
bokkichat

Release 0.0.1

Dec 14, 2018

CONTENTS

1 bokkichat package	3
1.1 Subpackages	3
1.2 Submodules	8
1.3 bokkichat.exceptions module	8
1.4 Module contents	8
2 bokkichat	9
3 Indices and tables	11
Python Module Index	13
Index	15

Contents:

BOKKICHAT PACKAGE

1.1 Subpackages

1.1.1 bokkichat.address package

Submodules

bokkichat.address.Address module

class bokkichat.address.Address.**Address** (*address: str*)

Bases: object

Class that models an Address.

__init__ (*address: str*)

Initializes the address object :param address: The actual address to which messages can be sent

Module contents

1.1.2 bokkichat.connection package

Submodules

bokkichat.connection.CliConnection module

class bokkichat.connection.CliConnection.**CliConnection** (*settings: bok-*
kichat.connection.Settings.Settings)

Bases: *bokkichat.connection.Connection.Connection*

Class that implements a CLI connection which can be used in testing

address

A CLI connection has no real address, so a dummy address is generated. :return: The address of the connection

close ()

Disconnects the Connection. :return: None

receive () → List[bokkichat.message.Message.Message]

A CLI Connection receives messages by listening to the input :return: A list of pending Message objects

send (*message: bokkichat.message.Message.Message*)

Prints a “sent” message :param message: The message to “send” :return: None

bokkichat.connection.CliSettings module

class bokkichat.connection.CliSettings.CliSettings

Bases: *bokkichat.connection.Settings.Settings*

Class that defines a Settings object for a CLI connection

classmethod **deserialize** (*_*: *str*)

Deserializes a string and generates a Settings object from it :param *_*: The serialized string :return: The deserialized Settings object

serialize () → *str*

Serializes the settings to a string :return: The serialized Settings object

bokkichat.connection.Connection module

class bokkichat.connection.Connection.Connection (*settings*: *bokkichat.connection.Settings.Settings*)

Bases: *object*

Class that defines methods a Connection must implement. A connection is the central class in bokkichat and handles all communications with the chat services.

__init__ (*settings*: *bokkichat.connection.Settings.Settings*)

Initializes the connection, with credentials provided by a Settings object. :param *settings*: The settings for the connection

address

A connection must be able to specify its own address :return: The address of the connection

close ()

Disconnects the Connection. :return: *None*

loop (*callback*: *Callable*, *sleep_time*: *int = 1*)

Starts a loop that periodically checks for new messages, calling a provided callback function in the process. :param *callback*: The callback function to call for each

received message. The callback should have the following format:

lambda connection, message: do_stuff()

Parameters **sleep_time** – The time to sleep between loops

Returns *None*

receive () → List[bokkichat.message.Message.Message]

Receives all pending messages. :return: A list of pending Message objects

send (*message*: *bokkichat.message.Message.Message*)

Sends a message. A message may be either a TextMessage or a MediaMessage. :param *message*: The message to send :return: *None*

bokkichat.connection.Settings module

class bokkichat.connection.Settings.Settings

Bases: *object*

Class that defines what methods a Settings class must implement. Settings objects are used to initialize a Connection.

classmethod deserialize (*serialized: str*)

Deserializes a string and generates a Settings object from it :param serialized: The serialized string :return: The deserialized Settings object

serialize () → str

Serializes the settings to a string :return: The serialized Settings object

bokkichat.connection.TelegramBotConnection module

class bokkichat.connection.TelegramBotConnection.**TelegramBotConnection** (*settings: bok-*
kichat.connection.TelegramB

Bases: *bokkichat.connection.Connection.Connection*

Class that implements a Telegram bot connection

__init__ (*settings: bokkichat.connection.TelegramBotSettings.TelegramBotSettings*)

Initializes the connection, with credentials provided by a Settings object. :param settings: The settings for the connection

address

A connection must be able to specify its own address :return: The address of the connection

close ()

Disconnects the Connection. :return: None

receive () → List[bokkichat.message.Message.Message]

