

Hachat Documentation

{rohrer, dittler, wedel}@informatik.hu-berlin.de

January 27, 2013

Contents

Contents	1
1 Module gui	3
1.1 Variables	3
1.2 Class gui	3
1.2.1 Methods	3
1.2.2 Properties	4
2 Module hachat	5
2.1 Variables	5
3 Module host	6
3.1 Variables	6
3.2 Class Host	6
3.2.1 Methods	6
4 Module message	7
4.1 Functions	7
4.2 Variables	7
4.3 Class ByeMessage	7
4.3.1 Methods	7
4.3.2 Properties	8
4.4 Class DeadMessage	8
4.4.1 Methods	8
4.4.2 Properties	8
4.5 Class HeloMessage	9
4.5.1 Methods	9
4.5.2 Properties	9
4.6 Class History	9
4.6.1 Methods	10
4.7 Class HistoryExchangeMessage	10
4.7.1 Methods	11
4.7.2 Properties	11
4.8 Class HostExchangeMessage	11
4.8.1 Methods	12
4.8.2 Properties	12
4.9 Class Message	12

4.9.1	Methods	12
4.9.2	Properties	13
4.10	Class MessageException	13
4.10.1	Methods	13
4.10.2	Properties	14
4.11	Class TextMessage	14
4.11.1	Methods	14
4.11.2	Properties	15
5	Module peer	16
5.1	Variables	16
5.2	Class Peer	16
5.2.1	Methods	16
Index		18


1 Module gui

this is the module for the GUI

1.1 Variables

Name	Description
CODEC	Value: 'utf8'
--package--	Value: None

1.2 Class gui

object  **gui.gui**

GUI Class for Hachat User Interface

1.2.1 Methods

__init__(*self*, *parent*)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

check_queue(*self*)

resieve a txt-msg and push it in the gui uses a Queue which is checked every 50ms

ende(*self*)

stops hachat and gui

ende_*(self, event)*

event: calls ende() to stop gui and hachat

run(*self*)

start gui

senden(*self*)

send a txt-msg from the gui

senden _(<i>self</i> , <i>event</i>)

event: calls <code>senden()</code> to send a txt-msg from the gui

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

1.2.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

2 Module hachat

This is the basic hachat module which starts a Peer with the given options from command line

2.1 Variables

Name	Description
<code>__package__</code>	Value: None
<code>args</code>	Value: Namespace(Test=False, ip=None, link=None, name=None, port...
<code>name</code>	Value: 'user43075'
<code>nr</code>	Value: 43075
<code>parser</code>	Value: ArgumentParser(prog='(imported)', usage=None, description...

3 Module host

This module provides the Host Class which represents connections to other Peers in a Hachat-Peer

3.1 Variables

Name	Description
<code>__package__</code>	Value: None

3.2 Class Host

Class representing a connection to another peer

3.2.1 Methods

<code>__del__(self)</code>
<code>__init__(self, peer, hostIP, hostPort)</code>
<code>addToMsgQueue(self, msg)</code> check if message is type Message and add to Queue
<code>constructKey(cls, hostIP, hostPort)</code> Class method: construct key to identify hosts in hostlist
<code>sendHello(self)</code> will send a HELO-Message to the corresponding peer

4 Module message

module which provides all message types for hachat

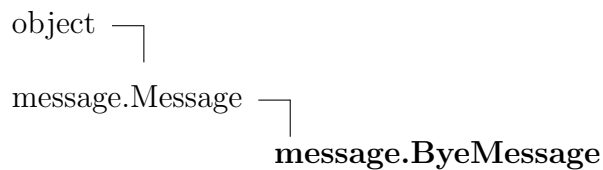
4.1 Functions

toMessage (<i>string</i>)
construct Message type from string

4.2 Variables

Name	Description
<code>__package__</code>	Value: None

4.3 Class ByeMessage



Message Type to be send when leaving

4.3.1 Methods

__init__ (<i>self</i> , <i>origin</i> , <i>uid</i> =None)
build Message with supplied uid or otherwise get a random uid
Overrides: object.__init__ exitit(inherited documentation)

