

Package: audit.workbench (via r-universe)

October 5, 2026

Type Package

Title RStudio/Workbench User Acceptance Tests

Version 0.6.9

Description Testing whether data scientists can do what they expect in
Posit Workbench.

License file LICENSE

URL <https://jumpingrivers.github.io/audit.workbench/>

Imports audit.base (>= 0.6.26), BiocManager, cli, dplyr, fs, glue,
httr, processx, purrr, quarto (>= 1.3), remotes, rlang,
rmarkdown, stringr, tibble, utils, withr, yaml

Suggests reticulate, rstudioapi, rvest, testthat (>= 3.0.0)

Remotes jumpingrivers/audit.base

Config/roxygen2/version 8.0.0

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

LazyData false

Config/pak/sysreqs

cmake git make libicu-dev libuv1-dev libxml2-dev libssl-dev libnode-dev libx11-dev

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

Date/Publication 2026-06-15 12:50:53 UTC

RemoteUrl <https://github.com/jumpingrivers/audit.workbench>

RemoteRef HEAD

RemoteSha 5dcb8dab1d62a53f341d83f206a634115d1a2d5e

Contents

check	2
check_bioconductor	2

check_core_r_pkgs	3
check_cran	4
check_file_permissions	4
check_git_cloning	5
check_github	6
check_python_pip	6
check_python_reticulate	7
check_rmd	8
create_config	8
Index	9

check	<i>Run UAT Checks</i>
-------	-----------------------

Description

This functions runs all UAT tests. To skip tests, set check to ‘no’ in the config yaml file. See ‘create_config()’

Usage

```
check(server, dir = ".", debug_level = 0:2)
```

Arguments

server	Posit Workbench server. If NULL, use the ENV variable WORKBENCH_SERVER
dir	directory location of the the config file
debug_level	Integer, 0 to 2.

Details

Debug level description * 0: clean-up all files; suppress all noise * 1: clean-up all files, but display build steps * 2: No clean-up and display build steps

check_bioconductor	<i>R6 classes</i>
--------------------	-------------------

Description

Check classes

Super classes

```
audit.base::logger -> audit.base::base_check -> check_bioconductor
```

Methods**Public methods:**

- `check_bioconductor$check()`
- `check_bioconductor$clone()`

`check_bioconductor$check()`: Checks that Bioconductor URLs are accessible

Usage:

`check_bioconductor$check(debug_level)`

Arguments:

`debug_level` See `check()` for details

`check_bioconductor$clone()`: The objects of this class are cloneable with this method.

Usage:

`check_bioconductor$clone(deep = FALSE)`

Arguments:

`deep` Whether to make a deep clone.

check_core_r_pkgs	R6 classes
-------------------	------------

Description

Checks that core R packages are pre-installed as described in https://docs.posit.co/ide/server-pro/reference/r_package_dependencies.html

Super classes

`audit.base::logger` -> `audit.base::base_check` -> `check_core_r_pkgs`

Methods**Public methods:**

- `check_core_r_pkgs$check()`
- `check_core_r_pkgs$clone()`

`check_core_r_pkgs$check()`: Test for core R packages

Usage:

`check_core_r_pkgs$check(debug_level)`

Arguments:

`debug_level` See `check()` for details

`check_core_r_pkgs$clone()`: The objects of this class are cloneable with this method.

Usage:

`check_core_r_pkgs$clone(deep = FALSE)`

Arguments:

`deep` Whether to make a deep clone.

 check_cran

R6 classes

Description

Check classes

Super classes

`audit.base::logger` -> `audit.base::base_check` -> `check_cran`

Methods

Public methods:

- `check_cran$check()`
- `check_cran$clone()`

`check_cran$check()`: Installs a package from CRAN

Usage:

`check_cran$check(debug_level)`

Arguments:

`debug_level` See `check()` for details

`check_cran$clone()`: The objects of this class are cloneable with this method.

Usage:

`check_cran$clone(deep = FALSE)`

Arguments:

`deep` Whether to make a deep clone.

 check_file_permissions

R6 classes

Description

Check classes

Super classes

`audit.base::logger` -> `audit.base::base_check` -> `check_file_permissions`

Methods**Public methods:**

- [check_file_permissions\\$check\(\)](#)
- [check_file_permissions\\$clone\(\)](#)

`check_file_permissions$check()`: Check

Usage:

`check_file_permissions$check(debug_level)`

Arguments:

`debug_level` See `check()` for details

`check_file_permissions$clone()`: The objects of this class are cloneable with this method.

