

Package: rapid (via r-universe)

September 13, 2026

Title R 'API' Descriptions

Version 0.0.0.9004

Description Convert an 'API' description ('APID'), such as one that follows the 'OpenAPI Specification', to an R 'API' description object (a ``rapid"). The rapid object follows the 'OpenAPI Specification' to make it easy to convert to and from 'API' documents.

License MIT + file LICENSE

URL <https://rapid.api2r.org/>, <https://github.com/api2r/rapid>

BugReports <https://github.com/api2r/rapid/issues>

Depends R (>= 3.5.0)

Imports cli, glue, jsonlite, purrr, rlang (>= 1.1.0), S7 (>= 0.2.2.9000), snakecase, stbl (>= 0.3.0.9000), tibble, tibblify (>= 0.4.0), xml2, yaml

Suggests htr2, testthat (>= 3.0.0)

Remotes wranglezone/stbl, wranglezone/tibblify, RConsortium/S7

Config/roxygen2/version 8.0.0

Config/testthat/edition 3

Config/testthat/parallel true

Encoding UTF-8

Language en-US

Roxygen list(markdown = TRUE)

Collate 'aaa-conditions.R' 'aaa-shared_params.R' 'properties.R' 'security.R' 'paths.R' 'components-security_scheme_details.R' 'components-security_schemes.R' 'components.R' 'servers-server_variables.R' 'servers-string_replacements.R' 'servers.R' 'info-origin.R' 'info-license.R' 'info-contact.R' 'info.R' 'zz-rapid.R' 'absolute_paths.R' 'as.R' 'components-reference.R' 'components-schema.R' 'components-security_scheme.R'

```
'components-security_scheme-api_key.R'
'components-security_scheme-oauth2-scopes.R'
'components-security_scheme-oauth2-flow.R'
'components-security_scheme-oauth2-authorization_code_flow.R'
'components-security_scheme-oauth2-implicit_flow.R'
'components-security_scheme-oauth2-token_flow.R'
'components-security_scheme-oauth2.R' 'rapid-package.R'
'swagger_to_openapi.R' 'urls.R' 'utils.R' 'validate_in.R'
'validate_lengths.R' 'validate_parallel.R' 'zzz.R'
```

Config/pak/sysreqs libicu-dev libxml2-dev

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

Date/Publication 2026-05-31 22:03:31 UTC

RemoteUrl <https://github.com/jonthegeek/rapid>

RemoteRef HEAD

RemoteSha bb37723115d282ccb5c8d8bf562719f5c302d03b

Contents

as_api_key_security_scheme	3
as_api_object	4
as_components	5
as_contact	7
as_info	8
as_license	9
as_oauth2_authorization_code_flow	10
as_oauth2_implicit_flow	11
as_oauth2_security_scheme	12
as_oauth2_token_flow	13
as_origin	14
as_paths	15
as_rapid	15
as_reference	16
as_schema	17
as_scopes	18
as_security	19
as_security_scheme	20
as_security_scheme_details	21
as_security_schemes	23
as_server_variables	25
as_servers	26
as_string_replacements	27
class_api_key_security_scheme	28
class_components	29
class_contact	30
class_info	31
class_license	32

class_oauth2_authorization_code_flow	33
class_oauth2_implicit_flow	34
class_oauth2_security_scheme	36
class_oauth2_token_flow	37
class_origin	38
class_paths	39
class_rapid	40
class_reference	41
class_schema	42
class_scopes	43
class_security	44
class_security_scheme_details	45
class_security_schemes	47
class_server_variables	48
class_servers	49
class_string_replacements	50
expand_servers	51
Index	53

as_api_key_security_scheme

Coerce lists and character vectors to API key security schemes

Description

as_api_key_security_scheme() turns an existing object into an api_key_security_scheme. This is in contrast with [class_api_key_security_scheme\(\)](#), which builds an api_key_security_scheme from individual properties.

Usage

```
as_api_key_security_scheme(x, ..., arg = caller_arg(x), call = caller_env())
```

Arguments

x	The object to coerce. Must be empty or be a list or character vector with names "name" and either "in" or "location", or names that can be coerced to those names via snakecase::to_snake_case() . Additional names are ignored.
...	These dots are for future extensions and must be empty.
arg	An argument name as a string. This argument will be mentioned in error messages as the input that is at the origin of a problem.
call	The execution environment of a currently running function, e.g. caller_env(). The function will be mentioned in error messages as the source of the error. See the call argument of abort() for more information.

Value

An `api_key_security_scheme` as returned by `class_api_key_security_scheme()`.

See Also

security scheme objects: `as_oauth2_authorization_code_flow()`, `as_oauth2_implicit_flow()`, `as_oauth2_security_scheme()`, `as_oauth2_token_flow()`, `as_scopes()`, `as_security_scheme()`, `as_security_scheme_details()`, `as_security_schemes()`, `class_api_key_security_scheme()`, `class_oauth2_authorization_code_flow()`, `class_oauth2_implicit_flow()`, `class_oauth2_security_scheme()`, `class_oauth2_token_flow()`, `class_scopes()`, `class_security_scheme_details()`, `class_security_schemes()`

components object: `as_components()`, `as_oauth2_authorization_code_flow()`, `as_oauth2_implicit_flow()`, `as_oauth2_security_scheme()`, `as_oauth2_token_flow()`, `as_reference()`, `as_schema()`, `as_scopes()`, `as_security_scheme()`, `as_security_scheme_details()`, `as_security_schemes()`, `class_api_key_security_sche`, `class_components()`, `class_oauth2_authorization_code_flow()`, `class_oauth2_implicit_flow()`, `class_oauth2_security_scheme()`, `class_oauth2_token_flow()`, `class_reference()`, `class_schema()`, `class_scopes()`, `class_security_scheme_details()`, `class_security_schemes()`

as_api_object

Convert to a rapid-style object

Description

Convert an object into an object with a rapid-style class.

Usage

```
as_api_object(
  x,
  target_class,
  ...,
  alternate_names = NULL,
  arg = caller_arg(x),
  call = caller_env()
)
```

Arguments

<code>x</code>	The object to coerce. Must be empty, or be a named list or character vector having names corresponding to the parameter of the <code>target_class</code> , or names that can be coerced to those names via <code>snakecase::to_snake_case()</code> . Extra names are ignored.
<code>target_class</code>	The S7 class to which the object should be converted.
<code>...</code>	These dots are for future extensions and must be empty.
<code>alternate_names</code>	Character vector (optional). An optional named character vector, where the names are the names as they might appear in <code>x</code> , and the values are the corresponding properties.

arg	An argument name as a string. This argument will be mentioned in error messages as the input that is at the origin of a problem.
call	The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the <code>call</code> argument of <code>abort()</code> for more information.

Value

An object with the specified `target_class`.

See Also

rapid object: `as_rapid()`, `class_rapid()`, `expand_servers()`

as_components	<i>Coerce lists to components objects</i>
---------------	---

Description

`as_components()` turns an existing object into a components object. This is in contrast with `class_components()`, which builds a components from individual properties.

Usage

```
as_components(x, ..., arg = caller_arg(x), call = caller_env())
```

Arguments

x	The object to coerce. Must be empty or be a list containing a single list named "security_schemes", or a name that can be coerced to "security_schemes" via <code>snakecase::to_snake_case()</code> . Additional names are ignored.
...	These dots are for future extensions and must be empty.
arg	An argument name as a string. This argument will be mentioned in error messages as the input that is at the origin of a problem.
call	The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the <code>call</code> argument of <code>abort()</code> for more information.

Value

A components object as returned by `class_components()`.

See Also

`class_components()` for creating components objects.

components object: `as_api_key_security_scheme()`, `as_oauth2_authorization_code_flow()`, `as_oauth2_implicit_flow()`, `as_oauth2_security_scheme()`, `as_oauth2_token_flow()`, `as_reference()`, `as_schema()`, `as_scopes()`, `as_security_scheme()`, `as_security_scheme_details()`, `as_security_schemes()`, `class_api_key_security_scheme()`, `class_components()`, `class_oauth2_authorization_code_flow()`, `class_oauth2_implicit_flow()`, `class_oauth2_security_scheme()`, `class_oauth2_token_flow()`, `class_reference()`, `class_schema()`, `class_scopes()`, `class_security_scheme_details()`, `class_security_schemes()`

Examples

```
as_components()
as_components(list(
  securitySchemes = list(
    accountAuth = list(
      description = "Account JWT token",
      flows = list(
        password = list(
          scopes = list(
            Catalog = "Access all read-only content",
            Commerce = "Perform account-level transactions",
            Playback = "Allow playback of restricted content",
            Settings = "Modify account settings"
          ),
          tokenUrl = "/account/authorization"
        )
      ),
      type = "oauth2"
    ),
    profileAuth = list(
      description = "Profile JWT token",
      flows = list(
        password = list(
          scopes = list(
            Catalog = "Modify profile preferences and activity"
          ),
          tokenUrl = "/account/profile/authorization"
        )
      ),
      type = "oauth2"
    ),
    resetPasswordAuth = list(
      `in` = "header",
      name = "authorization",
      type = "apiKey"
    ),
    verifyEmailAuth = list(
      `in` = "header",
      name = "authorization",
      type = "apiKey"
    )
  )
)
```

```
)
)
))
```

as_contact

Coerce lists and character vectors to contacts

Description

`as_contact()` turns an existing object into a contact. This is in contrast with `class_contact()`, which builds a contact from individual properties.

Usage

```
as_contact(x, ..., arg = caller_arg(x), call = caller_env())
```

Arguments

<code>x</code>	The object to coerce. Must be empty or have names "name", "email", and/or "url", or names that can be coerced to those names via <code>snakecase::to_snake_case()</code> . Extra names are ignored. This object should describe a single point of contact.
<code>...</code>	These dots are for future extensions and must be empty.
<code>arg</code>	An argument name as a string. This argument will be mentioned in error messages as the input that is at the origin of a problem.
<code>call</code>	The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the <code>call</code> argument of <code>abort()</code> for more information.

Value

A contact as returned by `class_contact()`.

See Also

info object: `as_info()`, `as_license()`, `as_origin()`, `class_contact()`, `class_info()`, `class_license()`, `class_origin()`

Examples

```
as_contact()
as_contact(list(name = "Jon Harmon", email = "jonthegeek@gmail.com"))
```

as_info

*Coerce lists and character vectors to info objects***Description**

as_info() turns an existing object into an info object. This is in contrast with [class_info\(\)](#), which builds an info from individual properties.

Usage

```
as_info(x, ..., arg = caller_arg(x), call = caller_env())
```

Arguments

x	The object to coerce. Must be empty or have names "title", "version", "contact", "description", "license", "summary", "terms_of_service", and/or "origin" (or "x-origin", which will be coerced to "origin"), or names that can be coerced to those names via snakecase::to_snake_case() . Extra names are ignored. This object should describe a single API.
...	These dots are for future extensions and must be empty.
arg	An argument name as a string. This argument will be mentioned in error messages as the input that is at the origin of a problem.
call	The execution environment of a currently running function, e.g. caller_env(). The function will be mentioned in error messages as the source of the error. See the call argument of abort() for more information.

