THINKR

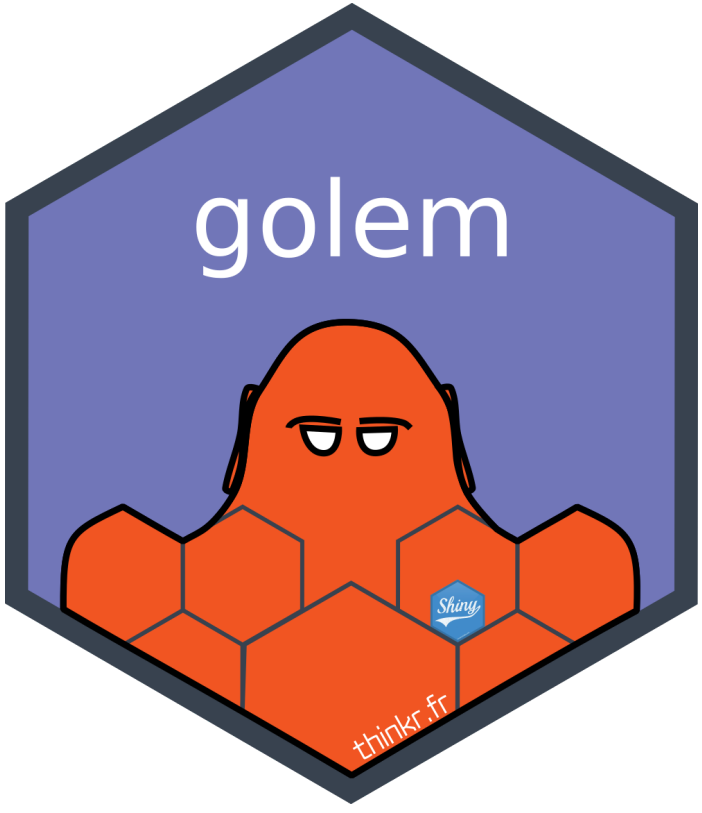
{golem}

Un framework pour construire des applications shiny qui tiennent la route

Vincent Guyader

Data Scientist, R expert.

- https://rtask.thinkr.fr
- https://github.com/ThinkR-open
- https://twitter.com/thinkr_fr



> Créer, maintenir et déployer facilement une application Shiny

Pourquoi utiliser Golem ?

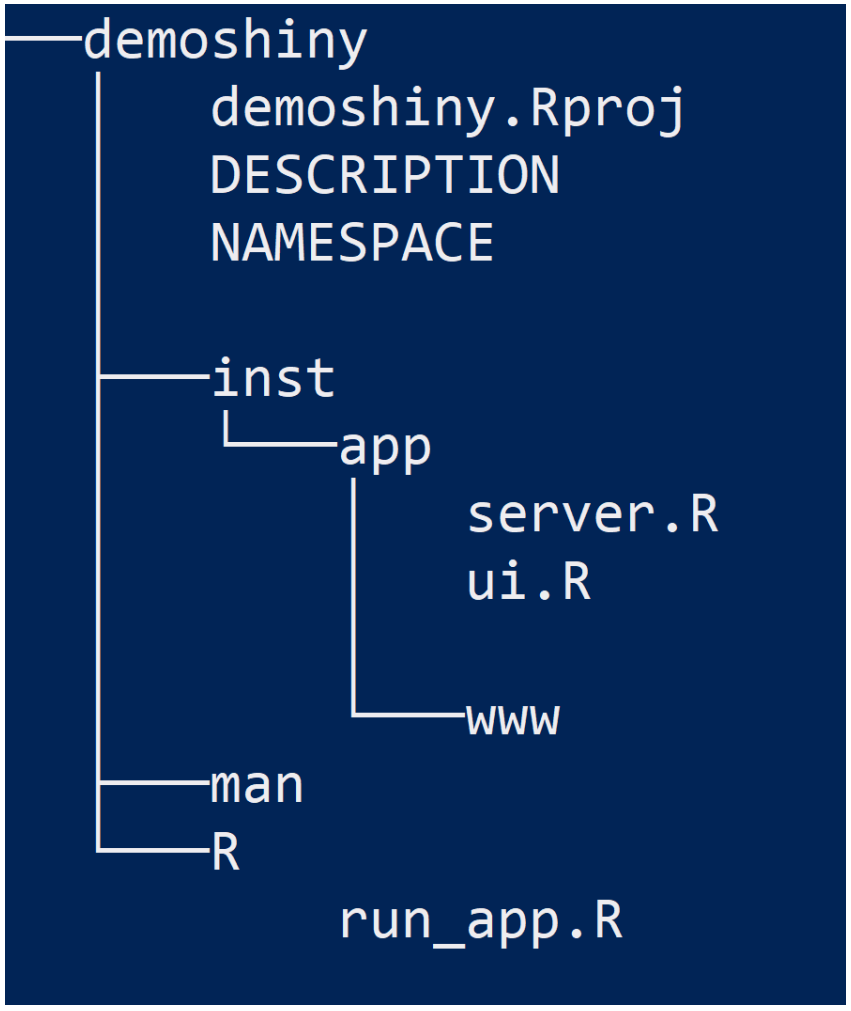
- Gagner du temps
- Travailler proprement
- Travailler à plusieurs
- Simplifier le déploiement des applications
- Faciliter la maintenance des applications
- Avoir une bonne documentation

