

# Package ‘shiny.worker’

January 21, 2021

**Type** Package

**Title** Delegate Jobs for Shiny Web Applications

**Version** 0.0.1

**Description** It allows you to delegate heavy computation tasks to a separate process, such that it does not freeze your Shiny app.

**Encoding** UTF-8

**LazyData** true

**License** MIT + file LICENSE

**Imports** future, shiny, R6

**RoxygenNote** 7.1.1

**Suggests** testthat, covr

**NeedsCompilation** no

**Author** Paweł Przytuła [aut],  
Dominik Krzemiński [cre],  
Appsilon [cph]

**Maintainer** Dominik Krzemiński <dominik@appsilon.com>

**Repository** CRAN

**Date/Publication** 2021-01-21 11:00:02 UTC

## R topics documented:

|                   |   |
|-------------------|---|
| initialize_worker | 2 |
| Worker            | 2 |

|              |          |
|--------------|----------|
| <b>Index</b> | <b>4</b> |
|--------------|----------|

`initialize_worker`      *Creates new shiny.worker object*

---

### Description

Creates new shiny.worker object

### Usage

```
initialize_worker()
```

### Value

shiny.worker object

### Examples

```
if(interactive()){  
  worker <- initialize_worker()  
}
```

---

Worker      *Shiny Worker R6 Class*

---

### Description

Shiny Worker R6 Class

Shiny Worker R6 Class

### Methods

#### Public methods:

- [Worker\\$new\(\)](#)
- [Worker\\$run\\_job\(\)](#)
- [Worker\\$get\\_job\\_registry\(\)](#)
- [Worker\\$clone\(\)](#)

**Method** `new()`: Initialize the worker's registry

*Usage:*

```
Worker$new()
```

**Method** `run_job()`: Add job to the registry

*Usage:*

```
Worker$run_job(  
  id,  
  fun,  
  args_reactive,  
  value_until_resolved = NULL,  
  invalidate_time = 1000  
)
```

*Arguments:*

id character with job id

fun function to calculate

args\_reactive reactive arguments that trigger the job

value\_until\_resolved default value returned until the job is completed

invalidate\_time wait time before invalidating reactive context (msec)

*Returns:* reactive expression with promise of job completion

**Method** get\_job\_registry(): Displays jobs registry.

*Usage:*

```
Worker$get_job_registry()
```

**Method** clone(): The objects of this class are cloneable with this method.

*Usage:*

```
Worker$clone(deep = FALSE)
```

*Arguments:*

deep Whether to make a deep clone.

# Index

`initialize_worker`, [2](#)

`Worker`, [2](#)