

Package: fastjmespath (via r-universe)

October 9, 2026

Title Fast 'JMESPath' Queries for R Objects and JSON

Version 0.0.1

Description Evaluate 'JMESPath' expressions against R lists, data frames and JSON strings. Built on the Rust 'jmespath' crate through 'extendr', with compiled expressions that can be reused across queries.

License MIT + file LICENSE

Config/roxygen2/version 8.1.0

Encoding UTF-8

Roxygen list(markdown = TRUE)

Config/rextendr/version 0.5.0

SystemRequirements Cargo (Rust's package manager), rustc >= 1.65.0, xz

Suggests jsonlite, testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 4.2)

Config/pak/sysreqs xz-utils libclang-dev

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

Date/Publication 2026-10-09 15:16:21 UTC

RemoteUrl <https://github.com/DyfanJones/fastjmespath>

RemoteRef HEAD

RemoteSha 23d70c0d464ec80197b5668458032ee8445e7b6a

Contents

jmespath_search	2
Index	4

jmespath_search	<i>Search data with a JMESPath expression</i>
-----------------	---

Description

jmespath_search() evaluates a **JMESPath** expression against an R object, and jmespath_search_json() against a JSON string. Use jmespath_compile() to parse an expression once and reuse it.

Usage

```
jmespath_search(data, expression, simplify = TRUE)

jmespath_search_json(data, expression, simplify = TRUE)

jmespath_compile(expression)
```

Arguments

data	An R object (jmespath_search()) or a single JSON string (jmespath_search_json()).
expression	A JMESPath expression string, or an object returned by jmespath_compile().
simplify	Whether to simplify arrays of scalars to atomic vectors.

Value

The query result as an R object.

Converting R to JSON

- Length-1 atomic vectors become scalars, longer ones arrays. Wrap a single value in `list()` to force an array.
- Named lists become objects, unnamed lists arrays.
- Data frames become arrays of row objects.
- Factors become strings.
- NULL, NA, NaN and Inf become null.

Converting results to R

- Objects become named lists. Keys come back in alphabetical order.
- With `simplify = TRUE`, arrays of scalars of one type become atomic vectors (null becomes NA). Otherwise arrays are lists.
- Integers outside the 32-bit range are returned as doubles.

Errors

Failures signal conditions of class `jmespath_error`, with subclasses `jmespath_syntax_error` (bad expression), `jmespath_search_error` (evaluation failed, e.g. a type error) and `jmespath_json_error` (data is not valid JSON).

Examples

```
people <- list(people = list(
  list(name = "a", age = 20L),
  list(name = "b", age = 30L)
))
jmespath_search(people, "people[?age > `25`].name")

expr <- jmespath_compile("people[*].age")
jmespath_search(people, expr)
jmespath_search_json('{"a": {"b": 1}}', "a.b")
```

Index

`jmespath_compile (jmespath_search)`, [2](#)
`jmespath_search`, [2](#)
`jmespath_search_json (jmespath_search)`,
[2](#)
`list()`, [2](#)