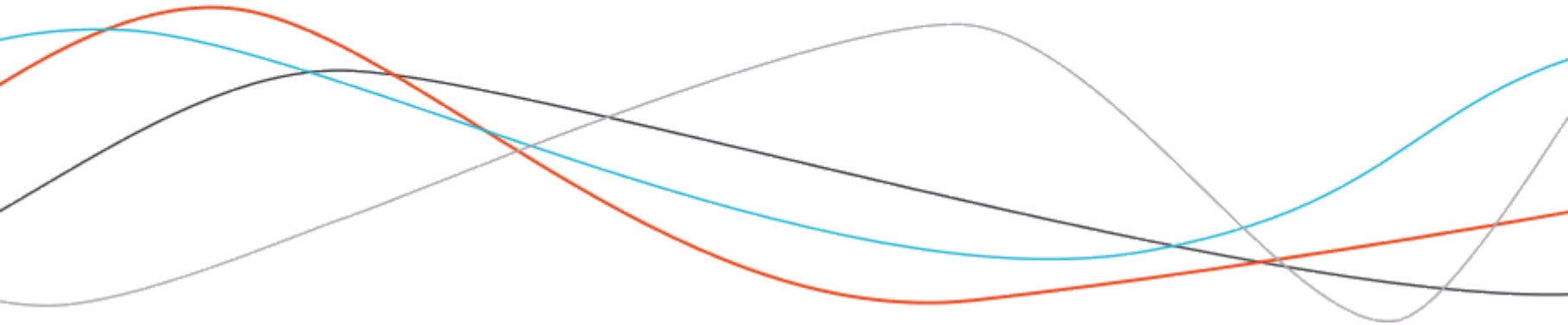
THINKR

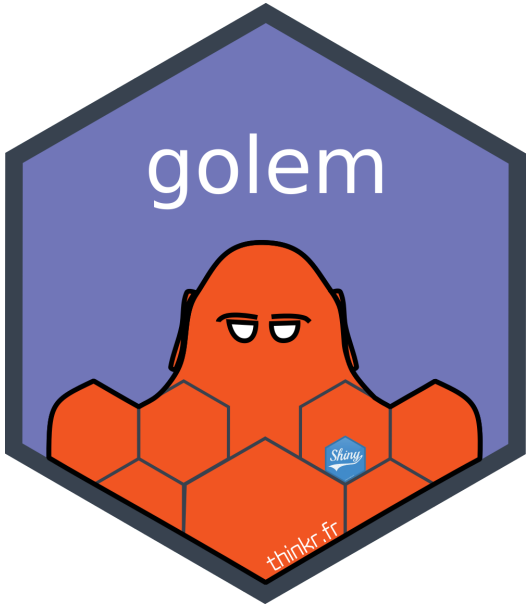
Building Successful Shiny Apps with {golem}

Colin Fay - ThinkR

PART 01 - ABOUT GOLEM



Here comes {golem}



`{golem}` is an **R package** that contains a framework for building production-ready Shiny Applications.

Why {golem}?

Why using `{golem}`?

- Automate the ~~boring stuff~~ repetitive tasks
- Work with reliable tools
- Gain time developing
- Simplify deployment
- Standardize team work

About `{golem}` at ThinkR:

- Internal need, used on a daily basis
- Need reliable tooling for deploying to our clients' environments
- Build and share good practices globally
- Promote R & Shiny in production

Shiny App As a Package 📦

What's a "prod-ready" Shiny App?

- Has meta data (`DESCRIPTION`)
- Divided in functions (`R/`)
- Tested (`tests/`)
- With dependencies (`NAMESPACE`)
- Documented (`man/` & `vignettes`)

