

Package ‘goodpractice’

July 22, 2025

Title Advice on R Package Building

Version 1.0.5

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://docs.ropensci.org/goodpractice/>,
<https://github.com/ropensci-review-tools/goodpractice>

BugReports <https://github.com/ropensci-review-tools/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 7.3.1

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 Mark Padgham [aut, cre] (ORCID:
<<https://orcid.org/0000-0003-2172-5265>>),
Ascent Digital Services UK Limited [cph] (GitHub: MangoTheCat),
Karina Marks [aut] (GitHub: KarinaMarks),
Daniel de Bortoli [aut] (GitHub: ddbortoli),
Gabor Csardi [aut],

Hannah Frick [aut],
Owen Jones [aut] (GitHub: owenjonesuob),
Hannah Alexander [aut],
Ana Simmons [ctb] (GitHub: anasimmons),
Fabian Scheipl [ctb] (GitHub: fabian-s)

Maintainer Mark Padgham <mark@ropensci.org>

Repository CRAN

Date/Publication 2024-06-04 15:40:06 UTC

Contents

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

Index	10
--------------	-----------

goodpractice-package	<i>goodpractice: Advice on R Package Building</i>
----------------------	---

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: Mark Padgham <mark@ropensci.org> (**ORCID**)

Authors:

- Karina Marks <karina.marks@ascent.io> (KarinaMarks)
- 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://ropensci-review-tools.github.io/goodpractice/>
- <https://github.com/ropensci-review-tools/goodpractice>
- Report bugs at <https://github.com/ropensci-review-tools/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

Export failed checks to JSON

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	<i>Run good practice checks</i>
----	---------------------------------

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 all_checks to list all checks.
extra_preps	Custom preparation functions. See make_prep on creating preparation functions.
extra_checks	Custom checks. See make_check 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
```

<code>print.goodPractice</code>	<i>Print goodpractice results</i>
---------------------------------	-----------------------------------

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.

<code>results</code>	<i>Return all check results in a data frame</i>
----------------------	---

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, [6](#)
- results, [8](#)

all_checks, [3](#), [7](#)

base::print(), [8](#)

checks, [3](#), [6](#), [8](#)

customization, [4](#)

export_json, [5](#)

failed_checks, [4](#), [6](#), [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, [4](#), [6](#), [7](#), [8](#)