

Package ‘SafeMapper’

November 7, 2025

Type Package

Title Fault-Tolerant and Resumable Functional Programming with Automatic Recovery

Version 1.0.0

Author Zaoqu Liu <liuzaoqu@163.com>

Maintainer Zaoqu Liu <liuzaoqu@163.com>

Description SafeMapper provides drop-in replacements for purrr and furrr mapping functions with built-in fault tolerance, automatic checkpointing, and seamless recovery capabilities. It extends the popular purrr and furrr packages by adding robust error handling and progress persistence. SafeMapper solves the critical problem of losing computation progress when long-running mapping operations fail due to network issues, memory errors, system crashes, or other interruptions. The package automatically saves progress at configurable intervals and can resume from the exact point of failure, making it indispensable for large-scale data processing, API calls, web scraping, machine learning workflows, and other time-intensive operations where reliability and resumability are paramount.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenVersion 7.3.2

RoxygenNote 7.3.2

Depends R (>= 3.5.0)

Imports purrr (>= 0.3.0),
furrr (>= 0.2.0),
future (>= 1.20.0),
tools

Suggests testthat,
knitr,
rmarkdown

VignetteBuilder knitr

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

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

Contents

s_clean_sessions	2
s_configure	3
s_debug_session	4
s_future_map	4
s_future_map2	6
s_imap	9
s_list_sessions	9
s_map	10
s_map2	11
s_pmap	12
s_possibly	13
s_quietly	13
s_recover_session	14
s_safely	14
s_session_stats	15
s_version	16
s_walk	16
Index	18

s_clean_sessions	<i>Clean Old SafeMapper Sessions</i>
------------------	--------------------------------------

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 Session***Description**

Set global configuration for safe mapping operations with automatic recovery. This function allows you to customize batch processing, caching, and retry behavior.

Usage

```
s_configure(
  session_id = NULL,
  batch_size = 50L,
  cache_dir = ".safe_purrr_cache",
  retry_attempts = 3L,
  auto_recover = TRUE
)
```

Arguments

session_id	Character. Unique session identifier for recovery. If NULL, auto-generated based on timestamp and operation.
batch_size	Integer. Number of items to process per batch before checkpointing. Smaller values provide more frequent saves but may slow processing.
cache_dir	Character. Directory path for storing checkpoint files.
retry_attempts	Integer. Number of retry attempts for failed batches.
auto_recover	Logical. Whether to automatically resume from checkpoints when restarting operations.

Value

Invisible list of current configuration settings.

Examples

```
# Basic configuration
s_configure()

# Custom configuration for large operations
s_configure(
```

```
        batch_size = 100,
        cache_dir = "my_cache",
        retry_attempts = 5
    )

    # Configuration for specific session
    s_configure(session_id = "data_processing_2024")
```

s_debug_session	<i>Debug a Specific SafeMapper Session</i>
-----------------	--

Description

Provides detailed diagnostic information about a specific session, including progress, errors, and suggested fixes.

Usage

```
s_debug_session(session_id)
```

Arguments

session_id Character. Session ID to debug.

Examples

```
# Debug a failed session
s_debug_session("map_20240101_120000_abcd1234")
```

s_future_map	<i>Safe Future Map - Parallel with Auto-Recovery</i>
--------------	--

Description

Safe Future Map - Parallel with Auto-Recovery

Usage

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

```
s_future_map_chr(  
  .x,  
  .f,  
  ...,  
  .options = furrr::furrr_options(),  
  .env_globals = parent.frame(),  
  .progress = FALSE,  
  .id = NULL,  
  .session_id = NULL  
)  
  
s_future_map_dbl(  
  .x,  
  .f,  
  ...,  
  .options = furrr::furrr_options(),  
  .env_globals = parent.frame(),  
  .progress = FALSE,  
  .id = NULL,  
  .session_id = NULL  
)  
  
s_future_map_int(  
  .x,  
  .f,  
  ...,  
  .options = furrr::furrr_options(),  
  .env_globals = parent.frame(),  
  .progress = FALSE,  
  .id = NULL,  
  .session_id = NULL  
)  
  
s_future_map_lgl(  
  .x,  
  .f,  
  ...,  
  .options = furrr::furrr_options(),  
  .env_globals = parent.frame(),  
  .progress = FALSE,  
  .id = NULL,  
  .session_id = NULL  
)  
  
s_future_map_dfr(  
  .x,  
  .f,  
  ...,  
  .options = furrr::furrr_options(),  
  .env_globals = parent.frame(),  
  .progress = FALSE,  
  .id = NULL,  
  .session_id = NULL  
)
```

```

    .session_id = NULL
  )

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

