Package: btw (via r-universe)

March 31, 2025

Title Describe R Stuff to Large Language Models

Version 0.0.1.9000

Description Provides a number of utilities for describing R objects and package documentation in plain text. For interactive use, this is especially powerful for describing relevant pieces of context to large language models. When used programmatically, these utilities can be registered with 'ellmer' chats as tool calls, enabling language models to peruse package documentation and explore your computational environment.

License MIT + file LICENSE

URL https://github.com/posit-dev/btw, https://posit-dev.github.io/btw/

BugReports https://github.com/posit-dev/btw/issues

Imports cli, clipr, dplyr, ellmer, fs, jsonlite, rlang (>= 1.1.0), rmarkdown, rstudioapi, S7, sessioninfo, skimr, tibble, withr

Suggests testthat (>= 3.0.0)

Config/Needs/website tidyverse/tidytemplate

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Config/pak/sysreqs make libicu-dev libssl-dev libx11-dev

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

RemoteUrl https://github.com/posit-dev/btw

RemoteRef main

RemoteSha 1af8227cc0f2937e970761a2d5dd13da715a80c4

Contents

`btw` *Plain-text descriptions of R objects*

Description

This function allows you to quickly describe your computational environment to a model by concatenating plain-text descriptions of ”R stuff”, from data frames to packages to function documentation.

There are two key ways to use `btw()`:

1. Use it interactively at the console to gather information about your environment into prompt text that you can paste into the chat interface of an LLM, like ChatGPT or Claude. By default, `btw()` copies the prompt to the clipboard for you.

   ```
   btw(vignette("colwise", "dplyr"), dplyr::across, dplyr::starwars)
   #>  btw copied to the clipboard!
   ```

2. Pair `btw()` with ellmer::Chat during a chat session to create a prompt that includes additional context drawn from your environment and help pages.

   ```
   library(ellmer)

   chat <- chat_claude() # requires an Anthropic API key
   chat <- chat_ollama(model = "llama3.1:8b") # requires ollama and a local model

   chat$chat(btw(
     vignette("colwise", "dplyr"),
     dplyr::across,
     dplyr::starwars,
   ```

```
    "Create a few interesting examples that use `dplyr::across()`",
    "with the `starwars` data set."
  ))
```

Usage

```
btw(..., clipboard = TRUE)
```

Arguments

`...` Objects to describe from your R environment. You can pass objects themselves, like data frames or functions, or the function also accepts output from `btw_tool_*()` functions like `btw_tool_docs_package_help_topics()`, `btw_tool_docs_help_page()`, etc. If omitted, this function will just describe the elements in your global R environment.

`clipboard` Whether to write the results to the clipboard. A single logical value; will default to `TRUE` when run interactively.

Value

Returns an ellmer::ContentText object with the collected prompt. If `clipboard = TRUE`, the prompt text is copied to the clipboard when the returned object is printed for the first time (e.g. calling `btw()` without assignment).

Examples

```
btw()

btw(mtcars)

btw(btw::btw)

if (FALSE) {
  # btw() can also be used directly in {ellmer} chats
  library(ellmer)

  chat <- chat_ollama(model = "llama3.1:8b")
  chat$chat(
    btw(mtcars, "Are there cars with 8 cylinders in this dataset?")
  )
}
```

`btw_client` *Create a btw-enhanced ellmer chat client*

Description

Creates an ellmer::Chat client, enhanced with the tools from `btw_register_tools()`. Use `btw_client()` to create the chat client for general or interactive use at the console, or `btw_app()` to create a chat client and launch a Shiny app for chatting with a btw-enhanced LLM in your local workspace.

Project Context:

You can keep track of project-specific rules, guidance and context by adding a `btw.md` file in your project directory. Any time you start a chat client with `btw_client()` or launch a chat session with `btw_app()`, btw will automatically find and include the contents of the `btw.md` file in your chat.

