

# Package: jsonstat (via r-universe)

October 21, 2024

**Type** Package

**Title** Interface to 'JSON-stat'

**Version** 0.0.2

**Author** Kim Seonghyun [aut, cre]

**Maintainer** Kim Seonghyun <zedoul@gmail.com>

**Description** Interface to 'JSON-stat' <<https://json-stat.org/>>, a simple lightweight 'JSON' format for data dissemination.

**URL** <https://github.com/zedoul/jsonstat>

**BugReports** <https://github.com/zedoul/jsonstat/issues>

**Depends** R (>= 3.3.0)

**Imports** cli, dplyr, jsonlite, rlang

**Suggests** knitr, testthat

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.0.1

**VignetteBuilder** knitr

**NeedsCompilation** no

**Date/Publication** 2018-08-17 20:40:03 UTC

**Repository** <https://zedoul.r-universe.dev>

**RemoteUrl** <https://github.com/cran/jsonstat>

**RemoteRef** HEAD

**RemoteSha** 92128a9e29a1325e9bfb697211fb34d108e47748

## Contents

as.collection . . . . .	2
as.dataset . . . . .	3
autounbox . . . . .	3
compress_plan . . . . .	4
dimension . . . . .	4
dimension.jsonstat.compress.plan . . . . .	5
galicia . . . . .	6
jsonstat . . . . .	6
toJSON . . . . .	7
toJSON.jsonstat.collection . . . . .	7
toJSON.jsonstat.dataset . . . . .	8
<b>Index</b>	<b>10</b>

---

as.collection	<i>Convert data set into JSON-stat collection</i>
---------------	---

---

### Description

Convert data set into JSON-stat collection

### Usage

```
as.collection(..., label = "", href = "", src = NULL, extension = NULL,
  updated = NULL)
```

### Arguments

...	a named list, the function creates a collection based on user input like the same way as 'list'.
label	label of dataset
href	href of dataset, "" by default
src	source of dataset, NULL by default
extension	user data, NULL by default
updated	a timestamp for data, NULL by default

---

as.dataset	<i>Convert data frame into JSON-stat dataset</i>
------------	--

---

**Description**

Convert data frame into JSON-stat dataset

**Usage**

```
as.dataset(x, .plan, label, href = "", src = NULL, extension = NULL,
          updated = NULL)
```

**Arguments**

x	data frame
.plan	compress plan
label	label of dataset
href	href of dataset, "" by default
src	source of dataset, NULL by default
extension	user data, NULL by default
updated	a timestamp for data, NULL by default

---

autounbox	<i>Unbox list object</i>
-----------	--------------------------

---

**Description**

This function marks atomic vectors in given list as a singleton, so that it will not turn into an 'array' when encoded into JSON.

**Usage**

```
autounbox(.list)
```

**Arguments**

.list	a list contains atomic vectors
-------	--------------------------------

---

compress_plan	<i>compress plan</i>
---------------	----------------------

---

**Description**

This function constructs a plan to compress data frame into jsonstat

**Usage**

```
compress_plan(dim_name, role_name = c("time", "geo", "metric",
  "classification", "value"), label = NULL)
```

**Arguments**

dim_name	name of dimension
role_name	role of dimension
label	label of dimension

**Value**

jsonstat.compress.plan object

**Examples**

```
library(jsonstat)

.plan <- compress_plan("place.of.birth", "geo", "Place of Birth")
```

---

dimension	<i>Create dimension object</i>
-----------	--------------------------------

---

**Description**

Create dimension object

**Usage**

```
dimension(.plan, dim_name, role_name, label)
```

**Arguments**

.plan	jsonstat.compress.plan object
dim_name	name of dimension
role_name	role of dimension
label	label of dimension

**Value**

jsonstat.compress.plan object

---

dimension.jsonstat.compress.plan  
*dimension*

---

**Description**

This function adds another dimension into compress plan

**Usage**

```
## S3 method for class 'jsonstat.compress.plan'  
dimension(.plan, dim_name,  
  role_name = c("time", "geo", "metric", "classification", "value"),  
  label = NULL)
```

**Arguments**

.plan	jsonstat.compress.plan object
dim_name	name of dimension
role_name	role of dimension
label	label of dimension

**Value**

jsonstat.compress.plan object

**Examples**

```
library(jsonstat)  
  
.plan <- compress_plan("place.of.birth", "geo", "Place of Birth")  
.plan <- dimension(.plan, "age.group", "classification", "Age Group")
```

---

galicia

*Galicia data*

---

### Description

Galicia data

### Usage

```
galicia
```

### Format

a data frame

### Examples

```
galicia
```

---

jsonstat

*jsonstat: R package for JSON-stat*

---

### Description

jsonstat provides useful functions to work with JSON-stat format.