Value

An info object as returned by [class_info\(\)](#).

See Also

info object: [as_contact\(\)](#), [as_license\(\)](#), [as_origin\(\)](#), [class_contact\(\)](#), [class_info\(\)](#), [class_license\(\)](#), [class_origin\(\)](#)

Examples

```
as_info()
as_info(list(title = "My Cool API", version = "1.0.0"))
```

as_license

*Coerce lists and character vectors to licenses***Description**

`as_license()` turns an existing object into a license. This is in contrast with `class_license()`, which builds a license from individual properties.

Usage

```
as_license(x, ..., arg = caller_arg(x), call = caller_env())
```

Arguments

<code>x</code>	The object to coerce. Must be empty or have names "name", "identifier", and/or "url", or names that can be coerced to those names via <code>snakecase::to_snake_case()</code> . Extra names are ignored. This object should describe a single license.
<code>...</code>	These dots are for future extensions and must be empty.
<code>arg</code>	An argument name as a string. This argument will be mentioned in error messages as the input that is at the origin of a problem.
<code>call</code>	The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the <code>call</code> argument of <code>abort()</code> for more information.

Value

A license as returned by `class_license()`.

See Also

info object: `as_contact()`, `as_info()`, `as_origin()`, `class_contact()`, `class_info()`, `class_license()`, `class_origin()`

Examples

```
as_license()
as_license(list(name = "Apache 2.0", identifier = "Apache-2.0"))
```

as_oauth2_authorization_code_flow

Coerce lists and character vectors to OAuth2 authorization code flows

Description

as_oauth2_authorization_code_flow() turns an existing object into an oauth2_authorization_code_flow.

This is in contrast with [class_oauth2_authorization_code_flow\(\)](#), which builds an oauth2_authorization_code_flow from individual properties.

Usage

```
as_oauth2_authorization_code_flow(
  x,
  ...,
  arg = caller_arg(x),
  call = caller_env()
)
```

Arguments

x	The object to coerce. Must be empty or be a list of named lists, each with names "refresh_url", "scopes", "authorization_url", and/or "token_url", or names that can be coerced to those names via snakecase::to_snake_case() . Additional names are ignored.
...	These dots are for future extensions and must be empty.
arg	An argument name as a string. This argument will be mentioned in error messages as the input that is at the origin of a problem.
call	The execution environment of a currently running function, e.g. caller_env(). The function will be mentioned in error messages as the source of the error. See the call argument of abort() for more information.

Value

An oauth2_authorization_code_flow as returned by [class_oauth2_authorization_code_flow\(\)](#).

See Also

security scheme objects: [as_api_key_security_scheme\(\)](#), [as_oauth2_implicit_flow\(\)](#), [as_oauth2_security_scheme\(\)](#), [as_oauth2_token_flow\(\)](#), [as_scopes\(\)](#), [as_security_scheme\(\)](#), [as_security_scheme_details\(\)](#), [as_security_schemes\(\)](#), [class_api_key_security_scheme\(\)](#), [class_oauth2_authorization_code_flow\(\)](#), [class_oauth2_implicit_flow\(\)](#), [class_oauth2_security_scheme\(\)](#), [class_oauth2_token_flow\(\)](#), [class_scopes\(\)](#), [class_security_scheme_details\(\)](#), [class_security_schemes\(\)](#)

components object: [as_api_key_security_scheme\(\)](#), [as_components\(\)](#), [as_oauth2_implicit_flow\(\)](#), [as_oauth2_security_scheme\(\)](#), [as_oauth2_token_flow\(\)](#), [as_reference\(\)](#), [as_schema\(\)](#), [as_scopes\(\)](#), [as_security_scheme\(\)](#), [as_security_scheme_details\(\)](#), [as_security_schemes\(\)](#), [class_api_key_security_sche](#)

[class_components\(\)](#), [class_oauth2_authorization_code_flow\(\)](#), [class_oauth2_implicit_flow\(\)](#), [class_oauth2_security_scheme\(\)](#), [class_oauth2_token_flow\(\)](#), [class_reference\(\)](#), [class_schema\(\)](#), [class_scopes\(\)](#), [class_security_scheme_details\(\)](#), [class_security_schemes\(\)](#)

as_oauth2_implicit_flow

Coerce lists and character vectors to OAuth2 implicit flows

Description

`as_oauth2_implicit_flow()` turns an existing object into an `oauth2_implicit_flow`. This is in contrast with [class_oauth2_implicit_flow\(\)](#), which builds an `oauth2_implicit_flow` from individual properties.

Usage

```
as_oauth2_implicit_flow(x, ..., arg = caller_arg(x), call = caller_env())
```

Arguments

<code>x</code>	The object to coerce. Must be empty or be a list of named lists, each with names "refresh_url", "scopes", and/or "authorization_url", or names that can be coerced to those names via snakecase::to_snake_case() . Additional names are ignored.
<code>...</code>	These dots are for future extensions and must be empty.
<code>arg</code>	An argument name as a string. This argument will be mentioned in error messages as the input that is at the origin of a problem.
<code>call</code>	The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the <code>call</code> argument of abort() for more information.

Value

An `oauth2_implicit_flow` as returned by [class_oauth2_implicit_flow\(\)](#).

See Also

security scheme objects: [as_api_key_security_scheme\(\)](#), [as_oauth2_authorization_code_flow\(\)](#), [as_oauth2_security_scheme\(\)](#), [as_oauth2_token_flow\(\)](#), [as_scopes\(\)](#), [as_security_scheme\(\)](#), [as_security_scheme_details\(\)](#), [as_security_schemes\(\)](#), [class_api_key_security_scheme\(\)](#), [class_oauth2_authorization_code_flow\(\)](#), [class_oauth2_implicit_flow\(\)](#), [class_oauth2_security_scheme\(\)](#), [class_oauth2_token_flow\(\)](#), [class_scopes\(\)](#), [class_security_scheme_details\(\)](#), [class_security_schemes\(\)](#)

components object: [as_api_key_security_scheme\(\)](#), [as_components\(\)](#), [as_oauth2_authorization_code_flow\(\)](#), [as_oauth2_security_scheme\(\)](#), [as_oauth2_token_flow\(\)](#), [as_reference\(\)](#), [as_schema\(\)](#), [as_scopes\(\)](#), [as_security_scheme\(\)](#), [as_security_scheme_details\(\)](#), [as_security_schemes\(\)](#), [class_api_key_security_sche](#), [class_components\(\)](#), [class_oauth2_authorization_code_flow\(\)](#), [class_oauth2_implicit_flow\(\)](#), [class_oauth2_security_scheme\(\)](#), [class_oauth2_token_flow\(\)](#), [class_reference\(\)](#), [class_schema\(\)](#), [class_scopes\(\)](#), [class_security_scheme_details\(\)](#), [class_security_schemes\(\)](#)

as_oauth2_security_scheme

Coerce lists to oauth2 security schemes

Description

as_oauth2_security_scheme() turns an existing object into an oauth2_security_scheme. This is in contrast with [class_oauth2_security_scheme\(\)](#), which builds an oauth2_security_scheme from individual properties.

Usage

```
as_oauth2_security_scheme(x, ...)
```

Arguments

x	The object to coerce. Must be empty or be a list with an object named "flows" or a name that can be coerced to "flows" via snakecase::to_snake_case() . Additional names are ignored.
...	These dots are for future extensions and must be empty.

Value

An oauth2_security_scheme as returned by [class_oauth2_security_scheme\(\)](#).

See Also

security scheme objects: [as_api_key_security_scheme\(\)](#), [as_oauth2_authorization_code_flow\(\)](#), [as_oauth2_implicit_flow\(\)](#), [as_oauth2_token_flow\(\)](#), [as_scopes\(\)](#), [as_security_scheme\(\)](#), [as_security_scheme_details\(\)](#), [as_security_schemes\(\)](#), [class_api_key_security_scheme\(\)](#), [class_oauth2_authorization_code_flow\(\)](#), [class_oauth2_implicit_flow\(\)](#), [class_oauth2_security_scheme\(\)](#), [class_oauth2_token_flow\(\)](#), [class_scopes\(\)](#), [class_security_scheme_details\(\)](#), [class_security_schemes\(\)](#)

components object: [as_api_key_security_scheme\(\)](#), [as_components\(\)](#), [as_oauth2_authorization_code_flow\(\)](#), [as_oauth2_implicit_flow\(\)](#), [as_oauth2_token_flow\(\)](#), [as_reference\(\)](#), [as_schema\(\)](#), [as_scopes\(\)](#), [as_security_scheme\(\)](#), [as_security_scheme_details\(\)](#), [as_security_schemes\(\)](#), [class_api_key_security_sche](#), [class_components\(\)](#), [class_oauth2_authorization_code_flow\(\)](#), [class_oauth2_implicit_flow\(\)](#), [class_oauth2_security_scheme\(\)](#), [class_oauth2_token_flow\(\)](#), [class_reference\(\)](#), [class_schema\(\)](#), [class_scopes\(\)](#), [class_security_scheme_details\(\)](#), [class_security_schemes\(\)](#)

as_oauth2_token_flow *Coerce lists and character vectors to OAuth2 token flows*

Description

as_oauth2_token_flow() turns an existing object into an oauth2_token_flow. This is in contrast with [class_oauth2_token_flow\(\)](#), which builds an oauth2_token_flow from individual properties.

Usage

```
as_oauth2_token_flow(x, ..., arg = caller_arg(x), call = caller_env())
```

Arguments

x	The object to coerce. Must be empty or be a list of named lists, each with names "refresh_url", "scopes", and/or "token_url", or names that can be coerced to those names via snakecase::to_snake_case() . Additional names are ignored.
...	These dots are for future extensions and must be empty.
arg	An argument name as a string. This argument will be mentioned in error messages as the input that is at the origin of a problem.
call	The execution environment of a currently running function, e.g. caller_env(). The function will be mentioned in error messages as the source of the error. See the call argument of abort() for more information.

Value

An oauth2_token_flow as returned by [class_oauth2_token_flow\(\)](#).

See Also

security scheme objects: [as_api_key_security_scheme\(\)](#), [as_oauth2_authorization_code_flow\(\)](#), [as_oauth2_implicit_flow\(\)](#), [as_oauth2_security_scheme\(\)](#), [as_scopes\(\)](#), [as_security_scheme\(\)](#), [as_security_scheme_details\(\)](#), [as_security_schemes\(\)](#), [class_api_key_security_scheme\(\)](#), [class_oauth2_authorization_code_flow\(\)](#), [class_oauth2_implicit_flow\(\)](#), [class_oauth2_security_scheme\(\)](#), [class_oauth2_token_flow\(\)](#), [class_scopes\(\)](#), [class_security_scheme_details\(\)](#), [class_security_schemes\(\)](#)

components object: [as_api_key_security_scheme\(\)](#), [as_components\(\)](#), [as_oauth2_authorization_code_flow\(\)](#), [as_oauth2_implicit_flow\(\)](#), [as_oauth2_security_scheme\(\)](#), [as_reference\(\)](#), [as_schema\(\)](#), [as_scopes\(\)](#), [as_security_scheme\(\)](#), [as_security_scheme_details\(\)](#), [as_security_schemes\(\)](#), [class_api_key_security_scheme\(\)](#), [class_components\(\)](#), [class_oauth2_authorization_code_flow\(\)](#), [class_oauth2_implicit_flow\(\)](#), [class_oauth2_security_scheme\(\)](#), [class_oauth2_token_flow\(\)](#), [class_reference\(\)](#), [class_schema\(\)](#), [class_scopes\(\)](#), [class_security_scheme_details\(\)](#), [class_security_schemes\(\)](#)

as_origin

*Coerce lists and character vectors to origin***Description**

as_origin() turns an existing object into an origin. This is in contrast with [class_origin\(\)](#), which builds an origin from individual properties.

