Package: audit.connect (via r-universe)

August 24, 2024

Type Package

Title Posit Connect Health Check

Version 0.7.6

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

License file LICENSE

Imports audit.base (>= 0.6.15), cli, connectapi (>= 0.2.0), dplyr, fs, httr, jsonlite, lubridate, pins (>= 1.1.0), processx, purrr, quarto (>= 1.3), rlang, rsconnect (>= 1.3.1), stringr, tibble

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 7.3.1

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

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

RemoteRef HEAD

RemoteSha 2b0b577fe537852958c20d526266a02aede5a071

Contents

`check` *Run Posit Checks*

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_pluumber_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 Quarto document with HTML output

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_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 an R Markdown document with Word Docx output

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

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

deploy_python *Deploy Python output*

Description

Deploy Python output

Usage

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

Arguments

python_dir	Location of python deployment files
python_files	Either an app or a qmd file
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 or index.qmd. index.qmd is for quarto only. General idea is to

- copy dir of contents to R_TMP_DIR
- Use the python interface to deploy
- Use the python interface to grab the guid
- Clean up. @noRd

---

get_quarto_old_users *Quarto Helper Functions*

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