Package 'rcodeagent'

November 11, 2026

Title Coding Assistant for 'RStudio' with 'ChatGPT' Authentication

Version 0.1.0

Description Provides an interactive coding assistant in the 'RStudio' Viewer to explain R code, draft scripts, and propose changes to a selected editor target. Supports read-only chat, explicitly authorized edits, and user-requested submission of code to the native R console. Authenticates through 'ChatGPT' using browser-based OAuth 2.0 authorization with Proof Key for Code Exchange (PKCE) and stores credentials using the operating system credential store. Model requests require Internet access and an eligible 'ChatGPT' Plus or Pro account under the service's current terms <https://developers.openai.com/siwc/quickstart>. Neither an API key nor 'Codex CLI' is required.

License GPL (>= 3)

Encoding UTF-8

Depends R (>= 4.2.0)

SystemRequirements RStudio Desktop (interactive use), web browser, Internet access, operating system credential store, eligible ChatGPT Plus or Pro account (model requests), Cairo support (PDF export), XQuartz (macOS PDF export)

Imports shiny, htmltools, miniUI, rstudioapi, httpuv, httr2 (>= 1.2.0), openssl, jose (>= 2.0.0), keyring, sodium, jsonlite, commonmark, xml2, callr, filelock (>= 1.0.3), later, utils, grid

Suggests testthat (>= 3.2.1), withr, pkgload

Config/testthat/edition 3

RoxygenNote 7.3.2

NeedsCompilation no

Author José Gamarra-Moncayo [aut, cre]

Maintainer José Gamarra-Moncayo <gamarramoncayoj@gmail.com>

Repository CRAN

Date/Publication 2026-11-11 20:10:02 UTC

Contents

rcodeagent 2

Index **4**

`rcodeagent` *Open the rcodeagent Gadget or Inspect Local Authentication Status*

Description

Starts the RStudio coding assistant in a native Background Job, displays it in the Viewer, and returns control to the R console. The status function inspects saved local credentials without returning tokens.

Usage

```
rcodeagent()
rcodeagent_login()
rcodeagent_status()
```

Details

The gadget and login alias require an interactive RStudio Desktop session. Authentication requires a web browser, Internet access, and an available operating system credential store. Model requests require Internet access and an eligible ChatGPT Plus or Pro account under the service's current terms; see https://developers.openai.com/siwc/quickstart. Signing in alone does not grant permission to use models. Available models, limits, and service availability depend on the authorized account and provider. Normal generates and revises code only in chat. Auto permits edits to the selected target and submits code to the main R console only on explicit request. Explicitly requested ordinary code and package installation do not require a second confirmation. Classified administrative operations require an exact preview and one-use confirmation; RStudio may queue submissions. Stopping the chat does not withdraw submitted code. Saving remains manual. The background process does not import or export the user's global environment. Editor operations use RStudio's native interprocess API with a restricted client which waits for request completion and never resends a write. A busy main R session may delay these operations or cause a timeout. An uncertain operation disables further channel access and revokes Auto; inspect the target and restart the assistant before trying again. Reopening reuses the live job. When other output replaces the Viewer page, the job retains its recoverable chat state in memory for 30 minutes. Reopen the addin to restore the conversation, attachments, model, effort, draft and editing target. Auto retains its previous state only in this temporary return to the same job, after revalidating the saved document ID, path and full contents. Conflicts or an unavailable target revoke Auto. Permissions are never saved in chat history. In-flight requests are cancelled on disconnect and their draft can be resubmitted; no editor writes occur while the page is absent. Explicit Close stops the server and pending workers. A Viewer which never connects stops the job after 60 seconds. Connect using the gadget's login button. The available models and permissions depend on the authorized account. No API key or Codex CLI is required. Auto edits only the selected target. Administrative calls in function defaults are included in the additional review. Changes are not saved to disk automatically. Console output

is not automatically collected or shared with the model. Closing the gadget does not remove saved OAuth credentials. Local status is not a connectivity test. Conversation export to PDF requires Cairo support in R. On macOS, this also requires XQuartz; see https://www.xquartz.org/.

Value

`rcodeagent()` and `rcodeagent_login()` invisibly return the native job ID. `rcodeagent_status()` returns a character string describing local status.

Examples

```
if (interactive() && rstudioapi::isAvailable()) {
  rcodeagent()
  rcodeagent_status()
  # The login alias reopens the same live assistant job.
  rcodeagent_login()
}
```

Index

rcodeagent, 2
rcodeagent_login *(*rcodeagent*)*, 2
rcodeagent_status *(*rcodeagent*)*, 2