Receives all pending messages. :return: A list of pending Message objects

send (*message: bokkichat.message.Message.Message*)

Sends a message. A message may be either a TextMessage or a MediaMessage. :param message: The message to send :return: None

bokkichat.connection.TelegramBotSettings module

class bokkichat.connection.TelegramBotSettings.**TelegramBotSettings** (*api_key: str*)

Bases: *bokkichat.connection.Settings.Settings*

Class that defines a Settings object for a Telegram bot connection

__init__ (*api_key: str*)

Initializes the Telegram Connection. :param api_key: The API key used for authentication

classmethod deserialize (*serialized: str*)

Deserializes a string and generates a Settings object from it :param serialized: The serialized string :return: The deserialized Settings object

serialize () → str

Serializes the settings to a string :return: The serialized Settings object

bokkichat.connection.TelegramConnection module

class bokkichat.connection.TelegramConnection.**TelegramConnection** (*settings: bok-*
kichat.connection.TelegramSettings.Te

Bases: *bokkichat.connection.Connection.Connection*

Class that implements a Telegram connection using telethon

__init__ (*settings: bokkichat.connection.TelegramSettings.TelegramSettings*)
Initializes the connection, with credentials provided by a Settings object. :param settings: The settings for the connection

address
A connection must be able to specify its own address :return: The address of the connection

close ()
Disconnects the Connection. :return: None

receive () → List[bokkichat.message.Message.Message]
Receives all pending messages. :return: A list of pending Message objects

send (*message: bokkichat.message.Message.Message*)
Sends a message. A message may be either a TextMessage or a MediaMessage. :param message: The message to send :return: None

bokkichat.connection.TelegramSettings module

```
class bokkichat.connection.TelegramSettings.TelegramSettings (api_id: str,  
api_hash: str,  
session_name: str  
= 'bokkichat')
```

Bases: *bokkichat.connection.Settings.Settings*

Class that defines a Settings object for a Telegram connection

__init__ (*api_id: str, api_hash: str, session_name: str = 'bokkichat')*
Initializes the Telegram Connection. :param api_id: The API ID :param api_hash: The API hash :param session_name: The name of the telethon session

classmethod deserialize (*serialized: str*)
Deserializes a string and generates a Settings object from it :param serialized: The serialized string :return: The deserialized Settings object

serialize () → str
Serializes the settings to a string :return: The serialized Settings object

Module contents

1.1.3 bokkichat.message package

Submodules

bokkichat.message.MediaMessage module

```
class bokkichat.message.MediaMessage.MediaMessage (sender: bok-  
kichat.address.Address.Address,  
receiver: bok-  
kichat.address.Address.Address,  
media_type: bok-  
kichat.message.MediaType.MediaType,  
data: bytes, caption: str)
```

Bases: *bokkichat.message.Message.Message*

Class that defines an interface for media messages. Each media message has a media type, data and caption.

```
__init__(sender: bokkichat.address.Address.Address, receiver: bokkichat.address.Address.Address,
         media_type: bokkichat.message.MediaType.MediaType, data: bytes, caption: str)
```

Initializes the TextMessage object :param sender: The sender of the message :param receiver: The receiver of the message :param media_type: The type of the contained media :param data: The data of the attached media :param caption: The caption attached to the media

bokkichat.message.MediaType module

```
class bokkichat.message.MediaType.MediaType
```

Bases: `enum.Enum`

Enum that specifies the various types of media that can be sent using MediaMessage objects.

```
AUDIO = 2
```

```
IMAGE = 3
```

```
VIDEO = 1
```

bokkichat.message.Message module

```
class bokkichat.message.Message.Message (sender: bokkichat.address.Address.Address, receiver: bokkichat.address.Address.Address)
```

Bases: `object`

Class that defines common attributes for a Message object.

```
__init__(sender: bokkichat.address.Address.Address, receiver: bokkichat.address.Address.Address)
```

