
mongo-memoize Documentation

Release 0.0.4

Ikuya Yamada

November 09, 2014

| | | |
|----------|--------------------------------|-----------|
| 1 | Basic Usage | 3 |
| 2 | Customization | 5 |
| 3 | Using Capped Collection | 7 |
| 4 | Contents | 9 |
| 4.1 | API Reference | 9 |
| 5 | Indices and tables | 11 |

A Python decorator library for instantly caching function results in MongoDB.

Basic Usage

```
from mongo_memoize import memoize

@memoize()
def func():
    ...
```

Customization

You can specify custom *serializer* and *key_generator*. *serializer* is used to serialize function results in order to convert them into formats that can be stored in MongoDB. *key_generator* generates a cache key from the function arguments. `PickleSerializer` and `PickleMD5KeyGenerator` are used by default.

```
from mongo_memoize import memoize, NoopSerializer, PickleMD5KeyGenerator

@memoize(serializer=NoopSerializer, key_generator=PickleMD5KeyGenerator)
def func():
    ...
```

Using Capped Collection

Capped collection is a MongoDB feature which allows to limit the maximum size of the collection. By setting *capped=True*, a capped collection is created automatically.

```
from mongo_memoize import memoize

@memoize(capped=True, capped_size=100000000)
def func():
    ...
```

For further information, please refer the *API Reference*.

4.1 API Reference

```
mongo_memoize.memoize(db_name='mongo_memoize',      host='localhost',      port=27017,
                      collection_name=None,        prefix='memoize',     capped=False,
                      capped_size=100000000,       capped_max=None,     connection_options={},
                      key_generator=None, serializer=None)
```

A decorator that caches results of the function in MongoDB.

Usage:

```
>>> from mongo_memoize import memoize
>>> @memoize()
... def some_function():
...     pass
... 
```

Parameters

- **db_name** (*str*) – MongoDB database name.
- **host** (*str*) – MongoDB host name.
- **port** (*int*) – MongoDB port.
- **collection_name** (*str*) – MongoDB collection name. If not specified, the collection name is generated automatically using the prefix, the module name, and the function name.
- **prefix** (*str*) – Prefix of the MongoDB collection name. This argument is only valid when the `collection_name` argument is not specified.
- **capped** (*bool*) – Whether to use the capped collection.
- **capped_size** (*int*) – The maximum size of the capped collection in bytes.
- **capped_max** (*int*) – The maximum number of items in the capped collection.
- **connection_options** (*dict*) – Additional parameters for establishing MongoDB connection.
- **key_generator** – Key generator instance. `PickleMD5KeyGenerator` is used by default.
- **serializer** – Serializer instance. `PickleSerializer` is used by default.

```
class mongo_memoize.NoopSerializer
```

Serializer that does nothing.

Note: It is required that the result of the function can be stored directly in MongoDB when you use this serializer.

class `mongo_memoize.PickleSerializer` (*protocol=-1*)
Serializer using Pickle.

Parameters `protocol` (*int*) – Pickle protocol version.

class `mongo_memoize.PickleMD5KeyGenerator` (*protocol=-1*)
Cache key generator using Pickle and MD5.

Parameters `protocol` (*int*) – Pickle protocol version.

Indices and tables

- *genindex*
- *modindex*
- *search*

M

memoize() (in module mongo_memoize), 9

N

NoopSerializer (class in mongo_memoize), 9

P

PickleMD5KeyGenerator (class in mongo_memoize), 10

PickleSerializer (class in mongo_memoize), 10