
persistentlist Documentation

Release 0.4

Carl Chenet <chaica@ohmytux.com.org>

November 16, 2016

1	Guide	3
1.1	How to install persistentlist	3
1.2	Use persistentlist	3
1.3	The persistentlist module	4
1.4	License	5
1.5	Authors	6
2	Indices and tables	7
	Python Module Index	9

Persistentlist is a simple persistent list carrying n objects, the last objects being appended at the end of the list.

When n objects are in the list, before appending the $n+1$ object at then end of the list, the first object is removed from the list.

You'll find below anything you need to install, configure and use the persistentlist library.

1.1 How to install persistentlist

1.1.1 From PyPI

```
$ pip3 install persistentlist
```

1.1.2 From sources

- You need at least Python 3.4.
- Untar the tarball and go to the source directory with the following commands:

```
$ tar zxvf persistentlist-0.4.tar.gz  
$ cd persistentlist
```

- Next, to install persistentlist on your computer, type the following command with the root user:

```
$ python3 setup.py install
```

1.2 Use persistentlist

How to use the persistentlist library. Start by importing the module:

```
>>> from persistentlist import PersistentList
```

Then initiate a persistentlist object, providing the path to the persistent file and the length of the list:

```
>>> cache = PersistentList('cache', 3)
```

Remember the goal of this object persistentlist is to store persistently N objects (here 3) and the appended n+1 object will remove the first object of the list:

```
>>> cache  
[]
```

You can use `append()`:

```
>>> cache.append(1)
>>> cache
[1]
>>> cache.append(2)
>>> cache
[1, 2]
>>> cache.append(3)
>>> cache
[1, 2, 3]
>>> cache.append(4)
>>> cache
[2, 3, 4]
```

And extend():

```
>>> cache.extend([5, 6])
>>> cache
[4, 5, 6]
```

Use your persistent list like a traditional list:

```
>>> mylist = cache
>>> mylist
[4, 5, 6]
```

Once you are finished using the persistentlist object, do not forget to close it (requirement of the standard library module `shelve` underneath):

```
>>> cache.close()
```

One last important point: the name of your persistent file on the disk will be the path indicated while initializing the object, with the suffix `".db"`, e.g if your path is `/tmp/cache`, the path on your disk will be `/tmp/cache.db`

1.3 The persistentlist module

The persistentlist module provides classes to use a PersistentList.

Example

```
>>> from persistentlist import PersistentList
>>> cache = PersistentList('cache', 3)
>>> cache.append(1)
>>> cache
[1]
>>> cache.append(2)
>>> cache
[1, 2]
>>> cache.append(3)
>>> cache
[1, 2, 3]
>>> cache.append(4)
>>> cache
[2, 3, 4]
```

class persistentlist.**PersistentList** (*dbpath*, *maxitems*)

The PersistentList class. Instantiate a PersistentList object.

Parameters

- **dbpath** – The path to the cache file. If the empty string is provided, defaults to ‘cache’
- **maxitems** – The maximum number of elements i your PersistentList. Next ones are dropped.

Returns A PersistentList object

Return type *PersistentList*

Example

```
>>> from persistentlist import PersistentList
>>> cache = PersistentList('cache', 3)
>>> cache
[]
>>> type(cache)
<class 'persistentlist.PersistentList'>
```

append (*item*)

Append a new element at the end of the PersistentList object. When the max size of the PersistentList is reached, the first element is dropped

Parameters **item** – An item to append at the end of the PersistentList element.

Example

```
>>> from persistentlist import PersistentList
>>> cache = PersistentList('cache', 3)
>>> cache
[]
>>> cache.append(1)
>>>
```

close ()

Close a PersistentList object.

Example

```
>>> from persistentlist import PersistentList
>>> cache = PersistentList('cache', 3)
>>> cache.close()
>>>
```

extend (*items*)

Extend a PersistentList object with a list of elements. When the max size of the PersistentList is reached, the first element is dropped

Parameters **items** – List of items to append at the end of the PersistentList object.

Example

```
>>> from persistentlist import PersistentList
>>> cache = PersistentList('cache', 3)
>>> cache.extend([1,2,3])
>>> cache
[1, 2, 3]
```

1.4 License

This software comes under the terms of the **GPLv3+**. See the LICENSE file for the complete text of the license.

1.5 Authors

- Carl Chenet <chaica@ohmytux.com>
- Florent Gallaire <fgallaire@gmail.com>

Indices and tables

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

p

`persistentlist`, 4

A

`append()` (`persistentlist.PersistentList` method), [5](#)

C

`close()` (`persistentlist.PersistentList` method), [5](#)

E

`extend()` (`persistentlist.PersistentList` method), [5](#)

P

`PersistentList` (class in `persistentlist`), [4](#)

`persistentlist` (module), [4](#)