Usage

```
as_origin(x, ..., arg = caller_arg(x), call = caller_env())
```

Arguments

x	The object to coerce. Must be empty or have names "url", "format", and/or "version", or names that can be coerced to those names via snakecase::to_snake_case() . Extra names are ignored. This object should describe a single origin for this API description.
...	These dots are for future extensions and must be empty.
arg	An argument name as a string. This argument will be mentioned in error messages as the input that is at the origin of a problem.
call	The execution environment of a currently running function, e.g. caller_env(). The function will be mentioned in error messages as the source of the error. See the call argument of abort() for more information.

Value

An origin as returned by [class_origin\(\)](#).

See Also

info object: [as_contact\(\)](#), [as_info\(\)](#), [as_license\(\)](#), [class_contact\(\)](#), [class_info\(\)](#), [class_license\(\)](#), [class_origin\(\)](#)

Examples

```
as_origin()
as_origin(
  list(
    list(
      format = "openapi",
      url = "https://api.open.fec.gov/swagger/",
      version = "3.0"
    )
  )
)
```

as_paths	<i>Coerce objects to paths</i>
----------	--------------------------------

Description

as_paths() turns an existing object into a paths object. This is in contrast with [class_paths\(\)](#), which builds a paths object from individual properties. In practice, [class_paths\(\)](#) and as_paths() are currently functionally identical. However, in the future, as_paths() will coerce other valid objects to the expected shape.

Usage

```
as_paths(x, ...)
```

Arguments

x	The object to coerce. Must be empty or be a data.frame().
...	These dots are for future extensions and must be empty.

Value

A paths object as returned by [class_paths\(\)](#).

See Also

paths object: [class_paths\(\)](#)

Examples

```
as_paths()  
as_paths(mtcars)
```

as_rapid	<i>Coerce lists and urls to rapid objects</i>
----------	---

Description

as_rapid() turns an existing object into a rapid object. This is in contrast with [class_rapid\(\)](#), which builds a rapid object from individual properties.

Usage

```
as_rapid(x, ...)
```

Arguments

x	The object to coerce. Must be empty or have names "info", "servers", "components", "paths", and/or "security", or names that can be coerced to those names via <code>snakecase::to_snake_case()</code> . Extra names are ignored. <code>url()</code> objects are read with <code>jsonlite::fromJSON()</code> or <code>yaml::read_yaml()</code> before conversion.
...	These dots are for future extensions and must be empty.

Value

A rapid object as returned by `class_rapid()`.

See Also

rapid object: `as_api_object()`, `class_rapid()`, `expand_servers()`

Examples

```
as_rapid()
```

as_reference

Coerce lists and character vectors to references

Description

`as_reference()` turns an existing object into a reference. This is in contrast with `class_reference()`, which builds a reference from individual properties.

Usage

```
as_reference(x, ..., arg = caller_arg(x), call = caller_env())
```

Arguments

x	The object to coerce. Must be empty or have names "type", "nullable", "description", and/or "format", or names that can be coerced to those names via <code>snakecase::to_snake_case()</code> . Extra names are ignored. This object should describe a single reference.
...	These dots are for future extensions and must be empty.
arg	An argument name as a string. This argument will be mentioned in error messages as the input that is at the origin of a problem.
call	The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the <code>call</code> argument of <code>abort()</code> for more information.

Value

A reference as returned by [class_reference\(\)](#).

See Also

reference objects: [class_reference\(\)](#)

components object: [as_api_key_security_scheme\(\)](#), [as_components\(\)](#), [as_oauth2_authorization_code_flow\(\)](#), [as_oauth2_implicit_flow\(\)](#), [as_oauth2_security_scheme\(\)](#), [as_oauth2_token_flow\(\)](#), [as_schema\(\)](#), [as_scopes\(\)](#), [as_security_scheme\(\)](#), [as_security_scheme_details\(\)](#), [as_security_schemes\(\)](#), [class_api_key_security_scheme\(\)](#), [class_components\(\)](#), [class_oauth2_authorization_code_flow\(\)](#), [class_oauth2_implicit_flow\(\)](#), [class_oauth2_security_scheme\(\)](#), [class_oauth2_token_flow\(\)](#), [class_reference\(\)](#), [class_schema\(\)](#), [class_scopes\(\)](#), [class_security_scheme_details\(\)](#), [class_security_schemes\(\)](#)

Examples

```
as_reference()
as_reference(list(`$ref` = "#/components/schemas/Pet"))
```

as_schema	<i>Coerce lists to schemas</i>
-----------	--------------------------------

Description

[as_schema\(\)](#) turns an existing object into a schema. This is in contrast with [class_schema\(\)](#), which builds a schema from individual properties.

Usage

```
as_schema(x, ..., arg = caller_arg(x), call = caller_env())
```

Arguments

x	The object to coerce. Must be empty or have names "type", "nullable", "description", and/or "format", or names that can be coerced to those names via snakecase::to_snake_case() . Extra names are ignored. This object should describe a single schema.
...	These dots are for future extensions and must be empty.
arg	An argument name as a string. This argument will be mentioned in error messages as the input that is at the origin of a problem.
call	The execution environment of a currently running function, e.g. caller_env() . The function will be mentioned in error messages as the source of the error. See the call argument of abort() for more information.

Value

A schema as returned by [class_schema\(\)](#).

See Also

schema objects: `class_schema()`

components object: `as_api_key_security_scheme()`, `as_components()`, `as_oauth2_authorization_code_flow()`, `as_oauth2_implicit_flow()`, `as_oauth2_security_scheme()`, `as_oauth2_token_flow()`, `as_reference()`, `as_scopes()`, `as_security_scheme()`, `as_security_scheme_details()`, `as_security_schemes()`, `class_api_key_security_scheme()`, `class_components()`, `class_oauth2_authorization_code_flow()`, `class_oauth2_implicit_flow()`, `class_oauth2_security_scheme()`, `class_oauth2_token_flow()`, `class_reference()`, `class_schema()`, `class_scopes()`, `class_security_scheme_details()`, `class_security_schemes()`

Examples

```
as_schema()
as_schema(
  list(
    type = "string",
    format = "date-time",
    description = "Timestamp when the event will occur."
  )
)
```

as_scopes

Coerce lists and character vectors to scopes

Description

`as_scopes()` turns an existing object into a scopes. This is in contrast with `class_scopes()`, which builds a scopes from individual properties.

Usage

```
as_scopes(x, ...)
```

Arguments

`x` The object to coerce. Must be coercible to a named character vector.

`...` These dots are for future extensions and must be empty.

Value

A scopes as returned by `class_scopes()`.

See Also

security scheme objects: [as_api_key_security_scheme\(\)](#), [as_oauth2_authorization_code_flow\(\)](#), [as_oauth2_implicit_flow\(\)](#), [as_oauth2_security_scheme\(\)](#), [as_oauth2_token_flow\(\)](#), [as_security_scheme\(\)](#), [as_security_scheme_details\(\)](#), [as_security_schemes\(\)](#), [class_api_key_security_scheme\(\)](#), [class_oauth2_authorization_code_flow\(\)](#), [class_oauth2_implicit_flow\(\)](#), [class_oauth2_security_scheme\(\)](#), [class_oauth2_token_flow\(\)](#), [class_scopes\(\)](#), [class_security_scheme_details\(\)](#), [class_security_schemes\(\)](#)

components object: [as_api_key_security_scheme\(\)](#), [as_components\(\)](#), [as_oauth2_authorization_code_flow\(\)](#), [as_oauth2_implicit_flow\(\)](#), [as_oauth2_security_scheme\(\)](#), [as_oauth2_token_flow\(\)](#), [as_reference\(\)](#), [as_schema\(\)](#), [as_security_scheme\(\)](#), [as_security_scheme_details\(\)](#), [as_security_schemes\(\)](#), [class_api_key_security_scheme\(\)](#), [class_components\(\)](#), [class_oauth2_authorization_code_flow\(\)](#), [class_oauth2_implicit_flow\(\)](#), [class_oauth2_security_scheme\(\)](#), [class_oauth2_token_flow\(\)](#), [class_reference\(\)](#), [class_schema\(\)](#), [class_scopes\(\)](#), [class_security_scheme_details\(\)](#), [class_security_schemes\(\)](#)

as_security

Coerce lists to as_security objects

Description

[as_security\(\)](#) turns an existing object into a security object. This is in contrast with [class_security\(\)](#), which builds a security from individual properties.

Usage

```
as_security(x, ...)
```

Arguments

x	The object to coerce. Must be empty or be a list containing a single list named "security_schemes", or a name that can be coerced to "security_schemes" via snakecase::to_snake_case() . Additional names are ignored.
...	These dots are for future extensions and must be empty.

Value

A security object as returned by [class_security\(\)](#).

See Also

security object: [class_security\(\)](#)

Examples

```

as_security()
as_security(
  list(
    list(
      oauth2 = c("user", "user:email", "user:follow")
    ),
    list(internalApiKey = list())
  )
)

```

as_security_scheme	<i>Coerce lists to security_scheme objects</i>
--------------------	--

Description

as_security_scheme() turns an existing object into a security_scheme object. It uses the type element of such objects to determine which type of security scheme to construct.

Usage

```
as_security_scheme(x, ...)
```

Arguments

x	The object to coerce. Must be empty or be a named list, with at least an element type. The type element is processed through snakecase::to_snake_case() , and then must be one of "api_key", "oauth2", or "oauth_2".
...	These dots are for future extensions and must be empty.

Value

A security_scheme object as returned by [class_api_key_security_scheme\(\)](#) or [class_oauth2_security_scheme\(\)](#).

