

Package: charpente (via r-universe)

May 28, 2026

Title Seamlessly design robust 'shiny' extensions

Version 0.7.1

Description 'charpente' eases the creation of 'shiny' extensions like 'shinydashboard', 'bs4Dash', 'shinyMobile'. It provides helpers to quickly set up a relevant package structure, import all external web dependencies (JavaScript, CSS) as well as initialize input/output bindings and custom handlers boilerplates. 'charpente' offers tools to convert HTML code into R to dramatically speed up the development of the template components as well as an high level interface to 'htmltools'. 'charpente' is a chatty package relying on the same principle as 'usethis' and more recently 'golem', that is make ('shiny') developer's life much easier.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports magrittr, XML, stringr, purrr, htmltools, cli, styler, fs, jsonlite, rstudioapi, usethis, rlang, golem (>= 0.2.1), npm, utils, glue, readr, desc, gh

Remotes JohnCoene/npm

URL <https://github.com/RinterFace/charpente>,
<https://rinterface.github.io/charpente/>

BugReports <https://github.com/RinterFace/charpente/issues>

Suggests testthat

Config/pak/sysreqs

cmake git make libgit2-dev libicu-dev libuv1-dev libxml2-dev libssl-dev libx11-dev zlib1g-dev

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

Date/Publication 2024-07-18 15:05:09 UTC

RemoteUrl <https://github.com/RinterFace/charpente>

RemoteRef HEAD

RemoteSha 0faccd8106a98c8a8061cbd6567db9d65e50bf43

Contents

build_js	2
charpente_options	3
create_charpente	3
create_custom_dependency	4
create_dependency	4
create_input_binding	5
create_manifest	7
create_pwa_dependency	8
get_dependency_assets	8
get_dependency_versions	9
get_installed_dependency	10
get_pkg_name	10
html_2_R	11
set_esbuild	12
set_mocha	12
set_pwa	13
test_js	14
update_dependency	14
Index	15

build_js

Compress and optimize all files in the current folder

Description

Generates a minified file under `inst/pkg_name-pkg_version`, if mode is prod. If mode is dev, aggregates all js files without mangling or compression.

Usage

```
build_js(dir = "srcjs", mode = c("prod", "dev"), entry_points = "main.js")
```

Arguments

dir	Default to srcjs.
mode	Production or development mode. Choose either "prod" or "dev". "prod" bundles, aggregates and minifyies files. "dev" only bundles the code. Modules follow the ES6 format (import/export).
entry_points	Entry point(s) to use in esbuild configuration. In case of a monolithic bundle, only one endpoint is needed. This the default. In case of component based bundles, a vector of endpoints is needed. The output files will match the endpoints names.

charpente_options	<i>Configure charpente</i>
-------------------	----------------------------

Description

Configure charpente

Usage

```
charpente_options(local = TRUE)
```

Arguments

local	Whether to download files locally or to point to a CDN. Default to TRUE.
-------	--

Value

A list of options.

create_charpente	<i>Create a package using charpente and usethis conventions</i>
------------------	---

Description

A charpente powered package does not differ from other package structure. [create_charpente](#) calls a series of function to help you quickly setting up the package structure.

Usage

```
create_charpente(path, remote = NULL, private = FALSE, license)
```

Arguments

path	A path. If it exists, it is used. If it does not exist, it is created, provided that the parent path exists.
remote	Name of the remote Github organization to connect to.
private	Whether the repository is private. Default to FALSE.
license	Which license is your project under?

Examples

```
## Not run:
  create_charpente("bs4Dash")

## End(Not run)
```

create_custom_dependency

Imports Internal Dependencies

Description

Wrap internal scripts and stylesheets in one [htmlDependency](#).

Usage

```
create_custom_dependency(  
  name,  
  version,  
  entry_points,  
  open = interactive(),  
  mode  
)
```

Arguments

name	Package name.
version	Package version.
entry_points	Entry points to create dependency for.
open	Whether to allow rstudioapi to open the newly created script. Default to TRUE.
mode	Internal. Don't use.

create_dependency

Imports External Dependencies

Description

Download and create dependency function.

Usage

```
create_dependency(  
  name,  
  tag = NULL,  
  open = interactive(),  
  options = charpente_options()  
)
```

Arguments

name	Name of library.
tag	Library version. Default to NULL. If NULL, takes the latest version.
open	Whether to allow rstudioapi to open the newly created script. Default to TRUE.
options	See charpente_options .

