Package: SafeMapper (via r-universe)

May 25, 2026

Type Package

Title Fault-Tolerant Functional Programming with Automatic Checkpointing

Version 1.0.0

Description Provides drop-in replacements for 'purrr' and 'furrr' mapping functions with built-in fault tolerance, automatic checkpointing, and seamless recovery capabilities. When long-running computations are interrupted due to errors, system crashes, or other failures, simply re-run the same code to automatically resume from the last checkpoint. Ideal for large-scale data processing, API calls, web scraping, and other time-intensive operations where reliability is critical.

License MIT + file LICENSE

Encoding UTF-8

URL https://github.com/Zaoqu-Liu/SafeMapper

BugReports https://github.com/Zaoqu-Liu/SafeMapper/issues

RoxygenNote 7.3.3

Depends R (>= 3.5.0)

Imports purrr (>= 0.3.0), digest, tools

Suggests furrr (>= 0.2.0), future, testthat (>= 3.0.0), knitr, rmarkdown, pkgdown

VignetteBuilder knitr

Config/testthat/edition 3

Repository https://zaoqu-liu.r-universe.dev

Date/Publication 2026-01-22 20:17:38 UTC

RemoteUrl https://github.com/Zaoqu-Liu/SafeMapper

RemoteRef main

RemoteSha 5ffddad8a0862180dd3306e8e7abfa79a332c231

Contents

`s_clean_sessions` *Clean Old SafeMapper Sessions*

Description

Removes old checkpoint files to free up disk space. Can filter by age, specific session IDs, or status.

Usage

```
s_clean_sessions(older_than_days = 7, session_ids = NULL, status_filter = NULL)
```

Arguments

`older_than_days`	Integer. Remove sessions older than this many days.
`session_ids`	Character vector. Specific session IDs to remove.
`status_filter`	Character vector. Remove only sessions with these statuses ("in_progress", "failed", "corrupted").

Value

Integer. Number of files removed (invisible).

Examples

```
# Clean sessions older than 7 days
s_clean_sessions(older_than_days = 7)

# Clean only failed sessions
s_clean_sessions(status_filter = "failed")

# Clean specific sessions
```

```
s_clean_sessions(session_ids = c("session1", "session2"))
```

`s_configure` *Configure SafeMapper Settings*

Description

Optionally customize SafeMapper behavior. This function is NOT required for basic usage - SafeMapper works out of the box with sensible defaults.

Usage

```
s_configure(batch_size = 100L, retry_attempts = 3L, auto_recover = TRUE)
```

Arguments

`batch_size`	Integer. Number of items to process per batch before checkpointing. Smaller values provide more frequent saves but may slow processing. Default is 100.
`retry_attempts`	Integer. Number of retry attempts for failed batches. Default is 3.
`auto_recover`	Logical. Whether to automatically resume from checkpoints when restarting operations. Default is TRUE.

Value

Invisible list of current configuration settings.

Examples

```
# Basic usage - no configuration needed!
# result <- s_map(1:100, slow_function)

# Optional: customize for large operations
s_configure(batch_size = 50)

# Optional: customize for unstable operations
s_configure(retry_attempts = 5)
```

s_future_map *Safe Future Map - Parallel with Auto-Recovery*

Description

Parallel mapping with automatic checkpointing. Requires the furrr package.

Usage

```
s_future_map(
  .x,
  .f,
  ...,
  .options = NULL,
  .env_globals = parent.frame(),
  .progress = FALSE,
  .id = NULL,
  .session_id = NULL
)

s_future_map_chr(
  .x,
  .f,
  ...,
  .options = NULL,
  .env_globals = parent.frame(),
  .progress = FALSE,
  .id = NULL,
  .session_id = NULL
)

s_future_map_dbl(
  .x,
  .f,
  ...,
  .options = NULL,
  .env_globals = parent.frame(),
  .progress = FALSE,
  .id = NULL,
  .session_id = NULL
)

s_future_map_int(
  .x,
  .f,
  ...,
  .options = NULL,
```

```
  .env_globals = parent.frame(),
  .progress = FALSE,
  .id = NULL,
  .session_id = NULL
)

s_future_map_lgl(
  .x,
  .f,
  ...,
  .options = NULL,
  .env_globals = parent.frame(),
  .progress = FALSE,
  .id = NULL,
  .session_id = NULL
)

s_future_map_dfr(
  .x,
  .f,
  ...,
  .options = NULL,
  .env_globals = parent.frame(),
  .progress = FALSE,
  .id = NULL,
  .session_id = NULL
)

s_future_map_dfc(
  .x,
  .f,
  ...,
  .options = NULL,
  .env_globals = parent.frame(),
  .progress = FALSE,
  .id = NULL,
  .session_id = NULL
)
```

Arguments

.x	A list or atomic vector.
.f	A function, formula, or vector.
...	Additional arguments passed to .f.
.options	A furrr_options object (NULL uses defaults).
.env_globals	The environment to look for globals.
.progress	A single logical.

