

# Package ‘golem’

August 5, 2019

**Title** A Framework for Robust Shiny Applications

**Version** 0.1

**Description** An opinionated framework for building a production-ready 'Shiny' application. This package contains a series of tools for building a robust 'Shiny' application from start to finish.

**License** MIT + file LICENSE

**URL** <https://github.com/ThinkR-open/golem>

**BugReports** <https://github.com/ThinkR-open/golem/issues>

**Depends** R (>= 3.0)

**Imports** attempt, cli, crayon, desc, dockerfiler, DT, glue, htmltools, pkgload, processx, remotes, rlang, roxygen2, rsconnect, rstudioapi, shiny, stats, stringr, testthat, tools, usethis, utils, yesno

**Suggests** covr, knitr, pkgdown, purrr, rcmdcheck, rmarkdown, withr

**VignetteBuilder** knitr

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**NeedsCompilation** no

**Author** Vincent Guyader [cre, aut] (<<https://orcid.org/0000-0003-0671-9270>>),  
Colin Fay [aut] (<<https://orcid.org/0000-0001-7343-1846>>),  
Sébastien Rochette [aut] (<<https://orcid.org/0000-0002-1565-9313>>),  
Cervan Girard [aut] (<<https://orcid.org/0000-0002-4816-4624>>),  
ThinkR [cph]

**Maintainer** Vincent Guyader <[vincent@thinkr.fr](mailto:vincent@thinkr.fr)>

**Repository** CRAN

**Date/Publication** 2019-08-05 15:50:02 UTC

## R topics documented:

activate_js . . . . .	2
add_dockerfile . . . . .	3
add_js_file . . . . .	4
add_module . . . . .	5
add_rstudioconnect_file . . . . .	5
app_prod . . . . .	6
browser_button . . . . .	7
cat_dev . . . . .	7
create_golem . . . . .	8
detach_all_attached . . . . .	8
document_and_reload . . . . .	9
expect_shinytag . . . . .	9
fill_desc . . . . .	10
get_dependencies . . . . .	10
get_golem_options . . . . .	11
get_golem_wd . . . . .	11
golem . . . . .	12
make_dev . . . . .	12
set_golem_options . . . . .	13
use_favicon . . . . .	13
use_recommended_deps . . . . .	14
use_utils_ui . . . . .	14
with_golem_options . . . . .	15
<b>Index</b>	<b>16</b>

---

activate\_js

*Interact with JavaScript built-in Functions*

---

### Description

activate\_js is used in your UI to insert directly the JavaScript functions contained in golem. These functions can be called from the server with invoke\_js. invoke\_js can also be used to launch any JS function created inside a Shiny JavaScript handler.

### Usage

```
activate_js()
```

```
invoke_js(fun, ui_ref, session = shiny::getDefaultReactiveDomain())
```

**Arguments**

fun	JS function to be invoked.
ui_ref	The UI reference to call the JS function on.
session	The shiny session within which to call sendCustomMessage.
<b>show</b>	Show an element with the jQuery selector provided.
<b>hide</b>	Hide an element with the jQuery selector provided.
<b>showid</b>	Show an element with the id provided.
<b>hideid</b>	Hide an element with the id provided.
<b>showclass</b>	Same as showid, but with class.
<b>hideclass</b>	Same as hideid, but with class.
<b>showhref</b>	Same as showid, but with a[href*=
<b>hidehref</b>	Same as hideid, but with a[href*=
<b>clickon</b>	Click on an element. The full jQuery selector has to be used.
<b>disable</b>	Add "disabled" to an element. The full jQuery selector has to be used.
<b>reable</b>	Remove "disabled" from an element. The full jQuery selector has to be used.

---

add_dockerfile	<i>Create a Dockerfile for Shiny App</i>
----------------	--

---

**Description**

Build a container containing your Shiny App. `add_dockerfile()` creates a "classical" Dockerfile, while `add_dockerfile_shinyproxy()` and `add_dockerfile_heroku()` creates platform specific Dockerfile.

