memcache

An Long

Sep 18, 2022

CONTENTS

Experimental memcached client library for python. This project is in WIP status, please don't use it in production environment.

Key features:

- Based on memcached's new meta commands;
- Asyncio support;
- Type hints.

CHAPTER

ONE

INSTALLATION

```
$ pip install memcache
```

CHAPTER

TWO

API DOCUMENT

class memcache.**AsyncMemcache**(*addr: ~typing.Optional[~typing.Union[~typing.Tuple[str, int], ~typing.List[~typing.Tuple[str, int]]]] = None, *, pool_size: ~typing.Optional[int] = 23, pool_timeout: ~typing.Optional[int] = 1, load_func: ~typing.Callable[[~typing.Union[str, bytes], bytes, int], ~typing.Any] = <function load>, dump_func: ~typing.Callable[[~typing.Union[str, bytes], ~typing.Any], ~typing.Tuple[bytes, int]] = <function dump>, username: ~typing.Optional[str] = None, password: ~typing.Optional[str] = None*)

async delete(*key: Union[bytes, str]*) → None

async execute_meta_command(*command:* MetaCommand) → *MetaResult*

async flush_all() → None

async get(*key: Union[bytes, str]*) → Optional[Any]

async set(*key: Union[bytes, str]*, *value: Any*, *, *expire: Optional[int] = None*) → None

class memcache.**Memcache**(*addr: ~typing.Optional[~typing.Union[~typing.Tuple[str, int], ~typing.List[~typing.Tuple[str, int]]]] = None, *, pool_size: ~typing.Optional[int] = 23, pool_timeout: ~typing.Optional[int] = 1, load_func: ~typing.Callable[[~typing.Union[str, bytes], bytes, int], ~typing.Any] = <function load>, dump_func: ~typing.Callable[[~typing.Union[str, bytes], ~typing.Any], ~typing.Tuple[bytes, int]] = <function dump>, username: ~typing.Optional[str] = None, password: ~typing.Optional[str] = None*)

delete(*key: Union[bytes, str]*) → None

execute_meta_command(*command:* MetaCommand) → *MetaResult*

flush_all() → None

get(*key: Union[bytes, str]*) → Optional[Any]

set(*key: Union[bytes, str]*, *value: Any*, *, *expire: Optional[int] = None*) → None

exception memcache.**MemcacheError**

class memcache.**MetaCommand**(*cm: bytes*, *key: Union[bytes, str]*, *datalen: Union[int, NoneType] = None*, *flags: Union[List[bytes], NoneType] = None*, *value: Union[bytes, NoneType] = None*)

cm: bytes

datalen: Optional[int]

dump_header() → bytes

flags: List[bytes]

key: bytes

value: Optional[bytes]

class memcache.**MetaResult**(*rc: bytes*, *datalen: Union[int, NoneType]*, *flags: List[bytes]*, *value: Union[bytes, NoneType]*)

datalen: Optional[int]

flags: List[bytes]

static load_header(*line: bytes*) → *MetaResult*

rc: bytes

value: Optional[bytes]

CHAPTER

THREE

LICENSE

Memcache is distributed by a MIT license.

PYTHON MODULE INDEX

INDEX