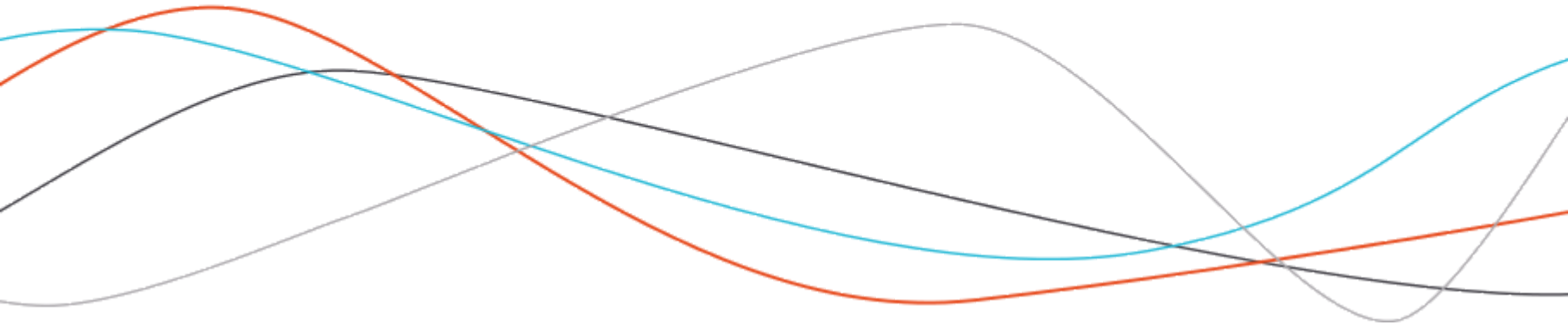
Shiny App As a Package

The plus side: everything you know about package development works with `{golem}`.

Notably:

