

Package ‘FaaSr’

May 7, 2026

Type Package

Title 'FaaSr' Local Test Development Package

Version 2.0.0

Maintainer Figueiredo Renato <renato.figueiredo@oregonstate.edu>

Description Provides a local execution environment for testing and developing the 'FaaSr' workflows without requiring cloud infrastructure. The 'FaaSr' package enables R developers to validate and test workflows locally before deploying to Function-as-a-Service (FaaS) platforms. Key features include:

- 1) Parsing and validating JSON workflow configurations compliant with the 'FaaSr' schema
- 2) Simulated S3 storage operations using local file system with local logging
- 3) Support for conditional branching
- 4) Support for parallel rank functions execution
- 5) Workflow cycle detection and validation
- 6) No cloud credentials or infrastructure required for testing

This package is designed for development and testing purposes. For production deployment to cloud FaaS platforms, use the main 'FaaSr' package available at <<https://faasr.io/>>.

License MIT + file LICENSE

URL <https://github.com/FaaSr/FaaSr-package-v2>

BugReports <https://github.com/FaaSr/FaaSr-package-v2/issues>

Depends R (>= 3.5.0)

Imports jsonlite, cli, jsonvalidate, uuid

Suggests testthat, knitr, rmarkdown

VignetteBuilder knitr

Encoding UTF-8

Language en-US

RoxygenNote 7.3.3

NeedsCompilation no

Author Figueiredo Renato [aut, cre, ths, cph] (ORCID: <<https://orcid.org/0000-0001-9841-6060>>),

Park Sungjae [aut] (ORCID: <<https://orcid.org/0009-0000-5357-804X>>),
 Mu Nan [ctb],
 Ku Yun-Jung [ctb],
 Daneshmand Vahid [ctb],
 Thomas R. Quinn [aut],
 Carey Cayelan [ctb],
 Tran Hoang [aut],
 Ramrakhiani Ashish Tulso [aut]

Repository CRAN

Date/Publication 2026-03-19 16:50:02 UTC

Contents

faasr_delete_file	2
faasr_get_file	3
faasr_get_folder_list	4
faasr_invocation_id	5
faasr_log	5
faasr_put_file	6
faasr_rank	7
faasr_test	7

Index **9**

faasr_delete_file	<i>Delete a file from local storage</i>
-------------------	---

Description

Deletes a file from the local FaaSr storage. This function mirrors the signature of the production FaaSr API but operates on local filesystem.

Usage

```
faasr_delete_file(server_name = NULL, remote_folder = "", remote_file)
```

Arguments

server_name	Character string specifying the server name (ignored in local implementation)
remote_folder	Character string path to the remote folder containing the file to delete
remote_file	Character string name of the remote file to delete

Value

Invisibly returns TRUE on success

Examples

```
## Not run:
# Delete a file from remote storage
faasr_delete_file(remote_file = "temp_data.csv")

# Delete a file from a specific remote folder
faasr_delete_file(remote_folder = "temp", remote_file = "data.csv")

## End(Not run)
```

faasr_get_file	<i>Get (download) a file from local storage</i>
----------------	---

Description

Downloads a file from the local FaaSr storage to the local filesystem. This function mirrors the signature of the production FaaSr API but operates on local filesystem.

Usage

```
faasr_get_file(
  server_name = NULL,
  remote_folder = "",
  remote_file,
  local_folder = ".",
  local_file
)
```

Arguments

server_name	Character string specifying the server name (ignored in local implementation)
remote_folder	Character string path to the remote folder containing the file to download
remote_file	Character string name of the remote file to download
local_folder	Character string path to the local folder where the file will be saved
local_file	Character string name for the file in local storage

Value

Invisibly returns TRUE on success

Examples

```
## Not run:
# Download a file from remote storage
faasr_get_file(remote_file = "data.csv", local_file = "downloaded_data.csv")

# Download to a specific local folder
```

```
faasr_get_file(remote_folder = "processed", remote_file = "data.csv",
              local_folder = "output", local_file = "data.csv")

## End(Not run)
```

`faasr_get_folder_list` *List files in local storage with optional prefix*

Description

Lists all files in the local FaaSr storage, optionally filtered by a prefix. This function mirrors the signature of the production FaaSr API but operates on local filesystem.

