

Package: audit.connect (via r-universe)

October 16, 2026

Type Package

Title Posit Connect Health Check

Version 0.8.15

Description Posit Connect Health Check. Deploys various content types to assess whether Connect is functioning correctly.

License MIT + file LICENSE

Depends R (>= 4.1.0)

Imports audit.base (>= 0.7.6), cli, connectapi (>= 0.12.1), dplyr, fs, httr, jsonlite, lubridate, pins (>= 1.1.0), processx, purrr, R6, quarto (>= 1.3), rlang, rsconnect (>= 1.10.0), stringr, tibble, utils, withr

Suggests glue, gt, gtExtras, htmltools, testthat (>= 3.0.0), yaml

Remotes jumpingrivers/audit.base, jumpingrivers/serverheaders

Config/testthat/edition 3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libxml2-dev libssl-dev libnode-dev libx11-dev

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

Date/Publication 2026-10-16 18:38:26 UTC

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

RemoteRef HEAD

RemoteSha f26a46c77875ce868025403532db596a5ef027fa

Contents

check	2
check_deploy_github	3

check_deploy_pins_rds	4
check_deploy_plumber_api	5
check_deploy_python_flask	6
check_deploy_python_streamlit	7
check_deploy_python_voila	8
check_deploy_quarto_python	8
check_deploy_shiny	9
create_config	10
deploy_python	11
get_quarto_old_users	11
sanitise	12
Index	13

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

Description

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

Usage

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

Arguments

server	Connect server (URL). If NULL, use the ENV variable CONNECT_SERVER.
token	Connect api token. If NULL, use the ENV variable CONNECT_API_KEY
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 (on connect and on disk) and display build steps

check_deploy_github	R6 classes
---------------------	------------

Description

R6 classes

R6 classes

Details

Check classes

Super classes

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

Methods

Public methods:

- `check_deploy_github$check()`
- `check_deploy_github$clone()`

Method `check()`: Deploy a shiny application from github

Usage:

`check_deploy_github$check(debug_level)`

Arguments:

`debug_level` See `check()` for details

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

Usage:

`check_deploy_github$clone(deep = FALSE)`

Arguments:

`deep` Whether to make a deep clone.

check_deploy_pins_rds *R6 classes*

Description

R6 classes

R6 classes

Details

Check classes

Super classes

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

Methods

Public methods:

- `check_deploy_pins_rds$check()`
- `check_deploy_pins_rds$clone()`

Method `check()`: Checks a pin can be written, read and deleted

Usage:

`check_deploy_pins_rds$check(debug_level)`

Arguments:

`debug_level` See `check()` for details

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

Usage:

`check_deploy_pins_rds$clone(deep = FALSE)`

Arguments:

`deep` Whether to make a deep clone.

`check_deploy_plumber_api`*R6 classes*

Description

R6 classes

R6 classes

Details

Check classes

Super classes

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

Methods

Public methods:

- `check_deploy_plumber_api$check()`
- `check_deploy_plumber_api$clone()`

Method `check()`: Checks deployment of a Plumber API

Usage:

```
check_deploy_plumber_api$check(debug_level)
```

Arguments:

`debug_level` See `check()` for details

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

Usage:

```
check_deploy_plumber_api$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

check_deploy_python_flask

R6 classes

Description

R6 classes

R6 classes

Details

Check classes

Super classes

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

Methods

Public methods:

- `check_deploy_python_flask$check()`
- `check_deploy_python_flask$clone()`

Method `check()`: Checks deployment of a Python Flask API

Usage:

`check_deploy_python_flask$check(debug_level)`

Arguments:

`debug_level` See `check()` for details

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

Usage:

`check_deploy_python_flask$clone(deep = FALSE)`

Arguments:

`deep` Whether to make a deep clone.

check_deploy_python_streamlit
R6 classes

Description

R6 classes

R6 classes

Details

Check classes

Super classes

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

Methods

Public methods:

- `check_deploy_python_streamlit$check()`
- `check_deploy_python_streamlit$clone()`

Method `check()`: Checks deployment of Python Streamlit app

Usage:

`check_deploy_python_streamlit$check(debug_level)`

Arguments:

`debug_level` See `check()` for details

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

Usage:

`check_deploy_python_streamlit$clone(deep = FALSE)`

Arguments:

`deep` Whether to make a deep clone.

check_deploy_python_voila

R6 classes

Description

Check classes

Super classes

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

Methods

Public methods:

- `check_deploy_python_voila$check()`
- `check_deploy_python_voila$clone()`

`check_deploy_python_voila$check()`: Checks deployment of a Voila notebook

Usage:

`check_deploy_python_voila$check(debug_level)`

Arguments:

`debug_level` See `check()` for details

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

Usage:

`check_deploy_python_voila$clone(deep = FALSE)`

Arguments:

`deep` Whether to make a deep clone.

check_deploy_quarto_python

R6 classes

Description

R6 classes

R6 classes

Details

Check classes.

We use the `deploy_python()` mechanism for this quarto document, as we need to specify the `requirements.txt` file

Super classes

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

Methods**Public methods:**

- `check_deploy_quarto_python$check()`
- `check_deploy_quarto_python$clone()`

Method `check()`: Deploy a quarto document that uses a jupyter engine

Usage:

`check_deploy_quarto_python$check(debug_level, account = NULL)`

Arguments:

`debug_level` See `check()` for details

`account` Connect username

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

Usage:

`check_deploy_quarto_python$clone(deep = FALSE)`

Arguments:

`deep` Whether to make a deep clone.

check_deploy_shiny *R6 classes*

Description

R6 classes

R6 classes

Details

Check classes

Super classes

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

Methods

Public methods:

- `check_deploy_shiny$check()`
- `check_deploy_shiny$clone()`

Method `check()`: Checks deployment of a Shiny app

Usage:

