
iofree

GuYingbo

Mar 30, 2021

CONTENTS:

1	iofree package	1
1.1	Submodules	2
1.1.1	iofree.exceptions module	2
1.1.2	iofree.schema module	3
2	Indices and tables	7
	Python Module Index	9
	Index	11

IOFREE PACKAGE

iofree is an easy-to-use and powerful library to help you implement network protocols and binary parsers.

```
class iofree.LinkedList (parser: iofree.Parser, next_: Optional[LinkedList])
    Bases: object

    next

    parser

class iofree.Parser (gen: Generator)
    Bases: object

    finished () → bool

    get_result () → Any
        raises NoResult exception if no result has been set

    has_more_data () → bool
        indicate whether input has some bytes left

    property has_result

    parse (data: bytes, *, strict: bool = True) → Any
        parse bytes

    read_output_bytes () → bytes

    readall () → bytes
        retrieve data from input back

    respond (*, data: bytes = b'', close: bool = False, exc: Optional[Exception] = None, result: Any =
        <object object>) → None
        produce some event data to interact with a stream: data: bytes to send to the peer close: whether the socket
        should be closed exc: raise an exception to break the loop result: result to return

    run (sock: _socket.socket) → Any
        reference implementation of how to deal with socket

    send (data: bytes = b'') → None
        send data for parsing

    send_event (event: Any) → None

    set_result (result) → None

class iofree.ParserChain (*parsers: iofree.Parser)
    Bases: object

    send (data: bytes) → None
```

```
class iofree.State
    Bases: enum.IntEnum

    An enumeration.

class iofree.Traps
    Bases: enum.IntEnum

    An enumeration.

iofree.get_parser() → Generator[tuple, iofree.Parser, iofree.Parser]
    get current parser object

iofree.parser(generator_func: Callable) → Callable
    decorator function to wrap a generator

iofree.peek(nbytes: int = 1, *, from_=None) → Generator[tuple, bytes, bytes]
    peek many bytes without taking them away from buffer

iofree.read(nbytes: int = 0, *, from_=None) → Generator[tuple, bytes, bytes]
    if nbytes = 0, read as many as possible, empty bytes is valid; if nbytes > 0, read exactly nbytes

iofree.read_int(nbytes: int, byteorder: str = 'big', *, signed: bool = False, from_=None) → Generator[tuple, int, int]
    read some bytes as integer

iofree.read_more(nbytes: int = 1, *, from_=None) → Generator[tuple, bytes, bytes]
    read at least nbytes

iofree.read_raw_struct(struct_obj: Struct, *, from_=None) → Generator[tuple, tuple, tuple]
    read raw struct formatted data

iofree.read_struct(fmt: str, *, from_=None) → Generator[tuple, tuple, tuple]
    read specific formatted data

iofree.read_until(data: bytes, *, return_tail: bool = True, from_=None) → Generator[tuple, bytes, bytes]
    read until some bytes appear

iofree.wait() → Generator[tuple, bytes, Optional[object]]
    wait for next send event

iofree.wait_event() → Generator[tuple, Any, Any]
    wait for an event
```

1.1 Submodules

1.1.1 iofree.exceptions module

```
exception iofree.exceptions.NoResult
    Bases: Exception

exception iofree.exceptions.ParseError
    Bases: Exception
```

