

Package: webformscore (via r-universe)

October 22, 2026

Type Package

Title 'WebForms Core'

Version 2.2.0

Description A stateless server-driven user interface technology for building web interfaces through server-side user interface manipulation. 'WebForms Core' provides a common structure across multiple programming languages. The server-side class acts as a commander that generates commands, while 'WebFormsJS' serves as the client-side executor that applies them. Commands range from atomic operations, such as adding elements and modifying text and attributes, to higher-level operations such as iterative processing and render. See <https://github.com/webforms-core>.

License MIT + file LICENSE

Encoding UTF-8

Imports R6

NeedsCompilation no

Author Mohammad Rabie [aut, cre]

Maintainer Mohammad Rabie <info@elanat.net>

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

Date/Publication 2026-10-21 15:25:31 UTC

RemoteUrl <https://github.com/cran/webformscore>

RemoteRef HEAD

RemoteSha a91f57ba0973702efbd6eb9700a898a99a5b7da1

Contents

webformscore-package	2
Index	6

webformscore-package *WebForms Core for R*

Description

An R implementation of the server-side part of WebForms Core, a stateless server-driven user interface technology for building web interfaces through server-side UI manipulation. The server-side class acts as a Commander that generates commands, while WebFormsJS serves as the client-side Executor that applies them. Commands range from atomic operations, such as adding elements and modifying text and attributes, to higher-level operations such as iterative processing and render.

Usage

GS
US
RS
NUL

InputPlace
OutputPlace
Fetch
HtmlEvent
HtmlEventListener
WasmLanguage

ext_child(text, value)
ext_parent(text)
ext_criteria(text, value)
ext_append_fetch_replace(text, searchValue, value)
ext_line_break(text, encodeLine = FALSE)
ext_to_js_string(text)
ext_to_js_object(text)
ext_to_js_return_object(text)

Arguments

text	A character string.
value	A character string appended or substituted according to the function.
searchValue	A character string to search for.
encodeLine	Logical. When TRUE, line breaks are replaced by "\$[sln];".

Details

Commander and Executor: The package implements the server side of WebForms Core. A WebForms object accumulates commands. The result is sent to the browser using `response()`, `export_to_html_comment()`, or `export_to_line_break()`, where WebFormsJS applies the commands to the DOM.

The WebForms class: Create an instance with `WebForms$new()` and call its public methods to append commands. The main families of methods are:

- **Buffer:** `add()`, `add_to_up()`, `get_web_forms_data()`, `append_form()`, `clean()`, `get_line_by_index()`, `update_line_by_index()`.
- **Add helpers** (append to existing value): `add_id()`, `add_name()`, `add_value()`, `add_class()`, `add_style()`, `add_attribute()`, `add_text()`, `add_tag()`, `add_option_tag()`, `add_check_box_tag()`, and others.
- **Set helpers** (replace existing value): `set_id()`, `set_name()`, `set_value()`, `set_class()`, `set_style()`, `set_text()`, `set_attribute()`, `set_background_color()`, `set_text_color()`, `set_font_size()`, `set_visible()`, `set_disabled()`, and others.
- **Insert helpers** (create only, never replace): `insert_id()`, `insert_name()`, `insert_value()`, `insert_class()`, `insert_style()`, `insert_text()`, `insert_attribute()`, and others.
- **Delete:** `delete()`, `delete_parent()`, `delete_id()`, `delete_name()`, `delete_value()`, `delete_class()`, `delete_style()`, `delete_attribute()`, and others.
- **Events:** `set_post_event()`, `set_post_event_listener()`, `set_get_event()`, `set_put_event()`, `set_patch_event()`, `set_delete_event()`, `set_send_event()`, `set_comment_event()`, `set_sse_event()`, `set_web_socket_event()`, `set_wasm_event()`, `set_front_event()`, `set_method_event()`, `assign_confirm_event()`, `trigger_event()`, and matching `remove_*_event()` methods.
- **Flow control:** `start_index()`, `start_state()`, `go_to()`, `is_equal_to()`, `is_not_equal_to()`, `is_greater_than()`, `is_less_than()`, `exist()`, `not_exist()`, `is_true()`, `is_false()`, `include()`, `not_include()`, `element_exists()`, `for_each()`, `then()`, `else_()`, `start_bracket()`, `end_bracket()`, `break_()`, `repeat_()`, `isole()`, `render()`, `async()`, `delay()`.
- **Save and cache:** `save_id()`, `save_value()`, `save_text()`, `cache_id()`, `cache_value()`, `cache_text()`, and the `set_*_save_value()` / `set_*_cache_value()` family.
- **Pre-runner modifiers:** `assign_delay()`, `assign_interval()`, `assign_repeat()`, `delete_interval()`.
- **Browser:** `change_url()`, `set_head_title()`, `clipboard_write_text()`, `scroll_to()`, `history_go()`, `reload_page()`, `redirect()`.
- **Call and message:** `load_url()`, `call_script()`, `call_method()`, `call_post_back()`, `call_get_back()`, `call_sse_back()`, `alert()`, `message()`, `console_message()`, `create_debugger()`, `comment()`.
- **Output:** `response()`, `get_forms_action_data()`, `get_forms_action_data_line_break()`, `export_to_html_comment()`, `export_to_line_break()`.

