

# Package ‘shiny.pwa’

September 25, 2020

**Type** Package

**Title** Progressive Web App Support for Shiny

**Version** 0.2.0

**Description** Adds Progressive Web App support for Shiny apps, including desktop and mobile installations.

**License** MIT + file LICENSE

**URL** <https://github.com/pedrocoutinhosilva/shiny.pwa>

**Encoding** UTF-8

**LazyData** true

**Imports** shiny, htmltools, readr, utils, urltools

**Suggests** testthat

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**Author** Pedro Silva [aut, cre]

**Maintainer** Pedro Silva <pedrocoutinhosilva@gmail.com>

**Repository** CRAN

**Date/Publication** 2020-09-25 08:20:02 UTC

## R topics documented:

applyTemplate . . . . .	2
createDirectories . . . . .	2
createIcon . . . . .	3
createManifest . . . . .	3
createOfflinePage . . . . .	4
createServiceWorker . . . . .	4
generateFiles . . . . .	5
getTemplate . . . . .	6
loadDependencies . . . . .	6
pwa . . . . .	7

validateDomain . . . . .	8
validateIcon . . . . .	9
validateLocation . . . . .	9

## Index 10

---

applyTemplate	<i>Applies a provided template string to a list of arguments. Any arguments gives will be replaced in the template via placeholders wrapped in «argument»</i>
---------------	---

---

### Description

Applies a provided template string to a list of arguments. Any arguments gives will be replaced in the template via placeholders wrapped in «argument»

### Usage

```
applyTemplate(template, arguments = list())
```

### Arguments

template	A string with placeholders that can be replaced with the given arguments.
arguments	Named list with values used for the template placeholders.

### Value

A string based on the template with the diferent argumetns applied.

---

createDirectories	<i>Generates the folder structure required for the pwa files.</i>
-------------------	---

---

### Description

Generates the folder structure required for the pwa files.

### Usage

```
createDirectories()
```

### See Also

[createServiceWorker()], [createIcon()], [createOfflinePage()], [createManifest()]

Other fileGen: [createIcon\(\)](#), [createManifest\(\)](#), [createOfflinePage\(\)](#), [createServiceWorker\(\)](#)

---

createIcon	<i>Creates the pwa icon file based on the given path.</i>
------------	---

---

**Description**

Creates the pwa icon file based on the given path.

**Usage**

```
createIcon(icon)
```

**Arguments**

icon	Path to the icon used for PWA installations.
------	--

**See Also**

[createDirectories()], [createServiceWorker()], [createOfflinePage()], [createManifest()]

Other fileGen: [createDirectories\(\)](#), [createManifest\(\)](#), [createOfflinePage\(\)](#), [createServiceWorker\(\)](#)

---

createManifest	<i>Creates the manifest file.</i>
----------------	-----------------------------------

---

**Description**

Creates the manifest file.

**Usage**

```
createManifest(title, start_url, color)
```

**Arguments**

title	title of the page when the pwa is running in offline mode.
start_url	The ull url where the app is hosted.
color	An argument that can be used in the offline template to show a custom message.

**See Also**

[createDirectories()], [createServiceWorker()], [createIcon()], [createOfflinePage()]

Other fileGen: [createDirectories\(\)](#), [createIcon\(\)](#), [createOfflinePage\(\)](#), [createServiceWorker\(\)](#)

---

`createOfflinePage`      *Creates the offline landing page.*

---

**Description**

Creates the offline landing page.

**Usage**

```
createOfflinePage(title, template, offline_message)
```

**Arguments**

<code>title</code>	title of the page when the pwa is running in offline mode.
<code>template</code>	HTML template to use. If null a default template is used.
<code>offline_message</code>	An argument that can be used in the offline template to show a custom message.

**See Also**

[[createDirectories\(\)](#)], [[createServiceWorker\(\)](#)], [[createIcon\(\)](#)], [[createManifest\(\)](#)]

Other fileGen: [createDirectories\(\)](#), [createIcon\(\)](#), [createManifest\(\)](#), [createServiceWorker\(\)](#)

---

`createServiceWorker`      *Creates the service worker file based of the package template file.*

---

**Description**

Creates the service worker file based of the package template file.

**Usage**

```
createServiceWorker(output)
```

**Arguments**

<code>output</code>	Relative folder where to create the service worker file. Usually corresponds to the folder used by shiny to serve static files, this folder must exist and is usually the www folder of your shiny project.
---------------------	---

**See Also**

[[createDirectories\(\)](#)], [[createIcon\(\)](#)], [[createOfflinePage\(\)](#)], [[createManifest\(\)](#)]

Other fileGen: [createDirectories\(\)](#), [createIcon\(\)](#), [createManifest\(\)](#), [createOfflinePage\(\)](#)

---

generateFiles	<i>Generates the required files for enabling the pwa.</i>
---------------	---

---

### Description

Generates the required files for enabling the pwa.

