API Documentation

API Documentation

June 14, 2009

Contents

1 Package egoist

Author: Vassilis Lekakis

Organization: Institute of Computer Science, F.O.R.T.H

Contact: lekakis@ics.forth.gr/lekakis@gmail.com/lex@umd.edu

See Also: *http://csr.bu.edu/sns/*, *http://www.ics.forth.gr/˜lekakis*

1.1 Modules

2   Package egoist.constants

Author: Vassilis Lekakis

Organization: Institute of Computer Science, F.O.R.T.H

Contact: lekakis@ics.forth.gr/lekakis@gmail.com/lex@umd.edu

See Also: *http://csr.bu.edu/sns/* , *http://www.ics.forth.gr/˜lekakis*

2.1   Modules

- **ntwrsemantics** *(Section 3, p. 4)*

3 Module egoist.constants.ntwrsemantics

Author: Vassilis Lekakis

Organization: Institute of Computer Science, F.O.R.T.H

Contact: lekakis@ics.forth.gr/lekakis@gmail.com/lex@umd.edu

See Also: *http://csr.bu.edu/sns/*, *http://www.ics.forth.gr/˜lekakis*

3.1 Variables

Name	Description
REGISTER	**Value:** `'REG'`
UPDATE	**Value:** `'UPT'`
REMOVE	**Value:** `'RMV'`
TEARDOWN	**Value:** `'TRD'`
PORT_DELIMITER	**Value:** `':'`
PACK_DELIMITER	**Value:** `'@'`
UPDT_DELIMITER	**Value:** `'**'`
NEIGHBOR_DELIMITER	**Value:** `'|'`
OVERLAY_DELIMITER	**Value:** `'#'`
NACK	**Value:** `'NACK'`
REMOVE_PERIOD	**Value:** 50
PACKET_TYPE	**Value:** 0
BOOTSTRAP_PORT	**Value:** 61223
BOOT_DELAY	**Value:** 500
HUGE_DELAY	**Value:** 2147482647
NBR_IP	**Value:** `'NEIGHBOR_IP'`
NBR_PORT	**Value:** `'NEIGHBOR_PORT'`
NBR_FACTORY	**Value:** `'NEIGHBOR_FACTORY'`
NBR_CONNECTOR	**Value:** `'NEIGHBOR_CONNECTOR'`
PKT_IDS_LIMIT	**Value:** 5000
UPDATE_HASH	**Value:** 1
UPDATE_ID	**Value:** 2
UPDATE_SRC	**Value:** 3
MIL	**Value:** 1000
PING_PERIOD	**Value:** 35
REWIRING_PERIOD	**Value:** 60
ERASE_HASH	**Value:** 300
PING_TIMEOUT	**Value:** 3
PING_RESULTS	**Value:** `'PING_MEASURMENTS'`
PING_FACTORY	**Value:** `'PING_FACTORY'`
SEG	**Value:** `'SEG'`
OVERLAY_PACKET	**Value:** `'OVER'`
NET_PACKET	**Value:** `'NET'`
PKT_LIMIT	**Value:** 8190

4 Package egoist.msgutils

Author: Vassilis Lekakis

Organization: Institute of Computer Science, F.O.R.T.H

Contact: lekakis@ics.forth.gr/lekakis@gmail.com/lex@umd.edu

See Also: *http://csr.bu.edu/sns/*, *http://www.ics.forth.gr/~lekakis*

4.1 Modules

- **msgprocess** *(Section 5, p. 6)*

5   Module egoist.msgutils.msgprocess

Author: Vassilis Lekakis

Organization: Institute of Computer Science, F.O.R.T.H

Contact: lekakis@ics.forth.gr/lekakis@gmail.com/lex@umd.edu

See Also: *http://csr.bu.edu/sns/*, *http://www.ics.forth.gr/˜lekakis*

5.1   Functions

---

set2str(*overlay, MSG_TYPE*)

converts the overlay set to a string

Parameters

    `overlay:` the set that holds the alive nodes of the egoist network
            (type=python set)

Return Value

    The string representation of the set argument
    (type=String)

---

dict2str(*network, MSG_TYPE*)

converts a dictionary to a string

Parameters

    `network:` the graph representation of the egoist overlay
            (type=python dictionary)