1.1.2 iofree.schema module

```

class iofree.schema.BinarySchema(*args)
    Bases: object

    The main class for users to define their own binary structures

    property binary

    member_get(name)

    member_set(name, value, binary)

class iofree.schema.BinarySchemaMetaclass
    Bases: type

    get_parser() → iofree.Parser

    get_value() → Generator[tuple, Any, iofree.schema.BinarySchema]
        get BinarySchema object from bytes

    parse(data: bytes, *, strict: bool = True) → iofree.schema.BinarySchema

class iofree.schema.Bytes(length: int)
    Bases: iofree.schema.Unit

    get_value()
        get object you want from bytes

class iofree.schema.Convert(unit: iofree.schema.Unit, *, encode: Callable = None, decode:
                                Callable = None)
    Bases: iofree.schema.Unit

    get_value()
        get object you want from bytes

class iofree.schema.EndWith(bytes_: bytes)
    Bases: iofree.schema.Unit

    get_value()
        get object you want from bytes

iofree.schema.Group(**fields: Dict[str, Union[Type[iofree.schema.BinarySchema],
                                                iofree.schema.Unit]]) → Type[iofree.schema.BinarySchema]

class iofree.schema.IntUnit(length: int, byteorder: str, signed: bool = False)
    Bases: iofree.schema.Unit

    get_value()
        get object you want from bytes

class iofree.schema.LengthPrefixed(length_unit: Union[iofree.schema.StructUnit,
                                                        iofree.schema.IntUnit],
                                    object_unit: Union[Type[iofree.schema.BinarySchema],
                                                        iofree.schema.Unit])
    Bases: iofree.schema.Unit

    get_value()
        get object you want from bytes

class iofree.schema.LengthPrefixedBytes(length_unit: Union[iofree.schema.StructUnit,
                                                            iofree.schema.IntUnit])
    Bases: iofree.schema.Unit

```

```
    get_value()
        get object you want from bytes

class iofree.schema.LengthPrefixedObject (length_unit: Union[iofree.schema.StructUnit,
                                                             iofree.schema.IntUnit],
                                           object_unit: Union[Type[iofree.schema.BinarySchema],
                                                             iofree.schema.Unit])

    Bases: iofree.schema.LengthPrefixed

class iofree.schema.LengthPrefixedObjectList (length_unit:
                                                Union[iofree.schema.StructUnit,
                                                iofree.schema.IntUnit],
                                                object_unit: Union[Type[iofree.schema.BinarySchema],
                                                                    iofree.schema.Unit])

    Bases: iofree.schema.LengthPrefixed

class iofree.schema.LengthPrefixedString (length_unit: Union[iofree.schema.StructUnit,
                                                             iofree.schema.IntUnit],
                                           encoding='utf-8')

    Bases: iofree.schema.Convert

class iofree.schema.MemberDescriptor (key: str, member: Union[Type[iofree.schema.BinarySchema],
                                                             iofree.schema.Unit])

    Bases: object

    key
    member

class iofree.schema.MustEqual (unit: iofree.schema.Unit, value: Any)

    Bases: iofree.schema.Unit

    get_value()
        get object you want from bytes

class iofree.schema.SizedIntEnum (size_unit: Union[iofree.schema.StructUnit,
                                                    iofree.schema.IntUnit],
                                  enum_class: Type[enum.IntEnum])

    Bases: iofree.schema.Unit

    get_value()
        get object you want from bytes

class iofree.schema.String (length: int, encoding='utf-8')

    Bases: iofree.schema.Convert

class iofree.schema.StructUnit (format_: str)

    Bases: iofree.schema.Unit

    get_value()
        get object you want from bytes

class iofree.schema.Switch (ref: str,
                             cases: Mapping[Any, Union[Type[iofree.schema.BinarySchema], iofree.schema.Unit]])

    Bases: iofree.schema.Unit

    get_value()
        get object you want from bytes

class iofree.schema.Unit

    Bases: abc.ABC

    Unit is the base class of all units. If you can build your own unit class, you must inherit from it

    abstract get_value() → Generator
        get object you want from bytes
```


parse (*data: bytes, *, strict: bool = True*)

a convenient function to help you parse fixed bytes

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

i

`iofree`, [1](#)
`iofree.exceptions`, [2](#)
`iofree.schema`, [3](#)

B

`binary()` (*iofree.schema.BinarySchema* property), 3
BinarySchema (class in *iofree.schema*), 3
BinarySchemaMetaclass (class in *iofree.schema*), 3
Bytes (class in *iofree.schema*), 3

