

# Package ‘goodpractice’

October 13, 2022

**Title** Advice on R Package Building

**Version** 1.0.4

**Description** Give advice about good practices when building R packages.  
Advice includes functions and syntax to avoid, package structure, code complexity, code formatting, etc.

**License** MIT + file LICENSE

**URL** <https://github.com/mangothecat/goodpractice>

**BugReports** <https://github.com/mangothecat/goodpractice/issues>

**Imports** clisymbols, covr, crayon, cyclocomp (>= 1.1.0), desc, jsonlite, lintr (>= 3.0.0), praise, rcmdcheck, rstudioapi, tools, utils, whoami, withr, xml2, xmlparsedata (>= 1.0.1)

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Encoding** UTF-8

**RoxygenNote** 7.1.2

**Config/testthat/edition** 3

**Collate** 'api.R' 'customization.R' 'lists.R' 'chk\_covr.R' 'chk\_cyclocomp.R' 'chk\_description.R' 'chk\_lintr.R' 'chk\_namespace.R' 'chk\_rcmdcheck.R' 'chk\_tnf.R' 'gp.R' 'my\_linters.R' 'package.R' 'prep\_covr.R' 'prep\_cyclocomp.R' 'prep\_description.R' 'prep\_expressions.R' 'prep\_lintr.R' 'prep\_namespace.R' 'prep\_rcmdcheck.R' 'print.R' 'rstudio\_markers.R' 'utils.R'

**NeedsCompilation** no

**Author** Ascent Digital Services UK Limited [cph] (MangoTheCat),  
Karina Marks [aut, cre] (KarinaMarks),  
Daniel de Bortoli [aut] (ddbortoli),  
Gabor Csardi [aut],  
Hannah Frick [aut],  
Owen Jones [aut] (owenjonesuob),  
Hannah Alexander [aut],

Ana Simmons [ctb] (anasimmons),  
 Fabian Scheipl [ctb] (fabian-s)

**Maintainer** Karina Marks <karina.marks@ascent.io>

**Repository** CRAN

**Date/Publication** 2022-08-30 19:00:02 UTC

## R topics documented:

goodpractice-package . . . . .	2
all_checks . . . . .	3
checks . . . . .	3
customization . . . . .	4
export_json . . . . .	5
failed_checks . . . . .	5
failed_positions . . . . .	6
gp . . . . .	7
print.goodPractice . . . . .	8
results . . . . .	8

**Index** **10**

goodpractice-package *goodpractice: Advice on R Package Building*

## Description

Give advice about good practices when building R packages. Advice includes functions and syntax to avoid, package structure, code complexity, code formatting, etc.

## Author(s)

**Maintainer:** Karina Marks <karina.marks@ascent.io> (KarinaMarks)

Authors:

- Daniel de Bortoli (ddbortoli)
- Gabor Csardi <csardi.gabor@gmail.com>
- Hannah Frick <hannah.frick@gmail.com>
- Owen Jones <owenjonesuob@gmail.com> (owenjonesuob)
- Hannah Alexander <halexander@mango-solutions.com>

Other contributors:

- Ascent Digital Services UK Limited (MangoTheCat) [copyright holder]
- Ana Simmons <ana.simmons@ascent.io> (anasimmons) [contributor]
- Fabian Scheipl (fabian-s) [contributor]

**See Also**

Useful links:

- <https://github.com/mangothecat/goodpractice>
- Report bugs at <https://github.com/mangothecat/goodpractice/issues>

---

all_checks	<i>List the names of all checks</i>
------------	-------------------------------------

---

**Description**

List the names of all checks

**Usage**

```
all_checks()
```

**Value**

Character vector of checks

---

checks	<i>List all checks performed</i>
--------	----------------------------------

---

**Description**

List all checks performed

**Usage**

```
checks(gp)
```

**Arguments**

gp [gp](#) output.

**Value**

Character vector of check names.

**See Also**

Other API: [failed\\_checks\(\)](#), [results\(\)](#)

**Examples**

```
path <- system.file("bad1", package = "goodpractice")
# run a subset of all checks available
g <- gp(path, checks = all_checks()[3:16])
checks(g)
```

---

 customization

*Defining custom preparations and checks*


---

**Description**

Defining custom preparations and checks

**Usage**

```
make_prep(name, func)

make_check(description, check, gp, ...)
```

**Arguments**

name	Name of the preparation function.
func	A function that takes two arguments: The path to the root directory of the package, and a logical argument: quiet. If quiet is true, the preparation function may print out diagnostic messages. The output of this function will be saved as the " name" entry of state, i.e. of the input for the check-functions (see example).
description	A description of the check.
check	A function that takes the state as an argument.
gp	A short description of what is good practice.
...	Further arguments. Most important: A preps argument that contains the names of all the preparation functions required for the check.

