

Package: blockr.ai (via r-universe)

May 26, 2026

Title AI-Powered Blocks for 'blockr.core'

Version 0.0.1

Description Provides AI-powered functionality as blocks for 'blockr.core'. Enables integration with various AI services and models to enhance data analysis workflows.

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports shiny, blockr.core (>= 0.1.1), ellmer (>= 0.3.0), htmltools, jsonlite, utils, shinychat (>= 0.2.0.9000), evaluate

Suggests testthat (>= 3.0.0), blockr.dplyr

Remotes BristolMyersSquibb/blockr.core,
BristolMyersSquibb/blockr.dplyr, posit-dev/shinychat/pkg-r

Config/testthat/edition 3

URL <https://github.com/BristolMyersSquibb/blockr.ai>

BugReports <https://github.com/BristolMyersSquibb/blockr.ai/issues>

Config/pak/sysreqs cmake make libuv1-dev libssl-dev zlib1g-dev

Repository <https://cynkra.r-universe.dev>

Date/Publication 2026-05-06 11:31:45 UTC

RemoteUrl <https://github.com/cynkra/blockr.ai>

RemoteRef HEAD

RemoteSha 7441767266cd982858bef0c633bcad5597698a0a

Contents

ai_ctrl_block	2
data_exploration_backend	3
data_schema	3

discover_block_args	4
new_llm_tool	6
print_conversation	7
reporter_console	7
reporter_shiny	8
reporter_silent	8

Index	9
--------------	----------

ai_ctrl_block	<i>AI-powered control block plugin</i>
---------------	--

Description

Replaces the default ctrl_block with an AI chat interface. Users can describe what they want in natural language and the LLM will configure the block parameters.

Usage

```
ai_ctrl_block()

ai_ctrl_ui(id, x)

ai_ctrl_server(id, x, vars, data, eval)
```

Arguments

id	Namespace ID
x	Block object
vars	Reactive state values (pre-filtered to externally controllable vars)
data	Input data as list of reactive values
eval	Reactive that evaluates block expression against input data

Value

A ctrl_block plugin object

Examples

```
## Not run:
serve(
  new_board(new_dataset_block("iris")),
  plugins = custom_plugins(ai_ctrl_block())
)

## End(Not run)
```

```
data_exploration_backend
      Data exploration backend
```

Description

Creates a backend that controls how the LLM can explore input data during block argument discovery.

Usage

```
data_exploration_backend(type = c("none", "manual", "tools"), ...)
```

Arguments

type	Backend type: "none" (default, no exploration), "manual" (LLM requests R code execution via tagged code blocks), or "tools" (ellmer native tool calling). Also accepts a backend list directly for custom backends.
...	Passed to backend constructor (e.g. max_probes, structured).

Value

A list with `setup(client, data)`, `process(response, data)`, and `probes_used()`.

```
data_schema      Produce a text schema of a data object for LLM consumption
```

Description

S3 generic that summarises an object's structure (dimensions, column types, sample rows, keys, etc.) as a character string suitable for inclusion in an LLM prompt. Packages can define methods for custom types (e.g. **blockr.dm** registers a `dm` method).

Usage

```
data_schema(x, ...)

## S3 method for class 'data.frame'
data_schema(x, ...)

## Default S3 method:
data_schema(x, ...)

## S3 method for class 'ggplot'
data_schema(x, ...)
```

Arguments

x Object to summarise.
 ... Additional arguments passed to methods.

Value

Character string.

Future

Consider moving this generic to **blockr.core** so that any package can provide methods without depending on **blockr.ai**.

discover_block_args *Discover Block Arguments via LLM*

Description

Uses an LLM to discover block arguments from a natural language prompt. The function iteratively queries the LLM, validates responses, and refines until successful or max iterations reached.

