Package 'thriftr'

March 18, 2019

Type Package

Title Apache Thrift Client Server

Version 1.1.5

Date 2019-03-12

Author Marek Jagielski [aut, cre, cph],
Lixin Yu [aut, cph]

Maintainer Marek Jagielski <marek.jagielski@gmail.com>

Description Pure R implementation of Apache Thrift.
This library doesn't require any code generation.
To learn more about Thrift go to <https://thrift.apache.org>.

License MIT + file LICENSE

URL https://github.com/systemincloud/thriftr

BugReports https://github.com/systemincloud/thriftr/issues

Suggests testthat

Encoding UTF-8

Imports R6, rly, stringi

RoxygenNote 6.0.1

Collate 'thrift.R' 'transport.R' 'rpc.R' 'parser.R'
'protocol_binary.R' 'server.R' 'transport_buffered.R'
'transport_memory.R' 'transport_socket.R' 'utils.R'

NeedsCompilation no

Repository CRAN

Date/Publication 2019-03-18 15:30:03 UTC

R topics documented:

`binary_read_val` *Binary protocol: read value from binary buffer*

Description

Binary protocol: read value from binary buffer

Usage

```
binary_read_val(inbuf, ttype, spec = NA, decode_response = TRUE)
```

Arguments

`inbuf`	binary buffor
`ttype`	type of value
`spec`	specification of value
`decode_response`	for string decode binary as chars

Value

value of type ttype

binary_write_val *Binary protocol: write value to binary buffer*

Description

Binary protocol: write value to binary buffer

Usage

```
binary_write_val(outbuf, ttype, val, spec = NA)
```

Arguments

outbuf	binary buffor
ttype	type of value
val	value to write
spec	specification of value

hexlify *hexlify*

Description

String representation of raw array

Usage

```
hexlify(byte_array, delimeter = " ")
```

Arguments

byte_array	raw array
delimeter	separation character

Value

string

make_client *Create client side thrift API*

Description

Create client side thrift API

Usage

```
make_client(service, host = "localhost", port = 9090,
  proto_factory = TBinaryProtocolFactory$new(),
  trans_factory = TBufferedTransportFactory$new())
```

Arguments

service	parsed service
host	server host
port	server tcp port
proto_factory	factory that generates protocol implementation
trans_factory	factory that generates transport implementation

Examples

```
## Not run:
# File calc.thrift content:
# service Calculator {
#   i32 add(1:i32 a, 2:i32 b);
#   i32 sub(1:i32 a, 2:i32 b);
#   i32 mult(1:i32 a, 2:i32 b);
#   i32 div(1:i32 a, 2:i32 b);
# }
#

calc_thrift <- thriftr::t_load("calc.thrift", module_name="calc_thrift")

cal <- thriftr::make_client(
    calc_thrift$Calculator,
    "127.0.0.1",
    6000)

a <- cal$mult(5, 2)
b <- cal$sub(7, 3)
c <- cal$sub(6, 4)
d <- cal$mult(b, 10)
e <- cal$add(a, d)
f <- cal$div(e, c)
print(f)

## End(Not run)
```

make_server *Create server side thrift API*

Description

Create server side thrift API

Usage

```
make_server(service, handler, host = "localhost", port = 9090,
  proto_factory = TBinaryProtocolFactory$new(),
  trans_factory = TBufferedTransportFactory$new())
```

Arguments

service	parsed service
handler	R6 class implementing service
host	server host
port	port server tcp port
proto_factory	factory that generates protocol implementation
trans_factory	factory that generates transport implementation