Use `btw.md` to inform the LLM of your preferred code style, to provide domain-specific terminology or definitions, to establish project documentation, goals and constraints, to include reference materials such or technical specifications, or more. Storing this kind of information in `btw.md` may help you avoid repeating yourself and can be used to maintain coherence across many chat sessions.

The `btw.md` file, when present, is included as part of the system prompt for your chat conversation. You can structure the file in any way you wish.

You can also use the `btw.md` file to choose default chat settings for your project in a YAML block at the top of the file. In this YAML block you can choose the default `provider`, `model` and `tools` for `btw_client()` or `btw_app()`. `provider` chooses the `ellmer::chat_*()` function, e.g. `provider: openai` or `provider: chat_openai` to use `ellmer::chat_openai()`. `tools` chooses which btw tools are included in the chat, and all other values are passed to the `ellmer::chat_*()` constructor, e.g. `model: gpt-4o`, `seed: 42`, or ‘echo: all“.

Here’s an example `btw.md` file:

```
---
provider: claude
model: claude-3-7-sonnet-20250219
tools: [data, docs, environment]
---

Follow these important style rules for any R code in this project:

* Prefer solutions that use {tidyverse}
* Always use `<-` for assignment
* Always use the native base-R pipe `|>` for piped expressions
```

Client Options:

- `btw.client`: The ellmer::Chat client to use as the basis for new `btw_client()` or `btw_app()` chats.
- `btw.tools`: The btw tools to include by default when starting a new btw chat, see `btw_register_tools()` for details.

Usage

```
btw_client(..., client = NULL, tools = NULL, path_btw = NULL)

btw_app(..., client = NULL, tools = NULL, path_btw = NULL)
```

Arguments

`...` Objects and documentation to be included as context in the chat, via `btw()`.

`client` An ellmer::Chat client, defaults to `ellmer::chat_claude()`. You can use the `btw.client` option to set a default client for new `btw_client()` calls, or use a `btw.md` project file for default chat client settings, like provider and model. We check the `client` argument, then the `btw.client` R option, and finally the `btw.md` project file, using only the client definition from the first of these that is available.

`tools` Optional names of tools or tool groups to include in the chat client. By default, all btw tools are included. For example, use `include = "docs"` to include only the documentation related tools, or `include = c("env", "docs")`, etc. `btw_client()` also supports `tools = FALSE` to skip registering **btw** tools with the chat client.

`path_btw` A path to a `btw.md` project context file. If `NULL`, btw will find a project-specific `btw.md` file in the parents of the current working directory.

Value

Returns an ellmer::Chat object with additional tools registered by `btw_register_tools()`. `btw_app()` returns the chat object invisibly, and the chat object with the messages added during the chat session.

Functions

- `btw_client()`: Create a btw-enhanced ellmer::Chat client
- `btw_app()`: Create a btw-enhanced client and launch a Shiny app to chat

Examples

```
if (interactive()) {
  withr::local_options(list(
    btw.client = ellmer::chat_ollama(model="llama3.1:8b")
  ))

  chat <- btw_client()
  chat$chat("How can I replace `stop()` calls with functions from the cli package?")
}
```

`btw_register_tools` *Tools: Register tools from btw*

Description

The `btw_register_tools()` function equips an ellmer chat to interface with your computational environment. Chats returned by this function have access to the tools:

Name	Group	Description
btw_tool_docs_available_vignettes	docs	List available vignettes for an R package.
btw_tool_docs_help_page	docs	Get help page from package.
btw_tool_docs_package_help_topics	docs	Get available help topics for an R package.
btw_tool_docs_vignette	docs	Get a package vignette in plain text.
btw_tool_env_describe_data_frame	env	Show the data frame or table or get information about the struc
btw_tool_env_describe_environment	env	List and describe items in an environment.
btw_tool_files_list_files	files	List files in the current working directory or in subfolders in the
btw_tool_files_read_text_file	files	Read an entire text file.
btw_tool_ide_read_current_editor	ide	Read the contents of the editor that is currently open in the user
btw_tool_session_package_info	session	Verify that a specific package is installed, or find out which pack
btw_tool_session_platform_info	session	Describes the R version, operating system, language and locale s

Usage

```
btw_register_tools(chat, tools = NULL)
```

Arguments

`chat` An ellmer `Chat` object.