**Usage**

```
add_dockerfile(input = "DESCRIPTION", output = "Dockerfile",
  pkg = get_golem_wd(), from = paste0("rocker/tidyverse:",
  R.Version()$major, ".", R.Version()$minor), as = NULL, port = 80,
  host = "0.0.0.0")

add_dockerfile_shinyproxy(input = "DESCRIPTION", output = "Dockerfile",
  pkg = get_golem_wd(), from = paste0("rocker/tidyverse:",
  R.Version()$major, ".", R.Version()$minor), as = NULL)

add_dockerfile_heroku(input = "DESCRIPTION", output = "Dockerfile",
  pkg = get_golem_wd(), from = paste0("rocker/tidyverse:",
  R.Version()$major, ".", R.Version()$minor), as = NULL)
```

**Arguments**

input	path to the DESCRIPTION file to use as an input.
output	name of the Dockerfile output.
pkg	Path to the root of the package. Default is ".".
from	The FROM of the Dockerfile. Default is FROM rocker/tidyverse: with R.Version()\$major and R.Version()\$minor.
as	The AS of the Dockerfile. Default it NULL.
port	The options('shiny.port') on which to run the Shiny App. Default is 80.
host	The options('shiny.host') on which to run the Shiny App. Default is 0.0.0.0.

**Examples**

```
# Add a standard Dockerfile
if (interactive()){
  add_dockerfile()
}
# Add a Dockerfile for ShinyProxy
if (interactive()){
  add_dockerfile_shinyproxy()
}
# Add a Dockerfile for Heroku
if (interactive()){
  add_dockerfile_heroku()
}
```

---

 add\_js\_file

 Create Files
 

---

**Description**

These functions create files inside the inst/app folder.

**Usage**

```
add_js_file(name, pkg = get_golem_wd(), dir = "inst/app/www",
  open = TRUE, dir_create = TRUE)

add_js_handler(name, pkg = get_golem_wd(), dir = "inst/app/www",
  open = TRUE, dir_create = TRUE)

add_css_file(name, pkg = get_golem_wd(), dir = "inst/app/www",
  open = TRUE, dir_create = TRUE)

add_ui_server_files(pkg = get_golem_wd(), dir = "inst/app",
  dir_create = TRUE)
```

**Arguments**

name	The name of the module
pkg	Path to the root of the package. Default is ".".
dir	Path to the dir of the file
open	Should the file be opened?
dir_create	Creates the directory if it doesn't exist, default is TRUE.

---

add_module	<i>Create a module</i>
------------	------------------------

---

**Description**

This function creates a module inside the R/ folder, based on a specific module structure.

**Usage**

```
add_module(name, pkg = get_golem_wd(), open = TRUE,
           dir_create = TRUE)
```

**Arguments**

name	The name of the module
pkg	Path to the root of the package. Default is ".".
open	Should the file be opened?
dir_create	Creates the directory if it doesn't exist, default is TRUE.

**Note**

This function will prefix the name argument with mod\_.

---

add_rstudioconnect_file	<i>Add an app.R at the root of your package to deploy on RStudio Connect</i>
-------------------------	--

---

**Description**

Add an app.R at the root of your package to deploy on RStudio Connect

**Usage**

```
add_rstudioconnect_file(pkg = get_golem_wd(), open = TRUE)
```

```
add_shinyappsio_file(pkg = get_golem_wd(), open = TRUE)
```

```
add_shinyserver_file(pkg = get_golem_wd(), open = TRUE)
```

**Arguments**

pkg	Where to put the app.R.
open	Open the file

**Note**

In previous versions, this function was called `add_rconnect_file`.

**Examples**

```
# Add a file for Connect
if (interactive()){
  add_rstudioconnect_file()
}
# Add a file for Shiny Server
if (interactive()){
  add_shinyserver_file()
}
# Add a file for Shinyapps.io
if (interactive()){
  add_shinyappsio_file()
}
```

---

app\_prod

*Is the app in dev mode or prod mode?*

---

**Description**

Is the app in dev mode or prod mode?

**Usage**

```
app_prod()
```

```
app_dev()
```

**Value**

TRUE or FALSE depending on the status of `getOption("golem.app.prod")`

---

browser_button	<i>Insert an hidden browser button</i>
----------------	--

---

**Description**

See <https://rtask.thinkr.fr/blog/a-little-trick-for-debugging-shiny/> for more context.

**Usage**

```
browser_button()
```

**Value**

Prints the code to the console.

---

cat_dev	<i>Functions already made dev dependant</i>
---------	---

---

**Description**

This functions will be run only if `golem::app_dev()` returns TRUE.

**Usage**

```
cat_dev(...)  
print_dev(...)  
message_dev(...)  
warning_dev(...)  
browser_dev(...)
```

**Arguments**

... R objects (see ‘Details’ for the types of objects allowed).