Dans R, tout devrait être package!

Créer un package R en 6 minutes and Dockeriser son application shiny

- Gestion des dépendances.
- Gestion des versions.
- Installation et déploiement faciles.
- Gestion de la documentation

Dans R, tout devrait être package!

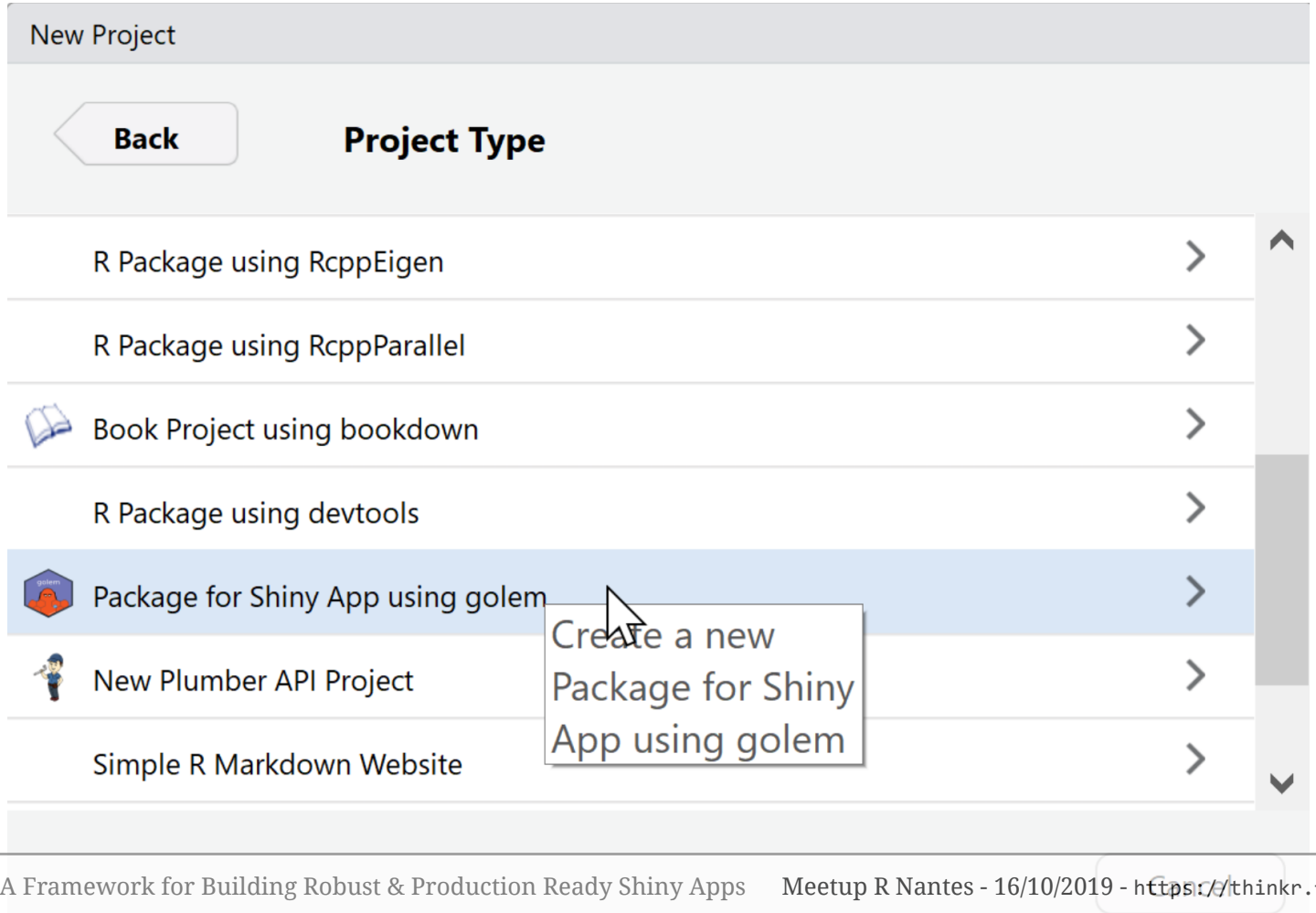