Examples

```
## Not run:
create_dependency("tabler")
# Use CDNs
create_dependency(
  "framework7",
  options = charpente_options(local = FALSE)
)

## End(Not run)
```

create_input_binding *Create a shiny custom input binding boilerplate*

Description

Creates a script in inst and the R part in ./R

Usage

```
create_input_binding(
  name,
  pkg = ".",
  dir = "srcjs",
  open = TRUE,
  initialize = FALSE,
  dev = FALSE,
  events = list(name = "click", rate_policy = FALSE),
  add_reference = TRUE
)

create_output_binding(
  name,
  pkg = ".",
  dir = "srcjs",
  open = TRUE,
  add_reference = TRUE
)
```

```

create_custom_handler(
  name,
  pkg = ".",
  dir = "srcjs",
  open = TRUE,
  add_reference = TRUE
)

create_js(
  name,
  dir = "srcjs",
  open = TRUE,
  with_doc_ready = FALSE,
  template = golem::js_template,
  ...,
  add_reference = TRUE
)

create_scss(
  name,
  dir = "styles",
  open = TRUE,
  template = golem::sass_template,
  ...,
  add_reference = TRUE
)

```

Arguments

name	The name of the module.
pkg	Path to the root of the package. Default is <code>get_golem_wd()</code> .
dir	Path to the dir where the file will be created.
open	Should the created file be opened?
initialize	For JS file - Whether to add the initialize method. Default to FALSE. Some JavaScript API require to initialize components before using them.
dev	Whether to insert <code>console.log</code> calls in the most important methods of the binding. This is only to help building the input binding. Default is FALSE.
events	List of events to generate event listeners in the subscribe method. For instance, <code>list(name = c("click", "keyup"), rate_policy = c(FALSE, TRUE))</code> . The list contain names and rate policies to apply to each event. If a rate policy is found, the debounce method with a default delay of 250 ms is applied. You may edit manually according to https://shiny.rstudio.com/articles/building-inputs.html
add_reference	Whether to add an import statement in <code>main.js</code> . Defaults to TRUE.
with_doc_ready	For JS file - Should the default file include <code>\$(document).ready()</code> ?
template	Function writing in the created file. You may overwrite this with your own template function.

... Arguments to be passed to the template function.

create_manifest *Create a manifest for your shiny app*

Description

This is a central piece if you want to have your app standalone for instance

Usage

```
create_manifest(
  path,
  name = "My Progressive Web App",
  shortName = "My App",
  description = "What it does!",
  lang = "en-US",
  startUrl = "/",
  display = c("standalone", "minimal-ui", "fullscreen", "browser"),
  background_color = "#ffffff",
  theme_color = "#ffffff"
)
```

Arguments

path	App path.
name	App name.
shortName	App short name.
description	App description
lang	App language (en-US by default).
startUrl	Page to open at start.
display	Display mode. Choose among <code>c("minimal-ui", "standalone", "fullscreen", "browser")</code> . In practice, you want the standalone mode so that the app looks like a native app.
background_color	The <code>background_color</code> property is used on the splash screen when the application is first launched.
theme_color	The <code>theme_color</code> sets the color of the tool bar, and may be reflected in the app's preview in task switchers.

Value

This function creates a `www` folder for your shiny app. Must specify the path. It creates 1 folders to contain icons and the `manifest.json` file.

Note

See <https://developer.mozilla.org/en-US/docs/Web/Manifest> for more informations.

Examples

```
create_manifest(  
  path = tempdir(),  
  name = "My App",  
  shortName = "My App",  
  description = "What it does!",  
  lang = "en-US",  
  startUrl = "https://www.google.com/",  
  display = "standalone",  
  background_color = "#3367D6",  
  theme_color = "#3367D6"  
)
```

create_pwa_dependency *Create a PWA dependency*

Description

List all relevant pwa resources in a custom dependency. Includes the web manifest + icons using the head parameter of `htmltools::htmlDependency`.

Usage

```
create_pwa_dependency(open = interactive())
```

Arguments

`open` Whether to allow `rstudioapi` to open the newly created script. Default to `TRUE`.

get_dependency_assets *Get all links to dependencies*

Description

Query from <https://data.jsdelivr.com/v1/package/npm/> under the hood.

Usage

```
get_dependency_assets(dep, tag = "latest")
```

Arguments

dep	Library name.
tag	Library version. Default to latest.

Value

A list of url containing links to CSS and JS dependencies for the given library.

Examples

```
## Not run:  
get_dependency_assets("bootstrap")  
get_dependency_assets("framework7", tag = "5.5.5")  
  
## End(Not run)
```

get_dependency_versions

Get all version for the current dependency

Description

Query from <https://data.jsdelivr.com/v1/package/npm/> under the hood.

Usage

```
get_dependency_versions(dep, latest = FALSE)
```

Arguments

dep	Library name.
latest	Whether to get the last version. Default to FALSE.