See Also

security scheme objects: [as_api_key_security_scheme\(\)](#), [as_oauth2_authorization_code_flow\(\)](#), [as_oauth2_implicit_flow\(\)](#), [as_oauth2_security_scheme\(\)](#), [as_oauth2_token_flow\(\)](#), [as_scopes\(\)](#), [as_security_scheme_details\(\)](#), [as_security_schemes\(\)](#), [class_api_key_security_scheme\(\)](#), [class_oauth2_authorization_code_flow\(\)](#), [class_oauth2_implicit_flow\(\)](#), [class_oauth2_security_scheme\(\)](#), [class_oauth2_token_flow\(\)](#), [class_scopes\(\)](#), [class_security_scheme_details\(\)](#), [class_security_schemes\(\)](#)

components object: [as_api_key_security_scheme\(\)](#), [as_components\(\)](#), [as_oauth2_authorization_code_flow\(\)](#), [as_oauth2_implicit_flow\(\)](#), [as_oauth2_security_scheme\(\)](#), [as_oauth2_token_flow\(\)](#), [as_reference\(\)](#), [as_schema\(\)](#), [as_scopes\(\)](#), [as_security_scheme_details\(\)](#), [as_security_schemes\(\)](#), [class_api_key_security_scheme\(\)](#), [class_components\(\)](#), [class_oauth2_authorization_code_flow\(\)](#), [class_oauth2_implicit_flow\(\)](#), [class_oauth2_security_scheme\(\)](#), [class_oauth2_token_flow\(\)](#), [class_reference\(\)](#), [class_schema\(\)](#), [class_scopes\(\)](#), [class_security_scheme_details\(\)](#), [class_security_schemes\(\)](#)

Examples

```

as_security_scheme(
  list(
    description = "Account JWT token",
    flows = list(
      password = list(
        scopes = list(
          Catalog = "Access all read-only content",
          Commerce = "Perform account-level transactions",
          Playback = "Allow playback of restricted content",
          Settings = "Modify account settings"
        ),
        tokenUrl = "/account/authorization"
      )
    ),
    type = "oauth2"
  )
)
as_security_scheme(
  list(
    description = "Profile JWT token",
    flows = list(
      password = list(
        scopes = list(
          Catalog = "Modify profile preferences and activity"
        ),
        tokenUrl = "/account/profile/authorization"
      )
    ),
    type = "oauth2"
  )
)
as_security_scheme(
  list(
    `in` = "header",
    name = "authorization",
    type = "apiKey"
  )
)

```

as_security_scheme_details

Coerce lists to security_scheme_details objects

Description

as_security_scheme_details() turns an existing object into a security_scheme_details object. This is in contrast with `class_security_scheme_details()`, which builds a security_scheme_details from individual properties.

Usage

```
as_security_scheme_details(x, ...)
```

Arguments

x The object to coerce. Must be empty or be a named list, where each element describes a security scheme object. This object should describe the security schemes for a single API.

... These dots are for future extensions and must be empty.

Value

A security_scheme_details object as returned by `class_security_scheme_details()`.

See Also

security scheme objects: `as_api_key_security_scheme()`, `as_oauth2_authorization_code_flow()`, `as_oauth2_implicit_flow()`, `as_oauth2_security_scheme()`, `as_oauth2_token_flow()`, `as_scopes()`, `as_security_scheme()`, `as_security_schemes()`, `class_api_key_security_scheme()`, `class_oauth2_authorization_code_flow()`, `class_oauth2_implicit_flow()`, `class_oauth2_security_scheme()`, `class_oauth2_token_flow()`, `class_scopes()`, `class_security_scheme_details()`, `class_security_schemes()`

components object: `as_api_key_security_scheme()`, `as_components()`, `as_oauth2_authorization_code_flow()`, `as_oauth2_implicit_flow()`, `as_oauth2_security_scheme()`, `as_oauth2_token_flow()`, `as_reference()`, `as_schema()`, `as_scopes()`, `as_security_scheme()`, `as_security_schemes()`, `class_api_key_security_scheme()`, `class_components()`, `class_oauth2_authorization_code_flow()`, `class_oauth2_implicit_flow()`, `class_oauth2_security_scheme()`, `class_oauth2_token_flow()`, `class_reference()`, `class_schema()`, `class_scopes()`, `class_security_scheme_details()`, `class_security_schemes()`

Examples

```
as_security_scheme_details()
as_security_scheme_details(
  list(
    accountAuth = list(
      description = "Account JWT token",
      flows = list(
        password = list(
          scopes = list(
            Catalog = "Access all read-only content",
            Commerce = "Perform account-level transactions",
            Playback = "Allow playback of restricted content",
            Settings = "Modify account settings"
          ),
          tokenUrl = "/account/authorization"
        )
      ),
      type = "oauth2"
    ),
    profileAuth = list(
      description = "Profile JWT token",
```

```

        flows = list(
            password = list(
                scopes = list(
                    Catalog = "Modify profile preferences and activity"
                ),
                tokenUrl = "/account/profile/authorization"
            )
        ),
        type = "oauth2"
    ),
    resetPasswordAuth = list(
        `in` = "header",
        name = "authorization",
        type = "apiKey"
    ),
    verifyEmailAuth = list(
        `in` = "header",
        name = "authorization",
        type = "apiKey"
    )
)
)
)

```

as_security_schemes *Coerce lists to security_schemes objects*

Description

as_security_schemes() turns an existing object into a security_schemes object. This is in contrast with [class_security_schemes\(\)](#), which builds a security_schemes from individual properties.

Usage

```
as_security_schemes(x, ...)
```

Arguments

x	The object to coerce. Must be empty or be a named list, where each element describes a security scheme object. This object should describe the security schemes for a single API.
...	These dots are for future extensions and must be empty.

Value

A security_schemes object as returned by [class_security_schemes\(\)](#).

See Also

security scheme objects: `as_api_key_security_scheme()`, `as_oauth2_authorization_code_flow()`, `as_oauth2_implicit_flow()`, `as_oauth2_security_scheme()`, `as_oauth2_token_flow()`, `as_scopes()`, `as_security_scheme()`, `as_security_scheme_details()`, `class_api_key_security_scheme()`, `class_oauth2_authorization_code_flow()`, `class_oauth2_implicit_flow()`, `class_oauth2_security_scheme()`, `class_oauth2_token_flow()`, `class_scopes()`, `class_security_scheme_details()`, `class_security_schemes()`

components object: `as_api_key_security_scheme()`, `as_components()`, `as_oauth2_authorization_code_flow()`, `as_oauth2_implicit_flow()`, `as_oauth2_security_scheme()`, `as_oauth2_token_flow()`, `as_reference()`, `as_schema()`, `as_scopes()`, `as_security_scheme()`, `as_security_scheme_details()`, `class_api_key_security_scheme()`, `class_components()`, `class_oauth2_authorization_code_flow()`, `class_oauth2_implicit_flow()`, `class_oauth2_security_scheme()`, `class_oauth2_token_flow()`, `class_reference()`, `class_schema()`, `class_scopes()`, `class_security_scheme_details()`, `class_security_schemes()`

Examples

```
as_security_schemes()
as_security_schemes(
  list(
    accountAuth = list(
      description = "Account JWT token",
      flows = list(
        password = list(
          scopes = list(
            Catalog = "Access all read-only content",
            Commerce = "Perform account-level transactions",
            Playback = "Allow playback of restricted content",
            Settings = "Modify account settings"
          ),
          tokenUrl = "/account/authorization"
        ),
        type = "oauth2"
      ),
      profileAuth = list(
        description = "Profile JWT token",
        flows = list(
          password = list(
            scopes = list(
              Catalog = "Modify profile preferences and activity"
            ),
            tokenUrl = "/account/profile/authorization"
          ),
          type = "oauth2"
        ),
        resetPasswordAuth = list(
          `in` = "header",
          name = "authorization",
          type = "apiKey"
        ),
        verifyEmailAuth = list(
```

```

      `in` = "header",
      name = "authorization",
      type = "apiKey"
    )
  )
)

```

as_server_variables *Coerce lists to server_variables*

Description

as_server_variables() turns an existing object into a server_variables object. This is in contrast with [class_server_variables\(\)](#), which builds a server_variables object from individual properties.

Usage

```
as_server_variables(x, ...)
```

Arguments

x	The object to coerce. Must be empty or be a list of objects that can be coerced to string_replacements objects via as_string_replacements() .
...	These dots are for future extensions and must be empty.

Value

A server_variables object as returned by [class_server_variables\(\)](#).

See Also

servers object: [as_servers\(\)](#), [as_string_replacements\(\)](#), [class_server_variables\(\)](#), [class_servers\(\)](#), [class_string_replacements\(\)](#)

Examples

```

as_server_variables()
as_server_variables(
  list(
    list(
      username = c(default = "demo", description = "Name of the user.")
    ),
    list(
      username = c(
        default = "demo",
        description = "Name of the user."
      ),
      port = list(

```

```

        default = "8443",
        enum = c("8443", "443")
      )
    )
  )
)

```

as_servers

Coerce lists and character vectors to servers

Description

as_servers() turns an existing object into a servers object. This is in contrast with [class_servers\(\)](#), which builds a servers object from individual properties.

Usage

```
as_servers(x, ...)
```

Arguments

x	The object to coerce. Must be empty or have names "name", "email", and/or "url", or names that can be coerced to those names via snakecase::to_snake_case() . Extra names are ignored.
...	These dots are for future extensions and must be empty.

Value

A servers object as returned by [class_servers\(\)](#).

See Also

servers object: [as_server_variables\(\)](#), [as_string_replacements\(\)](#), [class_server_variables\(\)](#), [class_servers\(\)](#), [class_string_replacements\(\)](#)

Examples

```

as_servers()
as_servers(
  list(
    list(
      url = "https://development.gigantic-server.com/v1",
      description = "Development server"
    ),
    list(
      url = "https://staging.gigantic-server.com/v1",
      description = "Staging server"
    ),
    list(

```

```

        url = "https://api.gigantic-server.com/v1",
        description = "Production server"
    )
}

```

as_string_replacements

Coerce lists to string_replacements

Description

as_string_replacements() turns an existing object into a string_replacements. This is in contrast with [class_string_replacements\(\)](#), which builds a string_replacements from individual properties.

Usage

```
as_string_replacements(x, ...)
```

Arguments

x	The object to coerce. Must be empty or be a list of named lists, each with names "enum", "default", or "description", or names that can be coerced to those names via snakecase::to_snake_case() . Additional names are ignored.
...	These dots are for future extensions and must be empty.

Value

A string_replacements as returned by [class_string_replacements\(\)](#).

See Also

servers object: [as_server_variables\(\)](#), [as_servers\(\)](#), [class_server_variables\(\)](#), [class_servers\(\)](#), [class_string_replacements\(\)](#)

Examples

```

as_string_replacements()
as_string_replacements(
  list(
    username = c(
      default = "demo",
      description = "Name of the user."
    ),
    port = list(
      default = "8443",
      enum = c("8443", "443")
    )
  )
)

```

```
)
)
```

```
class_api_key_security_scheme
    API key security schemes
```

Description

Defines an API key security scheme that can be used by the operations.

Usage

```
class_api_key_security_scheme(
  parameter_name = character(),
  location = c("query", "header", "cookie")
)
```

Arguments

parameter_name Character vector (required). The names of the header, query or cookie parameters to be used.

location Character vector (required). The location of the API key. Valid values are "query", "header" or "cookie".

Value

An `api_key_security_scheme` S7 object, with fields `parameter_name` and `location`.

