

Package: sakura (via r-universe)

March 4, 2025

Type Package

Title Extension to R Serialization

Version 0.1.0.9000

Description Extends the functionality of R serialization by augmenting the built-in reference hook system. This enhanced implementation allows an integrated single-pass operation that combines R serialization with third-party serialization methods. Facilitates the serialization of even complex R objects, which contain non-system reference objects, such as those accessed via external pointers, to enable their use in parallel and distributed computing.

License MIT + file LICENSE

BugReports <https://github.com/shikokuchuo/sakura/issues>

URL <https://shikokuchuo.net/sakura/>,
<https://github.com/shikokuchuo/sakura/>

Encoding UTF-8

Depends R (>= 3.5)

Suggests arrow

RoxygenNote 7.3.2

Config/build/compilation-database true

Repository <https://shikokuchuo.r-universe.dev>

RemoteUrl <https://github.com/shikokuchuo/sakura>

RemoteRef HEAD

RemoteSha 60946b639ecd5d609ce453bb8bfa2475417e8939

Contents

sakura-package	2
serialize	2
serial_config	3

Index	5
--------------	----------

sakura-package

sakura: Extension to R Serialization

Description

Extends the functionality of R serialization by augmenting the built-in reference hook system. This enhanced implementation allows an integrated single-pass operation that combines R serialization with third-party serialization methods. Facilitates the serialization of even complex R objects, which contain non-system reference objects, such as those accessed via external pointers, to enable their use in parallel and distributed computing.

Author(s)

Charlie Gao <charlie.gao@shikokuchuo.net> ([ORCID](#))

See Also

Useful links:

- <https://shikokuchuo.net/sakura/>
- <https://github.com/shikokuchuo/sakura/>
- Report bugs at <https://github.com/shikokuchuo/sakura/issues>

serialize

Serialize

Description

An extension of R native serialization using the 'refhook' system for custom serialization and unserialization of non-system reference objects.

Usage

```
serialize(x, hook = NULL)
```

```
unserialize(x, hook = NULL)
```

Arguments

x an object.
hook [default NULL] optionally, a configuration returned by [serial_config](#).

Value

For `serialize`: a raw vector. For `unserialize`: the unserialized object.

Examples

```
vec <- serialize(data.frame())
vec
unserialize(vec)

obj <- list(arrow::as_arrow_table(iris), arrow::as_arrow_table(mtcars))
cfg <- serial_config(
  "ArrowTabular",
  arrow::write_to_raw,
  function(x) arrow::read_ipc_stream(x, as_data_frame = FALSE)
)
raw <- serialize(obj, cfg)
unserialize(raw, cfg)
```

serial_config

Create Serialization Configuration

Description

Returns a serialization configuration for custom serialization and unserialization of non-system reference objects, using the 'refhook' system of R native serialization. This allows their use across different R sessions.

Usage

```
serial_config(class, sfunc, ufunc, vec = FALSE)
```

Arguments

class	character string of the class of object custom serialization functions are applied to, e.g. 'ArrowTabular' or 'torch_tensor'.
sfunc	a function that accepts a reference object inheriting from 'class' (or a list of such objects) and returns a raw vector.
ufunc	a function that accepts a raw vector and returns a reference object (or list of such objects).
vec	[default FALSE] whether or not the serialization functions are vectorized. If FALSE, they should accept and return reference objects individually e.g. <code>arrow::write_to_raw</code> and <code>arrow::read_ipc_stream</code> . If TRUE, they should accept and return a list of reference objects, e.g. <code>torch::torch_serialize</code> and <code>torch::torch_load</code> .

Value

A pairlist comprising the configuration. This may be provided to the 'hook' argument of [serialize](#) and [unserialize](#).

Examples

```
serial_config("test_class", base::serialize, base::unserialize)
```

Index

sakura (sakura-package), [2](#)
sakura-package, [2](#)
serial_config, [2](#), [3](#)
serialize, [2](#), [3](#)

unserialize, [3](#)
unserialize (serialize), [2](#)