__str__ (<i>self</i>)
implements interface
Overrides: object.__str__

Inherited from object

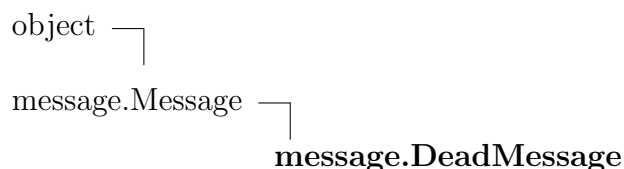
`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,

`__repr__()`, `__setattr__()`, `__sizeof__()`, `__subclasshook__()`

4.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

4.4 Class *DeadMessage*



Message that tells a Peer that another Peer is dead

4.4.1 Methods

`__init__`(*self*, *origin*, *peer*, *uid*=None)

build Message with supplied uid or otherwise get a random uid

Overrides: `object.__init__` extit(inherited documentation)

`__str__`(*self*)

implements interface

Overrides: `object.__str__`

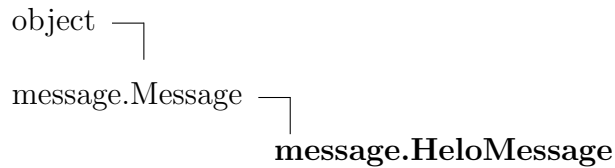
Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__subclasshook__()`

4.4.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

4.5 Class HeloMessage



regularly sent HELO Message which exchange information on IP and Port Message layout:
 | type | uid | recipientIP | recipientPort | senderIP | senderPort |

4.5.1 Methods

__init__(*self*, *recipientIP*, *recipientPort*, *senderIP*, *senderPort*, *uid=None*)
 build Message with supplied uid or otherwise get a random uid
 Overrides: object.__init__ extit(inherited documentation)

__str__(*self*)
 implements interface
 Overrides: object.__str__

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__subclasshook__()`

4.5.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

4.6 Class History

Klasse History speichert und ueberprueft Text-Msgs

4.6.1 Methods

__init__ (<i>self</i> , <i>msgLimit</i> , <i>hashLimit</i>)
--

addMsg (<i>self</i> , <i>msg</i>)
--

add TXTMessage to History

getListMsgObjects (<i>self</i> , <i>msgQuant</i>)
--

get a list of x Messages

getMsgHashes (<i>self</i> , <i>msgQuant</i>)

get a list of hashes from x Messages

getMsgObjects (<i>self</i> , <i>msgHash</i>)

get Message object from hash

msgExists (<i>self</i> , <i>msghash</i>)

Message is already in History (test on hash)
--

msgSafed (<i>self</i> , <i>msg</i>)
--

Message is already in History (test on TMsage object)

removeMsg (<i>self</i> , <i>msgQuant</i> =0, <i>hashQuant</i> =0)

remove x Messages from History

4.7 Class HistoryExchangeMessage

object └─

message.Message └─

message.HistoryExchangeMessage

for request and pushing Hosts Message layout: | type | uid | recipientIP | recipientPort |
origin key | level | quant | liste |

4.7.1 Methods

```
__init__(self, recipientIP, recipientPort, origin, level, quant=None, liste=None, uid=None)
```

build Message with supplied uid or otherwise get a random uid

Overrides: object.__init__ extit(inherited documentation)

```
__str__(self)
```

implements interface

Overrides: object.__str__

Inherited from object

```
__delattr__(), __format__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
__repr__(), __setattr__(), __sizeof__(), __subclasshook__()
```

4.7.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

4.8 Class HostExchangeMessage



for request and pushing Hosts Message layout: | type | uid | recipientIP | recipientPort |
origin key | level | quant | listofHosts |

4.8.1 Methods

```
__init__(self, recipientIP, recipientPort, origin, level, quant=None,
listofHosts=None, uid=None)
```

build Message with supplied uid or otherwise get a random uid

Overrides: object.__init__ extit(inherited documentation)

```
__str__(self)
```

implements interface

Overrides: object.__str__

Inherited from object

```
__delattr__(), __format__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
__repr__(), __setattr__(), __sizeof__(), __subclasshook__()
```