See Also

security scheme objects: `as_api_key_security_scheme()`, `as_oauth2_authorization_code_flow()`, `as_oauth2_implicit_flow()`, `as_oauth2_security_scheme()`, `as_oauth2_token_flow()`, `as_scopes()`, `as_security_scheme()`, `as_security_scheme_details()`, `as_security_schemes()`, `class_oauth2_authorization_code_flow()`, `class_oauth2_implicit_flow()`, `class_oauth2_security_scheme()`, `class_oauth2_token_flow()`, `class_scopes()`, `class_security_scheme_details()`, `class_security_schemes()`

components object: `as_api_key_security_scheme()`, `as_components()`, `as_oauth2_authorization_code_flow()`, `as_oauth2_implicit_flow()`, `as_oauth2_security_scheme()`, `as_oauth2_token_flow()`, `as_reference()`, `as_schema()`, `as_scopes()`, `as_security_scheme()`, `as_security_scheme_details()`, `as_security_schemes()`, `class_components()`, `class_oauth2_authorization_code_flow()`, `class_oauth2_implicit_flow()`, `class_oauth2_security_scheme()`, `class_oauth2_token_flow()`, `class_reference()`, `class_schema()`, `class_scopes()`, `class_security_scheme_details()`, `class_security_schemes()`

Examples

```
class_api_key_security_scheme(
  parameter_name = "Authorization",
  location = "header"
)
```

class_components	<i>An element to hold various schemas for the API</i>
------------------	---

Description

Holds a set of reusable objects for different aspects of the OAS. All objects defined within the components object will have no effect on the API unless they are explicitly referenced from properties outside the components object. We currently only support the security_schemes object within the components object (see [class_security_schemes\(\)](#)).

Usage

```
class_components(..., security_schemes = class_security_schemes())
```

Arguments

... These dots are for future extensions and must be empty.

security_schemes An object to hold reusable security scheme objects created by [class_security_schemes\(\)](#).

Value

A components S7 object with various schemas for the API.

See Also

[as_components\(\)](#) for coercing objects to components objects.

components object: [as_api_key_security_scheme\(\)](#), [as_components\(\)](#), [as_oauth2_authorization_code_flow\(\)](#), [as_oauth2_implicit_flow\(\)](#), [as_oauth2_security_scheme\(\)](#), [as_oauth2_token_flow\(\)](#), [as_reference\(\)](#), [as_schema\(\)](#), [as_scopes\(\)](#), [as_security_scheme\(\)](#), [as_security_scheme_details\(\)](#), [as_security_schemes\(\)](#), [class_api_key_security_scheme\(\)](#), [class_oauth2_authorization_code_flow\(\)](#), [class_oauth2_implicit_flow\(\)](#), [class_oauth2_security_scheme\(\)](#), [class_oauth2_token_flow\(\)](#), [class_reference\(\)](#), [class_schema\(\)](#), [class_scopes\(\)](#), [class_security_scheme_details\(\)](#), [class_security_schemes\(\)](#)

Examples

```
class_components()
class_components(
  security_schemes = class_security_schemes(
    name = "a",
    details = class_security_scheme_details(
      class_api_key_security_scheme("param", "query")
    )
  )
)
```

class_contact	<i>Contact information for the API</i>
---------------	--

Description

A contact object provides contact information for the API.

Usage

```
class_contact(name = character(0), email = character(0), url = character(0))
```

Arguments

name	Character scalar (optional). The identifying name of the contact person or organization.
email	Character scalar (optional). The email address of the contact person/organization. This must be in the form of an email address.
url	Character scalar (optional). The URL pointing to the contact information.

Value

A contact S7 object describing who to contact for information about the API, with fields name, email, and url.

See Also

info object: [as_contact\(\)](#), [as_info\(\)](#), [as_license\(\)](#), [as_origin\(\)](#), [class_info\(\)](#), [class_license\(\)](#), [class_origin\(\)](#)

Examples

```
class_contact(  
  "API Support",  
  "support@example.com",  
  "https://www.example.com/support"  
)
```

class_info

*Information about the API***Description**

The object provides metadata about the API. The metadata may be used by the clients if needed, and may be presented in editing or documentation generation tools for convenience.

Usage

```
class_info(
  title = character(),
  version = character(),
  ...,
  contact = class_contact(),
  description = character(),
  license = class_license(),
  summary = character(),
  terms_of_service = character(),
  origin = class_origin()
)
```

Arguments

title	Character scalar. The title of the API. Required when the object is not empty.
version	Character scalar. The version of the API description (which is distinct from the OpenAPI Specification version or the API implementation version). Required when the object is not empty.
...	These dots are for future extensions and must be empty.
contact	The contact information for the exposed API, generated via class_contact() .
description	Character scalar (optional). A description of the API. CommonMark syntax may be used for rich text representation.
license	The license information for the exposed API, generated via license() .
summary	Character scalar (optional). A short summary of the API.
terms_of_service	Character scalar (optional). A URL to the Terms of Service for the API.
origin	The url and related information about the document used to build the API description. This is used to resolve relative paths in the API description. Note: This is not part of the OpenAPI Specification, but is sometimes supplied as "x-origin".

Value

An info S7 object with metadata describing a single API.

See Also

info object: [as_contact\(\)](#), [as_info\(\)](#), [as_license\(\)](#), [as_origin\(\)](#), [class_contact\(\)](#), [class_license\(\)](#), [class_origin\(\)](#)

Examples

```
class_info()
class_info(
    title = "My Cool API",
    version = "1.0.2",
    license = class_license(
        name = "Apache 2.0",
        url = "https://opensource.org/license/apache-2-0/"
    )
)
class_info(
    title = "My Abbreviated API",
    version = "2.0.0",
    origin = "https://root.url"
)
```

class_license	<i>License information for the API</i>
---------------	--

Description

A license object provides license information for the API to describe how the API can be used.

Usage

```
class_license(
    name = character(),
    ...,
    identifier = character(),
    url = character()
)
```

Arguments

name	Character scalar (optional). The license name used for the API.
...	These dots are for future extensions and must be empty.
identifier	Character scalar (optional). An SPDX license expression for the API. The identifier field is mutually exclusive of the url field.
url	Character scalar (optional). A URL to the license used for the API. This must be in the form of a URL. The url field is mutually exclusive of the identifier field.

Value

A license S7 object describing allowed usage of the API, with fields name, identifier, and url.

See Also

info object: [as_contact\(\)](#), [as_info\(\)](#), [as_license\(\)](#), [as_origin\(\)](#), [class_contact\(\)](#), [class_info\(\)](#), [class_origin\(\)](#)

Examples

```
class_license(
  "Apache 2.0",
  identifier = "Apache-2.0"
)
class_license(
  "Apache 2.0",
  url = "https://opensource.org/license/apache-2-0/"
)
```

class_oauth2_authorization_code_flow

OAuth2 authorization code flow object

Description

An oauth2_authorization_code_flow object describes the configuration for the OAuth Authorization Code flow. Previously called accessCode in OpenAPI 2.0.

Usage

```
class_oauth2_authorization_code_flow(
  authorization_url = character(),
  token_url = character(),
  ...,
  refresh_url = character(),
  scopes = class_scopes()
)
```

Arguments

authorization_url	Character vector (required). The authorization URL to be used for this flow. This must be in the form of a URL. The OAuth2 standard requires the use of TLS.
token_url	Character vector (required). The token URL to be used for this flow. This must be in the form of a URL. The OAuth2 standard requires the use of TLS.
...	These dots are for future extensions and must be empty.

refresh_url	Character scalar (optional). The URL to be used for obtaining refresh tokens. This must be in the form of a URL. The OAuth2 standard requires the use of TLS.
scopes	An optional <code>class_scopes()</code> object with the available scopes for the OAuth2 security scheme.

Value

An oauth2_authorization_code_flow S7 object, with fields authorization_url, token_url, refresh_url, and scopes.

See Also

security scheme objects: `as_api_key_security_scheme()`, `as_oauth2_authorization_code_flow()`, `as_oauth2_implicit_flow()`, `as_oauth2_security_scheme()`, `as_oauth2_token_flow()`, `as_scopes()`, `as_security_scheme()`, `as_security_scheme_details()`, `as_security_schemes()`, `class_api_key_security_scheme()`, `class_oauth2_implicit_flow()`, `class_oauth2_security_scheme()`, `class_oauth2_token_flow()`, `class_scopes()`, `class_security_scheme_details()`, `class_security_schemes()`

components object: `as_api_key_security_scheme()`, `as_components()`, `as_oauth2_authorization_code_flow()`, `as_oauth2_implicit_flow()`, `as_oauth2_security_scheme()`, `as_oauth2_token_flow()`, `as_reference()`, `as_schema()`, `as_scopes()`, `as_security_scheme()`, `as_security_scheme_details()`, `as_security_schemes()`, `class_api_key_security_scheme()`, `class_components()`, `class_oauth2_implicit_flow()`, `class_oauth2_security_scheme()`, `class_oauth2_token_flow()`, `class_reference()`, `class_schema()`, `class_scopes()`, `class_security_scheme_details()`, `class_security_schemes()`

Examples

```
class_oauth2_authorization_code_flow(
  authorization_url = "https://example.com/authorize",
  token_url = "https://example.com/token",
  refresh_url = "https://example.com/refresh",
  scopes = class_scopes(
    name = c("server:read", "server:write"),
    description = c("Read server settings", "Write server settings")
  )
)
```

```
class_oauth2_implicit_flow
      OAuth2 implicit flow object
```

Description

An oauth2_implicit_flow object describes the configuration for the OAuth Implicit flow.

Usage

```
class_oauth2_implicit_flow(
  authorization_url = character(),
  ...,
  refresh_url = character(),
  scopes = class_scopes()
)
```

Arguments

authorization_url	Character vector (required). The authorization URL to be used for this flow. This must be in the form of a URL. The OAuth2 standard requires the use of TLS.
...	These dots are for future extensions and must be empty.
refresh_url	Character scalar (optional). The URL to be used for obtaining refresh tokens. This must be in the form of a URL. The OAuth2 standard requires the use of TLS.
scopes	An optional <code>class_scopes()</code> object with the available scopes for the OAuth2 security scheme.

Value

An `oauth2_implicit_flow` S7 object, with fields `authorization_url`, `refresh_url`, and `scopes`.

See Also

security scheme objects: `as_api_key_security_scheme()`, `as_oauth2_authorization_code_flow()`, `as_oauth2_implicit_flow()`, `as_oauth2_security_scheme()`, `as_oauth2_token_flow()`, `as_scopes()`, `as_security_scheme()`, `as_security_scheme_details()`, `as_security_schemes()`, `class_api_key_security_scheme()`, `class_oauth2_authorization_code_flow()`, `class_oauth2_security_scheme()`, `class_oauth2_token_flow()`, `class_scopes()`, `class_security_scheme_details()`, `class_security_schemes()`

components object: `as_api_key_security_scheme()`, `as_components()`, `as_oauth2_authorization_code_flow()`, `as_oauth2_implicit_flow()`, `as_oauth2_security_scheme()`, `as_oauth2_token_flow()`, `as_reference()`, `as_schema()`, `as_scopes()`, `as_security_scheme()`, `as_security_scheme_details()`, `as_security_schemes()`, `class_api_key_security_scheme()`, `class_components()`, `class_oauth2_authorization_code_flow()`, `class_oauth2_security_scheme()`, `class_oauth2_token_flow()`, `class_reference()`, `class_schema()`, `class_scopes()`, `class_security_scheme_details()`, `class_security_schemes()`

Examples

```
class_oauth2_implicit_flow(
  authorization_url = "https://example.com/authorize",
  refresh_url = "https://example.com/refresh"
)
```

```
class_oauth2_security_scheme
    OAuth2 security schemes
```

Description

Defines an OAuth2 security scheme that can be used by the operations.