```
check_deploy_shiny$check(debug_level)
```

Arguments:

`debug_level` See `check()` for details

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

Usage:

```
check_deploy_shiny$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

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

<code>dir</code>	Directory of the config file
<code>default</code>	Default value for if a test should run (logical)
<code>type</code>	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

deploy_python	<i>Deploy Python output</i>
---------------	-----------------------------

Description

Deploy Python output

Usage

```
deploy_python(
    python_dir,
    python_files = c("app.py", "index.qmd", "hash.ipynb"),
    rsconnect_type = c("api", "streamlit", "quarto", "voila"),
    debug_level
)
```

Arguments

python_dir	Location of python deployment files
python_files	An app, a qmd file or a notebook
rsconnect_type	Type of deployment
debug_level	Integer, 0 to 2.

Details

Python deploy function All python content must have a requirements.txt file and either a app.py, index.qmd or hash.ipynb. index.qmd is for quarto only, and hash.ipynb for voila only. General idea is to

- copy dir of contents to R_TMP_DIR
- Use the python interface to deploy
- Clean up, finding the content on Connect by its title. @noRd

get_quarto_old_users	<i>Quarto Helper Functions</i>
----------------------	--------------------------------

Description

In general not used by users.

Usage

```
get_quarto_old_users(out)

get_quarto_user_roles(out)

get_quarto_locked_user_apps(out)
```

Arguments

out Output from check

sanitise *Sanitise audit.connect object*

Description

This function removes user-identifying data from an audit.connect object

Usage

```
sanitise(audit_connect_check)
```

Arguments

audit_connect_check
 An object from `audit.connect::check()`

Index

- audit.base::base_check, [3–9](#)
- audit.base::logger, [3–9](#)

- check, [2](#)
- check_deploy_github, [3](#)
- check_deploy_pins_rds, [4](#)
- check_deploy_plumber_api, [5](#)
- check_deploy_python_flask, [6](#)
- check_deploy_python_streamlit, [7](#)
- check_deploy_python_voila, [8](#)
- check_deploy_quarto_beamer
 - (check_deploy_quarto_python), [8](#)
- check_deploy_quarto_docx
 - (check_deploy_quarto_python), [8](#)
- check_deploy_quarto_html
 - (check_deploy_quarto_python), [8](#)
- check_deploy_quarto_observable
 - (check_deploy_quarto_python), [8](#)
- check_deploy_quarto_pdf
 - (check_deploy_quarto_python), [8](#)
- check_deploy_quarto_python, [8](#)
- check_deploy_quarto_rsvg_convert
 - (check_deploy_quarto_python), [8](#)
- check_deploy_rmd_html
 - (check_deploy_quarto_python), [8](#)
- check_deploy_rmd_pdf
 - (check_deploy_quarto_python), [8](#)
- check_deploy_rmd_word
 - (check_deploy_quarto_python), [8](#)
- check_deploy_shiny, [9](#)
- create_config, [10](#)

- deploy_python, [11](#)

- get_quarto_locked_user_apps
 - (get_quarto_old_users), [11](#)
- get_quarto_old_users, [11](#)
- get_quarto_user_roles
 - (get_quarto_old_users), [11](#)

- sanitise, [12](#)