Dans R, tout devrait être package!


```
run_app <- function() {
  appdir <- system.file("app", package = "demoshiny")
  shiny::runApp(appdir, display.mode = "normal")
}
```

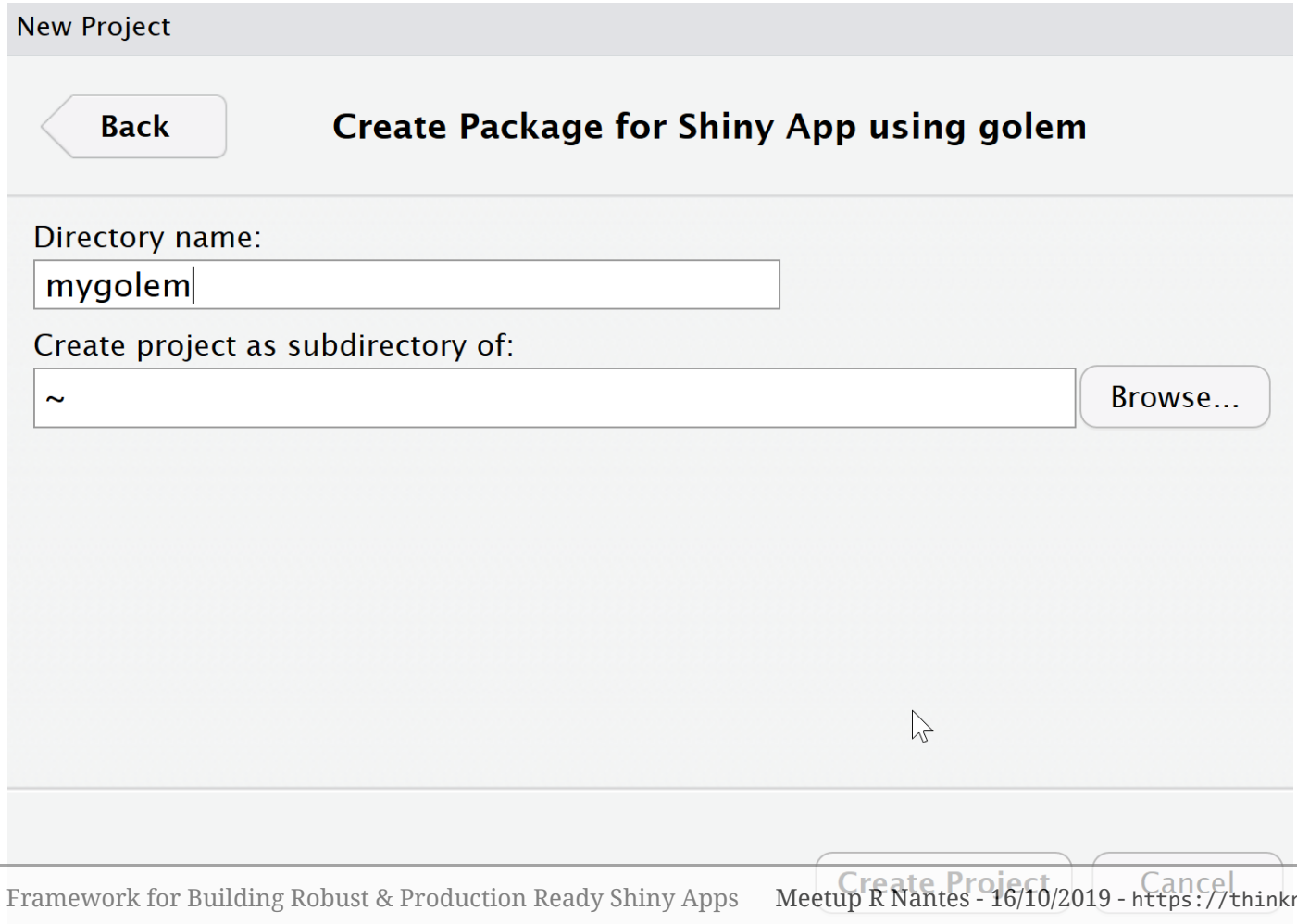
Let's start

```
remotes::install_github("thinkr-open/golem")
```

