Basilisk Documentation

Release 0.1

Bonnier Business Polska

October 30, 2015

Contents

1 Indices and tables 7

Python Module Index 9

Contents: Basilisk enables Pythonic use of Redis hashes, lists and sorted sets with simple class interface as well as provides an ORM Model-like class using Redis hash or Elasticsearch inside.

A simple example:

```
>>> import basilisk
>>>
>>> basilisk.Config.load(redis={'host': 'localhost',
>>>                             'port': 6379,
>>>                             'db': 0,
>>>                             'max_connections': 10},
>>>                      elastic={})
>>>
>>>
>>> class Item(basilisk.RedisModel):
>>>     id = basilisk.MapField(key=True)
>>>     name = basilisk.MapField()
>>>     content = basilisk.MapField()
>>>     attachments = basilisk.JsonMapField()
>>>
>>>     @classmethod
>>>     def select(cls):
>>>         redis_items = basilisk.RedisList('items')
>>>         variables = basilisk.RedisHash('items_variables')
>>>         last_modified = int(variables['last_modified'] or 0)
>>>
>>>         if (not len(redis_items) or
>>>             not last_modified or
>>>             last_modified + 30 < time.time()):
>>>             items = DownloadNewItemsFromDatabase(last_modified)
>>>             for item in items:
>>>                 redis_items.append(item.id)
>>>             variables['last_modified'] = int(time.time())
>>>             variables.save()
>>>         return list(redis_items)
>>>
>>> items = Item.select()
>>> for item in items:
>>>     print(Item.get(item).content)
```

class basilisk.**Config**

This class keeps appropriate config data.

It enables access as a class property and as an item (like in dict).

classmethod load(***namespaces*)

Initialize config from a list of params.

class basilisk.**MapField**(***kwargs*)

This is a base class for all NoSQL store fields. It supports data-based initialisation, default values and prepping values to serialization.

get_default()

It returns field's default value. Surprise surprise.

Returns field's default value.

get_name()

Returns field's name.

Returns this field's name.

is_primary()
Is the field model's primary key.

Returns boolean.

pythonize(*data*)
This function pythonizes data coming from NoSQL store.

Parameters **data** – data fetched from NoSQL store.

Returns data in Python format.

static serialize(*data*)
Let's prepare the data to writing it in NoSQL store. By default it returns data without any changes.

Parameters **data** – input data

Returns data ready to be written in NoSQL store.

set_name(*name*)
It sets field's name to make it self-conscious.

Parameters **name** – field name as in model.

Returns self

class basilisk.**JsonMapField**(***kwargs*)
This class enables keeping JSON as field value.

pythonize(*data*)
This function loads JSON data to Python objects.

Parameters **data** – data fetched from NoSQL store.

Returns data in Python format.

static serialize(*data*)
This function dumps data to JSON.

Parameters **data** – input data.

Returns data dumped to JSON.

class basilisk.**MapModelBase**(***kwargs*)
This is the base class for NoSQL store models.

This class enables reading object with given id, saving object and data (de)serialization.

Redis connection is available in connect property. Dict of fields is available in _fields property.

Reserved property names, apart from methods, are _fields, id_field and connect.

exception MapModelException
Exception raised when errors related to NoSQL store handling are encountered.

classmethod MapModelBase.**get**(*oid*)
This method gets a model instance with given id from NoSQL store.

Parameters **oid** – id of object to get.

Returns hydrated model instance.

classmethod MapModelBase.**get_fields**()
This function returns the dict of model fields.

Returns dict containing name: field pairs.

MapModelBase.**get_instance_key**()
This function returns a key in which the instance will live in NoSQL store.

Returns Redis key name containing instance hash

classmethod MapModelBase.**get_key**(*oid*)
This function creates a key in which NoSQL store will save the instance with given id.

Parameters **oid** – id of object for which a key should be created.

Returns NoSQL store key.

classmethod MapModelBase.**pythonize**(*data*, *loads=False*)
This method prepares the data fetched from NoSQL store for new instance.

Parameters **data** – values to convert.

Returns dict of values ready to pass to __init__.

MapModelBase.**save**(*create_id=True*)
Let's save instance's current state to NoSQL store.

Parameters **create_id** – whether id should be created automatically if it's not set yet.

Returns self

MapModelBase.**serialize**(*dump=False*)
We try to call serialize for each field, if it doesn't exist then field's value is not converted.

Parameters **dump** – whether the result should be json (True) or python dict (False).

