

Package: tinyqueue (via r-universe)

May 21, 2026

Type Package

Title A Simple Task Queue

Version 0.0.1

Date 2025-07-04

Description A task queue is provided with simple operator modeled after existing task queue functionality. By relying on either Redis or Valkey as a backend, a suitable degree of concurrency of worker nodes is provided.

Imports methods, RcppRedis

URL <https://github.com/eddelbuettel/tinyqueue>

BugReports <https://github.com/eddelbuettel/tinyqueue/issues>

License GPL (>= 2)

RoxygenNote 6.0.1

Config/pak/sysreqs libhiredis-dev

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

Date/Publication 2026-02-19 03:23:14 UTC

RemoteUrl <https://github.com/eddelbuettel/tinyqueue>

RemoteRef HEAD

RemoteSha bdf83c54165cc5a44f54f754362c0c2a4bc61057

Contents

tinyqueue-package	2
simpletask	2
tinyqueue	3

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

tinyqueue-package *A Simple Task Queue*

Description

A task queue is provided with simple operator modeled after existing task queue functionality. By relying on either Redis or Valkey as a backend, a suitable degree of concurrency of worker nodes is provided.

Package Content

Index: This package was not yet installed at build time.

Maintainer

Dirk Eddelbuettel <edd@debian.org>

Author(s)

Dirk Eddelbuettel [aut, cre]

simpletask *simpletask: A Simple Task Object*

Description

Represent a simpletask object suitable for queuing

Usage

```
simpletask(pkg, ver, ...)

## Default S3 method:
simpletask(pkg, ver, ...)

## S3 method for class 'simpletask'
format(x, ...)

## S3 method for class 'simpletask'
print(x, ...)

enqueue_task(q, st)

start_task(q)

end_task(q, st, res)
```

Arguments

pkg	character A package name
ver	character A package version number
...	dots Currently ignored
x	simpletask A task object
q	tinyqueue A tinyqueue object
st	object An R object describing the task that is serialized
res	numeric The result code of the completed task, must be between 0 and 2

Details

A simpletask objects describes a task, here usually a package name for a package to be tested along with a version number. It can be any R object as the simplequeue serializes and de-serializes the object appropriately

See Also

tinyqueue

tinyqueue

tinyqueue: A Simple Workqueue Object

Description

Create a tinyqueue object

Usage

```
tinyqueue(name, ...)  
  
## Default S3 method:  
tinyqueue(name, ...)  
  
## S3 method for class 'tinyqueue'  
print(x, ...)  
  
## S3 method for class 'tinyqueue'  
format(x, ...)  
  
## S3 method for class 'tinyqueue'  
summary(object, ...)  
  
ensure_queue(name)  
  
publish(queue, message)
```

```
list_messages(queue)

try_consume(queue)

ack(queue, message, res = 0)

cleanup(queue)
```

Arguments

name	character	The task name for the queue
...		other arguments
x	tinyqueue	A tinyqueue object task name for
object	tinyqueue	A tinyqueue object
queue	tinyqueue	A tinyqueue object task name for
message	object	A R object describing a task that is serialized
res	numeric	The task result that is acknowledged where default is zero for success, one signals failure, and two that the task was skipped

Details

A tinyqueue object manages a job queue for a given and named task, using either **Redis** or **Valkey** as a backend. The (basic) functionality is modelled after the **liteq** package.

A tinyqueue is created and returned by function `ensure_queue`. The function `publish_queue`, `try_consume` and `ack` serve to, respectively, enqueue a task, dequeue it in order to do the designated work and then acknowledge completion. `list_message` reports the status, and `cleanup` finalize and cleans up.

The tinyqueue object has several associated methods to print, summarize, or format the object.

The functionality takes advantage of package **RcppRedis** and its ability to process arbitrary R objects. This allows task descriptions to be `data.frame` or `list` objects as needed by the application. For the same reason, no ‘schema’ or alike has to be created as the queued task objects are self-sufficient as R objects.

Examples

```
# This requires Redis or Valkey to run
## Not run: tinyqueue("sometask")
```

Index

* package

[tinyqueue-package](#), 2

[ack \(tinyqueue\)](#), 3

[cleanup \(tinyqueue\)](#), 3

[end_task \(simpletask\)](#), 2

[enqueue_task \(simpletask\)](#), 2

[ensure_queue \(tinyqueue\)](#), 3

[format.simpletask \(simpletask\)](#), 2

[format.tinyqueue \(tinyqueue\)](#), 3

[list_messages \(tinyqueue\)](#), 3

[print.simpletask \(simpletask\)](#), 2

[print.tinyqueue \(tinyqueue\)](#), 3

[publish \(tinyqueue\)](#), 3

[simpletask](#), 2

[start_task \(simpletask\)](#), 2

[summary.tinyqueue \(tinyqueue\)](#), 3

[tinyqueue](#), 3

[tinyqueue-package](#), 2

[try_consume \(tinyqueue\)](#), 3