

---

**iofree**

**GuYingbo**

**Apr 08, 2020**



**CONTENTS:**

<b>1</b>	<b>iofree package</b>	<b>1</b>
1.1	Submodules . . . . .	2
1.1.1	iofree.exceptions module . . . . .	2
1.1.2	iofree.schema module . . . . .	2
<b>2</b>	<b>Indices and tables</b>	<b>5</b>
	<b>Python Module Index</b>	<b>7</b>
	<b>Index</b>	<b>9</b>



## IOFREE PACKAGE

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

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

    finished() → bool

    get_result()
        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(nbytes: int = 0) → bytes
        read at most nbytes

    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)
        reference implementation of how to deal with socket

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

    set_result(result)

    write(data: bytes) → None

class iofree.State
    Bases: enum.IntEnum

    An enumeration.

class iofree.Traps
    Bases: enum.IntEnum

    An enumeration.
```

`iofree.get_parser()` → *iofree.Parser*  
get current parser object

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

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

`iofree.read(nbytes: int = 0, *, from_=None)` → 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=False, from_=None)` → int  
read some bytes as integer

`iofree.read_more(nbytes: int = 1, *, from_=None)` → bytes  
read *at least* `nbytes`

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

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

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

`iofree.wait()` → None  
wait for next send or get\_result event

`iofree.write(data: bytes)` → None  
write some bytes to the output buffer

## 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

**classmethod** `get_parser()` → *iofree.Parser*

**classmethod** `get_value()` → Generator  
get *BinarySchema* object from bytes

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

**class** `iofree.schema.BinarySchemaMetaclass(name, bases, namespace)`  
Bases: type

---

```

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

    get_value ()
        get object you want from bytes

class iofree.schema.Convert (unit: Union[iofree.schema.BinarySchemaMetaclass,
                                         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_)
    Bases: iofree.schema.Unit

    get_value ()
        get object you want from bytes

iofree.schema.Group (**fields: Dict[str, Union[iofree.schema.BinarySchemaMetaclass,
                                                iofree.schema.Unit]]) → 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.LengthPrefixedBytes (length_unit: iofree.schema.StructUnit)
    Bases: iofree.schema.Unit

    get_value ()
        get object you want from bytes

class iofree.schema.LengthPrefixedObject (length_unit: iofree.schema.StructUnit, ob-
                                         ject_unit: Union[iofree.schema.BinarySchemaMetaclass,
                                                         iofree.schema.Unit])
    Bases: iofree.schema.LengthPrefixedObjectList

class iofree.schema.LengthPrefixedObjectList (length_unit:
                                                iofree.schema.StructUnit, object_unit:
                                                Union[iofree.schema.BinarySchemaMetaclass,
                                                         iofree.schema.Unit])
    Bases: iofree.schema.Unit

    get_value ()
        get object you want from bytes

class iofree.schema.LengthPrefixedString (length_unit: iofree.schema.StructUnit,
                                         encoding='utf-8')
    Bases: iofree.schema.Unit

    get_value ()
        get object you want from bytes

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

    get_default ()

    get_value ()
        get object you want from bytes

```

```
class iofree.schema.SizedIntEnum (size_unit: iofree.schema.StructUnit, enum_class)
    Bases: iofree.schema.Unit

    get_value()
        get object you want from bytes

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

    get_value()
        get object you want from bytes

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[iofree.schema.BinarySchemaMetaclass, 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=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`, [2](#)



## INDEX

### B

BinarySchema (class in *iofree.schema*), 2  
BinarySchemaMetaclass (class in *iofree.schema*),  
2  
Bytes (class in *iofree.schema*), 2

### C

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

### E

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

### F

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

### G

get\_default() (*iofree.schema.MustEqual* method), 3  
get\_parser() (in module *iofree*), 1  
get\_parser() (*iofree.schema.BinarySchema* class  
method), 2  
get\_result() (*iofree.Parser* method), 1  
get\_value() (*iofree.schema.BinarySchema* class  
method), 2  
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.LengthPrefixedBytes*  
method), 3  
get\_value() (*iofree.schema.LengthPrefixedObjectList*  
method), 3  
get\_value() (*iofree.schema.LengthPrefixedString*  
method), 3  
get\_value() (*iofree.schema.MustEqual* method), 3  
get\_value() (*iofree.schema.SizedIntEnum* method), 4  
get\_value() (*iofree.schema.String* 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, 2

### L

LengthPrefixedBytes (class in *iofree.schema*), 3  
LengthPrefixedObject (class in *iofree.schema*), 3  
LengthPrefixedObjectList (class in  
*iofree.schema*), 3  
LengthPrefixedString (class in *iofree.schema*), 3

### M

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

### N

NoResult, 2

### P

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

### R

read() (in module *iofree*), 2

`read()` (*iofree.Parser method*), 1  
`read_int()` (*in module iofree*), 2  
`read_more()` (*in module iofree*), 2  
`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  
`set_result()` (*iofree.Parser method*), 1  
`SizedIntEnum` (*class in iofree.schema*), 3  
`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*), 1

## U

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

## W

`wait()` (*in module iofree*), 2  
`write()` (*in module iofree*), 2  
`write()` (*iofree.Parser method*), 1