
scrapy-mongoengine-item Documentation

Release 0.1.5

Artur Barseghyan <artur.barseghyan@gmail.com>

Sep 08, 2019

Contents

1 Prerequisites	3
2 Installation	5
3 Introduction	7
4 Usage	9
5 Development	11
5.1 Testing	11
5.2 Running MongoDB	11
5.3 Writing documentation	11
6 License	13
7 Support	15
8 Author	17
9 Documentation	19
9.1 Release history and notes	19
9.1.1 0.1.5	19
9.1.2 0.1.4	19
9.1.3 0.1.3	20
9.1.4 0.1.2	20
9.2 scrapy_mongoengine_item package	20
9.2.1 Module contents	20
10 Indices and tables	21
Python Module Index	23
Index	25

Bringing Scrapy and MongoEngine together.

scrapy-mongoengine-item is an extension that allows you to define Scrapy items using existing MongoEngine documents.

Documentation is available on [Read the Docs](#).

CHAPTER 1

Prerequisites

Both Python 2.7 and Python 3.5/3.6 are supported. For Python 3 you need Scrapy v1.1 or above.
Latest tested MongoEngine version is MongoEngine 0.17.0.

(1) Install latest stable version from PyPI:

```
pip install scrapy-mongoengine-item
```

or latest stable version from GitHub:

```
pip install https://github.com/barseghyanartur/scrapy-mongoengine-item/archive/  
↪stable.tar.gz
```

or latest stable version from BitBucket:

```
pip install https://bitbucket.org/barseghyanartur/scrapy-mongoengine-item/get/  
↪stable.tar.gz
```


CHAPTER 3

Introduction

`MongoEngineItem` is a class of item that gets its fields definition from a `MongoEngine` document, you simply create a `MongoEngineItem` and specify what `MongoEngine` document it relates to.

Besides of getting the document fields defined on your item, `MongoEngineItem` provides a method to create and populate a `MongoEngine` document instance with the item data.

MongoEngineItem works as follows: you create a subclass and define its `mongoengine_document` attribute to be a valid MongoEngine document. With this you will get an item with a field for each MongoEngine document field.

In addition, you can define fields that aren't present in the document and even override fields that are present in the model defining them in the item.

Let's see some examples:

Creating a MongoEngine document for the examples:

```
from mongoengine import fields, document

class Person(document.Document):

    name = fields.StringField(max_length=255)
    age = fields.IntField()
```

Defining a basic MongoEngineItem:

```
from scrapy_mongoengine_item import MongoEngineItem

class PersonItem(MongoEngineItem):

    mongoengine_document = Person
```

MongoEngineItem works just like Scrapy items:

```
p = PersonItem()
p['name'] = 'John'
p['age'] = 22
```

To obtain the MongoEngine document from the item, we call the extra method `MongoEngineItem.save()` of the MongoEngineItem:

```
person = p.save()
person.name
# 'John'
person.age
# 22
person.id
# 1
```

The document is already saved when we call `MongoEngineItem.save()`, we can prevent this by calling it with `commit=False`. We can use `commit=False` in `MongoEngineItem.save()` method to obtain an unsaved document:

```
person = p.save(commit=False)
person.name
# 'John'
person.age
# 22
person.id
# None
```

As said before, we can add other fields to the item:

```
import scrapy
from scrapy_mongoengine_item import MongoEngineItem

class PersonItem(MongoEngineItem):

    mongoengine_document = Person
    sex = scrapy.Field()
```

```
p = PersonItem()
p['name'] = 'John'
p['age'] = 22
p['sex'] = 'M'
```

And we can override the fields of the document with your own:

```
class PersonItem(MongoEngineItem):

    mongoengine_document = Person
    name = scrapy.Field(default='No Name')
```

This is useful to provide properties to the field, like a default or any other property that your project uses. Those additional fields won't be taken into account when doing a `MongoEngineItem.save()`.

5.1 Testing

To run tests in your working environment type:

```
./runtests.py
```

To test with all supported Python versions type:

```
tox
```

5.2 Running MongoDB

The easiest way is to run it via Docker:

```
docker pull mongo:latest
docker run -p 27017:27017 mongo:latest
```

5.3 Writing documentation

Keep the following hierarchy.

```
====
title
====

header
=====
```

(continues on next page)

(continued from previous page)

```
sub-header
-----

sub-sub-header
~~~~~

sub-sub-sub-header
^^^^^

sub-sub-sub-sub-header
+++++

sub-sub-sub-sub-sub-header
*****
```


CHAPTER 6

License

GPL-2.0-only OR LGPL-2.1-or-later

CHAPTER 7

Support

For any issues contact me at the e-mail given in the *Author* section.

CHAPTER 8

Author

Artur Barseghyan <artur.barseghyan@gmail.com>

Contents:

9.1 Release history and notes

Sequence based identifiers are used for versioning (schema follows below):

```
major.minor[.revision]
```

- It's always safe to upgrade within the same minor version (for example, from 0.3 to 0.3.4).
- Minor version changes might be backwards incompatible. Read the release notes carefully before upgrading (for example, when upgrading from 0.3.4 to 0.4).
- All backwards incompatible changes are mentioned in this document.

9.1.1 0.1.5

2019-05-29

- Minor fixes in docs.
- Added travis.

9.1.2 0.1.4

2019-03-16

- Clean up. Add proper docs.

9.1.3 0.1.3

2019-03-14

- Beta release.

9.1.4 0.1.2

2019-03-14

- Initial alpha release.

9.2 scrapy_mongoengine_item package

9.2.1 Module contents

Scrapy extension to write scraped items using *mongoengine* documents

```
class scrapy_mongoengine_item.MongoEngineItem(*args, **kwargs)
    Bases: scrapy.item.Item

    errors
    fields = {}
    instance
    is_valid()
    mongoengine_document = None
    save (commit=True)
```


CHAPTER 10

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

S

`scrapy_mongoengine_item`, 20

E

errors (*scrapy_mongoengine_item.MongoEngineItem*
attribute), 20

F

fields (*scrapy_mongoengine_item.MongoEngineItem*
attribute), 20

I

instance (*scrapy_mongoengine_item.MongoEngineItem*
attribute), 20

is_valid() (*scrapy_mongoengine_item.MongoEngineItem*
method), 20

M

mongoengine_document
(*scrapy_mongoengine_item.MongoEngineItem*
attribute), 20

MongoEngineItem (class in
scrapy_mongoengine_item), 20

S

save() (*scrapy_mongoengine_item.MongoEngineItem*
method), 20

scrapy_mongoengine_item (module), 20