`tools` Optional names of tools or tool groups to include when registering tools. By default all btw tools are included. For example, use `tools = "docs"` to include only the documentation related tools, or `tools = c("env", "docs", "session")`, etc.

Value

Registers the tools with `chat`, updating the `chat` object in place. The `chat` input is returned invisibly.

See Also

Other Tools: `btw_tool_env_describe_data_frame()`, `btw_tool_env_describe_environment()`, `btw_tool_files_list_files()`, `btw_tool_files_read_text_file()`, `btw_tool_ide_read_current_editor()`, `btw_tool_package_docs`, `btw_tool_session_package_info()`, `btw_tool_session_platform_info()`

Examples

```
# requires an ANTHROPIC_API_KEY
## Not run:
ch <- ellmer::chat_claude()

btw_register_tools(ch)

## End(Not run)
```

`btw_this` *Describe something for use by an LLM*

Description

A generic function used to describe an object for use by LLM.

Usage

```
btw_this(x, ...)
```

Arguments

`x`	The thing to describe.
`...`	Additional arguments passed down to underlying methods. Unused arguments are silently ignored.

Value

A character vector of lines describing the object.

See Also

Other `btw_this()` methods: `btw_this.character()`, `btw_this.data.frame()`, `btw_this.environment()`

Examples

```
btw_this(mtcars)
btw_this(dplyr::mutate)
btw_this("{dplyr}")

# Files ----
btw_this("./") # list files in the current working directory
```

`btw_this.character` *Describe objects*

Description

Character strings in `btw_this()` are used as shortcuts to many underlying methods. `btw_this()` detects specific formats in the input string to determine which method to call, or by default it will try to evaluate the character string as R code and return the appropriate object description.

`btw_this()` knows about the following special character string formats:

- `"./path"`
  Any string starting with `./` is treated as a relative path. If the path is a file, we call `btw_tool_files_read_text_file()` and if the path is a directory we call `btw_tool_files_list_files()` on the path.
  - `btw_this("./data")` lists the files in `data/`.
  - `btw_this("./R/load_data.R")` reads the source of the `R/load_data.R` file.
- `"{pkgName}"`
  A package name wrapped in braces. Returns either the introductory vignette for the package (`btw_tool_docs_vignette()`) or a list of help topics if no such vignette exists (`btw_tool_docs_package_help_topics()`).
  - `btw_this("{dplyr}")` includes dplyr's introductory vignette.
  - `btw_this("{btw}")` returns the package help index (because `btw` doesn't have an intro vignette, yet).
- `"?help_topic"`
  When the string starts with `?`, btw searches R's help topics using `btw_tool_docs_help_page()`.
  - `btw_this("?dplyr::across")` includes the reference page for `dplyr::across`.
- `"@current_file"` or `"@current_selection"`
  When used in RStudio or Positron, or anywhere else that the **rstudioapi** is supported, `btw("@current_file")` includes the contents of the file currently open in the editor using `rstudioapi::getSourceEditorContext()`.
- `"@platform_info"`
  Includes information about the current platform, such as the R version, operating system, IDE or UI being used, as well as language, locale, timezone and current date.
- `"@attached_packages"`, `"@loaded_packages"`, `"@installed_packages"`
  Includes information about the attached, loaded, or installed packages in your R session, using `sessioninfo::package_info()`.
- `"@last_error"`
  Includes the message from the last error that occurred in your session. To reliably capture the last error, you need to enable `rlang::global_entrace()` in your session.
- `"@last_value"`
  Includes the `.Last.value`, i.e. the result of the last expression evaluated in your R console.