.id	Optional name for ID column (dfr/dfc variants).
.session_id	Character. Optional session ID.

Value

A list.

Examples

```
## Not run:
library(future)
plan(multisession)
s_future_map(1:100, ~ .x^2)

## End(Not run)
```

`s_future_map2` *Safe Future Map2 - Parallel Two-Input with Auto-Recovery*

Description

Safe Future Map2 - Parallel Two-Input with Auto-Recovery

Usage

```
s_future_map2(
  .x,
  .y,
  .f,
  ...,
  .options = NULL,
  .env_globals = parent.frame(),
  .progress = FALSE,
  .id = NULL,
  .session_id = NULL
)

s_future_map2_chr(
  .x,
  .y,
  .f,
  ...,
  .options = NULL,
  .env_globals = parent.frame(),
  .progress = FALSE,
  .id = NULL,
  .session_id = NULL
)
```

```
s_future_map2_dbl(
  .x,
  .y,
  .f,
  ...,
  .options = NULL,
  .env_globals = parent.frame(),
  .progress = FALSE,
  .id = NULL,
  .session_id = NULL
)

s_future_map2_int(
  .x,
  .y,
  .f,
  ...,
  .options = NULL,
  .env_globals = parent.frame(),
  .progress = FALSE,
  .id = NULL,
  .session_id = NULL
)

s_future_map2_lgl(
  .x,
  .y,
  .f,
  ...,
  .options = NULL,
  .env_globals = parent.frame(),
  .progress = FALSE,
  .id = NULL,
  .session_id = NULL
)

s_future_map2_dfr(
  .x,
  .y,
  .f,
  ...,
  .options = NULL,
  .env_globals = parent.frame(),
  .progress = FALSE,
  .id = NULL,
  .session_id = NULL
)
```

```
s_future_map2_dfc(
  .x,
  .y,
  .f,
  ...,
  .options = NULL,
  .env_globals = parent.frame(),
  .progress = FALSE,
  .id = NULL,
  .session_id = NULL
)
```

Arguments

.x, .y	Vectors of the same length.
.f	A function, formula, or vector.
...	Additional arguments passed to .f.
.options	A furrr_options object.
.env_globals	The environment to look for globals.
.progress	A single logical.
.id	Optional name for ID column (dfr/dfc variants).
.session_id	Character. Optional session ID.

Value

A list.

---

s_imap *Safe IMap - Drop-in Replacement for purrr::imap with Auto-Recovery*

Description

Safe IMap - Drop-in Replacement for purrr::imap with Auto-Recovery

Usage

```
s_imap(.x, .f, ..., .session_id = NULL)

s_imap_chr(.x, .f, ..., .session_id = NULL)

s_future_imap(
  .x,
  .f,
  ...,
```

```
  .options = NULL,
  .env_globals = parent.frame(),
  .progress = FALSE,
  .session_id = NULL
)
```

Arguments

.x	A list or atomic vector.
.f	A function, formula, or vector.
...	Additional arguments passed to .f.
.session_id	Character. Optional session ID.
.options	A furrr_options object (NULL uses defaults).
.env_globals	The environment to look for globals.
.progress	A single logical.

Value

A list.

A character vector.

---

s_map *Safe Map - Drop-in Replacement for purrr::map with Auto-Recovery*

Description

Apply a function to each element of a list or vector with automatic checkpointing and recovery. If interrupted, simply run the same code again to resume from where it left off.

Usage

```
s_map(.x, .f, ..., .id = NULL, .session_id = NULL)

s_map_chr(.x, .f, ..., .id = NULL, .session_id = NULL)

s_map_dbl(.x, .f, ..., .id = NULL, .session_id = NULL)

s_map_int(.x, .f, ..., .id = NULL, .session_id = NULL)

s_map_lgl(.x, .f, ..., .id = NULL, .session_id = NULL)

s_map_dfr(.x, .f, ..., .id = NULL, .session_id = NULL)

s_map_dfc(.x, .f, ..., .id = NULL, .session_id = NULL)
```

Arguments

.x	A list or atomic vector to map over.
.f	A function, formula, or vector.
...	Additional arguments passed to .f.
.id	Either a string or NULL (used for dfr/dfc variants).
.session_id	Character. Optional session ID for this operation. If NULL (default), a session ID is automatically generated from the input data, enabling seamless recovery without user intervention.

Value

A list, same as purrr::map.

A character vector.

A double vector.

An integer vector.

A logical vector.

A data frame (row bind).

A data frame (column bind).