4.8.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

4.9 Class Message

```
object ┌
      │ message.Message
```

Known Subclasses: message.ByeMessage, message.DeadMessage, message.HeloMessage, message.HistoryExchangeMessage, message.HostExchangeMessage, message.TextMessage

abstract class all other message types will inherit from

4.9.1 Methods

```
__init__(self, uid)
```

build Message with supplied uid or otherwise get a random uid

Overrides: object.__init__

<code>--str--(self)</code>

cast Message to string

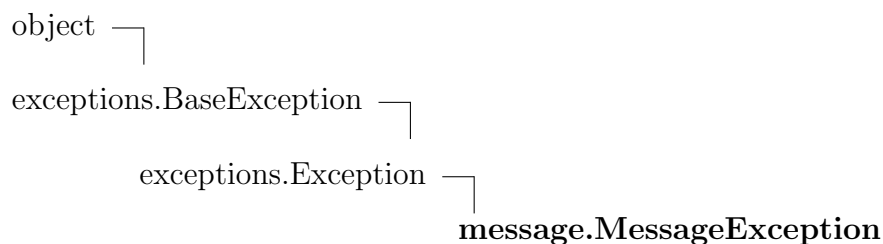
Overrides: <code>object.__str__</code>
--

Inherited from object

`--delattr--()`, `--format--()`, `--getattr--()`, `--hash--()`, `--new--()`, `--reduce--()`, `--reduce_ex--()`,
`--repr--()`, `--setattr--()`, `--sizeof--()`, `--subclasshook--()`

4.9.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

4.10 Class `MessageException`

Custom Exception Type for Messages

4.10.1 Methods

<code>--init--(self, value)</code>

<code>x.__init__(...)</code> initializes x; see <code>help(type(x))</code> for signature
--

Overrides: <code>object.__init__</code> <code>exitit</code> (inherited documentation)

<code>--str--(self)</code>

<code>str(x)</code>

Overrides: <code>object.__str__</code> <code>exitit</code> (inherited documentation)
--

Inherited from exceptions.Exception

`__new__()`

Inherited from `exceptions.BaseException`

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__unicode__()`

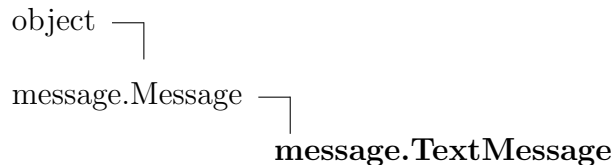
Inherited from `object`

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

4.10.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

4.11 Class `TextMessage`



normal Text Messages Message layout: | type | uid | hash | sender name | origin key | lastHop
key | text |

4.11.1 Methods

`__init__(self, name, origin, lastHop, text, uid=None)`

build Message with supplied uid or otherwise get a random uid

Overrides: `object.__init__` extit(inherited documentation)

`__str__(self)`

implements interface

Overrides: `object.__str__`

Inherited from `object`

`--delattr--()`, `--format--()`, `--getattr--()`, `--hash--()`, `--new--()`, `--reduce--()`, `--reduce_ex--()`,
`--repr--()`, `--setattr--()`, `--sizeof--()`, `--subclasshook--()`

4.11.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

5 Module peer

central module which defines the behaviour of a Hachat Peer

5.1 Variables

Name	Description
<code>--package--</code>	Value: None

5.2 Class Peer

Peer Klasse

5.2.1 Methods

HistoryControl(*self*, *neighbour*, *historyList*)

checks if the historyList from neighbour contains msgs wich are not in own History. If so, these lostMsg Hahses will be pushed back and by this the associated msgObjects are requested.