Usage

```
## S3 method for class 'character'
btw_this(x, ..., caller_env = parent.frame())
```

Arguments

`x`	A character string
`...`	Ignored.
`caller_env`	The caller environment.

Value

A character vector of lines describing the object.

See Also

Other `btw_this()` methods: `btw_this()`, `btw_this.data.frame()`, `btw_this.environment()`

`btw_this.data.frame` *Describe a data frame in plain text*

Description

Describe a data frame in plain text

Usage

```
## S3 method for class 'data.frame'
btw_this(
  x,
  ...,
  format = c("skim", "glimpse", "print", "json"),
  dims = c(5, 100)
)

## S3 method for class 'tbl'
btw_this(
  x,
  ...,
  format = c("skim", "glimpse", "print", "json"),
  dims = c(5, 100)
)
```

Arguments

`x` A data frame or tibble.

`...` Additional arguments are silently ignored.

`format` One of `"skim"`, `"glimpse"`, `"print"`, or `"json"`.

- `"skim"` is the most information-dense format for describing the data. It uses and returns the same information as `skimr::skim()` but formatting as a JSON object that describes the dataset.
- To glimpse the data column-by-column, use `"glimpse"`. This is particularly helpful for getting a sense of data frame column names, types, and distributions, when pairings of entries in individual rows aren't particularly important.
- To just print out the data frame, use `print()`.
- To get a json representation of the data, use `"json"`. This is particularly helpful when the pairings among entries in specific rows are important to demonstrate.

`dims` The number of rows and columns to show, as a numeric vector of length two. For example, the default `dims = c(5, 100)` shows the first 5 rows and 100 columns, whereas `dims = c(Inf, Inf)` would show all of the data.

Value

A character vector containing a representation of the data frame. Will error if the named data frame is not found in the environment.

Functions

- `btw_this(data.frame)`: Summarize a data frame.
- `btw_this(tbl)`: Summarize a `tbl`.

See Also

`btw_tool_env_describe_data_frame()`

Other `btw_this()` methods: `btw_this()`, `btw_this.character()`, `btw_this.environment()`

Other `btw_this()` methods: `btw_this()`, `btw_this.character()`, `btw_this.environment()`

Examples

```
btw_this(mtcars)

btw_this(mtcars, format = "print")

btw_this(mtcars, format = "json")
```

btw_this.environment *Describe the contents of an environment*

Description

Describe the contents of an environment

Usage

```
## S3 method for class 'environment'
btw_this(x, ..., items = NULL)
```

Arguments

`x`	An environment.
`...`	Additional arguments are silently ignored.
`items`	Optional. A character vector of objects in the environment to describe.

Value

A string describing the environment contents with `#>` prefixing each object's printed representation.

See Also

`btw_tool_env_describe_environment()`

Other `btw_this()` methods: `btw_this()`, `btw_this.character()`, `btw_this.data.frame()`

Examples

```
cyl_6 <- mtcars[mtcars$cyl == 6, ]
gear_5 <- mtcars[mtcars$gear == 5, ]
btw_this(environment())
```

btw_tool_env_describe_data_frame *Tool: Describe data frame*

Description

Tool: Describe data frame

Usage

```
btw_tool_env_describe_data_frame(
  data_frame,
  format = c("skim", "glimpse", "print", "json"),
  dims = c(5, 100)
)
```

Arguments

data_frame
: The data frame to describe

format
: One of "skim", "glimpse", "print", or "json".

- "skim" is the most information-dense format for describing the data. It uses and returns the same information as skimr::skim() but formatting as a JSON object that describes the dataset.
- To glimpse the data column-by-column, use "glimpse". This is particularly helpful for getting a sense of data frame column names, types, and distributions, when pairings of entries in individual rows aren't particularly important.
- To just print out the data frame, use print().
- To get a json representation of the data, use "json". This is particularly helpful when the pairings among entries in specific rows are important to demonstrate.

dims
: The number of rows and columns to show, as a numeric vector of length two. For example, the default dims = c(5, 100) shows the first 5 rows and 100 columns, whereas dims = c(Inf, Inf) would show all of the data.