Usage

```
class_oauth2_security_scheme(
    ...,
    implicit_flow = class_oauth2_implicit_flow(),
    password_flow = class_oauth2_token_flow(),
    client_credentials_flow = class_oauth2_token_flow(),
    authorization_code_flow = class_oauth2_authorization_code_flow()
)
```

Arguments

... These dots are for future extensions and must be empty.

implicit_flow An oauth2_implicit_flow object created with [class_oauth2_implicit_flow\(\)](#).

password_flow, client_credentials_flow An oauth2_token_flow object created with [class_oauth2_token_flow\(\)](#).

authorization_code_flow An oauth2_authorization_code_flow object created with [class_oauth2_authorization_code_flow\(\)](#).

Value

An oauth2_security_scheme S7 object, with fields implicit_flow, password_flow, client_credentials_flow, and authorization_code_flow.

See Also

security scheme objects: [as_api_key_security_scheme\(\)](#), [as_oauth2_authorization_code_flow\(\)](#), [as_oauth2_implicit_flow\(\)](#), [as_oauth2_security_scheme\(\)](#), [as_oauth2_token_flow\(\)](#), [as_scopes\(\)](#), [as_security_scheme\(\)](#), [as_security_scheme_details\(\)](#), [as_security_schemes\(\)](#), [class_api_key_security_scheme\(\)](#), [class_oauth2_authorization_code_flow\(\)](#), [class_oauth2_implicit_flow\(\)](#), [class_oauth2_token_flow\(\)](#), [class_scopes\(\)](#), [class_security_scheme_details\(\)](#), [class_security_schemes\(\)](#)

components object: [as_api_key_security_scheme\(\)](#), [as_components\(\)](#), [as_oauth2_authorization_code_flow\(\)](#), [as_oauth2_implicit_flow\(\)](#), [as_oauth2_security_scheme\(\)](#), [as_oauth2_token_flow\(\)](#), [as_reference\(\)](#), [as_schema\(\)](#), [as_scopes\(\)](#), [as_security_scheme\(\)](#), [as_security_scheme_details\(\)](#), [as_security_schemes\(\)](#), [class_api_key_security_scheme\(\)](#), [class_components\(\)](#), [class_oauth2_authorization_code_flow\(\)](#), [class_oauth2_implicit_flow\(\)](#), [class_oauth2_token_flow\(\)](#), [class_reference\(\)](#), [class_schema\(\)](#), [class_scopes\(\)](#), [class_security_scheme_details\(\)](#), [class_security_schemes\(\)](#)

Examples

```
class_oauth2_security_scheme()
class_oauth2_security_scheme(
  password_flow = class_oauth2_token_flow(token_url = "/tokens/passwords")
)
```

```
class_oauth2_token_flow
```

OAuth2 token flow object

Description

An oauth2_token_flow object describes the configuration for the OAuth Resource Owner Password flow or the OAuth Client Credentials flow (previously called application in OpenAPI 2.0).

Usage

```
class_oauth2_token_flow(
  token_url = character(),
  ...,
  refresh_url = character(),
  scopes = class_scopes()
)
```

Arguments

token_url	Character vector (required). The token URL to be used for this flow. This must be in the form of a URL. The OAuth2 standard requires the use of TLS.
...	These dots are for future extensions and must be empty.
refresh_url	Character scalar (optional). The URL to be used for obtaining refresh tokens. This must be in the form of a URL. The OAuth2 standard requires the use of TLS.
scopes	An optional class_scopes() object with the available scopes for the OAuth2 security scheme.

Value

An oauth2_token_flow S7 object, with fields token_url, refresh_url, and scopes.

See Also

security scheme objects: [as_api_key_security_scheme\(\)](#), [as_oauth2_authorization_code_flow\(\)](#), [as_oauth2_implicit_flow\(\)](#), [as_oauth2_security_scheme\(\)](#), [as_oauth2_token_flow\(\)](#), [as_scopes\(\)](#), [as_security_scheme\(\)](#), [as_security_scheme_details\(\)](#), [as_security_schemes\(\)](#), [class_api_key_security_sche](#), [class_oauth2_authorization_code_flow\(\)](#), [class_oauth2_implicit_flow\(\)](#), [class_oauth2_security_scheme\(\)](#), [class_scopes\(\)](#), [class_security_scheme_details\(\)](#), [class_security_schemes\(\)](#)

components object: `as_api_key_security_scheme()`, `as_components()`, `as_oauth2_authorization_code_flow()`, `as_oauth2_implicit_flow()`, `as_oauth2_security_scheme()`, `as_oauth2_token_flow()`, `as_reference()`, `as_schema()`, `as_scopes()`, `as_security_scheme()`, `as_security_scheme_details()`, `as_security_schemes()`, `class_api_key_security_scheme()`, `class_components()`, `class_oauth2_authorization_code_flow()`, `class_oauth2_implicit_flow()`, `class_oauth2_security_scheme()`, `class_reference()`, `class_schema()`, `class_scopes()`, `class_security_scheme_details()`, `class_security_schemes()`

Examples

```
class_oauth2_token_flow(  
  token_url = "https://example.com/token",  
  refresh_url = "https://example.com/refresh",  
  scopes = class_scopes(  
    name = c("server:read", "server:write"),  
    description = c("Read server settings", "Write server settings")  
  )  
)
```

class_origin	Source information for the API description
--------------	--

Description

An origin object provides information about the primary source document(s) used to build an API.

Usage

```
class_origin(  
  url = character(),  
  ...,  
  format = character(),  
  version = character()  
)
```

Arguments

url	Character (required). The URL(s) where the document(s) can be found.
...	These dots are for future extensions and must be empty.
format	Character scalar (optional). The format of the document. Presently this will likely always be "openapi".
version	Character scalar (optional). The specification version (relative to the format) used in defining the document. Not to be confused with the version of the API description itself. Most often this will be "3.0" (as in "OpenAPI Specification version 3.0"), "3.1", or a patch version of those.

Value

An origin S7 object describing where to find the API description, with fields `url`, `format`, and `version`.

See Also

info object: [as_contact\(\)](#), [as_info\(\)](#), [as_license\(\)](#), [as_origin\(\)](#), [class_contact\(\)](#), [class_info\(\)](#), [class_license\(\)](#)

Examples

```
class_origin(
  "https://api.open.fec.gov/swagger/",
  format = "openapi",
  version = "3.0"
)
```

class_paths

*The available paths and operations for the API***Description**

Holds the relative paths to the individual endpoints and their operations. The path is appended to the URL from the [class_servers\(\)](#) object in order to construct the full URL. The paths may be empty.

Usage

```
class_paths(...)
```

Arguments

... A data.frame, or arguments to pass to [tibble::tibble\(\)](#).

Value

A paths S7 object with details about API endpoints.

See Also

paths object: [as_paths\(\)](#)

Examples

```
class_paths()
class_paths(
  tibble::tibble(
    endpoint = c("/endpoint1", "/endpoint2"),
    operations = list(
      tibble::tibble(operation_properties = 1:2),
      tibble::tibble(operation_properties = 3:5)
    )
  )
)
```

class_rapid	<i>R API description object</i>
-------------	---------------------------------

Description

An object that represents an API.

Usage

```
class_rapid(
  info = class_info(),
  ...,
  servers = class_servers(),
  components = class_components(),
  paths = class_paths(),
  security = class_security()
)
```

Arguments

info	An info object defined by class_info() .
...	These dots are for future extensions and must be empty.
servers	A servers object defined by class_servers() .
components	A components object defined by class_components() .
paths	A paths object defined by class_paths() .
security	A security object defined by class_security() .

Value

A rapid S7 object, with properties info, servers, components, paths, and security.

See Also

rapid object: [as_api_object\(\)](#), [as_rapid\(\)](#), [expand_servers\(\)](#)

Examples

```
class_rapid()
class_rapid(
  info = class_info(title = "A", version = "1"),
  servers = class_servers(
    url = "https://development.gigantic-server.com/v1"
  )
)
class_rapid(
  info = class_info(title = "A", version = "1"),
  servers = class_servers(
```

```

url = c(
  "https://development.gigantic-server.com/v1",
  "https://staging.gigantic-server.com/v1",
  "https://api.gigantic-server.com/v1"
),
description = c(
  "Development server",
  "Staging server",
  "Production server"
),
components = class_components(
  security_schemes = class_security_schemes(
    name = "a",
    details = class_security_scheme_details(
      class_api_key_security_scheme("param", "query")
    )
  )
)
)
)
)

```

class_reference	<i>A simple object for referencing other components in the API description</i>
-----------------	--

Description

The reference object allows for reuse of components between different parts of the API description. These objects are currently simple character references, but may change structure in the future to automatically resolve references.

Usage

```

class_reference(
  ref_uri = character(0),
  summary = character(0),
  description = character(0)
)

```

Arguments

ref_uri	Character scalar. The reference identifier. This must be in the form of a URI.
summary	Character scalar (optional). A short summary which by default should override that of the referenced component. If the referenced object-type does not allow a summary field, then this field has no effect.
description	Character scalar (optional). A description which by default should override that of the referenced component. CommonMark syntax may be used for rich text representation. If the referenced object-type does not allow a description field, then this field has no effect.

Value

A reference S7 object pointing (by name) to another part of the rapid object.

See Also

reference objects: `as_reference()`
components object: `as_api_key_security_scheme()`, `as_components()`, `as_oauth2_authorization_code_flow()`, `as_oauth2_implicit_flow()`, `as_oauth2_security_scheme()`, `as_oauth2_token_flow()`, `as_reference()`, `as_schema()`, `as_scopes()`, `as_security_scheme()`, `as_security_scheme_details()`, `as_security_schemes()`, `class_api_key_security_scheme()`, `class_components()`, `class_oauth2_authorization_code_flow()`, `class_oauth2_implicit_flow()`, `class_oauth2_security_scheme()`, `class_oauth2_token_flow()`, `class_schema()`, `class_scopes()`, `class_security_scheme_details()`, `class_security_schemes()`

Examples

```
class_reference("#/components/schemas/Pet")
```

class_schema	<i>Reusable input and output data type definitions</i>
--------------	--

Description

The schema object allows the definition of input and output data types. These types can be objects, but also primitives and arrays. This object is a superset of the [JSON Schema Specification Draft 2020-12](#).

Usage

```
class_schema(  
  type = c("string", "number", "integer", "boolean", "array", "object"),  
  ...,  
  nullable = FALSE,  
  description = character(),  
  format = character()  
)
```

Arguments

- | | |
|-------------|---|
| type | Factor (or coercible to factor). The type of object being defined. Currently must be one of "string", "number", "integer", "boolean", "array", or "object". |
| ... | These dots are for future extensions and must be empty. |
| nullable | Logical scalar (default FALSE). Whether the parameter can be set to NULL. |
| description | Character scalar (optional). A description of the object defined by the schema. |
| format | Character scalar (optional). The format of the object. Essentially a sub-type. |

Value

A schema S7 object describing the data type, with fields type, nullable, description, and format.