The Security class: `Security$new()` creates an object with a single public method `safe_value(value)` that prepares a string for embedding in a command: it escapes a leading "@" to prevent command forgery, replaces newlines with "\$[ln];", replaces ",@" with "\$[co];@", and strips ASCII control characters 28–31.

Separator constants: GS separates fields inside one command. US joins list items inside one field. RS is used by Fetch helpers. NUL is the default splitter argument of `add_attribute()` and `insert_attribute()`.

Place criteria: `InputPlace` contains both constants (such as `document`, `window`, `html`, `head`, `parent`, `current`, `target`) and helper functions (such as `id()`, `name()`, `tag()`, `class()`, `attribute()`, `query()`). `OutputPlace` is a direct alias.

Fetch helpers: Fetch provides access to client-side values (for example `get_value()`, `get_text()`, `get_attribute()`), to the browser environment (such as `href`, `scroll_x`, `user_agent`, `screen_width`), to the current event (such as `event`, `event_key`, `event_client_x`), and to the save and cache storage (`save()`, `cache()`).

Extension helpers: `ext_child()`, `ext_parent()`, `ext_criteria()`, `ext_append_fetch_replace()`, `ext_line_break()`, `ext_to_js_string()`, `ext_to_js_object()`, and `ext_to_js_return_object()` mirror the extension methods used by the Commander API in other languages.

Value

WebForms An R6 class generator. Calling `WebForms$new()` returns a new `WebForms` object, the main Command builder.

Security An R6 class generator. Calling `Security$new()` returns a new `Security` object with the method `safe_value()`.

GS, US, RS, NUL Single-character strings. GS is ASCII 29 (group separator), RS is ASCII 30 (record separator), US is ASCII 31 (unit separator), and NUL is ASCII 0 (null character).

InputPlace, OutputPlace A named list of constants and helper functions for building `WebForms` Place Criteria (WPC) strings. `OutputPlace` is an alias of `InputPlace`.

Fetch A named list of helpers for reading client-side values or embedding dynamic values in commands.

HtmlEvent, HtmlEventListener Named lists of HTML event attribute names (with the `on` prefix) and event listener names (without the `on` prefix).

WasmLanguage A named list of WebAssembly language identifiers.

ext_* Each returns a character string.

See Also

Useful links:

- <https://github.com/webforms-core>

Examples

```
# Build a simple command payload without starting an HTTP server.
form <- WebForms$new()

form$set_font_size("<form>", 16)
form$set_background_color("<form>", "lightblue")
form$add_tag("<form>", "h3")
form$set_text("<h3>", "Welcome!")

cat(form$response())

# Separator constants.
GS
US
RS
nchar(NUL)
```

```
# Place criteria helpers.
InputPlace$id("my-id")
InputPlace$tag("div")
OutputPlace$class("active")

# Fetch helpers.
Fetch$get_value("<select>")
Fetch$date_hours

# Security helper.
sec <- Security$new()
sec$safe_value("line1")
sec$safe_value("@not-a-fetch-command")

# Extension helpers.
ext_child("a", "b")
ext_parent("a|b")
ext_line_break("a\r\nb", TRUE)
ext_to_js_string("value")
```

Index

* package

- webformscore-package, [2](#)
- ext_append_fetch_replace
 - (webformscore-package), [2](#)
- ext_child (webformscore-package), [2](#)
- ext_criteria (webformscore-package), [2](#)
- ext_line_break (webformscore-package), [2](#)
- ext_parent (webformscore-package), [2](#)
- ext_to_js_object
 - (webformscore-package), [2](#)
- ext_to_js_return_object
 - (webformscore-package), [2](#)
- ext_to_js_string
 - (webformscore-package), [2](#)
- Fetch (webformscore-package), [2](#)
- GS (webformscore-package), [2](#)
- HtmlEvent (webformscore-package), [2](#)
- HtmlEventListener
 - (webformscore-package), [2](#)
- InputPlace (webformscore-package), [2](#)
- NUL (webformscore-package), [2](#)
- OutputPlace (webformscore-package), [2](#)
- RS (webformscore-package), [2](#)
- Security (webformscore-package), [2](#)
- US (webformscore-package), [2](#)
- WasmLanguage (webformscore-package), [2](#)
- WebForms (webformscore-package), [2](#)
- webformscore (webformscore-package), [2](#)
- webformscore-package, [2](#)