Package 'shinypayload'

August 29, 2025

Type Package

Title Accept POST Data and URL Parameters in 'shiny' (Same-Port Integration)

Version 0.1.0

Author Pawan Rama Mali [aut, cre]

Maintainer Pawan Rama Mali <prm@outlook.in>

Description Handle POST requests on a custom path (e.g., /ingress) inside the same 'shiny' HTTP server using user interface functions and HTTP responses. Expose latest payload as a reactive and provide helpers for query parameters.

License MIT + file LICENSE

URL https://github.com/PawanRamaMali/shinypayload, https://pawanramamali.github.io/shinypayload/

BugReports https://github.com/PawanRamaMali/shinypayload/issues

Encoding UTF-8

Depends R (>= 4.1)

Imports shiny (>= 1.7.4), jsonlite

Suggests testthat (>= 3.0.0), covr, styler, roxygen2, DT

Config/testthat/edition 3

RoxygenNote 7.3.2

NeedsCompilation no

Repository CRAN

Date/Publication 2025-08-29 18:10:08 UTC

Contents

`params_get` *Get URL query parameters in Shiny*

Description

Get URL query parameters in Shiny

Usage

```
params_get(session, keys = NULL)
```

Arguments

`session`	Shiny session
`keys`	Optional character vector of keys to pull; if NULL return all

Value

A named list containing the URL query parameters. If `keys` is specified, only those parameters are returned. If no parameters exist or the specified keys are not found, returns an empty list or list with `NULL` values respectively.

Examples

```
if (interactive()) {
server <- function(input, output, session) {
  # Get all query parameters
  all_params <- params_get(session)

  # Get specific parameters
  user_params <- params_get(session, keys = c("user_id", "token"))

  # Use in outputs
  output$params_display <- renderText({
    params <- params_get(session)
    if (length(params) > 0) {
      paste("Parameters:", jsonlite::toJSON(params))
    } else {
      "No parameters provided"
    }
  })
}
}
```

payload_endpoint_url *Generate the absolute URL for the payload endpoint*

Description

Generate the absolute URL for the payload endpoint

Usage

```
payload_endpoint_url(session, path = "/ingress")
```

Arguments

session	The Shiny session object
path	The URL path (default "/ingress")

Value

A character string containing the complete URL (including protocol, hostname, port, and path) where POST requests should be sent to reach this endpoint.

Examples

```
if (interactive()) {
server <- function(input, output, session) {
  url <- payload_endpoint_url(session, "/data")
  print(paste("Send POST requests to:", url))
}
}
```

payload_last *Get a reactive that polls for new payload data*

Description

Get a reactive that polls for new payload data

Usage

```
payload_last(path = "/ingress", session, intervalMillis = 300)
```

Arguments

path	The URL path used in payload_ui() (default "/ingress")
session	The Shiny session object
intervalMillis	Polling interval in milliseconds (default 300)

Value

A reactive expression (class "reactive") that returns a list with two elements when new data is available: `payload` (the parsed request body) and `meta` (metadata including timestamp, remote address, headers, etc.), or `NULL` if no data has been received yet.

Examples

```
if (interactive()) {
server <- function(input, output, session) {
  latest_data <- payload_last("/data", session)

  observeEvent(latest_data(), {
    data <- latest_data()
    if (!is.null(data)) {
      print(data$payload)
      print(data$meta$timestamp)
    }
  })
}
}
```

---

payload_ui *Wrap an existing UI with an integrated POST handler on the same port*

Description

Wrap an existing UI with an integrated POST handler on the same port

Usage

```
payload_ui(base_ui, path = "/ingress", token = NULL)
```

Arguments

`base_ui`	The original UI (tagList, fluidPage, or a function(req) returning UI)
`path`	The URL path to handle POST requests (default "/ingress")
`token`	Optional authentication token for POST requests

Value

A function that takes a request object and returns either the regular UI (for GET requests) or an HTTP response (for POST requests). This function should be passed to shinyApp() as the ui parameter.

Examples

```
if (interactive()) {
ui <- payload_ui(
  fluidPage(h1("My App")),
  path = "/data",
  token = "secret123"
)
shinyApp(ui, server, uiPattern = ".*")
}
```

`setup_payload_endpoint` *Setup POST endpoint in server function - MUST be called in server*

Description

Setup POST endpoint in server function - MUST be called in server

Usage

```
setup_payload_endpoint(path = "/ingress", session, token = NULL)
```

Arguments

`path`	The URL path to handle POST requests (default "/ingress")
`session`	The Shiny session object
`token`	Optional authentication token for POST requests

Value

No return value, called for side effects. Registers a POST endpoint handler with the Shiny session that will process incoming requests.

Index