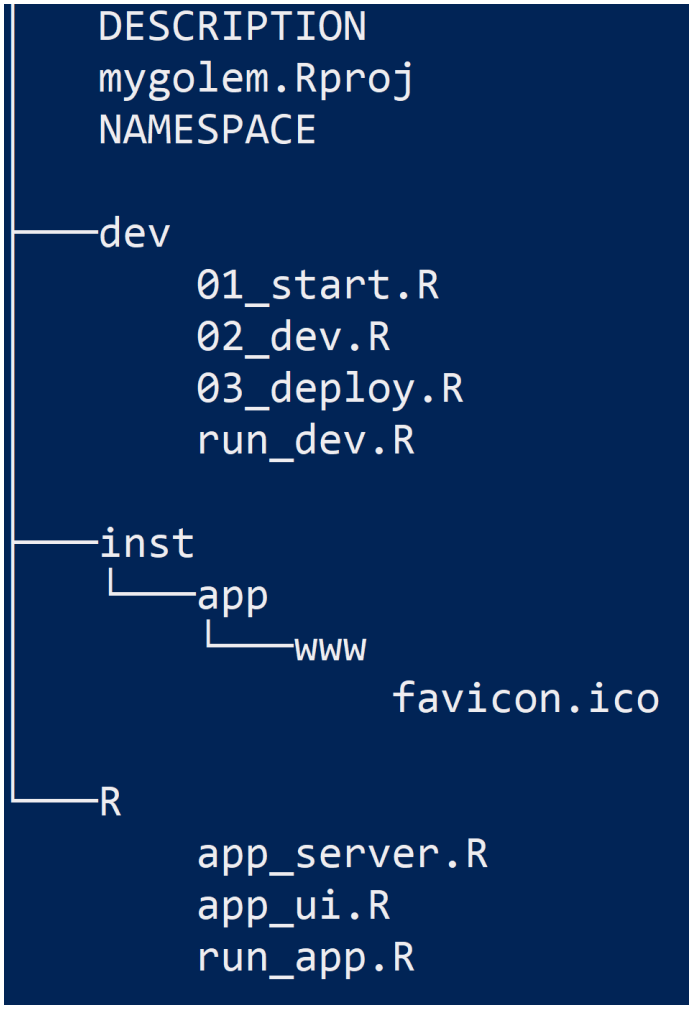


C'est parti

```
remotes::install_github("thinkr-open/golem")
```



C'est parti


DESCRIPTION
mygolem.Rproj
NAMESPACE
dev
01_start.R
02_dev.R
03_deploy.R
run_dev.R
inst
app
www
favicon.ico
R
app_server.R
app_ui.R
run_app.R

R/app_ui.R

```r
#' @import shiny
app_ui <- function() {
  tagList(
    golem_add_external_resources(),
          fluidPage(
            h1("demogolem")
            )
    )
}
```

```r
#' @import shiny
golem_add_external_resources <- function() {
  addResourcePath('www', system.file('app/www', package = 'demogolem'))
  tags$head(
    golem::activate_js(),
    golem::favicon()
    #tags$link(rel="stylesheet", type="text/css", href="www/custom.css")
    )
}
```

