\"b\":\"hello\"}")
jstring2 <- jsonString("{\"a\":[4,5,6],\"c\":\"goodbye\"}")
jsonUpdate(jstring1, jstring2)
```

print.jsonString *Print JSON string*

Description

Print a JSON string.

Usage

```
## S3 method for class 'jsonString'
print(x, pretty = getOption("jsonStrings.prettyPrint", FALSE), ...)
```

Arguments

x	a JSON string
pretty	logical value, whether to pretty-print
...	ignored

Index