Package 'cereal'

June 9, 2023

Title Serialize 'vctrs' Objects to 'JSON'

Version 0.1.0

Description The 'vctrs' package provides a concept of vector prototype
that can be especially useful when deploying models and code. Serialize
these object prototypes to 'JSON' so they can be used to check and coerce
data in production systems, and deserialize 'JSON' back to the correct
object prototypes.

License MIT + file LICENSE

URL <https://r-lib.github.io/cereal/>, <https://github.com/r-lib/cereal/>

BugReports <https://github.com/r-lib/cereal/issues>

Depends R (>= 3.6)

Imports jsonlite, rlang, tibble, vctrs

Suggests covr, testthat (>= 3.0.0), withr

Config/Needs/website tidyverse/tidytemplate

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.2.3

Language en-US

NeedsCompilation no

Author Julia Silge [cre, aut] (<<https://orcid.org/0000-0002-3671-836X>>),
Davis Vaughan [aut],
Posit Software, PBC [cph, fnd]

Maintainer Julia Silge <julia.silge@posit.co>

Repository CRAN

Date/Publication 2023-06-09 14:00:26 UTC

R topics documented:

---

cereal_decode   *Convert a JSON-serialized prototype to a vctrs prototype*

Description

Create a `vctrs::vec_ptype()` from a JSON-serialized prototype created with `cereal_encode()`.

Usage

```
cereal_decode(x)
```

Arguments

x   An object with class "cereal_*", such as "cereal_integer" or "cereal_factor"

Value

A vector of zero length, such as `integer()` or `vctrs::new_factor()`

Examples

```
cereal_decode(structure(list(), class = "cereal_integer"))
cereal_decode(structure(list(), class = "cereal_Date"))
```

---

cereal_details   *Find needed details for vctrs prototype*

Description

Find needed details for vctrs prototype

Usage

```
cereal_details(x)
```

Arguments

x   A vector

Value

A list

Examples

```
cereal_details(factor(letters[1:5], labels = "letter"))
cereal_details(factor(LETTERS[3:1], ordered = TRUE))
cereal_details(as.POSIXct("2023-01-01", tz = "America/New_York"))
```

---

cereal_encode   *Encode a vector as JSON*

Description

Create a list encoding the vctrs prototype (metadata) that can be stored as JSON.

Usage

```
cereal_encode(x)
```

Arguments

x   A vector

Details

Use the `digits` option to specify how many digits after the decimal point to record in JSON, for example via `withr::local_options()`.

Value

A list that can be converted to JSON with `jsonlite::toJSON()`

See Also

`vctrs::vec_ptype()`, `cereal_decode()`

Examples

```
cereal_encode(1:10)
cereal_encode(Sys.Date())
cereal_encode(sample(letters, 5))
cereal_encode(factor(letters[1:5], labels = "letter"))
cereal_encode(factor(LETTERS[3:1], ordered = TRUE))

## you can encode a ptype as well:
ptype <- vctrs::vec_ptype(factor(LETTERS[3:1], ordered = TRUE))
## but "example" is NULL:
cereal_encode(ptype)
```

---

cereal_to_json   *Serialize and deserialize the prototype of a data frame to JSON*

Description

The function `cereal_to_json()` serializes the vctrs prototype of a data frame to JSON, and the
function `cereal_from_json()` deserializes from a JSON prototype back to a vctrs prototype.

Usage

```
cereal_to_json(data)

cereal_from_json(x)
```

Arguments

data   A data frame
x   A JSON string

Value

`cereal_to_json()` returns a JSON string like `jsonlite::toJSON()`, and `cereal_from_json()`
returns a vctrs ptype, like `vctrs::vec_ptype()`.

See Also

`cereal_encode()`, `cereal_decode()`

Examples

```
df <- tibble::tibble(
    a = 1,
    b = 2L,
    c = Sys.Date(),
    d = as.POSIXct("2019-01-01", tz = "America/New_York"),
    e = "x",
    f = factor("blue", levels = c("blue", "green", "red")),
    g = ordered("small", levels = c("small", "medium", "large"))
)

json <- cereal_to_json(df)
json

str(cereal_from_json(json))
## same as:
str(vctrs::vec_ptype(df))
```

Index