```

Arguments

<code>.x</code>	A list or atomic vector.
<code>.f</code>	A function, formula, or vector.
<code>...</code>	Additional arguments passed to <code>.f</code> .
<code>.options</code>	A <code>furrr_options</code> object.
<code>.env_globals</code>	The environment to look for globals.
<code>.progress</code>	A single logical.
<code>.session_id</code>	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 = furrr::furrr_options(),  
  .env_globals = parent.frame(),  
  .progress = FALSE,  
  .id = NULL,  
  .session_id = NULL  
)  
  
s_future_map2_chr(  
  .x,  
  .y,  
  .f,  
  ...,  
  .options = furrr::furrr_options(),  
  .env_globals = parent.frame(),  
  .progress = FALSE,  
  .id = NULL,  
  .session_id = NULL  
)  
  
s_future_map2_dbl(  
  .x,  
  .y,  
  .f,  
  ...,  
  .options = furrr::furrr_options(),  
  .env_globals = parent.frame(),  
  .progress = FALSE,  
  .id = NULL,  
  .session_id = NULL  
)  
  
s_future_map2_int(  
  .x,  
  .y,  
  .f,  
  ...,  
  .options = furrr::furrr_options(),  
  .env_globals = parent.frame(),  
  .progress = FALSE,  
  .id = NULL,  
  .session_id = NULL  
)  
  
s_future_map2_lgl(  
  .x,  
  .y,
```

```

    .f,
    ...,
    .options = furrr::furrr_options(),
    .env_globals = parent.frame(),
    .progress = FALSE,
    .id = NULL,
    .session_id = NULL
  )

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

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

```

Arguments

<code>.x</code> , <code>.y</code>	Vectors of the same length.
<code>.f</code>	A function, formula, or vector.
<code>...</code>	Additional arguments passed to <code>.f</code> .
<code>.options</code>	A <code>furrr_options</code> object.
<code>.env_globals</code>	The environment to look for globals.
<code>.progress</code>	A single logical.
<code>.session_id</code>	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 = furr::furr_options(),
  .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 furr_options object.
.env_globals	The environment to look for globals.
.progress	A single logical.

Value

A list.

A character vector.

s_list_sessions

*List All Available SafeMapper Sessions***Description**

Returns a data frame with information about all stored checkpoint sessions, including progress, completion rates, and status.

Usage

```
s_list_sessions()
```

Value

Data frame with session information including:

- session_id: Unique identifier for the session
- created: Timestamp when session was created
- items_completed: Number of items successfully processed
- total_items: Total number of items in the operation
- completion_rate: Percentage of completion (0-1)
- status: Session status (in_progress, failed, corrupted)

Examples

```
# List all sessions
sessions <- s_list_sessions()
print(sessions)

# Find incomplete sessions
incomplete <- sessions[sessions$completion_rate < 1, ]
```

s_map

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

Description

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

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

<code>.x</code>	A list or atomic vector to map over.
<code>.f</code>	A function, formula, or vector.
<code>...</code>	Additional arguments passed to <code>.f</code> .
<code>.id</code>	Either a string or NULL for row binding.
<code>.session_id</code>	Character. Optional session ID for this operation.

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

```
result <- s_map(1:10, ~ .x^2)
```

s_map2	<i>Safe Map2 - Drop-in Replacement for purrr::map2 with Auto-Recovery</i>
--------	---

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

<code>.x, .y</code>	Vectors of the same length.
<code>.f</code>	A function, formula, or vector.
<code>...</code>	Additional arguments passed to <code>.f</code> .
<code>.id</code>	Either a string or NULL for row binding.
<code>.session_id</code>	Character. Optional session ID.

Value

A list.

Examples

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

s_pmap	<i>Safe PMap - Drop-in Replacement for purrr::pmap with Auto-Recovery</i>
--------	---

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 = furrr::furrr_options(),
  .env_globals = parent.frame(),
  .progress = FALSE,
  .session_id = NULL
)
```

Arguments

<code>.l</code>	A list of lists/vectors to map over.
<code>.f</code>	A function, formula, or vector.
<code>...</code>	Additional arguments passed to <code>.f</code> .
<code>.session_id</code>	Character. Optional session ID.