Examples

```
## Not run:
# File calc.thrift content:
# service Calculator {
#   i32 add(1:i32 a, 2:i32 b);
#   i32 sub(1:i32 a, 2:i32 b);
#   i32 mult(1:i32 a, 2:i32 b);
#   i32 div(1:i32 a, 2:i32 b);
# }
#

calc_thrift <- thriftr::t_load("calc.thrift", module_name="calc_thrift")

Dispatcher <- R6::R6Class("Dispatcher",
  public = list(
    add = function(a, b) {
      print(sprintf("add -> %s + %s", a, b))
      return(a + b)
    },
    sub = function(a, b) {
      print(sprintf("sub -> %s - %s", a, b))
      return(a - b)
    },
    mult = function(a, b) {
```

```
      print(sprintf("mult -> %s * %s", a, b))
      return(a * b)
    },
    div = function(a, b) {
      print(sprintf("div -> %s / %s", a, b))
      return(a / b)
    }
  )
)

server <- thriftr::make_server(
    calc_thrift$Calculator,
    Dispatcher$new(),
    "127.0.0.1",
    6000)

print("serving...")

server$serve()

## End(Not run)
```

`parse` *Parse a single thrift file to R6 class instance*

Description

Parse a single thrift file to R6 class instance

Usage

```
parse(path, module_name = NA, include_dirs = NA, lexer = NA,
  parser = NA, enable_cache = TRUE)
```

Arguments

`path`	file path to parse, should be a string ending with '.thrift'
`module_name`	the name for parsed module, the default is the basename without extension of 'path'
`include_dirs`	directories to find thrift files while processing the 'include' directive, by default: ['.']
`lexer`	rly lexer to use, if not provided, 'parse' will use a new one
`parser`	rly parser to use, if not provided, 'parse' will use a new one
`enable_cache`	if this is set to be 'TRUE', parsed module will be cached, this is enabled by default. If 'module_name' is provided, use it as cache key, else use the 'path'

Value

Thrift module

`parse_spec` *parse_spec*

Description

String representation of specification

Usage

```
parse_spec(ttype, spec = NA)
```

Arguments

`ttype`	type
`spec`	specification

Value

string representation

`TBinaryProtocol` *TBinaryProtocol*

Description

Binary implementation of the Thrift protocol driver.

Usage

```
TBinaryProtocol
```

Format

An `R6Class` generator object

`TBinaryProtocolFactory` *TBinaryProtocolFactory*

Description

TBinaryProtocolFactory generates TBinaryProtocol driver.

Usage

```
TBinaryProtocolFactory
```

Format

An `R6Class` generator object

`TBufferedTransport` *TBufferedTransport*

Description

Class that wraps another transport and buffers its I/O.

Usage

```
TBufferedTransport
```

Format

An `R6Class` generator object

`TBufferedTransportFactory` *TBufferedTransportFactory*

Description

TBufferedTransportFactory generates TBufferedTransport.

Usage

```
TBufferedTransportFactory
```

Format

An `R6Class` generator object

TClient *TClient*

Description

TClient implements client api of thrift service.

Usage

```
TClient
```

Format

An R6Class generator object

TMemoryBuffer *TMemoryBuffer*

Description

Wraps a raw array as a TTransport.

Usage

```
TMemoryBuffer
```

Format

An R6Class generator object

to_proper_struct *to_proper_struct*

Description

Help method for tests. It changes predefined structure to parsed thrift instead of parsing file.

Usage

```
to_proper_struct(thrift_spec_list, default_spec)
```

Arguments

`thrift_spec_list`	raw array
`default_spec`	separation character

Value

R6 class

`TPayload` *TPayload*

Description

Base class for all complex types of api.

Usage

```
TPayload
```

Format

An `R6Class` generator object

`TServerSocket` *TServerSocket*

Description

Socket implementation for server side.

Usage

```
TServerSocket
```

Format

An `R6Class` generator object

TSocket *TSocket*

Description

Socket implementation for client side.

Usage

```
TSocket
```

Format

An R6Class generator object

TType *TType*

Description

Identificator of value type.

Usage

```
TType
```

Format

An object of class environment of length 18.

t_load *Load thrift file as a R6 instance.*

Description

The module loaded and objects inside may only be pickled if module_name was provided.

Usage

```
t_load(path, module_name = NA, include_dirs = NA)
```

Arguments

`path`	file path to parse, should be a string ending with '.thrift'
`module_name`	the name for parsed module, the default is the basename without extension of 'path'
`include_dirs`	directories to find thrift files while processing the 'include' directive, by default: ['.']

Value

Thrift R6 class instance

Index