---

create_golem	<i>Create a package for Shiny App using golem</i>
--------------	---

---

**Description**

Create a package for Shiny App using golem

**Usage**

```
create_golem(path, check_name = TRUE, open = TRUE,  
             package_name = basename(path), ...)
```

**Arguments**

path	Name of the folder to create the package in. This will also be used as the package name.
check_name	When using this function in the console, you can prevent the package name from being checked.
open	boolean open the created project
package_name	package name to use
...	not used

---

detach_all_attached	<i>Detach all attached package</i>
---------------------	------------------------------------

---

**Description**

Detach all attached package

**Usage**

```
detach_all_attached()
```



---

document\_and\_reload     *Document and reload your package*

---

**Description**

This function calls `rstudioapi::documentSaveAll()`, `roxygen2::roxygenise()` and `pkgload::load_all()`.

**Usage**

```
document_and_reload(pkg = ".")
```

**Arguments**

pkg                    Path to the root of the package. Default is ".".

---

expect\_shinytag        *Test helpers*

---

**Description**

These functions are designed to be used inside the tests in your Shiny app package.

**Usage**

```
expect_shinytag(object)
```

```
expect_shinytaglist(object)
```

```
expect_html_equal(ui, html)
```

**Arguments**

object                the object to test

ui                    output of an UI function

html                  html file to compare to ui

**Value**

A testthat result

**Examples**

```
expect_shinytag(shiny::tags$span("1"))  
expect_shinytaglist(shiny::tagList(1))
```

---

fill_desc	<i>Fill your description</i>
-----------	------------------------------

---

**Description**

Fill your description

**Usage**

```
fill_desc(pkg_name, pkg_title, pkg_description, author_first_name,
  author_last_name, author_email, repo_url = NULL,
  pkg = get_golem_wd())
```

**Arguments**

pkg_name	The name of the package
pkg_title	The title of the package
pkg_description	Description of the package
author_first_name	First Name of the author
author_last_name	Last Name of the author
author_email	Email of the author
repo_url	URL (if needed)
pkg	Path to look for the DESCRIPTION

---

get_dependencies	<i>Return all package dependencies from current package</i>
------------------	---

---

**Description**

Return all package dependencies from current package

**Usage**

```
get_dependencies(path = "DESCRIPTION", pkg = get_golem_wd(),
  dput = FALSE, field = c("Imports"))
```

**Arguments**

path	path to the DESCRIPTION file
pkg	Path to the root of the package. Default is ".".
dput	if TRUE return a dput output instead of character vector
field	DESCRIPTION fields to parse. Default is Import

**Examples**

```
if (interactive()){
  get_dependencies()
}
```

---

get_golem_options	<i>Get all or one golem options</i>
-------------------	-------------------------------------

---

**Description**

This function is to be used inside the server and UI from your app, in order to call the parameters passed to `run_app()`.

**Usage**

```
get_golem_options(which = NULL)
```

**Arguments**

which	NULL (default), or the name of an option
-------	--

---

get_golem_wd	<i>Get and set {golem} working directory</i>
--------------	--

---

**Description**

Many {golem} functions rely on a specific working directory, most of the time the root of the package. This working directory is set by `set_golem_options` or the first time you create a file. It default to ".", the current directory when starting a golem. You can use these two functions if you need to manipulate this directory.

**Usage**

```
get_golem_wd()

set_golem_wd(path, talkative = TRUE)
```

**Arguments**

path	The path to set the golem woking directory. Note that it will be passed to <code>normalizePath</code> .
talkative	Should the function print where the new path is defined?

**Value**

The path to the working directory.

---

golem

*A package for building Shiny App*

---

### Description

Read more about building big shiny apps at <https://thinkr-open.github.io/building-shiny-apps-workflow/>.

### Author(s)

**Maintainer:** Vincent Guyader <vincent@thinkr.fr> (0000-0003-0671-9270)

Authors:

- Colin Fay <contact@colinfay.me> (0000-0001-7343-1846)
- Sébastien Rochette <sebastien@thinkr.fr> (0000-0002-1565-9313)
- Cervan Girard <cervan@thinkr.fr> (0000-0002-4816-4624)

Other contributors:

- ThinkR [copyright holder]

### See Also

Useful links:

- <https://github.com/ThinkR-open/golem>
- Report bugs at <https://github.com/ThinkR-open/golem/issues>

---

make\_dev

*Make a function dependent to dev mode*

---

### Description

The function returned will be run only if `golem::app_dev()` returns TRUE.

### Usage

```
make_dev(fun)
```

### Arguments

fun            A function

---

set_golem_options	<i>golem options</i>
-------------------	----------------------

---

**Description**

Set a series of options to be used internally by {golem}.

