

Package ‘rjson’

January 2, 2012

Version 0.2.6

Date 2011-10-17

Title JSON for R

Author Alex Couture-Beil <rjson_pkg@mofo.ca>

Maintainer Alex Couture-Beil <rjson_pkg@mofo.ca>

Depends R (>= 2.5.0)

Description Converts R object into JSON objects and vice-versa

License GPL-2

Repository CRAN

Date/Publication 2011-10-18 06:00:50

R topics documented:

fromJSON	2
newJSONParser	3
rjson	4
toJSON	4
Index	6

fromJSON

Convert JSON To R

Description

Convert a JSON object into an R object.

Usage

```
fromJSON( json_str, file, method = "C" )
```

Arguments

json_str	a JSON object to convert
file	the name of a file to read the json_str from; this can also be a URL. Only one of json_str or file must be supplied.
method	use the C implementation, or the older slower (and one day to be deprecated) R implementation

Value

R object that corresponds to the JSON object

See Also

[toJSON](#)

Examples

```
#watch out for R turning null into FALSE
fromJSON( "[true, false, null]" )
##[1] TRUE FALSE __FALSE__

#but if you use a hash, this will be avoided
fromJSON( '{"a":true, "b":false, "c":null}' )

#R vs C
x <- toJSON( iris )
system.time( y <- fromJSON(x) )
system.time( y2 <- fromJSON(x,method = "R") )
```

newJSONParser	<i>Convert buffered JSON objects To R</i>
---------------	---

Description

Convert a collection of JSON objects into R objects.

Usage

```
newJSONParser(method = "R")
```

Arguments

method use the C implementation, or the slower original R implementation

Value

A list of functions used for parsing objects

See Also

[toJSON](#)

Examples

```
sample_json <- '{
{
"breakfast" : [ "milk", "fruit loops", "juice" ],
"lunch" : [ "left over sushi" ]
}
}'

parser <- newJSONParser()

parser$addData( sample_json )
food <- parser$getObject()
print( food )

#This is equivalent to using FromJSON( sample_json )
#However, sample_json can be split into several parts:

### EXAMPLE 2:

part_1 <- '{ "breakfast" : [ "milk", "fruit loops", "juice" ], '
part_2 <- '"lunch" : [ "left over sushi" ]'
part_3 <- '}' [1,2,3,4,5]' #close off the first object, and create a 2nd JSON object, which is simply an array

parser <- newJSONParser()
parser$addData( part_1 )
```

```

parser$getObject() #returns NULL - since part_1 isn't complete
parser$addData( part_2 )
parser$getObject() #returns NULL - since part_2 still isn't complete
parser$addData( part_3 )
parser$getObject() #returns the first food object
parser$getObject() #returns the second array

```

rjson

JSON for R

Description

JSON (JavaScript Object Notation) is a lightweight data-interchange format. This package converts JSON objects into R objects and vice-versa. See json.org for an overview of JSON. Unicode is unfortunately not supported at the moment.

toJSON

Convert R To JSON

Description

Convert an R object into a corresponding JSON object.

Lists with unnamed components are not currently supported

Usage

```
toJSON( x )
```

Arguments

x a vector or list to convert into a JSON object

Value

a string containing the JSON object

See Also

[fromJSON](#)

Examples

```
x <- list( alpha = 1:5, beta = "Bravo",
           gamma = list(a=1:3, b=NULL),
           delta = c(TRUE, FALSE) )
json <- toJSON( x )
fromJSON( json )

#named vectors are treated as JSON objects (lists)
toJSON(islands[1:4])

#data.frames must be converted into a list before converting into JSON
plot(cars, pch=2)
json_cars <- toJSON(as.list(cars))
points( data.frame( fromJSON( json_cars ) ), col="red", pch=3 )

#special R types are encoded as strings
testString <- c(1,2,3,4,NA,NaN,Inf,8,9);
toJSON(testString);
```

Index

*Topic **interface**

fromJSON, [2](#)

newJSONParser, [3](#)

toJSON, [4](#)

*Topic **package**

rjson, [4](#)

fromJSON, [2](#), [4](#)

newJSONParser, [3](#)

rjson, [4](#)

rjson-package (rjson), [4](#)

toJSON, [2](#), [3](#), [4](#)