C

Convert (class in *iofree.schema*), 3

E

EndWith (class in *iofree.schema*), 3

F

`finished()` (*iofree.Parser* method), 1

G

`get_parser()` (in module *iofree*), 2
`get_parser()` (*iofree.schema.BinarySchemaMetaclass* method), 3
`get_result()` (*iofree.Parser* method), 1
`get_value()` (*iofree.schema.BinarySchemaMetaclass* method), 3
`get_value()` (*iofree.schema.Bytes* method), 3
`get_value()` (*iofree.schema.Convert* method), 3
`get_value()` (*iofree.schema.EndWith* method), 3
`get_value()` (*iofree.schema.IntUnit* method), 3
`get_value()` (*iofree.schema.LengthPrefixed* method), 3
`get_value()` (*iofree.schema.LengthPrefixedBytes* method), 3
`get_value()` (*iofree.schema.MustEqual* method), 4
`get_value()` (*iofree.schema.SizedIntEnum* method), 4
`get_value()` (*iofree.schema.StructUnit* method), 4
`get_value()` (*iofree.schema.Switch* method), 4
`get_value()` (*iofree.schema.Unit* method), 4
`Group()` (in module *iofree.schema*), 3

H

`has_more_data()` (*iofree.Parser* method), 1
`has_result()` (*iofree.Parser* property), 1

I

IntUnit (class in *iofree.schema*), 3
iofree
 module, 1
iofree.exceptions
 module, 2
iofree.schema
 module, 3

K

`key` (*iofree.schema.MemberDescriptor* attribute), 4

L

LengthPrefixed (class in *iofree.schema*), 3
LengthPrefixedBytes (class in *iofree.schema*), 3
LengthPrefixedObject (class in *iofree.schema*), 4
LengthPrefixedObjectList (class in *iofree.schema*), 4
LengthPrefixedString (class in *iofree.schema*), 4
LinkedNode (class in *iofree*), 1

M

`member` (*iofree.schema.MemberDescriptor* attribute), 4
`member_get()` (*iofree.schema.BinarySchema* method), 3
`member_set()` (*iofree.schema.BinarySchema* method), 3
MemberDescriptor (class in *iofree.schema*), 4
module
 iofree, 1
 iofree.exceptions, 2
 iofree.schema, 3
MustEqual (class in *iofree.schema*), 4

N

`next` (*iofree.LinkedNode* attribute), 1
NoResult, 2

P

`parse()` (*iofree.Parser* method), 1
`parse()` (*iofree.schema.BinarySchemaMetaclass* method), 3

`parse()` (*iofree.schema.Unit method*), 4
`ParseError`, 2
`Parser` (*class in iofree*), 1
`parser` (*iofree.LinkedList attribute*), 1
`parser()` (*in module iofree*), 2
`ParserChain` (*class in iofree*), 1
`peek()` (*in module iofree*), 2

R

`read()` (*in module iofree*), 2
`read_int()` (*in module iofree*), 2
`read_more()` (*in module iofree*), 2
`read_output_bytes()` (*iofree.Parser method*), 1
`read_raw_struct()` (*in module iofree*), 2
`read_struct()` (*in module iofree*), 2
`read_until()` (*in module iofree*), 2
`readall()` (*iofree.Parser method*), 1
`respond()` (*iofree.Parser method*), 1
`run()` (*iofree.Parser method*), 1

S

`send()` (*iofree.Parser method*), 1
`send()` (*iofree.ParserChain method*), 1
`send_event()` (*iofree.Parser method*), 1
`set_result()` (*iofree.Parser method*), 1
`SizedIntEnum` (*class in iofree.schema*), 4
`State` (*class in iofree*), 1
`String` (*class in iofree.schema*), 4
`StructUnit` (*class in iofree.schema*), 4
`Switch` (*class in iofree.schema*), 4

T

`Traps` (*class in iofree*), 2

U

`Unit` (*class in iofree.schema*), 4

W

`wait()` (*in module iofree*), 2
`wait_event()` (*in module iofree*), 2