R/app_server.R


```r
#' @import shiny
app_server <- function(input, output,session) {
# List the first level callModules here
}
```

R/run_app.R

```r
#' Run the Shiny Application
#'
#' @export
#' @importFrom shiny shinyApp
#' @importFrom golem with_golem_options
run_app <- function(...) {
  with_golem_options(
    app = shinyApp(ui = app_ui, server = app_server),
    golem_opts = list(...)
  )
}
```

Pour lancer votre application :

```r
remotes::install_local() # install your golem from sources
mygolem::run_app() # Launch your golem
```

dev/run_dev.R

Ce script vous permet de reconstruire et visualiser rapidement votre golem.

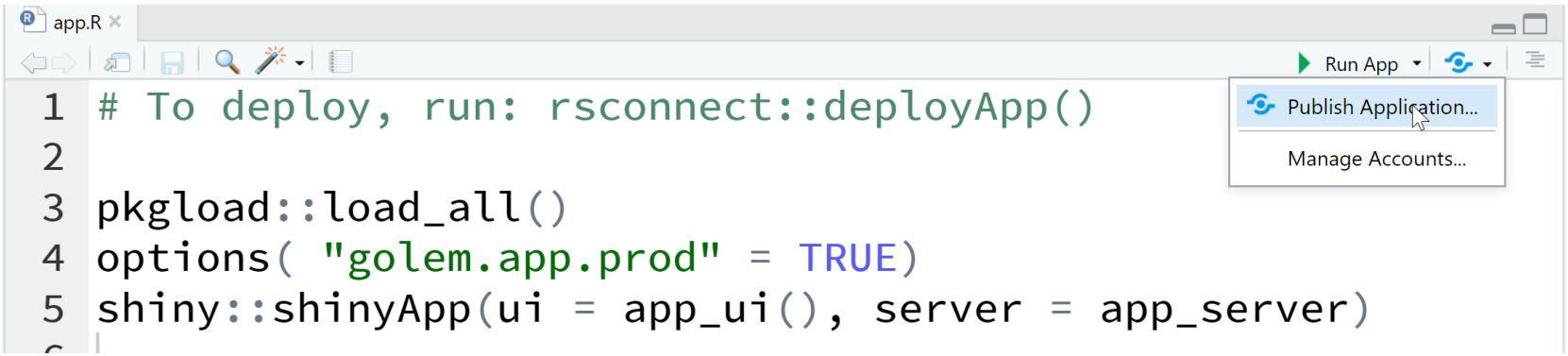
```
# Detach all loaded packages and clean your environment
golem::detach_all_attached()
# rm(list=ls(all.names = TRUE))

# Document and reload your package
golem::document_and_reload()

# Run the application
mygolem::run_app()
```

Déploiement

{golem} contient un ensemble d'outils qui facilite le déploiement.

sur : Rstudio Connect, shinyproxy, shiny server, heroku, ...

```
golem::add_dockerfile()
golem::add_dockerfile_heroku()
golem::add_dockerfile_shinyproxy()

golem::add_rstudioconnect_file()
golem::add_shinyappsio_file()
golem::add_shinyserveur_file()
```



dev/01_start.R : fill_desc()

Pour démarer vous n'avez qu'a suivre le contnu du fichier 01_start.R

- rempli le fichier DESCRIPTION

```
golem::fill_desc(
pkg_name = , # The Name of the package containing the App
pkg_title = , # The Title of the package containing the App
pkg_description = , # The Description of the package containing the App
author_first_name = , # Your First Name
author_last_name = , # Your Last Name
author_email = , # Your Email
repo_url = NULL) # The (optional) URL of the GitHub Repo
```

- rajoute les dépendances

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

dev/01_start.R : usethis::

Il est pré-rempli avec les appels les plus courants à {usethis}.

```
usethis::use_mit_license(name = "Your Name")
usethis::use_readme_rmd()
usethis::use_code_of_conduct()
usethis::use_lifecycle_badge("Experimental")
usethis::use_news_md()
```

dev/01_start.R : use_utils_ui()

Précharge des fonctions utiles pour le UI de votre golem

```
golem::use_utils_ui()
```