**Functions**

- make\_prep: Create a preparation function
- make\_check: Create a check function

**Examples**

```
# make a preparation function
url_prep <- make_prep(
  name = "desc",
  func = function(path, quiet) desc::description$new(path)
)
# and the corresponding check function
```

```

url_chk <- make_check(
  description = "URL field in DESCRIPTION",
  tags = character(),
  preps = "desc",
  gp = "have a URL field in DESCRIPTION",
  check = function(state) state$desc$has_fields("URL")
)
# use together in gp():
# (note that you have to list the name of your custom check in
# the checks-argument as well...)
bad1 <- system.file("bad1", package = "goodpractice")
res <- gp(bad1, checks = c("url", "no_description_depends"),
  extra_preps = list("desc" = url_prep),
  extra_checks = list("url" = url_chk))

```

---

export_json	<i>Export failed checks to JSON</i>
-------------	-------------------------------------

---

### Description

Export failed checks to JSON

### Usage

```
export_json(gp, file, pretty = FALSE)
```

### Arguments

gp	gp output.
file	Output connection or file.
pretty	Whether to pretty-print the JSON.

---

failed_checks	<i>Names of the failed checks</i>
---------------	-----------------------------------

---

### Description

Names of the failed checks

### Usage

```
failed_checks(gp)
```

### Arguments

gp	gp output.
----	------------

**Value**

Names of the failed checks.

**See Also**

Other API: [checks\(\)](#), [results\(\)](#)

**Examples**

```
path <- system.file("bad1", package = "goodpractice")
# run a subset of all checks available
g <- gp(path, checks = all_checks()[3:16])
failed_checks(g)
```

---

failed_positions	<i>Positions of check failures in the source code</i>
------------------	---

---

**Description**

Note that not all checks refer to the source code. For these the result will be NULL.

**Usage**

```
failed_positions(gp)
```

**Arguments**

gp                    [gp](#) output.

**Details**

For the ones that do, the results is a list, one for each failure. Since the same check can fail multiple times. A single failure is a list with entries: filename, line\_number, column\_number, ranges. ranges is a list of pairs of start and end positions for each line involved in the check.

**Value**

A list of lists of positions. See details below.

---

gp *Run good practice checks*

---

## Description

To see the results, just print it to the screen.

## Usage

```
gp(  
  path = ".",  
  checks = all_checks(),  
  extra_preps = NULL,  
  extra_checks = NULL,  
  quiet = TRUE  
)
```

## Arguments

path	Path to a package root.
checks	Character vector, the checks to run. Defaults to all checks. Use <a href="#">all_checks</a> to list all checks.
extra_preps	Custom preparation functions. See <a href="#">make_prep</a> on creating preparation functions.
extra_checks	Custom checks. See <a href="#">make_check</a> on creating checks.
quiet	Whether to suppress output from the preparation functions. Note that not all preparation functions produce output, even if this option is set to FALSE.

## Value

A goodpractice object that you can query with a simple API. See [results](#) to start.

## Examples

```
path <- system.file("bad1", package = "goodpractice")  
# run a subset of all checks available  
g <- gp(path, checks = all_checks()[3:16])  
g
```

---

```
print.goodPractice      Print goodpractice results
```

---

### Description

Print goodpractice results

### Usage

```
## S3 method for class 'goodPractice'
print(x, positions_limit = 5, ...)
```

### Arguments

`x`                    Object of class `goodPractice`, as returned by `gp()`.

`positions_limit`      How many positions to print at most.

`...`                 Unused, for compatibility with `base::print()` generic method.

---

```
results                 Return all check results in a data frame
```

---

### Description

Return all check results in a data frame

### Usage

```
results(gp)
```

### Arguments

`gp`                    `gp` output.

### Value

Data frame, with columns:

`check`                The name of the check.

`result`               Logical, whether it has failed or not.

### See Also

Other API: `checks()`, `failed_checks()`



### Examples

```
path <- system.file("bad1", package = "goodpractice")
# run a subset of all checks available
g <- gp(path, checks = all_checks()[3:16])
results(g)
```

# Index

## \* API

- checks, 3
- failed\_checks, 5
- results, 8

all\_checks, 3, 7

base::print(), 8

checks, 3, 6, 8

customization, 4

export\_json, 5

failed\_checks, 3, 5, 8

failed\_positions, 6

goodpractice(gp), 7

goodpractice-package, 2

gp, 3, 5, 6, 7, 8

gp(), 8

make\_check, 7

make\_check(customization), 4

make\_prep, 7

make\_prep(customization), 4

print.goodPractice, 8

results, 3, 6, 7, 8