--del--(*self*)

--init--(*self*, *firstHost*=None, *port*=None, *name*='temp', *ip*=None, *testmode*=False)

addToHosts(*self*, *addr*)

check if already in hostlist otherwise add

forwardMsg(*self*, *msg*, *Oneneighbour*=None)

forwarding TextMessage, but not to initial sender if host is set, it will only forward to this single host

generateMsgParts(*self*, *quant*=5, *length*=2000)

generates random TextMsgs, if length > 1000 there will be more then one msg-part

getHistory(*self*, *neighbour*, *initial=False*)

request History from neighbour; initial is true for a initial history exchange:
this will skip pushing Hashes and immediately request msg.objects

maintenanceLoop(*self*)

thread which runs regularly maintenance tasks

processMessage(*self*, *msg*, *fromAddr*)

processes the received messages

pushHistory(*self*, *neighbour*, *quant*)

push own List of History-Hashes to neighbour

pushHosts(*self*, *neighbour*, *quant*)

give Hosts from hostExchange to a neighbour

pushMsgObjects(*self*, *neighbour*, *lostMsgHashes=None*)

pushes requested msgObjects back to neighbour

requestHosts(*self*, *neighbour*, *quant=None*)

request Hosts from neighbour

sendAll(*self*, *msg*)

send Message Object to all your Peers

sendLoop(*self*, *test=False*)

send Message objects of all hosts from Queue as string

sendText(*self*, *text*)

make TXTMessage out of text and send to all hosts

startRecvLoop(*self*)

general receive loop of a peer

Index

- gui (*module*), 3–4
 - gui.gui (*class*), 3–4
 - gui.gui.check_queue (*method*), 3
 - gui.gui.ende (*method*), 3
 - gui.gui.ende_ (*method*), 3
 - gui.gui.run (*method*), 3
 - gui.gui.senden (*method*), 3
 - gui.gui.senden_ (*method*), 3
- hachat (*module*), 5
- host (*module*), 6
 - host.Host (*class*), 6
 - host.Host.__del__ (*method*), 6
 - host.Host.__init__ (*method*), 6
 - host.Host.addToMsgQueue (*method*), 6
 - host.Host.constructKey (*class method*), 6
 - host.Host.sendHello (*method*), 6
- message (*module*), 7–15
 - message.ByeMessage (*class*), 7–8
 - message.DeadMessage (*class*), 8
 - message.HeloMessage (*class*), 8–9
 - message.History (*class*), 9–10
 - message.History.__init__ (*method*), 10
 - message.History.addMsg (*method*), 10
 - message.History.getListMsgObjects (*method*), 10
 - message.History.getMsgHashes (*method*), 10
 - message.History.getMsgObjects (*method*), 10
 - message.History.msgExists (*method*), 10
 - message.History.msgSafed (*method*), 10
 - message.History.removeMsg (*method*), 10
 - message.HistoryExchangeMessage (*class*), 10–11
 - message.HostExchangeMessage (*class*), 11–12
 - message.Message (*class*), 12–13
 - message.MessageException (*class*), 13–14
 - message.TextMessage (*class*), 14–15
 - message.toMessage (*function*), 7
- peer (*module*), 16–17
 - peer.Peer (*class*), 16–17
 - peer.Peer.__del__ (*method*), 16
 - peer.Peer.__init__ (*method*), 16
 - peer.Peer.addToHosts (*method*), 16
 - peer.Peer.forwardMsg (*method*), 16
 - peer.Peer.generateMsgParts (*method*), 16
 - peer.Peer.getHistory (*method*), 16
 - peer.Peer.HistoryControl (*method*), 16
 - peer.Peer.maintenanceLoop (*method*), 17
 - peer.Peer.processMessage (*method*), 17
 - peer.Peer.pushHistory (*method*), 17
 - peer.Peer.pushHosts (*method*), 17
 - peer.Peer.pushMsgObjects (*method*), 17
 - peer.Peer.requestHosts (*method*), 17
 - peer.Peer.sendAll (*method*), 17
 - peer.Peer.sendLoop (*method*), 17
 - peer.Peer.sendText (*method*), 17
 - peer.Peer.startRecvLoop (*method*), 17