- 📖 Documentation
- 🔨 Testing

Understanding {golem}



Warning

```
packageVersion("golem")
```

```
[1] '0.2.0'
```

If not

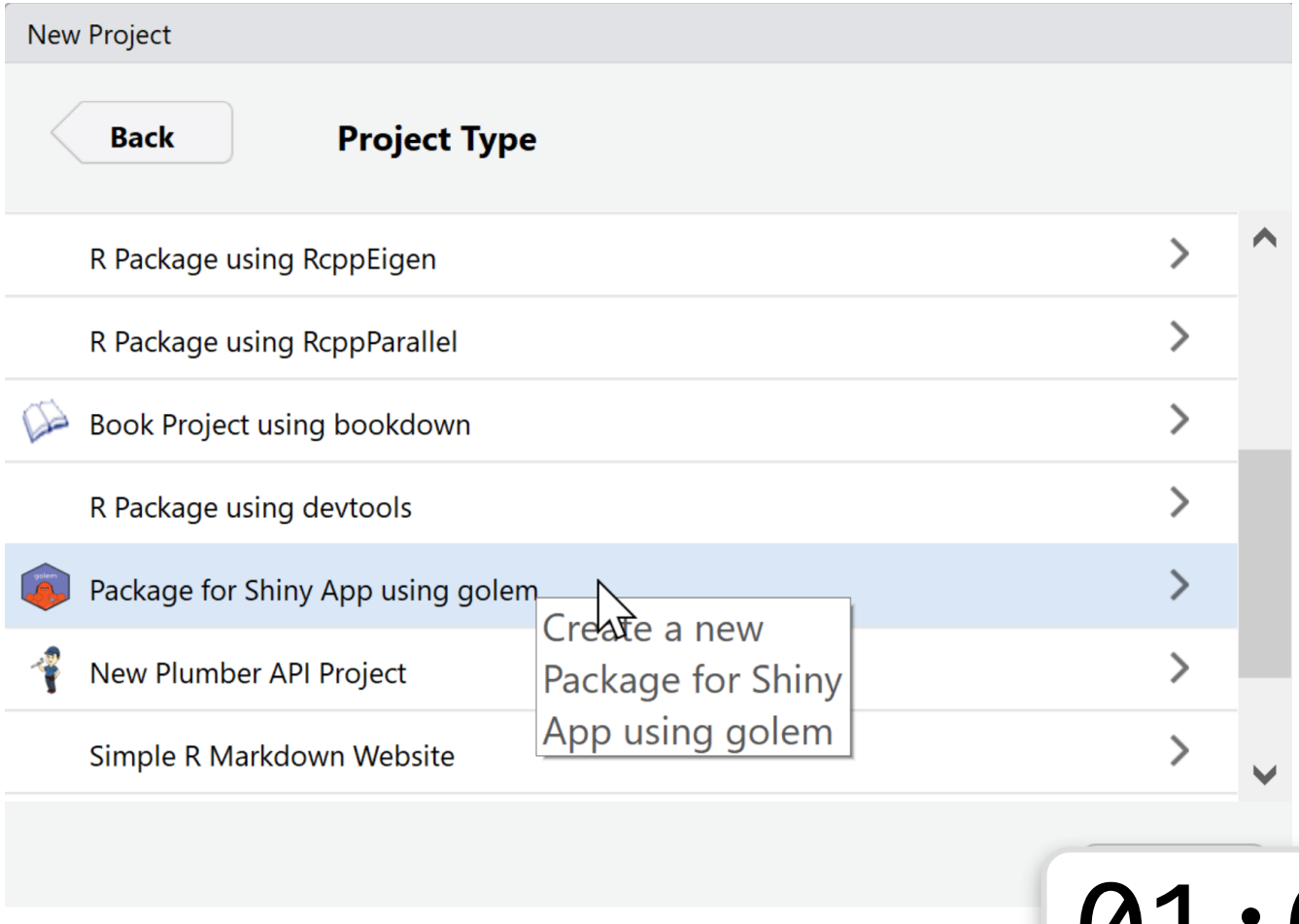
```
install.package("golem")
```

If still not

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

02:00

Create a {golem}



01:00

Understanding {golem}

```
fs::dir_tree("golex")
```

```
golex
├── DESCRIPTION
├── NAMESPACE
├── R
|   ├── app_config.R
|   ├── app_server.R
|   ├── app_ui.R
|   └── run_app.R
├── dev
|   ├── 01_start.R
|   ├── 02_dev.R
|   ├── 03_deploy.R
|   └── run_dev.R
├── inst
|   ├── app
|   |   └── www
|   |       └── favicon.ico
|   └── golem-config.yml
└── man
    └── run_app.Rd
```

Understanding {golem}

Standard package files (i.e. not `{golem}`-specific):

- `DESCRIPTION`: meta-data
- `NAMESPACE`: exported functions + functions from other 📦
- `R/`: functions (everything in `{golem}` is a function)
- `man/`: documentation

app_server.R

```
#' The application server-side
#'
#' @param input,output,session Internal parameters for {shiny}.
#'     DO NOT REMOVE.
#' @import shiny
#' @noRd
app_server <- function( input, output, session ) {
  # List the first level callModules here

}
```

- server logic
- can be thought of as a drop in replacement of `server.R`
- series of `callModule()` (if used)

app_ui.R

```
#' @param request Internal parameter for `{shiny}`.
#'     DO NOT REMOVE.
#' @import shiny
#' @noRd
app_ui <- function(request) {
  tagList(
    # Leave this function for adding external resources
    golem_add_external_resources(),
    # List the first level UI elements here
    fluidPage(
      h1("golex")
    )
  )
}
```

- UI counterpart
- put UI content after the `# List the first level UI elements here` line

app_ui.R

```
golem_add_external_resources <- function(){

  add_resource_path(
    'www', app_sys('app/www')
  )

  tags$head(
    favicon(),
    bundle_resources(
      path = app_sys('app/www'),
      app_title = 'golex'
    )
    # Add here other external resources
    # for example, you can add shinyalert::useShinyalert()
  )
}
```

- Used to add external resources
- Will integrate CSS and JS in the app

app_config.R

```
#' Access files in the current app
#'
#' @param ... Character vector specifying directory and or file to
#'     point to inside the current package.
#'
#' @noRd
app_sys <- function(...){
  system.file(..., package = "golex")
}
```

- `app_sys("x")` will refer to `inst/x`

app_config.R

```
get_golem_config <- function(
  value,
  config = Sys.getenv("R_CONFIG_ACTIVE", "default"),
  use_parent = TRUE
){
  config::get(
    value = value,
    config = config,
    # Modify this if your config file is somewhere else:
    file = app_sys("golem-config.yml"),
    use_parent = use_parent
  )
}
```

- Retrieves elements from `inst/golem-config.yml`

golem-config.yml

```
default:
  golem_name: golex
  golem_version: 0.0.0.9000
  app_prod: no
production:
  app_prod: yes
dev:
  golem_wd: !expr here::here()
```