Examples

```
# Basic usage - identical to purrr::map
result <- s_map(1:10, ~ .x^2)
```

---

s_map2 *Safe Map2 - Drop-in Replacement for purrr::map2 with Auto-Recovery*

Description

Safe Map2 - Drop-in Replacement for purrr::map2 with Auto-Recovery

Usage

```
s_map2(.x, .y, .f, ..., .id = NULL, .session_id = NULL)

s_map2_chr(.x, .y, .f, ..., .id = NULL, .session_id = NULL)

s_map2_dbl(.x, .y, .f, ..., .id = NULL, .session_id = NULL)

s_map2_int(.x, .y, .f, ..., .id = NULL, .session_id = NULL)

s_map2_lgl(.x, .y, .f, ..., .id = NULL, .session_id = NULL)
```

```
s_map2_dfr(.x, .y, .f, ..., .id = NULL, .session_id = NULL)

s_map2_dfc(.x, .y, .f, ..., .id = NULL, .session_id = NULL)
```

Arguments

.x, .y	Vectors of the same length.
.f	A function, formula, or vector.
...	Additional arguments passed to .f.
.id	Either a string or NULL (used for dfr/dfc variants).
.session_id	Character. Optional session ID.

Value

A list.

Examples

```
s_map2(1:5, 6:10, `+`)
```

---

s_pmap *Safe PMap - Drop-in Replacement for purrr::pmap with Auto-Recovery*

Description

Safe PMap - Drop-in Replacement for purrr::pmap with Auto-Recovery

Usage

```
s_pmap(.l, .f, ..., .session_id = NULL)

s_future_pmap(
  .l,
  .f,
  ...,
  .options = NULL,
  .env_globals = parent.frame(),
  .progress = FALSE,
  .session_id = NULL
)
```

Arguments

.l	A list of lists/vectors to map over.
.f	A function, formula, or vector.
...	Additional arguments passed to .f.
.session_id	Character. Optional session ID.
.options	A furrr_options object (NULL uses defaults).
.env_globals	The environment to look for globals.
.progress	A single logical for progress bar.

Value

A list.

---

s_possibly *Safe Possibly - Wrap Function to Return Default on Error*

Description

Drop-in replacement for purrr::possibly that returns a default value when an error occurs instead of throwing the error.

Usage

```
s_possibly(.f, otherwise, quiet = TRUE)
```

Arguments

.f	A function to wrap for safe execution.
otherwise	Default return value when an error occurs.
quiet	Logical. Hide errors from console if TRUE.

Value

A function that returns the result or the default value.

Examples

```
safe_log <- s_possibly(log, otherwise = NA)
safe_log(10) # Returns 2.30
safe_log("a") # Returns NA
```

s_quietly *Safe Quietly - Wrap Function to Capture Side Effects*

Description

Drop-in replacement for purrr::quietly that captures all side effects (output, messages, warnings) along with the result.

Usage

```
s_quietly(.f)
```

Arguments

.f	A function to wrap for quiet execution.

Value

A function that returns a list with 'result', 'output', 'warnings', and 'messages'.

Examples

```
quiet_summary <- s_quietly(summary)
result <- quiet_summary(cars)
# result$result contains the summary
# result$output contains any printed output
```

s_safely *Safe Safely - Wrap Function to Capture Errors*

Description

Drop-in replacement for purrr::safely that captures errors and returns them in a structured format.

Usage

```
s_safely(.f, otherwise = NULL, quiet = TRUE)
```

Arguments

.f	A function to wrap for safe execution.
otherwise	Default return value when an error occurs.
quiet	Logical. Hide errors from console if TRUE.

Value

A function that returns a list with 'result' and 'error' components.

Examples

```
safe_log <- s_safely(log)
safe_log(10) # Returns list(result = 2.30, error = NULL)
safe_log("a") # Returns list(result = NULL, error = <error>)
```

---

s_walk *Safe Walk - Drop-in Replacement for purrr::walk with Auto-Recovery*

Description

Safe Walk - Drop-in Replacement for purrr::walk with Auto-Recovery

Usage

```
s_walk(.x, .f, ..., .session_id = NULL)

s_walk2(.x, .y, .f, ..., .session_id = NULL)

s_future_walk(
  .x,
  .f,
  ...,
  .options = NULL,
  .env_globals = parent.frame(),
  .progress = FALSE,
  .session_id = NULL
)

s_future_walk2(
  .x,
  .y,
  .f,
  ...,
  .options = NULL,
  .env_globals = parent.frame(),
  .progress = FALSE,
  .session_id = NULL
)
```

Arguments

.x	A list or atomic vector.
.f	A function, formula, or vector.
...	Additional arguments passed to .f.
.session_id	Character. Optional session ID.
.y	A list or atomic vector (same length as .x).
.options	A furrr_options object (NULL uses defaults).
.env_globals	The environment to look for globals.
.progress	A single logical.

Value

Invisibly returns .x.

Index