Usage

```
faasr_get_folder_list(server_name = NULL, faasr_prefix = "")
```

Arguments

`server_name` Character string specifying the server name (ignored in local implementation)
`faasr_prefix` Character string prefix to filter file names (optional)

Value

Character vector of file names matching the criteria

Examples

```
## Not run:
# List all files in storage
files <- faasr_get_folder_list()

# List files with a specific prefix
csv_files <- faasr_get_folder_list(faasr_prefix = "data_")

## End(Not run)
```

faasr_invocation_id *Get the invocation ID for the current workflow*

Description

Returns the invocation ID for the current workflow execution. This function mirrors the signature of the production FaaSr API but operates on local filesystem.

Usage

```
faasr_invocation_id()
```

Value

Character string invocation ID, or NULL if not available

Examples

```
## Not run:  
# Get current invocation ID  
inv_id <- faasr_invocation_id()  
if (!is.null(inv_id)) {  
  cat("Invocation ID:", inv_id, "\n")  
}  
  
## End(Not run)
```

faasr_log *Append a log message to local logs*

Description

Appends a log message with timestamp to the local FaaSr log file. This function mirrors the signature of the production FaaSr API but operates on local filesystem.

Usage

```
faasr_log(log_message)
```

Arguments

log_message Character string message to log

Value

Invisibly returns TRUE on success

Examples

```
## Not run:
# Log a simple message
faasr_log("Starting data processing")

# Log with more detail
faasr_log(paste("Processing", nrow(data), "rows of data"))

## End(Not run)
```

faasr_put_file	<i>Put (upload) a file to local storage</i>
----------------	---

Description

Uploads a file from the local filesystem to the local FaaSr storage. This function mirrors the signature of the production FaaSr API but operates on local filesystem.

Usage

```
faasr_put_file(
  server_name = NULL,
  local_folder = ".",
  local_file,
  remote_folder = "",
  remote_file
)
```

Arguments

server_name	Character string specifying the server name (ignored in local implementation)
local_folder	Character string path to the local folder containing the file to upload
local_file	Character string name of the local file to upload
remote_folder	Character string path to the remote folder where the file will be stored
remote_file	Character string name for the file in remote storage

Value

Invisibly returns TRUE on success

Examples

```
## Not run:
# Upload a local file to remote storage
faasr_put_file(local_file = "data.csv", remote_file = "input/data.csv")

# Upload from a specific local folder
```

```
faasr_put_file(local_folder = "input", local_file = "data.csv",
              remote_folder = "processed", remote_file = "data.csv")

## End(Not run)
```

faasr_rank	<i>Get current rank information for the executing function</i>
------------	--

Description

Returns the current rank information for the executing function in a parallel workflow. This function mirrors the signature of the production FaaSr API but operates on local filesystem.

Usage

```
faasr_rank()
```

Value

List containing Rank and MaxRank, or empty list if no rank information is available

Examples

```
## Not run:
# Get current rank information
rank_info <- faasr_rank()
if (length(rank_info) > 0) {
  cat("Current rank:", rank_info$Rank, "of", rank_info$MaxRank, "\n")
}

## End(Not run)
```

faasr_test	<i>FaaSr test execution</i>
------------	-----------------------------

Description

Workflow execution that dynamically handles conditional branching and predecessor dependencies without precomputation.

Usage

```
faasr_test(json_path)
```

Arguments

json_path	path to workflow JSON
-----------	-----------------------

Value

TRUE if all functions run successfully; stops on error

Index

faasr_delete_file, 2
faasr_get_file, 3
faasr_get_folder_list, 4
faasr_invocation_id, 5
faasr_log, 5
faasr_put_file, 6
faasr_rank, 7
faasr_test, 7