See Also

schema objects: [as_schema\(\)](#)

components object: [as_api_key_security_scheme\(\)](#), [as_components\(\)](#), [as_oauth2_authorization_code_flow\(\)](#), [as_oauth2_implicit_flow\(\)](#), [as_oauth2_security_scheme\(\)](#), [as_oauth2_token_flow\(\)](#), [as_reference\(\)](#), [as_schema\(\)](#), [as_scopes\(\)](#), [as_security_scheme\(\)](#), [as_security_scheme_details\(\)](#), [as_security_schemes\(\)](#), [class_api_key_security_scheme\(\)](#), [class_components\(\)](#), [class_oauth2_authorization_code_flow\(\)](#), [class_oauth2_implicit_flow\(\)](#), [class_oauth2_security_scheme\(\)](#), [class_oauth2_token_flow\(\)](#), [class_reference\(\)](#), [class_scopes\(\)](#), [class_security_scheme_details\(\)](#), [class_security_schemes\(\)](#)

Examples

```
class_schema("string")
class_schema("string", nullable = TRUE, description = "A nullable string.")
```

class_scopes	<i>OAuth2 flow scopes objects</i>
--------------	-----------------------------------

Description

The available scopes for an OAuth2 flow.

Usage

```
class_scopes(name = character(), description = character())
```

Arguments

name Character vector (required). The name of the scope.

description Character vector (required). A short description of the scope.

Value

A scopes S7 object, with fields name and description.

See Also

security scheme objects: [as_api_key_security_scheme\(\)](#), [as_oauth2_authorization_code_flow\(\)](#), [as_oauth2_implicit_flow\(\)](#), [as_oauth2_security_scheme\(\)](#), [as_oauth2_token_flow\(\)](#), [as_scopes\(\)](#), [as_security_scheme\(\)](#), [as_security_scheme_details\(\)](#), [as_security_schemes\(\)](#), [class_api_key_security_sche](#), [class_oauth2_authorization_code_flow\(\)](#), [class_oauth2_implicit_flow\(\)](#), [class_oauth2_security_scheme\(\)](#), [class_oauth2_token_flow\(\)](#), [class_security_scheme_details\(\)](#), [class_security_schemes\(\)](#)

components object: `as_api_key_security_scheme()`, `as_components()`, `as_oauth2_authorization_code_flow()`, `as_oauth2_implicit_flow()`, `as_oauth2_security_scheme()`, `as_oauth2_token_flow()`, `as_reference()`, `as_schema()`, `as_scopes()`, `as_security_scheme()`, `as_security_scheme_details()`, `as_security_schemes()`, `class_api_key_security_scheme()`, `class_components()`, `class_oauth2_authorization_code_flow()`, `class_oauth2_implicit_flow()`, `class_oauth2_security_scheme()`, `class_oauth2_token_flow()`, `class_reference()`, `class_schema()`, `class_security_scheme_details()`, `class_security_schemes()`

Examples

```
class_scopes(
  name = c(
    "https://www.googleapis.com/auth/youtube",
    "https://www.googleapis.com/auth/youtube.channel-memberships.creator",
    "https://www.googleapis.com/auth/youtube.force-ssl",
    "https://www.googleapis.com/auth/youtube.readonly",
    "https://www.googleapis.com/auth/youtube.upload",
    "https://www.googleapis.com/auth/youtubepartner",
    "https://www.googleapis.com/auth/youtubepartner-channel-audit"
  ),
  description = c(
    "Manage your YouTube account",
    "See a list of your current active channel members",
    "See, edit, and permanently delete your YouTube videos",
    "View your YouTube account",
    "Manage your YouTube videos",
    "View and manage your assets and associated content on YouTube",
    "View private information of your YouTube channel"
  )
)
```

class_security

Security schemes required to execute an operation

Description

The object lists the required security schemes to execute an operation or operations.

Usage

```
class_security(name = character(), ..., required_scopes = list())
```

Arguments

name	Character vector (required). The names must correspond to security schemes declared in the <code>security_schemes</code> property of a <code>class_components()</code> .
...	These dots are for future extensions and must be empty.
required_scopes	A list of character vectors, each of which describe the scopes required for this security scheme. The vector corresponding to a given name can be empty.

Value

A security S7 object with references of security required for operations.

See Also

security object: [as_security\(\)](#)

Examples

```
class_security()  
class_security(  
  name = c("oauth2", "internalApiKey"),  
  required_scopes = list(  
    c("user", "user:email", "user:follow"),  
    character()  
  )  
)
```

class_security_scheme_details

Details of API security schemes

Description

The object provides a list of details of security schemes for the API. Each element within the list is an [abstract_security_scheme\(\)](#) object.

Usage

```
class_security_scheme_details(...)
```

Arguments

...

One or more [abstract_security_scheme\(\)](#) objects or a list of such objects. These objects must be generated by [class_api_key_security_scheme\(\)](#) or [class_oauth2_security_scheme\(\)](#) ([http_security_scheme\(\)](#) and [open_id_connect_security_scheme\(\)](#) coming soon).

Value

A security_scheme_details object, which is a validated list of [abstract_security_scheme\(\)](#) objects.

See Also

security scheme objects: `as_api_key_security_scheme()`, `as_oauth2_authorization_code_flow()`, `as_oauth2_implicit_flow()`, `as_oauth2_security_scheme()`, `as_oauth2_token_flow()`, `as_scopes()`, `as_security_scheme()`, `as_security_scheme_details()`, `as_security_schemes()`, `class_api_key_security_scheme()`, `class_oauth2_authorization_code_flow()`, `class_oauth2_implicit_flow()`, `class_oauth2_security_scheme()`, `class_oauth2_token_flow()`, `class_scopes()`, `class_security_schemes()`

components object: `as_api_key_security_scheme()`, `as_components()`, `as_oauth2_authorization_code_flow()`, `as_oauth2_implicit_flow()`, `as_oauth2_security_scheme()`, `as_oauth2_token_flow()`, `as_reference()`, `as_schema()`, `as_scopes()`, `as_security_scheme()`, `as_security_scheme_details()`, `as_security_schemes()`, `class_api_key_security_scheme()`, `class_components()`, `class_oauth2_authorization_code_flow()`, `class_oauth2_implicit_flow()`, `class_oauth2_security_scheme()`, `class_oauth2_token_flow()`, `class_reference()`, `class_schema()`, `class_scopes()`, `class_security_schemes()`

Examples

```
class_security_scheme_details(
  class_oauth2_security_scheme(
    password_flow = class_oauth2_token_flow(
      token_url = "/account/authorization",
      scopes = class_scopes(
        name = c("Catalog", "Commerce", "Playback", "Settings"),
        description = c(
          "Access all read-only content",
          "Perform account-level transactions",
          "Allow playback of restricted content",
          "Modify account settings"
        )
      )
    )
  ),
  class_oauth2_security_scheme(
    password_flow = class_oauth2_token_flow(
      token_url = "/account/profile/authorization",
      scopes = class_scopes(
        name = "Catalog",
        description = "Modify profile preferences and activity"
      )
    )
  ),
  class_api_key_security_scheme(
    parameter_name = "authorization",
    location = "header"
  ),
  class_api_key_security_scheme(
    parameter_name = "authorization",
    location = "header"
  )
)
```

class_security_schemes

Reusable security schemes for an API

Description

The object provides reusable security schemes for the API. These schemes may be referenced by name in the top-level security object or within paths objects.

Usage

```
class_security_schemes(
  name = character(),
  details = class_security_scheme_details(),
  ...,
  description = character()
)
```

Arguments

name	Character vector (required). Names by which security schemes will be referenced.
details	The details of each security scheme, as a <code>class_security_scheme_details()</code> object.
...	These dots are for future extensions and must be empty.
description	Character vector (optional). A short description for the security schemes. CommonMark syntax may be used for rich text representation.

Value

A security_schemes S7 object with details about security available for operations.

See Also

security scheme objects: `as_api_key_security_scheme()`, `as_oauth2_authorization_code_flow()`, `as_oauth2_implicit_flow()`, `as_oauth2_security_scheme()`, `as_oauth2_token_flow()`, `as_scopes()`, `as_security_scheme()`, `as_security_scheme_details()`, `as_security_schemes()`, `class_api_key_security_scheme()`, `class_oauth2_authorization_code_flow()`, `class_oauth2_implicit_flow()`, `class_oauth2_security_scheme()`, `class_oauth2_token_flow()`, `class_scopes()`, `class_security_scheme_details()`

components object: `as_api_key_security_scheme()`, `as_components()`, `as_oauth2_authorization_code_flow()`, `as_oauth2_implicit_flow()`, `as_oauth2_security_scheme()`, `as_oauth2_token_flow()`, `as_reference()`, `as_schema()`, `as_scopes()`, `as_security_scheme()`, `as_security_scheme_details()`, `as_security_schemes()`, `class_api_key_security_scheme()`, `class_components()`, `class_oauth2_authorization_code_flow()`, `class_oauth2_implicit_flow()`, `class_oauth2_security_scheme()`, `class_oauth2_token_flow()`, `class_reference()`, `class_schema()`, `class_scopes()`, `class_security_scheme_details()`

Examples

```

class_security_schemes()
class_security_schemes(
  name = c(
    "accountAuth",
    "resetPasswordAuth"
  ),
  details = class_security_scheme_details(
    class_oauth2_security_scheme(
      password_flow = class_oauth2_token_flow(
        token_url = "/account/authorization",
        scopes = class_scopes(
          name = c("Catalog", "Commerce", "Playback", "Settings"),
          description = c(
            "Access all read-only content",
            "Perform account-level transactions",
            "Allow playback of restricted content",
            "Modify account settings"
          )
        )
      )
    )
  ),
  class_api_key_security_scheme(
    parameter_name = "authorization",
    location = "header"
  )
)

```

class_server_variables

A collection of string replacements for multiple servers

Description

A list of string replacements objects, each of which is constructed with `class_string_replacements()`.

Usage

```
class_server_variables(...)
```

Arguments

... One or more `class_string_replacements()` objects, or a list of `class_string_replacements()` objects.

Value

A server_variables S7 object, which is a validated list of `class_string_replacements()` objects.

See Also

servers object: [as_server_variables\(\)](#), [as_servers\(\)](#), [as_string_replacements\(\)](#), [class_servers\(\)](#), [class_string_replacements\(\)](#)

Examples

```
class_server_variables(
  list(class_string_replacements(), class_string_replacements())
)
```

class_servers	<i>An object representing a collection of servers</i>
---------------	---

Description

The object provides connectivity information for the API. The described servers are intended to be used as the base urls for calls to the API.

Usage

```
class_servers(
  url = character(),
  description = character(),
  variables = class_server_variables()
)
```

Arguments

url	Character vector (required). The urls of the target hosts. These urls support class_string_replacements() . Variable substitutions will be made when a variable is named in {brackets}.
description	Character vector (optional). Strings describing the hosts designated by url. CommonMark syntax may be used for rich text representation.
variables	A class_server_variables() object.

Value

A servers S7 object, with properties url, description, and variables.