Initializes a Message object. :param sender: The sender of the message :param receiver: The receiver of the message

bokkichat.message.TextMessage module

```
class bokkichat.message.TextMessage.TextMessage (sender: bokkichat.address.Address.Address, receiver: bokkichat.address.Address.Address,
         body: str, title: str = "")
```

Bases: `bokkichat.message.Message.Message`

Class that defines an interface for text messages. Each text message has a title and a body. Some chat services don't allow titles for messages, in those cases, the title will be blank.

```
__init__(sender: bokkichat.address.Address.Address, receiver: bokkichat.address.Address.Address,
         body: str, title: str = "")
```

Initializes the TextMessage object :param sender: The sender of the message :param receiver: The receiver of the message :param body: The message body :param title: The title of the message. Defaults to an empty string

Module contents

1.1.4 bokkichat.test package

Submodules

bokkichat.test.TestTelegram module

class bokkichat.test.TestTelegram.**TestTelegram** (*methodName='runTest'*)

Bases: unittest.case.TestCase

Class that tests telegram connections, both the telethon and python-telegram-bot implementations. By using both, they can test each other.

setUp ()

Initializes the connections :return: None

tearDown ()

Closes the connections :return: None

test_sending_text_messages ()

Tests sending text messages from one connection to another :return: None

Module contents

1.2 Submodules

1.3 bokkichat.exceptions module

exception bokkichat.exceptions.**InvalidMessageData** (*message_data: Dict[str, Any]*)

Bases: Exception

Exception that indicates message data that's invalid or otherwise could not be correctly parsed.

__init__ (*message_data: Dict[str, Any]*)

Initializes the Exception :param message_data: The message data that caused the exception to be raised

1.4 Module contents

CHAPTER
TWO

BOKKICHAT

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

b

- bokkichat, 8
- bokkichat.address, 3
- bokkichat.address.Address, 3
- bokkichat.connection, 6
- bokkichat.connection.CliConnection, 3
- bokkichat.connection.CliSettings, 4
- bokkichat.connection.Connection, 4
- bokkichat.connection.Settings, 4
- bokkichat.connection.TelegramBotConnection,
5
- bokkichat.connection.TelegramBotSettings,
5
- bokkichat.connection.TelegramConnection,
5
- bokkichat.connection.TelegramSettings,
6
- bokkichat.exceptions, 8
- bokkichat.message, 8
- bokkichat.message.MediaMessage, 6
- bokkichat.message.MediaType, 7
- bokkichat.message.Message, 7
- bokkichat.message.TextMessage, 7
- bokkichat.test, 8
- bokkichat.test.TestTelegram, 8

INDEX

Symbols

`__init__()` (*bokkichat.address.Address.Address* method), 3
`__init__()` (*bokkichat.connection.Connection.Connection* method), 4
`__init__()` (*bokkichat.connection.TelegramBotConnection.TelegramBotConnection* method), 5
`__init__()` (*bokkichat.connection.TelegramBotSettings.TelegramBotSettings* method), 5
`__init__()` (*bokkichat.connection.TelegramConnection.TelegramConnection* method), 5
`__init__()` (*bokkichat.connection.TelegramSettings.TelegramSettings* method), 6
`__init__()` (*bokkichat.exceptions.InvalidMessageData* method), 8
`__init__()` (*bokkichat.message.MediaMessage.MediaMessage* method), 6
`__init__()` (*bokkichat.message.Message.Message* method), 7
`__init__()` (*bokkichat.message.TextMessage.TextMessage* method), 7
bokkichat.connection.CliSettings (module), 4
bokkichat.connection.Connection (module), 4
bokkichat.connection.Settings (module), 4
bokkichat.connection.TelegramBotConnection (module), 5
bokkichat.connection.TelegramBotSettings (module), 5
bokkichat.connection.TelegramConnection (module), 5
bokkichat.connection.TelegramSettings (module), 6
bokkichat.exceptions (module), 8
bokkichat.message (module), 8
bokkichat.message.MediaMessage (module), 6
bokkichat.message.MediaType (module), 7
bokkichat.message.Message (module), 7
bokkichat.message.TextMessage (module), 7
bokkichat.test (module), 8
bokkichat.test.TestTelegram (module), 8