Example :

```
# @examples
# rep_br(5)
rep_br <- function(times = 1) {
  HTML(rep("<br/>", times = times))
} #@examples
# enurl("https://www.thinkr.fr", "ThinkR")
enurl <- function(url, text) {
  tags$a(href = url, text)
}
```

dev/01_start.R : use_utils_server()

Précharge des fonctions utiles pour le server de votre golem

```
golem::use_utils_server()
```

Example :

```
`%not_in%` <- Negate(`%in%`)
not_null <- Negate(is.null)
not_na <- Negate(is.na)

# Removes the null from a vector
drop_nulls <- function(x) {
  x[!sapply(x, is.null)]
}

"%||%" <- function(x, y) {
  if (is.null(x)) { y
  } else { x }
}
```

dev/02_dev.R : add_js() & addcss()

Une grosse application, ambitieuse, nécessite souvent l'utilisation de fonctions javascript et de css personnalisés.

{golem} vous permet d'ajouter facilement de tels fichiers à votre projet.

```
golem::add_js_file("script")
golem::add_js_handler("script")
golem::add_css_file("custom")
```

dev/02_dev.R : add_module()

{golem} encourage l'utilisation des Shiny modules. La fonction `add_module` vous permet d'ajouter rapidement et efficacement un module à votre golem.

Pourquoi des modules ? Parce que les modules....

- divisent votre application en petits morceaux
- vous protégent des conflits de NAMESPACE
- sont réutilisables
- évitent d'avoir des fichiers trop longs

dev/02_dev.R : add_module()

{golem} encourage l'utilisation des Shiny modules. La fonction `add_module` vous permet d'ajouter rapidement et efficacement un module à votre golem.

```
golem::add_module("clock")
```

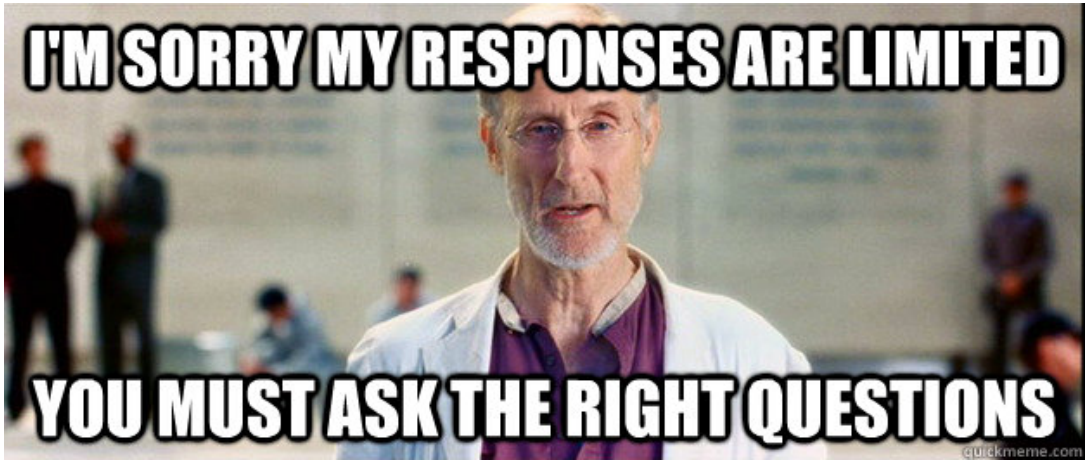
```
mod_clock_ui <- function(id){
  ns <- NS(id)
  tagList(
  )
}

mod_clock_server <- function(input, output, session){

  }

## To be copied in the UI
# mod_clock_ui("clock_ui_1")

## To be copied in the server
# callModule(mod_clock_server, "clock_ui_1")
```

Votre Golem est un package

N'oubliez pas que votre golem est un package, donc vous pouvez (devez) utiliser :

- documentation
- test unitaire
- intégration continue
- déploiement continu
- ...

Merci !



me retrouver:

- vincent@thinkr.fr
- http://twitter.com/vincentguyader
- http://twitter.com/thinkr_fr
- https://rtask.thinkr.fr/
- https://thinkr.fr/

pour aller plus loin :

- building-shiny-apps-workflow
- {golem}
- {shinypsum}
- {fakir}
- {shinysnippets}