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

<code>.f</code>	A function to wrap for safe execution.
<code>otherwise</code>	Default return value when an error occurs.
<code>quiet</code>	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

<code>.f</code>	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_recover_session	<i>Recover a Specific SafeMapper Session</i>
-------------------	--

Description

Recovers and returns the results from a specific checkpoint session. This is useful for examining partial results or continuing computation after fixing errors.

Usage

```
s_recover_session(session_id)
```

Arguments

session_id Character. Session ID to recover.

Value

List of results from the session, or NULL if session not found.

Examples

```
# Recover a specific session
results <- s_recover_session("map_20240101_120000_abcd1234")

# Check if recovery was successful
if (!is.null(results)) {
  cat("Recovered", length(results), "items\n")
}
```

s_safely	<i>Safe Safely - Wrap Function to Capture Errors</i>
----------	--

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

<code>.f</code>	A function to wrap for safe execution.
<code>otherwise</code>	Default return value when an error occurs.
<code>quiet</code>	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_session_stats`*Get SafeMapper Session Statistics*

Description

Provides summary statistics of all sessions including counts by status, overall completion rates, and timing information.

Usage

```
s_session_stats()
```

Value

Invisible data frame of session information.

Examples

```
# Show session statistics
s_session_stats()

# Capture statistics for further analysis
stats <- s_session_stats()
```

s_version

Get SafeMapper Package Version and Information

Description

Displays comprehensive information about the SafeMapper package including version, supported functions, and usage tips.

Usage

```
s_version()
```

Examples

```
# Show package information
s_version()
```

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 = furrr::furrr_options(),
  .env_globals = parent.frame(),
  .progress = FALSE,
  .session_id = NULL
)
```

```
s_future_walk2(
  .x,
  .y,
  .f,
  ...,
  .options = furrr::furrr_options(),
  .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 furr_options object.
.env_globals	The environment to look for globals.
.progress	A single logical.

Value

Invisibly returns .x.

Index

s_clean_sessions, [2](#)
s_configure, [3](#)
s_debug_session, [4](#)
s_future_imap(s_imap), [9](#)
s_future_map, [4](#)
s_future_map2, [6](#)
s_future_map2_chr(s_future_map2), [6](#)
s_future_map2_dbl(s_future_map2), [6](#)
s_future_map2_dfc(s_future_map2), [6](#)
s_future_map2_dfr(s_future_map2), [6](#)
s_future_map2_int(s_future_map2), [6](#)
s_future_map2_lgl(s_future_map2), [6](#)
s_future_map_chr(s_future_map), [4](#)
s_future_map_dbl(s_future_map), [4](#)
s_future_map_dfc(s_future_map), [4](#)
s_future_map_dfr(s_future_map), [4](#)
s_future_map_int(s_future_map), [4](#)
s_future_map_lgl(s_future_map), [4](#)
s_future_pmap(s_pmap), [12](#)
s_future_walk(s_walk), [16](#)
s_future_walk2(s_walk), [16](#)
s_imap, [9](#)
s_imap_chr(s_imap), [9](#)
s_list_sessions, [9](#)
s_map, [10](#)
s_map2, [11](#)
s_map2_chr(s_map2), [11](#)
s_map2_dbl(s_map2), [11](#)
s_map2_dfc(s_map2), [11](#)
s_map2_dfr(s_map2), [11](#)
s_map2_int(s_map2), [11](#)
s_map2_lgl(s_map2), [11](#)
s_map_chr(s_map), [10](#)
s_map_dbl(s_map), [10](#)
s_map_dfc(s_map), [10](#)
s_map_dfr(s_map), [10](#)
s_map_int(s_map), [10](#)
s_map_lgl(s_map), [10](#)
s_pmap, [12](#)
s_possibly, [13](#)
s_quietly, [13](#)
s_recover_session, [14](#)
s_safely, [14](#)
s_session_stats, [15](#)
s_version, [16](#)
s_walk, [16](#)
s_walk2(s_walk), [16](#)