

# Package ‘rtson’

August 29, 2016

**Type** Package

**Title** Typed JSON

**Version** 1.3

**Date** 2015-11-22

**Author** Alexandre Maurel

**Maintainer** Alexandre Maurel <alexandre.maurel@gmail.com>

**Description** TSON, short for Typed JSON, is a binary-encoded serialization of JSON like document that support JavaScript typed data (<https://github.com/tercen/TSON>).

**License** Apache License Version 2.0

**Suggests** testthat

**Imports** R6

**URL** <https://github.com/tercen/TSON>

**BugReports** <https://github.com/tercen/TSON/issues>

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2016-08-26 20:35:52

## R topics documented:

fromTSON . . . . .	2
readTSON . . . . .	3
toTSON . . . . .	4
tson.character . . . . .	5
tson.double . . . . .	5
tson.float32.vec . . . . .	6
tson.int . . . . .	6
tson.int16.vec . . . . .	7
tson.int8.vec . . . . .	7
tson.map . . . . .	8

tson.scalar . . . . .	8
tson.uint16.vec . . . . .	9
tson.uint32.vec . . . . .	9
tson.uint8.vec . . . . .	10
writeTSON . . . . .	10

<b>Index</b>	<b>12</b>
--------------	-----------

---

fromTSON	<i>Deserialize a raw vector</i>
----------	---------------------------------

---

### Description

This function convert a raw vector into a list following TSON specification binary-encoded format.

### Usage

```
fromTSON(bytes)
```

### Arguments

bytes	A raw vector
-------	--------------

### Value

A list

### Examples

```
## Example

library(rtson)

list = list(integer=42L,
            double=42,
            bool=TRUE,
            uint8=tson.uint8.vec(c(42,0)),
            uint16=tson.uint16.vec(c(42,0)),
            uint32=tson.uint32.vec(c(42,0)),
            int8=tson.int8.vec(c(42,0)),
            int16=tson.int16.vec(c(42,0)),
            int32=as.integer(c(42,0)),
            float32=tson.float32.vec(c(0.0, 42.0)),
            float64=c(42.0,42.0),
            map=list(x=42, y=42, label="42"),
            list=list("42",42)
)

bytes = toTSON(list)
object = fromTSON(bytes)
```

---

readTSON	<i>Deserialize a connection</i>
----------	---------------------------------

---

**Description**

Read TSON specification binary-encoded format from a connection.

**Usage**

```
readTSON(con)
```

**Arguments**

con                    A connection or a raw vector

**Value**

A list

**Examples**

```
## Example

library(rtson)

list = list(integer=42L,
            double=42,
            bool=TRUE,
            uint8=tson.uint8.vec(c(42,0)),
            uint16=tson.uint16.vec(c(42,0)),
            uint32=tson.uint32.vec(c(42,0)),
            int8=tson.int8.vec(c(42,0)),
            int16=tson.int16.vec(c(42,0)),
            int32=as.integer(c(42,0)),
            float32=tson.float32.vec(c(0.0, 42.0)),
            float64=c(42.0,42.0),
            map=list(x=42, y=42, label="42"),
            list=list("42",42)
)

con = rawConnection(raw(0), "r+")
writeTSON(list, con)
bytes = rawConnectionValue(con)
close(con)
con = rawConnection(bytes, "r")
object = readTSON(con)
```

toTSON

*Serialize a list*

---

**Description**

This function convert a list into raw following TSON specification binary-encoded format.

**Usage**

```
toTSON(object)
```

**Arguments**

object            A list

**Value**

A raw vector

**Examples**

```
## Example

library(rtson)

list = list(integer=42L,
            double=42,
            bool=TRUE,
            uint8=tson.uint8.vec(c(42,0)),
            uint16=tson.uint16.vec(c(42,0)),
            uint32=tson.uint32.vec(c(42,0)),
            int8=tson.int8.vec(c(42,0)),
            int16=tson.int16.vec(c(42,0)),
            int32=as.integer(c(42,0)),
            float32=tson.float32.vec(c(0.0, 42.0)),
            float64=c(42.0,42.0),
            map=list(x=42, y=42, label="42"),
            list=list("42",42)
)

bytes = toTSON(list)
```