Usage:

`check_file_permissions$clone(deep = FALSE)`

Arguments:

`deep` Whether to make a deep clone.

check_git_cloning	R6 classes
-------------------	------------

Description

Check classes

Super classes

`audit.base::logger` -> `audit.base::base_check` -> `check_git_cloning`

Methods**Public methods:**

- [check_git_cloning\\$check\(\)](#)
- [check_git_cloning\\$clone\(\)](#)

`check_git_cloning$check()`: Checking that we can access and clone from github.com

Usage:

`check_git_cloning$check(debug_level)`

Arguments:

`debug_level` See `'check()'` for details

`check_git_cloning$clone()`: The objects of this class are cloneable with this method.

Usage:

`check_git_cloning$clone(deep = FALSE)`

Arguments:

`deep` Whether to make a deep clone.

`check_github`*R6 classes*

Description

Check classes

Super classes

`audit.base::logger` -> `audit.base::base_check` -> `check_github`

Methods**Public methods:**

- `check_github$check()`
- `check_github$clone()`

`check_github$check()`: Checks installs a package from github#

Usage:

`check_github$check(debug_level)`

Arguments:

`debug_level` See `check()` for details

`check_github$clone()`: The objects of this class are cloneable with this method.

Usage:

`check_github$clone(deep = FALSE)`

Arguments:

`deep` Whether to make a deep clone.

`check_python_pip`*R6 classes*

Description

Check classes

Super classes

`audit.base::logger` -> `audit.base::base_check` -> `check_python`

Methods**Public methods:**

- [check_python\\$check\(\)](#)
- [check_python\\$clone\(\)](#)

`check_python$check()`: Checks python pip install

Usage:

`check_python$check(debug_level)`

Arguments:

`debug_level` See `check()` for details

`check_python$clone()`: The objects of this class are cloneable with this method.

Usage:

`check_python$clone(deep = FALSE)`

Arguments:

`deep` Whether to make a deep clone.

check_python_reticulate

R6 classes

Description

Check classes

Super classes

[audit.base::logger](#) -> [audit.base::base_check](#) -> `check_python`

Methods**Public methods:**

- [check_python\\$check\(\)](#)
- [check_python\\$clone\(\)](#)

`check_python$check()`: Checks python can be used via reticulate

Usage:

`check_python$check(debug_level)`

Arguments:

`debug_level` See `check()` for details

`check_python$clone()`: The objects of this class are cloneable with this method.

Usage:

`check_python$clone(deep = FALSE)`

Arguments:

`deep` Whether to make a deep clone.

check_rmd

R6 Quarto checks

Description

A short description...

Usage

types

create_config

Create test config

Description

This function creates an example config. By default all tests are TRUE. Some RSC servers intentionally won't have some features implemented.

Usage

```
create_config(dir = ".", default = TRUE, type = c("merge", "force", "error"))
```

Arguments

dir	Directory of the config file
default	Default value for if a test should run (logical)
type	Merge type, see details

Details

If a test is missing from the config file, it is assume to be TRUE. Therefore, the config file can be quite short and just list exceptions. If the config file is missing, then all tests are carried out.

When merging configs, we either * merge the new with existing * force: overwrite existing file * error: if a config file exists, raise an error

Index

`audit.base::base_check`, [2–7](#)
`audit.base::logger`, [2–7](#)

`check`, [2](#)
`check_bioconductor`, [2](#)
`check_core_r_pkgs`, [3](#)
`check_cran`, [4](#)
`check_file_permissions`, [4](#)
`check_git_cloning`, [5](#)
`check_github`, [6](#)
`check_python_pip`, [6](#)
`check_python_reticulate`, [7](#)
`check_quarto_beamer` (`check_cran`), [4](#)
`check_quarto_docx` (`check_cran`), [4](#)
`check_quarto_html` (`check_cran`), [4](#)
`check_quarto_observable` (`check_cran`), [4](#)
`check_quarto_pdf` (`check_cran`), [4](#)
`check_quarto_rsvg_convert` (`check_cran`),
[4](#)
`check_rmd`, [8](#)
`check_rmd_html` (`check_rmd`), [8](#)
`check_rmd_pdf` (`check_rmd`), [8](#)
`check_rmd_word` (`check_rmd`), [8](#)
`create_config`, [8](#)

`types` (`check_rmd`), [8](#)