Returns dictionary of values ready to be sent to NoSQL store.

MapModelBase.**to_dict**(**args*)
This method returns a dict containing fields and their values in this instance.

Returns values dict.

class basilisk.**ElasticsearchModel**(***kwargs*)
This is the base class for Elasticsearch models. Internally they are just a Elasticsearch entity. This class enables reading object with given id, saving object and data (de)serialization.

Elasticsearch connection is available in connect property. Dict of fields is available in _fields property.

Reserved property names, apart from methods, are _fields, id_field and connect.

exception **ElasticsearchModelException**
Exception raised when errors related to Redis handling are encountered.

ElasticsearchModel.**MapModelException**
alias of *ElasticsearchModelException*

classmethod ElasticsearchModel.**get**(*oid*)
This method gets a model instance with given id from Elasticsearch.

Parameters **oid** – id of object to get.

Returns hydrated model instance.

classmethod ElasticsearchModel.**get_key**(*oid=None*)
This function creates a key in which Elasticsearch will save the instance with given id.

Parameters **oid** – id of object for which a key should be created.

Returns Elasticsearch key (index, document and id).

ElasticsearchModel.**save**(*create_id=True*)
Let's save instance's current state to Elasticsearch.

Parameters **create_id** – whether id should be created automatically if it's not set yet.

class basilisk.**ElasticsearchModelException**
Exception raised when errors related to Redis handling are encountered.

class basilisk.**RedisModel**(***kwargs*)
This is the base class for Redis models. Internally they are just a Redis hash. This class enables reading object with given id, saving object and data (de)serialization. Redis connection is available in connect property. Dict of fields is available in _fields property.

Reserved property names, apart from methods, are _fields, id_field and connect.

MapModelException
alias of *RedisModelException*

exception RedisModelException
Exception raised when errors related to Redis handling are encountered.

classmethod RedisModel.**get**(*oid*)
This method gets a model instance with given id from Redis.

Parameters **oid** – id of object to get.

Returns hydrated model instance.

classmethod RedisModel.**get_key**(*oid*)
This function creates a key in which Redis will save the instance with given id.

Parameters **oid** – id of object for which a key should be created.

Returns Redis key.

RedisModel.**save**(*create_id=True*)
Let's save instance's current state to Redis.

Parameters **create_id** – whether id should be created automatically if it's not set yet.

Returns self

class basilisk.**RedisList**(*name*, *namespace=None*)
This class is a proxy for Redis List. It enables instant modifications to Redis entity. It has only basic operations pythonized at the moment.

append(*item*)
Adds element at the end of the list.

Parameters **item** – element to be appended.

clear()
This removes the list from Redis.

get_instance_key()
This function creates Redis's instance key.

Returns key in which instance will be saved.

classmethod **get_key**(*name*)
This method creates a Redis key in which instance with given name will be saved.

Parameters **name** – name of object for which a key is to be made.

Returns key used in Redis for given name.

pop(*first=False*)
Gets and removes an element from list's edge. By default it's the last element.

Parameters **first** – Should the first element be popped instead of the last.

prepend(*item*)
Adds element at the beginning of the list.

Parameters **item** – element to be prepended.

remove(*item*)
Removes all elements with given value.

Parameters **item** – value to be removed.

class basilisk.**RedisHash**(*name*, *namespace=None*)
This class acts as a proxy for Redis Hash. It enables delayed modifications. __setitem__ and __delitem__ methods don't modify Redis immediately, but are instead queued in a changelist.

clear()
This removes whole hash from Redis.

get(**fields*)
This gets one or many items from Redis.

Parameters **fields** – list of fields to get.

get_instance_key()
This function creates Redis's instance key.

Returns key in which instance will be saved.

classmethod **get_key**(*name*)
This method creates a Redis key in which instance with given name will be saved.

Parameters **name** – name of object for which a key is to be made.

Returns key used in Redis for given name.

items()
This returns key, value pairs available in this hash.

Returns hash's key, value pairs.

keys()
This returns list of keys in hash.

Returns list of keys in hash.

save()
This method analyzes changelist and using as few operations as possible propagates changes to Redis's Hash representing this instance.

class basilisk.**RedisSortedSet**(*name*, *namespace=None*)
This class is used to proxy Redis's Sorted Sets. It allows value search with pagination and delayed (lazy) key alterations. Indexing works with SCORE, not MEMBER or RANK.

set_score and delete_item methods don't interface with Redis directly, but are queued in a change list.

clear()
I'm tired of you, off you go. Disappear from Redis. NOW.

delete_item(*item*)
This method deletes given element. You need to call save() to propagate changes to Redis.