---

tson.character	<i>Make a tson character</i>
----------------	------------------------------

---

**Description**

Make a tson character

**Usage**

tson.character(object)

**Arguments**

object            A vector or list

**Value**

A tson character

---

tson.double	<i>Make a tson double</i>
-------------	---------------------------

---

**Description**

Make a tson double

**Usage**

tson.double(object)

**Arguments**

object            A vector or list

**Value**

A tson double

tson.float32.vec      *Make a tson float32 vector*

---

**Description**

Make a tson float32 vector

**Usage**

tson.float32.vec(object)

**Arguments**

object      A vector or list

**Value**

A tson float32 vector

---

tson.int      *Make a tson integer*

---

**Description**

Make a tson integer

**Usage**

tson.int(object)

**Arguments**

object      A vector or list

**Value**

A tson integer

---

tson.int16.vec      *Make a tson int16 vector*

---

**Description**

Make a tson int16 vector

**Usage**

tson.int16.vec(object)

**Arguments**

object      A vector or list

**Value**

A tson int16 vector

---

tson.int8.vec      *Make a tson int8 vector*

---

**Description**

Make a tson int8 vector

**Usage**

tson.int8.vec(object)

**Arguments**

object      A vector or list

**Value**

A tson int8 vector

---

`tson.map`*Make a tson map*

---

**Description**

Required to generate empty map.

**Usage**

```
tson.map(object)
```

**Arguments**

`object`            A vector or list

**Value**

A tson map

---

`tson.scalar`*Make a tson scalar (ie: singleton)*

---

**Description**

Make a tson scalar (ie: singleton)

**Usage**

```
tson.scalar(object)
```

**Arguments**

`object`            A vector or list

**Value**

A tson scalar



---

tson.uint16.vec      *Make a tson uint16 vector*

---

**Description**

Make a tson uint16 vector

**Usage**

tson.uint16.vec(object)

**Arguments**

object      A vector or list

**Value**

A tson uint16 vector

---

tson.uint32.vec      *Make a tson uint32 vector*

---

**Description**

Make a tson uint32 vector

**Usage**

tson.uint32.vec(object)

**Arguments**

object      A vector or list

**Value**

A tson uint32 vector

tson.uint8.vec      *Make a tson uint8 vector*

---

**Description**

Make a tson uint8 vector

**Usage**

```
tson.uint8.vec(object)
```

**Arguments**

object      A vector or list

**Value**

A tson uint8 vector

---

writeTSON      *Serialize a list*

---

**Description**

Write TSON specification binary-encoded format to a connection.

**Usage**

```
writeTSON(object, con)
```

**Arguments**

object      A list  
con      A connection

**Value**

invisibly NULL

**Examples**

```
## Example

library(rtson)

list = list(integer=42L,
            double=42,
            bool=TRUE,
            uint8=tson.uint8.vec(c(42,0)),
            uint16=tson.uint16.vec(c(42,0)),
            uint32=tson.uint32.vec(c(42,0)),
            int8=tson.int8.vec(c(42,0)),
            int16=tson.int16.vec(c(42,0)),
            int32=as.integer(c(42,0)),
            float32=tson.float32.vec(c(0.0, 42.0)),
            float64=c(42.0,42.0),
            map=list(x=42, y=42, label="42"),
            list=list("42",42)
)

con = rawConnection(raw(0), "r+")
writeTSON(list, con)
bytes = rawConnectionValue(con)
close(con)
con = rawConnection(bytes, "r")
object = readTSON(con)
```

# Index

[fromTSON](#), [2](#)

[readTSON](#), [3](#)

[toTSON](#), [4](#)

[tson.character](#), [5](#)

[tson.double](#), [5](#)

[tson.float32.vec](#), [6](#)

[tson.int](#), [6](#)

[tson.int16.vec](#), [7](#)

[tson.int8.vec](#), [7](#)

[tson.map](#), [8](#)

[tson.scalar](#), [8](#)

[tson.uint16.vec](#), [9](#)

[tson.uint32.vec](#), [9](#)

[tson.uint8.vec](#), [10](#)

[writeTSON](#), [10](#)