Return Value

    the string representation of the dictionary argument
    (type=String)

---

str2set(*netstr, MSG_TYPE*)

converts a string to a set

Parameters

    `netstr:` incoming data from network
            (type=String)

Return Value

    a set constructed from the contents of the input string
    (type=python set)

---

str2dict(*netstr, MSG_TYPE*)

converts a string to a dictionary

Parameters
> **netstr:** incoming data from network
>> *(type=String)*

Return Value
> a dictionary constructed from the contents of the input string
> *(type=python dictionary)*

---

segmentBootPkt(*over, net*)

segmentation method. Packets that exceed the maximum amount of data that can be transported from Twisted (see documentation) are segmented to more packets with smaller sizes

Parameters
> **over:** the nodes inside the egoist overlay
>> *(type=python set)*
>
> **net:** the egoist overlay graph
>> *(type=python dictionary)*

Return Value
> the packets that have been produced after the segmentation
> *(type=python list)*

---

pktBuildup(*pkt_list*)

when segmented packets are received this method builds up the initial packet

Parameters
> **pkt_list:** the segmented packets
>> *(type=python list)*

Return Value
> the actual data as extracted from the segmented packets
> *(type=python tuple)*

5.2 Variables

Name	Description
BOOTSTRAP_PORT	**Value:** 61223
BOOT_DELAY	**Value:** 500
ERASE_HASH	**Value:** 300
HUGE_DELAY	**Value:** 2147482647
MIL	**Value:** 1000
NACK	**Value:** 'NACK'
NBR_CONNECTOR	**Value:** 'NEIGHBOR_CONNECTOR'
NBR_FACTORY	**Value:** 'NEIGHBOR_FACTORY'
NBR_IP	**Value:** 'NEIGHBOR_IP'
NBR_PORT	**Value:** 'NEIGHBOR_PORT'
NEIGHBOR_DELIMITER	**Value:** '|'

Name	Description
NET_PACKET	**Value:** `'NET'`
OVERLAY_DELIMITER	**Value:** `'#'`
OVERLAY_PACKET	**Value:** `'OVER'`
PACKET_TYPE	**Value:** 0
PACK_DELIMITER	**Value:** `'@'`
PING_FACTORY	**Value:** `'PING_FACTORY'`
PING_PERIOD	**Value:** 35
PING_RESULTS	**Value:** `'PING_MEASURMENTS'`
PING_TIMEOUT	**Value:** 3
PKT_IDS_LIMIT	**Value:** 5000
PKT_LIMIT	**Value:** 8190
PORT_DELIMITER	**Value:** `':'`
REGISTER	**Value:** `'REG'`
REMOVE	**Value:** `'RMV'`
REMOVE_PERIOD	**Value:** 50
REWIRING_PERIOD	**Value:** 60
SEG	**Value:** `'SEG'`
TEARDOWN	**Value:** `'TRD'`
UPDATE	**Value:** `'UPT'`
UPDATE_HASH	**Value:** 1
UPDATE_ID	**Value:** 2
UPDATE_SRC	**Value:** 3
UPDT_DELIMITER	**Value:** `'**'`
maxint	**Value:** 2147483647

6   Package egoist.routing

Author: Vassilis Lekakis

Organization: Institute of Computer Science, F.O.R.T.H

Contact: lekakis@ics.forth.gr/lekakis@gmail.com/lex@umd.edu

See Also: *http://csr.bu.edu/sns/* , *http://www.ics.forth.gr/~lekakis*

6.1   Modules

- **dijkstra** *(Section 7, p. 10)*
- **priorityDictionary** *(Section 8, p. 11)*

7   Module egoist.routing.dijkstra

Author: Vassilis Lekakis

Organization: Institute of Computer Science, F.O.R.T.H

Contact: lekakis@ics.forth.gr/lekakis@gmail.com/lex@umd.edu

See Also: *http://csr.bu.edu/sns/*, *http://www.ics.forth.gr/˜lekakis*

7.1   Functions

Dijkstra(*G*, *start*, *end*=None)
The classic algorithm with the use of a pririotity dictionary