**Usage**

```
set_golem_options(golem_name = pkgload::pkg_name(),
  golem_version = pkgload::pkg_version(),
  golem_wd = pkgload::pkg_path(), app_prod = FALSE)
```

**Arguments**

golem_name	Name of the current golem.
golem_version	Version of the current golem.
golem_wd	Working directory of the current golem package.
app_prod	Is the {golem} in prod mode?

---

use_favicon	<i>Add a favicon to your shinyapp</i>
-------------	---------------------------------------

---

**Description**

This function adds the favicon from ico to your shiny app.

**Usage**

```
use_favicon(path, pkg = ".")

remove_favicon(path = "inst/app/www/favicon.ico")

favicon(ico = "www/favicon.ico", rel = "shortcut icon")
```

**Arguments**

path	Path to your favicon file (.ico or .png)
pkg	Path to the root of the package. Default is "."
ico	path to favicon file
rel	rel

**Examples**

```
if (interactive()){
  use_favicon()
  use_favicon(path='path/to/your/favicon.ico')
}
```

---

use\_recommended\_deps *Add recommended elements*

---

**Description**

**use\_recommended\_deps** Adds shiny, DT, attempt, glue, golem, htmltools to dependencies

**use\_recommended\_tests** Adds a test folder and copy the golem tests

**Usage**

```
use_recommended_deps(pkg = ".", recommended = c("shiny", "DT",
  "attempt", "glue", "htmltools", "golem"))
```

```
use_recommended_tests(pkg = ".")
```

**Arguments**

**pkg** Path to the root of the package. Default is ".".

**recommended** A vector of recommended packages.

---

use\_utils\_ui *Use the utils files*

---

**Description**

**use\_utils\_ui** Copies the golem\_utils\_ui.R to the R folder.

**use\_utils\_server** Copies the golem\_utils\_server.R to the R folder.

**Usage**

```
use_utils_ui(pkg = ".")
```

```
use_utils_server(pkg = ".")
```

**Arguments**

**pkg** Path to the root of the package. Default is ".".

---

with\_golem\_options     *Add Golem options to a Shiny App*

---

**Description**

Add Golem options to a Shiny App

**Usage**

```
with_golem_options(app, golem_opts)
```

**Arguments**

app                    the app object.  
golem\_opts            A list of Options to be added to the app

**Value**

a shiny.appObj object

**Note**

You'll probably never have to write this function as it is included in the golem template created on launch.

# Index

activate\_js, 2  
add\_css\_file (add\_js\_file), 4  
add\_dockerfile, 3  
add\_dockerfile\_heroku (add\_dockerfile),  
3  
add\_dockerfile\_shinyproxy  
(add\_dockerfile), 3  
add\_js\_file, 4  
add\_js\_handler (add\_js\_file), 4  
add\_module, 5  
add\_rconnect\_file  
(add\_rstudioconnect\_file), 5  
add\_rstudioconnect\_file, 5  
add\_shinyappsio\_file  
(add\_rstudioconnect\_file), 5  
add\_shinyserver\_file  
(add\_rstudioconnect\_file), 5  
add\_ui\_server\_files (add\_js\_file), 4  
app\_dev (app\_prod), 6  
app\_prod, 6  
  
browser\_button, 7  
browser\_dev (cat\_dev), 7  
  
cat\_dev, 7  
create\_golem, 8  
  
detach\_all\_attached, 8  
document\_and\_reload, 9  
  
expect\_html\_equal (expect\_shinytag), 9  
expect\_shinytag, 9  
expect\_shinytaglist (expect\_shinytag), 9  
  
favicon (use\_favicon), 13  
fill\_desc, 10  
  
get\_dependencies, 10  
get\_golem\_options, 11  
get\_golem\_wd, 11  
golem, 12  
golem-package (golem), 12  
invoke\_js (activate\_js), 2  
  
make\_dev, 12  
message\_dev (cat\_dev), 7  
  
print\_dev (cat\_dev), 7  
  
remove\_favicon (use\_favicon), 13  
  
set\_golem\_options, 13  
set\_golem\_wd (get\_golem\_wd), 11  
  
use\_favicon, 13  
use\_recommended\_deps, 14  
use\_recommended\_tests  
(use\_recommended\_deps), 14  
use\_utils\_server (use\_utils\_ui), 14  
use\_utils\_ui, 14  
  
warning\_dev (cat\_dev), 7  
with\_golem\_options, 15