Value

A character vector containing a representation of the data frame. Will error if the named data frame is not found in the environment.

See Also

btw_this.data.frame(), btw_register_tools()

Other Tools: btw_register_tools(), btw_tool_env_describe_environment(), btw_tool_files_list_files(
btw_tool_files_read_text_file(), btw_tool_ide_read_current_editor(), btw_tool_package_docs, btw_tool_session_package_info(), btw_tool_session_platform_info()

Examples

```
btw_tool_env_describe_data_frame(mtcars)
```

`btw_tool_env_describe_environment` *Tool: Describe an environment*

Description

Tool: Describe an environment

Usage

```
btw_tool_env_describe_environment(environment = global_env(), items = NULL)
```

Arguments

`environment`	An environment to describe.
`items`	Optional. A character vector of objects in the environment to describe.

Value

A string describing the environment contents with `#>` prefixing each object's printed representation.

See Also

`btw_this.environment()`, `btw_register_tools()`

Other Tools: `btw_register_tools()`, `btw_tool_env_describe_data_frame()`, `btw_tool_files_list_files()`, `btw_tool_files_read_text_file()`, `btw_tool_ide_read_current_editor()`, `btw_tool_package_docs`, `btw_tool_session_package_info()`, `btw_tool_session_platform_info()`

`btw_tool_files_list_files` *Tool: List files*

Description

Tool: List files

Usage

```
btw_tool_files_list_files(
  path = getwd(),
  type = c("any", "file", "directory"),
  regexp = ""
)
```

Arguments

path — Path to a directory or file for which to get information. The path must be in the current working directory. If path is a directory, we use fs::dir_info() to list information about files and directories in path (use type to pick only one or the other). If path is a file, we show information about that file.

type — File type(s) to return, one of "any" or "file" or "directory".

regexp — A regular expression (e.g. [.]csv$) passed on to grep() to filter paths.

Value

Returns a character table of file information.

See Also

Other Tools: btw_register_tools(), btw_tool_env_describe_data_frame(), btw_tool_env_describe_enviro
btw_tool_files_read_text_file(), btw_tool_ide_read_current_editor(), btw_tool_package_docs,
btw_tool_session_package_info(), btw_tool_session_platform_info()

---

btw_tool_files_read_text_file

Tool: Read a file

Description

Tool: Read a file

Usage

```
btw_tool_files_read_text_file(path, max_lines = 1000)
```

Arguments

path — Path to a file for which to get information. The path must be in the current working directory.

max_lines — Number of lines to include. Defaults to 1,000 lines.

Value

Returns a character vector of lines from the file.

See Also

Other Tools: btw_register_tools(), btw_tool_env_describe_data_frame(), btw_tool_env_describe_enviro
btw_tool_files_list_files(), btw_tool_ide_read_current_editor(), btw_tool_package_docs,
btw_tool_session_package_info(), btw_tool_session_platform_info()

btw_tool_ide_read_current_editor

Tool: Read current file

Description

Reads the current file using the **rstudioapi**, which works in RStudio, Positron and VS Code (with the vscode-r extension).

Usage

```
btw_tool_ide_read_current_editor(selection = TRUE, consent = FALSE)
```

Arguments

`selection` Should only the selected text be included? If no text is selected, the full file contents are returned.

`consent` Boolean indicating whether the user has consented to reading the current file. The tool definition includes language to induce LLMs to confirm with the user before calling the tool. Not all models will follow these instructions. Users can also include the string `@current_file` to induce the tool.

Value

Returns the contents of the current editor.