### Author(s)

**Maintainer:** Kim Seonghyun <zedoul@gmail.com>

### See Also

Useful links:

- <https://github.com/zedoul/jsonstat>
- Report bugs at <https://github.com/zedoul/jsonstat/issues>

---

toJSON	<i>Convert JSON-stat object into JSON-stat JSON string</i>
--------	--

---

**Description**

Convert JSON-stat object into JSON-stat JSON string

**Usage**

```
toJSON(x, auto_unbox)
```

**Arguments**

x	JSON-stat object
auto_unbox	this flag marks atomic vectors in given list as a singleton, so it will not turn into an 'array' when encoded into JSON. FALSE by default.

**Value**

JSON output

---

toJSON.jsonstat.collection	<i>Convert JSON-stat collection into JSON</i>
----------------------------	---

---

**Description**

Convert JSON-stat collection into JSON

**Usage**

```
## S3 method for class 'jsonstat.collection'  
toJSON(x, auto_unbox = F)
```

**Arguments**

x	JSON-stat collection
auto_unbox	this flag marks atomic vectors in given list as a singleton, so it will not turn into an 'array' when encoded into JSON. FALSE by default.

**Value**

JSON output

**Examples**

```

library(jsonstat)
library(dplyr)

.plan <- compress_plan("place.of.birth", "geo", "Place of Birth") %>%
  dimension("age.group", "classification", "Age Group") %>%
  dimension("gender", "classification", "Gender") %>%
  dimension("year", "time", "Year") %>%
  dimension("province.of.residence", "geo", "Province of Residence") %>%
  dimension("concept", "metric", "Concept") %>%
  dimension("value", "value", "value")

.dataset <- as.dataset(galicia, .plan,
  label = paste("Population by province of residence,",
    "place of birth, age, gender and year",
    "in Galicia"),
  href = "https://github.com/zedoul/jsonstat")

.collection <- as.collection(.dataset, label = "Comparison",
  href = "https://github.com/zedoul/jsonstat")

.jsonstat <- toJSON(.collection)

```

---

```
toJSON.jsonstat.dataset
```

*Convert JSON-stat dataset into JSON*

---

**Description**

Convert JSON-stat dataset into JSON

**Usage**

```
## S3 method for class 'jsonstat.dataset'
toJSON(x, auto_unbox = F)
```

**Arguments**

x	JSON-stat dataset
auto_unbox	this flag marks atomic vectors in given list as a singleton, so it will not turn into an 'array' when encoded into JSON. FALSE by default.

**Value**

JSON output



**Examples**

```
library(jsonstat)
library(dplyr)

.plan <- compress_plan("place.of.birth", "geo", "Place of Birth") %>%
  dimension("age.group", "classification", "Age Group") %>%
  dimension("gender", "classification", "Gender") %>%
  dimension("year", "time", "Year") %>%
  dimension("province.of.residence", "geo", "Province of Residence") %>%
  dimension("concept", "metric", "Concept") %>%
  dimension("value", "value", "value")

.dataset <- as.dataset(galicia, .plan,
  label = paste("Population by province of residence,",
    "place of birth, age, gender and year",
    "in Galicia"),
  href = "https://github.com/zedoul/jsonstat")

.jsonstat <- toJSON(.dataset)
```

# Index

`as.collection`, [2](#)

`as.dataset`, [3](#)

`autounbox`, [3](#)

`compress_plan`, [4](#)

`dimension`, [4](#)

`dimension.jsonstat.compress.plan`, [5](#)

`galicia`, [6](#)

`jsonstat`, [6](#)

`jsonstat-package (jsonstat)`, [6](#)

`toJSON`, [7](#)

`toJSON.jsonstat.collection`, [7](#)

`toJSON.jsonstat.dataset`, [8](#)