Package: shinyfa (via r-universe)

June 8, 2026

Title Analyze the File Contents of 'shiny' Directories

Version 0.0.1

Description Provides tools for analyzing and understanding the file contents of large 'shiny' application directories. The package extracts key information about render functions, reactive functions, and their inputs from app files, organizing them into structured data frames for easy reference. This streamlines the onboarding process for new contributors and helps identify areas for optimization in complex 'shiny' codebases with multiple files and sourcing chains.

URL https://github.com/dalyanalytics/shinyfa, https://dalyanalytics.github.io/shinyfa/

BugReports https://github.com/dalyanalytics/shinyfa/issues

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports stringr

Suggests dplyr, testthat (>= 3.0.0)

Config/testthat/edition 3

Config/pak/sysreqs libicu-dev

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

Date/Publication 2025-11-10 18:26:45 UTC

RemoteUrl https://github.com/dalyanalytics/shinyfa

RemoteRef HEAD

RemoteSha 9cda78d59ac9a3d73783d948265d5af68f059918

Contents

`analyze_shiny_reactivity` *Analyze Shiny server reactivity in a given file*

Description

This function processes a Shiny server script, identifying render functions, their reactivity types, and any input dependencies.

Usage

```
analyze_shiny_reactivity(file_path)
```

Arguments

`file_path`	A string representing the path to the R file.

Value

A data frame summarizing the reactivity structure of the file.

`classify_reactivity` *Classify the reactivity type of a render block*

Description

Determines whether a given block of Shiny code contains reactive elements such as `reactive()`, `eventReactive()`, `observeEvent()`, etc.

Usage

```
classify_reactivity(render_block)
```

Arguments

`render_block`	A character vector representing a block of Shiny server code.

Value

A string indicating the type of reactivity detected.

`extract_input_dependencies` *Extract input dependencies from a Shiny render block*

Description

Identifies all `input$` dependencies within a render block, including cases like `input$var` and `input[['var']]`.

Usage

```
extract_input_dependencies(render_block)
```

Arguments

`render_block`	A character vector representing a block of Shiny server code.

Value

A string of unique input IDs used within the render block, or `"None"` if no inputs are found.

`extract_named_reactives` *Extract named reactive expressions from a Shiny server file*

Description

Identifies `reactive()` assignments that are stored in variables and returns a data frame. Additionally, it checks for any input dependencies inside the `reactive()` function.

Usage

```
extract_named_reactives(server_code)
```

Arguments

`server_code`	A character vector representing lines of Shiny server code.

Value

A data frame containing named reactive expressions with their corresponding input dependencies.

`extract_output_assignments` *Extract output assignments from Shiny server code*

Description

Identifies `output$` assignments in a Shiny server file and extracts their associated render function types.

Usage

```
extract_output_assignments(server_code)
```

Arguments

`server_code`	A character vector representing lines of Shiny server code.

Value

A list of lists, where each list contains an index, output ID, and render function type.

`extract_ui_features` *Extract Shiny UI input and output elements and their ids (including custom Output functions)*

Description

This function identifies all Shiny input functions (e.g., selectInput, textInput, numericInput) and output functions (e.g., plotOutput, tableOutput, or custom output functions) in a UI script, and extracts the corresponding ids. It skips over lines that are source() calls.

Usage

```
extract_ui_features(ui_code)
```

Arguments

`ui_code`	A character vector representing lines of Shiny UI code.

Value

A data frame containing the input/output functions and their associated ids.

`read_shiny_file` *Read a Shiny file and determine if it should be skipped*

Description

This function reads a Shiny server file and checks whether it only contains `source()` calls or is empty. If so, it returns `NULL` to indicate the file should be skipped.

Usage

```
read_shiny_file(file_path)
```

Arguments

`file_path`	A string representing the path to the R file.

Value

A character vector containing the file's lines if valid, otherwise `NULL`.

Index