8 Module egoist.routing.priorityDictionary

Author: David Eppstein

Organization: UC Irvine

See Also: http://code.activestate.com/recipes/522995/

8.1 Class PriorityDictionary

object ⌐

     dict ⌐

         egoist.routing.priorityDictionary.PriorityDictionary
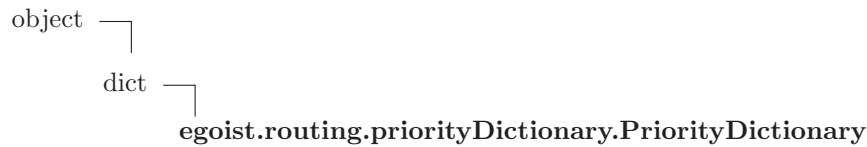
8.1.1 Methods

__init__(*self*)

Initialize priorityDictionary by creating binary heap of pairs (value,key). Note that changing or removing a dict entry will not remove the old pair from the heap until it is found by smallest() or until the heap is rebuilt.

Return Value
    ```new empty dictionary```

Overrides: dict.__init__

---

smallest(*self*)

Find smallest item after removing deleted items from heap.

__iter__(*self*)

Create destructive sorted iterator of priorityDictionary.

Overrides: dict.__iter__

__setitem__(*self*, *key*, *val*)

Change value stored in dictionary and add corresponding pair to heap. Rebuilds the heap if the number of deleted items grows too large, to avoid memory leakage.

Overrides: dict.__setitem__

---

setdefault(*self*, *key*, *val*)

Reimplement setdefault to call our customized __setitem__.

Return Value
    ```D.get(k,d), also set D[k]=d if k not in D```

Overrides: dict.setdefault

---

update(*self, other*)

Update D from E and F: for k in E: D[k] = E[k] (if E has keys else: for (k, v) in E: D[k] = v) then: for k in F: D[k] = F[k]

Return Value
> None

Overrides: dict.update extit(inherited documentation)

__cmp__(*x, y*)

cmp(x,y)

__contains__(*D, k*)
Return Value
> True if D has a key k, else False

__delattr__(*...*)

x.__delattr__('name') <==> del x.name

__delitem__(*x, y*)

del x[y]

__eq__(*x, y*)

x==y

__ge__(*x, y*)

x>=y

__getattribute__(*...*)

x.__getattribute__('name') <==> x.name

Overrides: object.__getattribute__

__getitem__(*x, y*)

x[y]

__gt__(*x, y*)

x>y

__hash__(*x*)

hash(x)

Overrides: object.__hash__

__le__(*x*, *y*)

x<=y

__len__(*x*)

len(x)

__lt__(*x*, *y*)

x<y

__ne__(*x*, *y*)

x!=y

__new__(*T*, *S*, ...)
Return Value
    a new object with type S, a subtype of T

Overrides: object.__new__

__reduce__(...)

helper for pickle

__reduce_ex__(...)

helper for pickle

__repr__(*x*)

repr(x)

Overrides: object.__repr__

__setattr__(...)

x.__setattr__('name', value) <==> x.name = value

__str__(*x*)

str(x)

---

clear(*D*)

Remove all items from D.

Return Value
    None

---

copy(D)
Return Value
    `a shallow copy of D`

---

fromkeys($dict$, S, $v=\ldots$)

v defaults to None.

Return Value
    `New dict with keys from S and values equal to v`

---

get(D, k, $d=\ldots$)

d defaults to None.

Return Value
    `D[k] if k in D, else d`

---

has_key(D, k)
Return Value
    `True if D has a key k, else False`

---

items(D)
Return Value
    `list of D's (key, value) pairs, as 2-tuples`

---

iteritems(D)
Return Value
    `an iterator over the (key, value) items of D`

---

iterkeys(D)
Return Value
    `an iterator over the keys of D`

---

itervalues(D)
Return Value
    `an iterator over the values of D`

---

keys(D)
Return Value
    `list of D's keys`

---

pop(D, k, $d=\ldots$)

If key is not found, d is returned if given, otherwise KeyError is raised

Return Value
    `v, remove specified key and return the corresponding value`

---

popitem(*D*)

2-tuple; but raise KeyError if D is empty

Return Value
    `(k, v), remove and return some (key, value) pair as a`

---

values(*D*)
Return Value
    `list of D's values`

8.1.2   Properties

Name	Description
__class__	**Value:** `<attribute '__class__' of 'object' objects>`

Index