See Also

Other Tools: `btw_register_tools()`, `btw_tool_env_describe_data_frame()`, `btw_tool_env_describe_envir`
`btw_tool_files_list_files()`, `btw_tool_files_read_text_file()`, `btw_tool_package_docs`, `btw_tool_session_package_info()`, `btw_tool_session_platform_info()`

btw_tool_package_docs

Tool: Describe R package documentation

Description

These functions describe package documentation in plain text:

Usage

```
btw_tool_docs_package_help_topics(package_name)

btw_tool_docs_help_page(topic, package_name = "")

btw_tool_docs_available_vignettes(package_name)

btw_tool_docs_vignette(package_name, vignette = package_name)
```

Arguments

`package_name`	The name of the package as a string, e.g. `"shiny"`.
`topic`	The `topic_id` or `alias` of the help page, e.g. `"withProgress"` or `"incProgress"`. Find `topic_id`s or `alias`es using `get_package_help()`.
`vignette`	The name (or index) of the vignette to retrieve. Defaults to the ”intro” vignette to the package (by the same rules as pkgdown.)

Value

- `btw_tool_docs_package_help_topics()` returns the `topic_id`, `title`, and `aliases` fields for every topic in a package's documentation as a json-formatted string.
- `btw_tool_docs_help_page()` return the help-page for a package topic as a string.

See Also

`btw_register_tools()`

Other Tools: `btw_register_tools()`, `btw_tool_env_describe_data_frame()`, `btw_tool_env_describe_envirc`
`btw_tool_files_list_files()`, `btw_tool_files_read_text_file()`, `btw_tool_ide_read_current_editor()`,
`btw_tool_session_package_info()`, `btw_tool_session_platform_info()`

Examples

```
cat(btw_tool_docs_package_help_topics("btw"))

cat(btw_tool_docs_help_page("btw", "btw"))

# show the TOC of vignettes in the dplyr package
cat(btw_tool_docs_available_vignettes("dplyr"))

# returns a whole bunch of output and relies on
# dplyr to have the mentioned vignettes available
## Not run:
# grab the intro vignette
cat(btw_tool_docs_vignette("dplyr"))

# grab the programming vignette specifically
cat(btw_tool_docs_vignette("dplyr", "programming"))

## End(Not run)
```

---

btw_tool_session_package_info

Tool: Gather information about a package or currently loaded packages

---

Description

Uses `sessioninfo::package_info()` to provide information about the loaded, attached, or installed packages. The primary use case is to verify that a package is installed; check the version number of a specific packages; or determine which packages are already in use in a session.

Usage

```
btw_tool_session_package_info(packages = "attached", dependencies = "")
```

Arguments

`packages` Which packages to show, or `"loaded"` to show all loaded packages, `"attached"` to show all attached packages, or `"installed"` to show all installed packages.

`dependencies` Whether to include the dependencies when listing package information.

Value

Returns a string describing the selected packages.

See Also

`btw_register_tools()`, `btw_tool_session_platform_info()`

Other Tools: `btw_register_tools`(), `btw_tool_env_describe_data_frame`(), `btw_tool_env_describe_enviro`
`btw_tool_files_list_files`(), `btw_tool_files_read_text_file`(), `btw_tool_ide_read_current_editor`(),
`btw_tool_package_docs`, `btw_tool_session_platform_info`()

Examples

```
cat(btw_tool_session_package_info("btw"))
```

btw_tool_session_platform_info

Tool: Describe user's platform

Description

Describes the R version, operating system, and language and locale settings for the user's system. When using `btw_client()` or `btw_app()`, this information is automatically included in the system prompt.

Usage

```
btw_tool_session_platform_info()
```

Value

Returns a string describing the user's platform.

See Also

`btw_register_tools()`

Other Tools: `btw_register_tools()`, `btw_tool_env_describe_data_frame()`, `btw_tool_env_describe_enviro`
`btw_tool_files_list_files()`, `btw_tool_files_read_text_file()`, `btw_tool_ide_read_current_editor()`,
`btw_tool_package_docs`, `btw_tool_session_package_info()`

Examples

```
cat(btw_tool_session_platform_info())
```

Index