

Package ‘ctrlvee’

May 8, 2026

Title Extract External 'R' Code and Insert Inline

Version 0.1.0

Description An 'RStudio' and 'Positron' add-in that prompts the user for a web 'URL', fetches the page content, extracts 'R' code chunks, and inserts those code chunks into the active editor at the current cursor position. Supports extraction of raw 'Markdown' or 'Quarto' source files, 'GitHub' Gist and rendered 'HTML' pages that have markup elements with 'R'-related classes.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports rstudioapi, httr2, xml2, purrr, magrittr

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

URL <https://github.com/vpnagraj/ctrlvee>

BugReports <https://github.com/vpnagraj/ctrlvee/issues>

NeedsCompilation no

Author VP Nagraj [aut, cre, cph]

Maintainer VP Nagraj <nagraj@nagraj.net>

Repository CRAN

Date/Publication 2026-04-21 18:02:14 UTC

Contents

addin_crawl_chunks	2
crawl_chunks	2
crawl_chunks_html	3
crawl_chunks_raw	4
detect_strategy	5
extract_r_chunks_from_html	6
extract_r_chunks_from_markdown	6
Index	8

addin_crawl_chunks *RStudio / Positron Addin: Crawl R Chunks from a URL*

Description

When invoked from the Addins menu or a keyboard shortcut, this function:

Usage

```
addin_crawl_chunks()
```

Details

1. Opens a dialog asking the user for a web URL.
2. Auto-detects the best extraction strategy.
3. Fetches the page and extracts R code chunks.
4. Inserts the extracted code into the active source editor at the current cursor position.

Value

Called for side effects (i.e., inserting text); returns NULL invisibly.

Keyboard shortcut

After installing the package, open **Tools > Modify Keyboard Shortcuts...** (RStudio) or the Command Palette in Positron, search for "*Extract External R Code and Insert Inline*", and assign your preferred shortcut (e.g. Ctrl+Shift+U / Cmd+Shift+U).

Examples

```
if (interactive() && rstudioapi::isAvailable()) {  
  addin_crawl_chunks()  
}
```

crawl_chunks *Fetch R code chunks from any URL*

Description

The primary engine of the ctrlvee package. Give it a URL to a raw source file, a GitHub Gist, or a rendered HTML page and it returns the R code chunks found there.

Usage

```
crawl_chunks(url, strategy = c("auto", "raw", "html"), verbose = TRUE)
```

Arguments

url	Character vector of length 1 with the URL to fetch
strategy	One of "auto" (default), "raw", or "html". When "auto", the strategy is detected from the URL pattern using detect_strategy() .
verbose	Logical; default is TRUE and progress messages are printed

Value

A character vector of R code chunk bodies. Returns character(0) if no R chunks are found.

Examples

```
## rendered Quarto book chapter (HTML strategy detected)
chunks <- crawl_chunks("https://r4ds.hadley.nz/data-visualize.html")

## GitHub Gist ... plain .R file (raw strategy detected)
chunks <- crawl_chunks(
  "https://gist.github.com/vpnagraj/59fa609c5adf47c8c7a5b156eb261be7"
)

## you can also dictate a specific strategy
chunks <- crawl_chunks("https://r4ds.hadley.nz/data-visualize.html", strategy = "html")
```

`crawl_chunks_html` *Fetch and extract R code blocks from a rendered HTML page*

Description

Downloads an HTML page (e.g., a rendered Quarto book chapter) and extracts code blocks that are tagged as R.

Usage

```
crawl_chunks_html(url, verbose = interactive())
```

Arguments

url	Character vector of length 1 with the URL to fetch
verbose	Logical; default is TRUE and progress messages are printed

Details

The function looks for `<code>` elements whose `class` attribute matches known patterns for R source code (e.g., `sourceCode r`, `language-r`). It then grabs the inner text, stripping any syntax highlighting `` elements.

It is important to note that:

- Chunk options (labels, `echo`, `eval`, etc.) are lost in rendered HTML because the renderer strips them.
- Output blocks that the renderer styled identically to source blocks may occasionally be captured. A heuristic filters out blocks where the majority (>50%) of lines look like R console output (e.g., `[1]`, `##`).