A

address (*bokkichat.connection.CliConnection.CliConnection* attribute), 3
address (*bokkichat.connection.Connection.Connection* attribute), 4
address (*bokkichat.connection.TelegramBotConnection.TelegramBotConnection* attribute), 5
address (*bokkichat.connection.TelegramConnection.TelegramConnection* attribute), 6
Address (class in *bokkichat.address.Address*), 3
AUDIO (*bokkichat.message.MediaType.MediaType* attribute), 7

B

bokkichat (module), 8
bokkichat.address (module), 3
bokkichat.address.Address (module), 3
bokkichat.connection (module), 6
bokkichat.connection.CliConnection (module), 3

C

CliConnection (class in *bokkichat.connection.CliConnection*), 3
CliSettings (class in *bokkichat.connection.CliSettings*), 4
close() (*bokkichat.connection.CliConnection.CliConnection* method), 3
close() (*bokkichat.connection.Connection.Connection* method), 4
close() (*bokkichat.connection.TelegramBotConnection.TelegramBotConnection* method), 5
close() (*bokkichat.connection.TelegramConnection.TelegramConnection* method), 6
Connection (class in *bokkichat.connection.Connection*), 4

D

deserialize() (*bokkichat.connection.CliSettings.CliSettings* class method), 4

deserialize() (*bokkichat.connection.Settings.Settings* class method), 4

deserialize() (*bokkichat.connection.TelegramBotSettings.TelegramBotSettings* class method), 5

deserialize() (*bokkichat.connection.TelegramSettings.TelegramSettings* class method), 6

I

IMAGE (*bokkichat.message.MediaType.MediaType* attribute), 7

InvalidMessageData, 8

L

loop() (*bokkichat.connection.Connection.Connection* method), 4

M

MediaMessage (class in *bokkichat.message.MediaMessage*), 6

MediaType (class in *bokkichat.message.MediaType*), 7

Message (class in *bokkichat.message.Message*), 7

R

receive() (*bokkichat.connection.CliConnection.CliConnection* method), 3

receive() (*bokkichat.connection.Connection.Connection* method), 4

receive() (*bokkichat.connection.TelegramBotConnection.TelegramBotConnection* method), 5

receive() (*bokkichat.connection.TelegramConnection.TelegramConnection* method), 6

S

send() (*bokkichat.connection.CliConnection.CliConnection* method), 3

send() (*bokkichat.connection.Connection.Connection* method), 4

send() (*bokkichat.connection.TelegramBotConnection.TelegramBotConnection* method), 5

send() (*bokkichat.connection.TelegramConnection.TelegramConnection* method), 6

serialize() (*bokkichat.connection.CliSettings.CliSettings* method), 4

serialize() (*bokkichat.connection.Settings.Settings* method), 5

serialize() (*bokkichat.connection.TelegramBotSettings.TelegramBotSettings* method), 5

serialize() (*bokkichat.connection.TelegramSettings.TelegramSettings* method), 6

Settings (class in *bokkichat.connection.Settings*), 4

setUp() (*bokkichat.test.TestTelegram.TestTelegram* class method), 8

T

tearDown() (*bokkichat.test.TestTelegram.TestTelegram* class method), 8

TelegramBotConnection (class in *bokkichat.connection.TelegramBotConnection*), 5

TelegramBotSettings (class in *bokkichat.connection.TelegramBotSettings*), 5

TelegramConnection (class in *bokkichat.connection.TelegramConnection*), 5

TelegramSettings (class in *bokkichat.connection.TelegramSettings*), 6

test_sending_text_messages() (*bokkichat.test.TestTelegram.TestTelegram* method), 8

TestTelegram (class in *bokkichat.test.TestTelegram*), 8

TextMessage (class in *bokkichat.message.TextMessage*), 7

V

VIDEO (*bokkichat.message.MediaType.MediaType* attribute), 7