Usage

```
discover_block_args(
  prompt,
  block,
  data = NULL,
  validate = NULL,
  max_iter = 5,
  verbose = FALSE,
  client = NULL,
  current_state = NULL,
  data_exploration = blockr.core::blockr_option("data_exploration", "manual"),
  reporter = NULL,
  images = NULL
)
```

Arguments

prompt User prompt describing what they want (e.g., "setosa only")
 block A block object (e.g., new_filter_block()).
 data Input data (data.frame, dm, or NULL for source blocks)
 validate Validation function. If NULL, uses standalone validator with testServer (for testing outside of Shiny).
 max_iter Maximum LLM iterations (default 5)

verbose	If TRUE, print conversation to console
client	An existing ellmer chat client to reuse. When NULL (default), a new client is created. Pass a persistent client to retain conversation memory across multiple calls.
current_state	Optional plain list of current block parameter values. When non-NULL, a "Current Configuration" section is included in the user message so the LLM can see what's already configured.
data_exploration	Data exploration strategy passed to <code>data_exploration_backend()</code> . Defaults to <code>blockr.core::blockr_option("data_exploration", "manual")</code> .
reporter	A progress reporter list (see reporter_silent , reporter_console , reporter_shiny). When NULL (default), auto-detects: console if interactive, silent otherwise.
images	Optional list of base64-encoded images to include with the first prompt.

Value

List with:

success	TRUE if args were discovered successfully
args	List of discovered arguments
result	The resulting object: data.frame, dm, ggplot, etc. (if successful)
error	Error message (if failed)
conversation	List of message exchanges (if verbose)
client	The ellmer chat client (R6). Pass to subsequent calls to retain conversation memory.

Examples

```
## Not run:
# Filter block
result <- discover_block_args(
  prompt = "setosa only",
  block = new_filter_block(),
  data = iris,
  verbose = TRUE
)
result$success
result$result

# Dataset block (no input data)
result <- discover_block_args(
  prompt = "use mtcars",
  block = new_dataset_block()
)

# Conversation memory: pass result$client to follow-up calls
r1 <- discover_block_args("use iris", new_dataset_block())
r2 <- discover_block_args("now mtcars", new_dataset_block(),
```

```
client = r1$client)

# Summarize block
result <- discover_block_args(
  prompt = "average sepal length by species",
  block = new_summarize_block(),
  data = iris
)

## End(Not run)
```

new_llm_tool

LLM tools

Description

Tools can be made available to LLMs in order to make them more powerful and in turn enabling them to create more accurate results.

Usage

```
new_llm_tool(..., prompt = character())

is_llm_tool(x)

get_tool(x)

get_prompt(x)
```

Arguments

...	Passed to <code>ellmer::tool()</code> .
prompt	A string with additional prompts that will be added to the system prompt.
x	Object

Value

All blocks constructed via `new_llm_tool()` inherit from `llm_tool`.

print_conversation *Print Conversation from Discovery Result*

Description

Prints the full conversation history from a `discover_block_args()` result in a readable format.

Usage

```
print_conversation(x)
```

Arguments

x Result from `discover_block_args()` (must have been called with `verbose = TRUE`)

Value

Invisibly returns x

Examples

```
## Not run:
result <- discover_block_args(
  prompt = "setosa only",
  block = new_filter_block(),
  data = iris,
  verbose = TRUE
)
print_conversation(result)

## End(Not run)
```

reporter_console *Progress reporter: console*

Description

Prints formatted progress to stdout. Useful for interactive standalone discovery and debugging.

Usage

```
reporter_console()
```

Value

A reporter list

reporter_shiny	<i>Progress reporter: Shiny</i>
----------------	---------------------------------

Description

Uses shinychat's chunk protocol to show live progress in the chat widget. Opens a single streaming assistant message and shows one active badge at a time using operation = "replace". When done() is called the message is closed.

Usage

```
reporter_shiny(chat_id, session)
```

Arguments

chat_id	The shinychat chat widget ID (namespaced)
session	The Shiny session

Value

A reporter list

reporter_silent	<i>Progress reporter: silent</i>
-----------------	----------------------------------

Description

No-op reporter for benchmarks and non-interactive use. All callbacks are empty functions.

Usage

```
reporter_silent()
```

Value

A reporter list

Index

`ai_ctrl_block`, 2
`ai_ctrl_server (ai_ctrl_block)`, 2
`ai_ctrl_ui (ai_ctrl_block)`, 2

`data_exploration_backend`, 3
`data_exploration_backend()`, 5
`data_schema`, 3
`discover_block_args`, 4
`discover_block_args()`, 7

`ellmer::tool()`, 6

`get_prompt (new_llm_tool)`, 6
`get_tool (new_llm_tool)`, 6

`is_llm_tool (new_llm_tool)`, 6

`new_llm_tool`, 6

`print_conversation`, 7

`reporter_console`, 5, 7
`reporter_shiny`, 5, 8
`reporter_silent`, 5, 8