Package 'httpproblems'

July 22, 2025

Title Report Errors in Web Applications with 'Problem Details' (RFC 7807)

Version 1.0.1

Description Tools for emitting the 'Problem Details' structure defined in 'RFC' 7807 <https://tools.ietf.org/html/rfc7807> for reporting errors from 'HTTP' servers in a standard way.

License Apache License (>= 2)

URL https://github.com/atheriel/httpproblems

BugReports https://github.com/atheriel/httpproblems/issues

Encoding UTF-8

RoxygenNote 7.1.1

NeedsCompilation no

Author Aaron Jacobs [aut, cre]

Maintainer Aaron Jacobs <atheriel@gmail.com>

Repository CRAN

Date/Publication 2021-06-16 13:40:02 UTC

Contents

`http_problem` *Describe a Problem with an HTTP Request*

Description

`http_problem()` creates the "Problem Details" structure defined in RFC 7807, used for reporting errors from HTTP APIs in a standard way.

There are also helper methods for the most common HTTP problems: HTTP 400 Bad Request, 404 Not Found, 401 Unauthorized, 403 Forbidden, 409 Conflict, and 500 Internal Server Error.

Usage

```
http_problem(
  detail = NULL,
  status = 500L,
  type = NULL,
  title = NULL,
  instance = NULL,
  ...
)

bad_request(detail = NULL, instance = NULL, ...)

unauthorized(detail = NULL, instance = NULL, ...)

forbidden(detail = NULL, instance = NULL, ...)

not_found(detail = NULL, instance = NULL, ...)

conflict(detail = NULL, instance = NULL, ...)

internal_server_error(detail = NULL, instance = NULL, ...)
```

Arguments

`detail`	A human-readable string giving more detail about the error, if possible.
`status`	The HTTP status code appropriate for the response.
`type`	A URL pointing to human-readable documentation for this type of problem. When `NULL`, the type is generated based on the status code; see `http_problem_types()` for a list of the defaults.
`title`	A "short, human-readable summary of the problem type". When `NULL`, the title is generated based on the status code; see `http_problem_types()` for a list of the defaults.
`instance`	A URL that identifies the specific occurrence of the problem, if possible. When `NULL` this field is simply excluded.
`...`	Additional fields added to the problem as Extension Members.

Value

An object of class "http_problem", which has fields corresponding to an RFC 7807 Problem Details structure.

See Also

stop_for_http_problem for issuing R errors with these structures.

Examples

```
body <- bad_request("Parameter 'id' must be a number.")
str(body)
```

---

http_problem_types *List Built-In Problem Types*

Description

Many APIs will not need to define custom problem "types", since HTTP status codes are usually illustrative enough. This function lists the default type and title information for a given status code.

Usage

```
http_problem_types()
```

Value

A data frame of HTTP status codes and their default title & type.

---

stop_for_http_problem *Signal an Error Caused by an HTTP Problem*

Description

The various `stop_for_*()` functions leverage R's condition system to signal an error with a custom type embedding the "Problem Details" structure defined in RFC 7807.

They can be used for reporting errors from HTTP APIs in a standard way.

There are also helper methods for the most common HTTP problems: HTTP 400 Bad Request, 404 Not Found, 401 Unauthorized, 403 Forbidden, 409 Conflict, and 500 Internal Server Error.

Usage

```
stop_for_http_problem(
  detail = NULL,
  status = 500L,
  type = NULL,
  title = NULL,
  instance = NULL,
  ...
)

stop_for_bad_request(detail = NULL, instance = NULL, ...)

stop_for_unauthorized(detail = NULL, instance = NULL, ...)

stop_for_forbidden(detail = NULL, instance = NULL, ...)

stop_for_not_found(detail = NULL, instance = NULL, ...)

stop_for_conflict(detail = NULL, instance = NULL, ...)

stop_for_internal_server_error(detail = NULL, instance = NULL, ...)
```

Arguments

`detail`	A human-readable string giving more detail about the error, if possible.
`status`	The HTTP status code appropriate for the response.
`type`	A URL pointing to human-readable documentation for this type of problem. When `NULL`, the type is generated based on the status code; see `http_problem_types()` for a list of the defaults.
`title`	A "short, human-readable summary of the problem type". When `NULL`, the title is generated based on the status code; see `http_problem_types()` for a list of the defaults.
`instance`	A URL that identifies the specific occurrence of the problem, if possible. When `NULL` this field is simply excluded.
`...`	Additional fields added to the problem as Extension Members.

Value

These functions call `stop()` with a custom condition (with class `"http_problem_error"`), so they do not return a value.

See Also

http_problem for creating the structure directly.

Examples

```
tryCatch(
  stop_for_bad_request("Parameter 'id' must be a number."),
  error = function(e) {
    str(e)
  }
)
```

Index