Value

A vector containing all versions

Examples

```
## Not run:  
get_dependency_versions("framework7")  
get_dependency_versions("bootstrap")  
get_dependency_versions("react", latest = TRUE)  
  
## End(Not run)
```

get_installed_dependency

Get the version of the current installed dependency

Description

Used by [update_dependency](#).

Usage

get_installed_dependency(name)

Arguments

name Library name

Value

A character containing the version number

get_pkg_name

Retrieves Package Name

Description

Retrieves Package Name

Usage

get_pkg_name()

Value

Name of package.

`html_2_R`*Convert HTML content to R Shiny tags*

Description

Convert HTML content to R Shiny tags

Usage

```
html_2_R(html, path = "/html/body/*", prefix = TRUE)
```

Arguments

<code>html</code>	HTML string
<code>path</code>	Path where to extract elements. Default to body content.
<code>prefix</code>	Whether to prefix elements by tag\$...

Value

A list of R Shiny tags

Author(s)

Alan Dipert, RStudio

Examples

```
if (interactive()) {
  library(charpente)
  bs4_card <- '<div class="card" style="width: 18rem;">
    
    <div class="card-body">
      <h5 class="card-title">Card title</h5>
      <p class="card-text">Some quick example text.</p>
      <a href="#" class="btn btn-primary">Go somewhere</a>
    </div>
  </div>'
  html_2_R(bs4_card)

  ## With non standard attributes
  tag <- "<div data-toggle='accordion'></div>"
  html_2_R(tag)
}
```

`set_esbuild`*Setup esbuild*

Description

Installs esbuild for the local project

Usage

```
set_esbuild(light = FALSE)
```

Arguments

`light` Used to only install Sass plugins. This is to workaround a breaking change in charpente where styles does not exist in old versions.

Value

Installs esbuild in `node_modules` (dev scope), if not existing, creates `srcjs + srcjs/main.js` and `styles + styles/main.scss`, and sets relevant files and folders in `.gitignore`. and `.Rbuildignore`.

`set_mocha`*Setup mocha*

Description

Installs mocha for the local project

Usage

```
set_mocha()
```

Value

Installs mocha in `node_modules` (dev scope), creates `srcjs/test` folder, write basic test in `test_basic.js`

 set_pwa

Utils to set up a PWA compatible structure

Description

Creates a web manifest, service-worker.js, icons and set the necessary dependencies. The app must be part of a package. Must not be used from the package root but from the app root.

Usage

```
set_pwa(
  path,
  name = "My Progressive Web App",
  shortName = "My App",
  description = "What it does!",
  lang = "en-US",
  startUrl = "/",
  display = c("standalone", "minimal-ui", "fullscreen", "browser"),
  background_color = "#ffffff",
  theme_color = "#ffffff",
  register_service_worker = TRUE,
  create_dependencies = TRUE
)
```

Arguments

path	App path.
name	App name.
shortName	App short name.
description	App description
lang	App language (en-US by default).
startUrl	Page to open at start.
display	Display mode. Choose among <code>c("minimal-ui", "standalone", "fullscreen", "browser")</code> . In practice, you want the standalone mode so that the app looks like a native app.
background_color	The <code>background_color</code> property is used on the splash screen when the application is first launched.
theme_color	The <code>theme_color</code> sets the color of the tool bar, and may be reflected in the app's preview in task switchers.
register_service_worker	Whether to register the service worker. Default to TRUE. Don't change the file name of service-worker.js!!!
create_dependencies	Default to TRUE. Relevant if used in a shinyMobile context. If used outside, you must set it to FALSE and handle the dependencies yourself.

test_js	<i>Test JS code</i>
---------	---------------------

Description

Test the entire srcjs/test folder

Usage

```
test_js()
```

Value

A message showing the test results

update_dependency	<i>Update the given dependency to a specific version or latest</i>
-------------------	--

Description

Update the given dependency to a specific version or latest

Usage

```
update_dependency(name, version_target = "latest")
```

Arguments

name Library name.
version_target Targeted version. Default to latest.

Examples

```
## Not run:  
update_dependency("framework7")  
  
## End(Not run)
```

Index

* Internal

- create_custom_dependency, 4
- build_js, 2
- charpente_options, 3, 5
- create_charpente, 3, 3
- create_custom_dependency, 4
- create_custom_handler
 - (create_input_binding), 5
- create_dependency, 4
- create_input_binding, 5
- create_js (create_input_binding), 5
- create_manifest, 7
- create_output_binding
 - (create_input_binding), 5
- create_pwa_dependency, 8
- create_scss (create_input_binding), 5
- get_dependency_assets, 8
- get_dependency_versions, 9
- get_installed_dependency, 10
- get_pkg_name, 10
- html_2_R, 11
- htmlDependency, 4
- set_esbuild, 12
- set_mocha, 12
- set_pwa, 13
- test_js, 14
- update_dependency, 10, 14