### Usage

```
generateFiles(  
  title,  
  domain,  
  location,  
  output,  
  offline_template,  
  offline_message,  
  icon,  
  color  
)
```

### Arguments

title	The title of your Shiny app
domain	the base URL where the app is hosted
location	subdirectory where the app is hosted. Only required if the app is not on the root domain.
output	Relative folder where to create the service worker file. Usually corresponds to the folder used by shiny to serve static files, this folder must exist and is usually the www folder of your shiny project.
offline_template	Path to the offline template you want to use. If left NULL the default template is used.
offline_message	When using the default offline page template, message to be displayed.
icon	Icon Path to be used for the app. Size should be 512x512px. If left NULL a default icon is provided.
color	Color of the app. Used to color the browser elements when the pwa is installed.

### Value

A UI definition that can be passed to the [shinyUI] function.

---

getTemplate	<i>Get the full path for a default file template</i>
-------------	--

---

**Description**

Get the full path for a default file template

**Usage**

```
getTemplate(file)
```

**Arguments**

file	Name of the file including the file extension.
------	--

---

loadDependencies	<i>Returns a set of tags required for registering the pwa.</i>
------------------	--

---

**Description**

Returns a set of tags required for registering the pwa.

**Usage**

```
loadDependencies(title, icon, color, location)
```

**Arguments**

title	Title of the app
icon	Name of the icon file to be user
color	color used for the UI on the installed pwa
location	subdirectory where the app is hosted. Only required if the app is not on the root domain.

**Value**

A UI definition that can be passed to the [shinyUI] function.

**See Also**

[generateFiles()]

---

pwa	<i>Adds PWA support. Takes care of all the support required for browsers to recognise the application as a Progressive Web app.</i>
-----	---

---

### Description

Adds PWA support. Takes care of all the support required for browsers to recognise the application as a Progressive Web app.

### Usage

```
pwa(  
  domain,  
  title = "My Shiny PWA",  
  output = NULL,  
  icon = NULL,  
  color = "#000000",  
  offline_template = NULL,  
  offline_message = "Looks like you are offline :/"  
)
```

### Arguments

domain	The base URL where the app is hosted
title	The title of your Shiny app
output	Relative folder where to create the service worker file. Usually corresponds to the folder used by shiny to serve static files, this folder must exist and is usually the www folder of your shiny project.
icon	Icon Path to be used for the app. Size should be 512x512px. If left NULL a default icon is provided.
color	Color of the app. Used to color the browser elements when the pwa is installed.
offline_template	Path to the offline template you want to use. If left NULL the default template is used.
offline_message	When using the default offline page template defines the message to be displayed.

### Value

A UI definition that can be passed to the [shinyUI] function.

### Note

For additional details on progressive web apps, visit [https://developer.mozilla.org/en-US/docs/Web/Progressive\\_web\\_apps](https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps)

## Examples

```
if (interactive()){  
  library(shiny.pwa)  
  ui <- fluidPage(  
    pwa("https://myapp.com", output = "www")  
  )  
  server <- function(input, output, session) {  
  }  
  shinyApp(ui, server)  
}
```

---

validateDomain	<i>Clears and validates the provided domain.</i>
----------------	--

---

## Description

Clears and validates the provided domain.

## Usage

```
validateDomain(domain)
```

## Arguments

domain            A string representing the domain of the application

## Value

A validated version of the domain

## See Also

[[validateLocation\(\)](#)], [[validateIcon\(\)](#)]

Other validators: [validateIcon\(\)](#), [validateLocation\(\)](#)



---

validateIcon	<i>Validates the provided icon. If the icon doesnt exist returns a default one.</i>
--------------	---

---

**Description**

Validates the provided icon. If the icon doesnt exist returns a default one.

**Usage**

```
validateIcon(icon)
```

**Arguments**

icon	Path location for an icon relative to the project root
------	--

**Value**

A valid icon path

**See Also**

[validateDomain()], [validateLocation()]

Other validators: [validateDomain\(\)](#), [validateLocation\(\)](#)

---

validateLocation	<i>Clears and validates the provided location</i>
------------------	---

---

**Description**

Clears and validates the provided location

**Usage**

```
validateLocation(location)
```

**Arguments**

location	a string representing the domain subfolder where the appication is deployed.
----------	--

**Value**

A validated version of the location

**See Also**

[validateDomain()], [validateIcon()]

Other validators: [validateDomain\(\)](#), [validateIcon\(\)](#)

# Index

## \* **dependencies**

loadDependencies, 6

## \* **fileGen**

createDirectories, 2

createIcon, 3

createManifest, 3

createOfflinePage, 4

createServiceWorker, 4

## \* **validators**

validateDomain, 8

validateIcon, 9

validateLocation, 9

applyTemplate, 2

createDirectories, 2, 3, 4

createIcon, 2, 3, 3, 4

createManifest, 2, 3, 3, 4

createOfflinePage, 2–4, 4

createServiceWorker, 2–4, 4

generateFiles, 5

getTemplate, 6

loadDependencies, 6

pwa, 7

validateDomain, 8, 9

validateIcon, 8, 9, 9

validateLocation, 8, 9, 9