

# Package: requal (via r-universe)

May 30, 2026

**Title** Shiny Application for Computer-Assisted Qualitative Data Analysis

**Version** 1.1.3.9001

**Description** PKG\_DESC.

**License** MIT + file LICENSE

**Imports** colourpicker, config (>= 0.3.1), DBI, dbplyr, dplyr, DT, fs, ggplot2, glue, golem (>= 0.3.1), htmltools, htmlwidgets, jsonlite, magrittr, pool, purrr, rlang, RPostgreSQL, RSQLite, shiny (>= 1.6.0), shinydashboard, shinydashboardPlus, shinyFiles, shinyjs, shinymanager, shinyWidgets, sortable, stringr, tibble, tidyr, tidyselect, viridisLite

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.2.3

**Suggests** rmarkdown, knitr, testthat (>= 3.0.0), spelling, shinytest2, rvest, devtools, pkgload, covr

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Language** en-US

**URL** <https://github.com/RE-QDA/requal>

**BugReports** <https://github.com/RE-QDA/requal/issues>

**Repository** <https://re-qda.r-universe.dev>

**Date/Publication** 2025-02-03 19:05:35 UTC

**RemoteUrl** <https://github.com/RE-QDA/requal>

**RemoteRef** v1.1.3

**RemoteSha** 64a0ec44f26bb2c1dfdc849dd626195b4c74a89f

## Contents

rql_import_rqda . . . . .	2
run_app . . . . .	2
sample_colours . . . . .	3

<b>Index</b>	<b>4</b>
--------------	----------

---

rql_import_rqda	<i>Import data from RQDA to Requal</i>
-----------------	--

---

### Description

Import data from RQDA database to requal. It uses tables RQDA tables "project", "source" (documents), "freecode" (codes) and "coding" (coded segments), "cases", "caselinkage", "codecat" (categories). Memos from "source", "freecode", "cases" and "codecat" are used as descriptions. Memos from "coding" are stored as memos.

### Usage

```
rql_import_rqda(rqda_file, requal_file)
```

### Arguments

rqda_file	Path to RQDA database
requal_file	Path to requal database to be created by import, it must not exist

---

run_app	<i>Run the Shiny Application</i>
---------	----------------------------------

---

### Description

Run the Shiny Application

### Usage

```
run_app(
  mode = "local",
  dbhost = "",
  dbport = 5432,
  dbname = NULL,
  dbusername = NULL,
  dbpassword = NULL,
  credentials_path = NULL,
  credentials_pass = NULL,
  uiPattern = "/",
  ...
)
```

**Arguments**

mode	Mode of application (local or server)
dbhost	Host of DB (for server use only)
dbport	Port of DB (for server use only)
dbname	Name of DB (for server use only)
dbusername	Username (for server use only)
dbpassword	Password to DB (for server use only)
credentials_path	Path to credentials DB (for server use only)
credentials_pass	Password to credentials DB (for server use only)
uiPattern	A regular expression that will be applied to each GET request to determine whether the ui should be used to handle the request. Note that the entire request path must match the regular expression in order for the match to be considered successful.
...	arguments to pass to golem_opts. See ‘?golem::get_golem_options‘ for more details.

---

sample\_colours

*Sample colours*


---

**Description**

Sample colours

**Usage**

```
sample_colours(n)
```

**Arguments**

n	N colours to be sampled
---	-------------------------

# Index

rql\_import\_rqda, [2](#)

run\_app, [2](#)

sample\_colours, [3](#)