See Also

servers object: [as_server_variables\(\)](#), [as_servers\(\)](#), [as_string_replacements\(\)](#), [class_server_variables\(\)](#), [class_string_replacements\(\)](#)

Examples

```

class_servers(
  url = c(
    "https://development.gigantic-server.com/v1",
    "https://staging.gigantic-server.com/v1",
    "https://api.gigantic-server.com/v1"
  ),
  description = c(
    "Development server",
    "Staging server",
    "Production server"
  )
)
class_servers(
  url = "https://{username}.gigantic-server.com:{port}/{basePath}",
  description = "The production API server",
  variables = class_server_variables(class_string_replacements(
    name = c("username", "port", "basePath"),
    default = c("demo", "8443", "v2"),
    description = c(
      "The active user's folder.",
      NA, NA
    ),
    enum = list(
      NULL,
      c("8443", "443"),
      NULL
    )
  ))
)

```

class_string_replacements

A set of variables for server URL template substitution

Description

A `string_replacements` object describes server variable properties used for substitution in a single server's URL template.

Usage

```

class_string_replacements(
  name = character(),
  default = character(),
  ...,
  enum = list(),
  description = character()
)

```

Arguments

name	Character vector (required). The names of the variables.
default	Character vector (required). The default value to use for substitution of each variable, which shall be sent if an alternate value is not supplied. Note this behavior is different than the Schema Object's treatment of default values, because in those cases parameter values are optional. If the enum is defined, the value must exist in the enum's values.
...	These dots are for future extensions and must be empty.
enum	List of (potentially empty) character vectors (optional). An enumeration of valid string values to be used if the substitution options are from a limited set.
description	Character vector (optional). An optional description for each server variable. CommonMark syntax may be used for rich text representation.

Value

A string_replacements S7 object describing a collection of string variables for a single server, with fields name, default, enum, and description.

See Also

servers object: [as_server_variables\(\)](#), [as_servers\(\)](#), [as_string_replacements\(\)](#), [class_server_variables\(\)](#), [class_servers\(\)](#)

Examples

```
class_string_replacements(
  "username",
  "demo",
  enum = c("demo", "other"),
  description = "The active user's folder."
)
```

expand_servers	<i>Expand server urls to absolute paths</i>
----------------	---

Description

expand_servers() uses the origin property of a rapid object to expand the servers url property to an absolute path.

Usage

```
expand_servers(x, ...)
```

Arguments

<code>x</code>	The object to update. Must be a rapid.
<code>...</code>	These dots are for future extensions and must be empty.

Value

A rapid object as returned by `class_rapid()`, with absolute server paths.

See Also

rapid object: `as_api_object()`, `as_rapid()`, `class_rapid()`

Index

- * **components_reference**
 - as_reference, [16](#)
 - class_reference, [41](#)
- * **components_schema**
 - as_schema, [17](#)
 - class_schema, [42](#)
- * **components_security_schemes**
 - as_api_key_security_scheme, [3](#)
 - as_oauth2_authorization_code_flow, [10](#)
 - as_oauth2_implicit_flow, [11](#)
 - as_oauth2_security_scheme, [12](#)
 - as_oauth2_token_flow, [13](#)
 - as_scopes, [18](#)
 - as_security_scheme, [20](#)
 - as_security_scheme_details, [21](#)
 - as_security_schemes, [23](#)
 - class_api_key_security_scheme, [28](#)
 - class_oauth2_authorization_code_flow, [33](#)
 - class_oauth2_implicit_flow, [34](#)
 - class_oauth2_security_scheme, [36](#)
 - class_oauth2_token_flow, [37](#)
 - class_scopes, [43](#)
 - class_security_scheme_details, [45](#)
 - class_security_schemes, [47](#)
- * **components**
 - as_api_key_security_scheme, [3](#)
 - as_components, [5](#)
 - as_oauth2_authorization_code_flow, [10](#)
 - as_oauth2_implicit_flow, [11](#)
 - as_oauth2_security_scheme, [12](#)
 - as_oauth2_token_flow, [13](#)
 - as_reference, [16](#)
 - as_schema, [17](#)
 - as_scopes, [18](#)
 - as_security_scheme, [20](#)
 - as_security_scheme_details, [21](#)
 - as_security_schemes, [23](#)
 - class_api_key_security_scheme, [28](#)
 - class_oauth2_authorization_code_flow, [33](#)
 - class_oauth2_implicit_flow, [34](#)
 - class_oauth2_security_scheme, [36](#)
 - class_oauth2_token_flow, [37](#)
 - class_reference, [41](#)
 - class_schema, [42](#)
 - class_scopes, [43](#)
 - class_security_scheme_details, [45](#)
 - class_security_schemes, [47](#)
- * **info**
 - as_contact, [7](#)
 - as_info, [8](#)
 - as_license, [9](#)
 - as_origin, [14](#)
 - class_contact, [30](#)
 - class_info, [31](#)
 - class_license, [32](#)
 - class_origin, [38](#)
- * **paths**
 - as_paths, [15](#)
 - class_paths, [39](#)
- * **rapid**
 - as_api_object, [4](#)
 - as_rapid, [15](#)
 - class_rapid, [40](#)
 - expand_servers, [51](#)
- * **security**
 - as_security, [19](#)
 - class_security, [44](#)
- * **servers**
 - as_server_variables, [25](#)
 - as_servers, [26](#)
 - as_string_replacements, [27](#)
 - class_server_variables, [48](#)
 - class_servers, [49](#)

- class_string_replacements, 50
- abort(), 3, 5, 7–11, 13, 14, 16, 17
- abstract_security_scheme(), 45
- as_api_key_security_scheme, 3
- as_api_key_security_scheme(), 6, 10–13, 17–20, 22, 24, 28, 29, 34–38, 42–44, 46, 47
- as_api_object, 4
- as_api_object(), 16, 40, 52
- as_components, 5
- as_components(), 4, 10–13, 17–20, 22, 24, 28, 29, 34–36, 38, 42–44, 46, 47
- as_contact, 7
- as_contact(), 8, 9, 14, 30, 32, 33, 39
- as_info, 8
- as_info(), 7, 9, 14, 30, 32, 33, 39
- as_license, 9
- as_license(), 7, 8, 14, 30, 32, 33, 39
- as_oauth2_authorization_code_flow, 10
- as_oauth2_authorization_code_flow(), 4, 6, 11–13, 17–20, 22, 24, 28, 29, 34–38, 42–44, 46, 47
- as_oauth2_implicit_flow, 11
- as_oauth2_implicit_flow(), 4, 6, 10, 12, 13, 17–20, 22, 24, 28, 29, 34–38, 42–44, 46, 47
- as_oauth2_security_scheme, 12
- as_oauth2_security_scheme(), 4, 6, 10, 11, 13, 17–20, 22, 24, 28, 29, 34–38, 42–44, 46, 47
- as_oauth2_token_flow, 13
- as_oauth2_token_flow(), 4, 6, 10–12, 17–20, 22, 24, 28, 29, 34–38, 42–44, 46, 47
- as_origin, 14
- as_origin(), 7–9, 30, 32, 33, 39
- as_paths, 15
- as_paths(), 39
- as_rapid, 15
- as_rapid(), 5, 40, 52
- as_reference, 16
- as_reference(), 4, 6, 10–13, 18–20, 22, 24, 28, 29, 34–36, 38, 42–44, 46, 47
- as_schema, 17
- as_schema(), 4, 6, 10–13, 17, 19, 20, 22, 24, 28, 29, 34–36, 38, 42–44, 46, 47
- as_scopes, 18
- as_scopes(), 4, 6, 10–13, 17, 18, 20, 22, 24, 28, 29, 34–38, 42–44, 46, 47
- as_security, 19
- as_security(), 45
- as_security_scheme, 20
- as_security_scheme(), 4, 6, 10–13, 17–19, 22, 24, 28, 29, 34–38, 42–44, 46, 47
- as_security_scheme_details, 21
- as_security_scheme_details(), 4, 6, 10–13, 17–20, 24, 28, 29, 34–38, 42–44, 46, 47
- as_security_schemes, 23
- as_security_schemes(), 4, 6, 10–13, 17–20, 22, 28, 29, 34–38, 42–44, 46, 47
- as_server_variables, 25
- as_server_variables(), 26, 27, 49, 51
- as_servers, 26
- as_servers(), 25, 27, 49, 51
- as_string_replacements, 27
- as_string_replacements(), 25, 26, 49, 51
- class_api_key_security_scheme, 28
- class_api_key_security_scheme(), 3, 4, 6, 10–13, 17–20, 22, 24, 29, 34–38, 42–47
- class_components, 29
- class_components(), 4–6, 11–13, 17–20, 22, 24, 28, 34–36, 38, 40, 42–44, 46, 47
- class_contact, 30
- class_contact(), 7–9, 14, 31–33, 39
- class_info, 31
- class_info(), 7–9, 14, 30, 33, 39, 40
- class_license, 32
- class_license(), 7–9, 14, 30, 32, 39
- class_oauth2_authorization_code_flow, 33
- class_oauth2_authorization_code_flow(), 4, 6, 10–13, 17–20, 22, 24, 28, 29, 35–38, 42–44, 46, 47
- class_oauth2_implicit_flow, 34
- class_oauth2_implicit_flow(), 4, 6, 10–13, 17–20, 22, 24, 28, 29, 34, 36–38, 42–44, 46, 47
- class_oauth2_security_scheme, 36
- class_oauth2_security_scheme(), 4, 6, 10–13, 17–20, 22, 24, 28, 29, 34, 35, 37, 38, 42–47
- class_oauth2_token_flow, 37

`class_oauth2_token_flow()`, 4, 6, 10–13,
17–20, 22, 24, 28, 29, 34–36, 42–44,
46, 47

`class_origin`, 38

`class_origin()`, 7–9, 14, 30, 32, 33

`class_paths`, 39

`class_paths()`, 15, 40

`class_rapid`, 40

`class_rapid()`, 5, 15, 16, 52

`class_reference`, 41

`class_reference()`, 4, 6, 11–13, 16–20, 22,
24, 28, 29, 34–36, 38, 43, 44, 46, 47

`class_schema`, 42

`class_schema()`, 4, 6, 11–13, 17–20, 22, 24,
28, 29, 34–36, 38, 42, 44, 46, 47

`class_scopes`, 43

`class_scopes()`, 4, 6, 10–13, 17–20, 22, 24,
28, 29, 34–38, 42, 43, 46, 47

`class_security`, 44

`class_security()`, 19, 40

`class_security_scheme_details`, 45

`class_security_scheme_details()`, 4, 6,
10–13, 17–22, 24, 28, 29, 34–38,
42–44, 47

`class_security_schemes`, 47

`class_security_schemes()`, 4, 6, 10–13,
17–20, 22–24, 28, 29, 34–38, 42–44,
46

`class_server_variables`, 48

`class_server_variables()`, 25–27, 49, 51

`class_servers`, 49

`class_servers()`, 25–27, 39, 40, 49, 51

`class_string_replacements`, 50

`class_string_replacements()`, 25–27, 48,
49

`expand_servers`, 51

`expand_servers()`, 5, 16, 40

`jsonlite::fromJSON()`, 16

`license()`, 31

`snakecase::to_snake_case()`, 3–5, 7–14,
16, 17, 19, 20, 26, 27

`tibble::tibble()`, 39

`url()`, 16

`yaml::read_yaml()`, 16