- Uses the `{config}` format
- Can be safely ignored if you don't feel like you need it 🤷‍♀️

run_app.R

```
#' Run the Shiny Application
#'
#' @param ... A series of options to be used inside the app.
#'
#' @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(...)
  )
}
```

- Launches the app

run_app.R

> About `with_golem_options`

- Allows to pass arguments to `run_app()`
- will later be callable with `golem::get_golem_options()`

Understanding {golem}

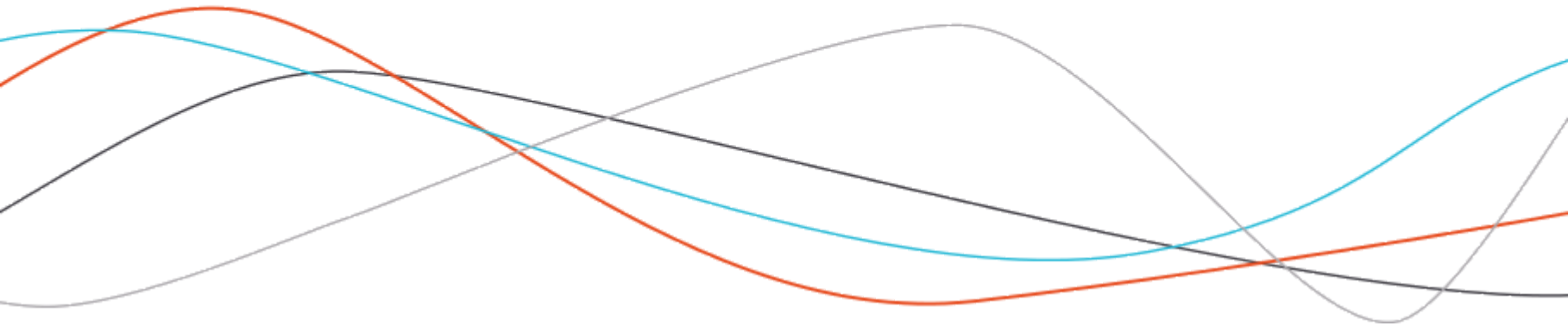
`inst/app/www/`

Host external files, notably the one created with:

- `golem::add_css_file()`
- `golem::add_js_file()`
- `golem::add_js_handler()`
- `golem::use_favicon()`

(More on the later...)

About the dev/ folder



About the dev/ folder

```
fs::dir_tree("golex/dev")
```

```
golex/dev
├── 01_start.R
├── 02_dev.R
├── 03_deploy.R
└── run_dev.R
```

Four files that bundle the golem workflow:

- `01_start.R`: run once at the beginning of the project
- `02_dev.R`: day to day development
- `03_deploy.R`: to use before sending to prod
- `run_dev.R`: to relaunch your app during development

01_start.R

- Fill the `DESCRIPTION` file

```
golem::fill_desc(
  pkg_name = "golex", # The Name of the package containing the App
  pkg_title = "PKG_TITLE", # The Title of the package containing the App
  pkg_description = "PKG_DESC.", # The Description of the package
containing the App
  author_first_name = "AUTHOR_FIRST", # Your First Name
  author_last_name = "AUTHOR_LAST",  # Your Last Name
  author_email = "AUTHOR@MAIL.COM",      # Your Email
  repo_url = NULL # The (optional) URL of the GitHub Repo
)
```

02:00

`01_start.R`

- To be launched for setting elements:
  - `golem::set_golem_options()`
    Fills the `yaml` file
  - `golem::use_recommended_tests()`
    Sets common dependencies

01_start.R

{usethis} commonly used calls

```
usethis::use_mit_license( name = "Golem User" )  # You can set another
license here
usethis::use_readme_rmd( open = FALSE )
usethis::use_code_of_conduct()
usethis::use_lifecycle_badge( "Experimental" )

usethis::use_news_md( open = FALSE )
usethis::use_git()

usethis::use_data_raw( name = "my_dataset", open = FALSE ) # Change
"my_dataset"
```

> We'll come back to data-raw later

`01_start.R`

- `golem::use_recommended_deps()`
- `golem::use_favicon()`
- `golem::use_utils_ui()` & `golem::use_utils_server()`

01_start.R

Your turn!

https://github.com/ColinFay/golem-joburg/tree/master/exo-part-1