Value

A character vector where each element is the text content of one R code block, in document order. Returns `character(0)` if none are found.

Examples

```
## extract R code from a rendered Quarto book chapter
chunks <- crawl_chunks_html("https://r4ds.hadley.nz/data-visualize.html")
length(chunks)
cat(chunks[[1]])
```

<code>crawl_chunks_raw</code>	<i>Fetch and extract R code from a raw source URL</i>
-------------------------------	---

Description

Downloads the raw text content of a URL (typically a `.qmd`, `.Rmd`, `.md`, or `.R` file served as plain text) and extracts R code.

Usage

```
crawl_chunks_raw(url, verbose = TRUE)
```

Arguments

<code>url</code>	Character vector of length 1 with the URL to fetch
<code>verbose</code>	Logical; default is TRUE and progress messages are printed

Details

For Markdown-family files, fenced R code blocks are extracted individually. For plain `.R` files (or any file where no fenced chunks are found), the entire file content is returned as a single chunk.

Value

A character vector where each element is the body of one R chunk. For plain .R files, a vector of length 1 containing the full file. Returns character(0) if the file is empty.

Examples

```
## fenced chunks from a Quarto source file
chunks <- crawl_chunks_raw(
  "https://raw.githubusercontent.com/hadley/r4ds/main/data-visualize.qmd"
)

## plain .R file from a GitHub Gist
chunks <- crawl_chunks_raw(
  "https://gist.githubusercontent.com/vpnagraj/59fa609c5adf47c8c7a5b156eb261be7/raw"
)
```

detect_strategy	<i>Detect the best extraction strategy for a URL</i>
-----------------	--

Description

Examines the URL pattern and returns a strategy string that tells the caller which extraction function to use. This will be one of either: "raw" (URL points directly to a raw source file or a GitHub Gist) or "html" (URL is a rendered web page like a Quarto book chapter or pkgdown site).

Usage

```
detect_strategy(url)
```

Arguments

url Character vector of length 1 with the URL to fetch

Details

Detection is entirely URL-based (no network requests) and intentionally conservative; when in doubt it falls back to "html".

Rules (evaluated in order):

1. Contains gist.githubusercontent.com or matches gist.github.com/<user>/<hash>: "raw"
2. Contains raw.githubusercontent.com or ends in .Rmd, .qmd, .md, .R, or .Rmarkdown: "raw"
3. Everything else: "html"

Value

One of "raw", "github", or "html".

Examples

```
detect_strategy("https://gist.github.com/vpnagraj/59fa609c5adf47c8c7a5b156eb261be7")  
detect_strategy("https://raw.githubusercontent.com/hadley/r4ds/main/data-visualize.qmd")  
detect_strategy("https://r4ds.hadley.nz/data-visualize.html")
```

extract_r_chunks_from_html

Extract R code blocks from an HTML string

Description

Helper function to do HTML-parsing in `crawl_chunks_html()`. Also exported so users can call it on local HTML strings if needed.

Usage

```
extract_r_chunks_from_html(html)
```

Arguments

html A character vector of length 1 containing HTML content

Value

Character vector of extracted R code bodies.

extract_r_chunks_from_markdown

Extract fenced R code chunks from a Markdown text string

Description

A helper to go line-by-line through a Markdown file to find the common R code fence styles used in R Markdown, Quarto, and GitHub Markdown. Note that non-R fenced blocks (e.g., python, sql, etc.) are skipped.

Usage

```
extract_r_chunks_from_markdown(text)
```

Arguments

text Character vector of length 1 with Markdown source text

Value

Character vector of extracted code bodies.

Index

`addin_crawl_chunks`, [2](#)

`crawl_chunks`, [2](#)

`crawl_chunks_html`, [3](#)

`crawl_chunks_html()`, [6](#)

`crawl_chunks_raw`, [4](#)

`detect_strategy`, [5](#)

`detect_strategy()`, [3](#)

`extract_r_chunks_from_html`, [6](#)

`extract_r_chunks_from_markdown`, [6](#)