Parameters **item** – element to be removed.

get_instance_key()
This function creates Redis's instance key.

Returns key in which instance will be saved.

classmethod **get_key**(*name*)
This method creates a Redis key in which instance with given name will be saved.

Parameters **name** – name of object for which a key is to be made.

Returns key used in Redis for given name.

highest()
Returns element with highest SCORE and its SCORE.

Returns element with highest SCORE and its SCORE.

lowest()
Returns element with lowest SCORE and its SCORE.

Returns element with lowest SCORE and its SCORE.

save()
This method analyzes changelist and using as few operations as possible propagates changes to Redis's Sorted Set representing this instance.

set_score(*item*, *score*)
This function adds a new element if it's not in Redis and sets its SCORE. You need to call save() to propagate changes to Redis.

Parameters

- **item** – element to be added or modified.
- **score** – element's SCORE.

class basilisk.**RedisModelException**
Exception raised when errors related to Redis handling are encountered.

CHAPTER 1

Indices and tables

- genindex
- modindex
- search

b

basilisk, 1

A

append() (basilisk.RedisList method), 4

B

basilisk (module), 1

C

clear() (basilisk.RedisHash method), 5
clear() (basilisk.RedisList method), 4
clear() (basilisk.RedisSortedSet method), 5
Config (class in basilisk), 1

D

delete_item() (basilisk.RedisSortedSet method), 6

E

ElasticsearchModel (class in basilisk), 3
ElasticsearchModel.ElasticsearchModelException, 3
ElasticsearchModelException (class in basilisk), 4

G

get() (basilisk.ElasticsearchModel class method), 3
get() (basilisk.MapModelBase class method), 2
get() (basilisk.RedisHash method), 5
get() (basilisk.RedisModel class method), 4
get_default() (basilisk.MapField method), 1
get_fields() (basilisk.MapModelBase class method), 2
get_instance_key() (basilisk.MapModelBase method), 2
get_instance_key() (basilisk.RedisHash method), 5
get_instance_key() (basilisk.RedisList method), 4
get_instance_key() (basilisk.RedisSortedSet method), 6
get_key() (basilisk.ElasticsearchModel class method), 3
get_key() (basilisk.MapModelBase class method), 3
get_key() (basilisk.RedisHash class method), 5
get_key() (basilisk.RedisList class method), 4
get_key() (basilisk.RedisModel class method), 4
get_key() (basilisk.RedisSortedSet class method), 6
get_name() (basilisk.MapField method), 1

H

highest() (basilisk.RedisSortedSet method), 6

I

is_primary() (basilisk.MapField method), 1
items() (basilisk.RedisHash method), 5

J

JsonMapField (class in basilisk), 2

K

keys() (basilisk.RedisHash method), 5

L

load() (basilisk.Config class method), 1
lowest() (basilisk.RedisSortedSet method), 6

M

MapField (class in basilisk), 1
MapModelBase (class in basilisk), 2
MapModelBase.MapModelException, 2
MapModelException (basilisk.ElasticsearchModel attribute), 3
MapModelException (basilisk.RedisModel attribute), 4

P

pop() (basilisk.RedisList method), 5
prepend() (basilisk.RedisList method), 5
pythonize() (basilisk.JsonMapField method), 2
pythonize() (basilisk.MapField method), 2
pythonize() (basilisk.MapModelBase class method), 3

R

RedisHash (class in basilisk), 5
RedisList (class in basilisk), 4
RedisModel (class in basilisk), 4
RedisModel.RedisModelException, 4
RedisModelException (class in basilisk), 6
RedisSortedSet (class in basilisk), 5

remove() (basilisk.RedisList method), 5

S

save() (basilisk.ElasticsearchModel method), 3
save() (basilisk.MapModelBase method), 3
save() (basilisk.RedisHash method), 5
save() (basilisk.RedisModel method), 4
save() (basilisk.RedisSortedSet method), 6
serialize() (basilisk.JsonMapField static method), 2
serialize() (basilisk.MapField static method), 2
serialize() (basilisk.MapModelBase method), 3
set_name() (basilisk.MapField method), 2
set_score() (basilisk.RedisSortedSet method), 6

T

to_dict() (basilisk.MapModelBase method), 3