

# Package ‘rjson’

November 1, 2009

**Version** 0.1.8

**Date** 2009-04-28

**Title** JSON for R

**Author** Alex Couture-Beil <rjson\_pkg@mofoc.ca>

**Maintainer** Alex Couture-Beil <rjson\_pkg@mofoc.ca>

**Depends** R (>= 2.5.0)

**Description** Converts R object into JSON objects and vice-versa

**License** GPL-2

**Repository** CRAN

**Date/Publication** 2009-11-01 17:34:06

## R topics documented:

fromJSON . . . . .	1
newJSONParser . . . . .	2
rjson . . . . .	3
toJSON . . . . .	4
<b>Index</b>	<b>5</b>

---

fromJSON                      *Convert JSON To R*

---

**Description**

Convert a JSON object into an R object.

**Usage**

```
fromJSON( json_str )
```

**Arguments**

json\_str            a JSON object to convert

**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}' )
```

---

newJSONParser                *Convert buffered JSON objects To R*

---

**Description**

Convert a collection of JSON objects into R objects.

**Usage**

```
newJSONParser()
```

**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 i

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
```

**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](http://json.org) for an overview of JSON.

---

`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, 1

newJSONParser, 2

toJSON, 3

\*Topic **package**

rjson, 3

fromJSON, 1, 4

newJSONParser, 2

rjson, 3

rjson-package (*rjson*), 3

toJSON, 2, 3