

---

# Django-money Documentation

*Release 0.12.1*

**Jacob Hansson**

Nov 20, 2017



---

# Contents

---

<b>1</b>	<b>Django-money</b>	<b>1</b>
1.1	Installation . . . . .	1
1.2	Model usage . . . . .	2
1.3	Field validation . . . . .	2
1.4	Adding a new Currency . . . . .	3
1.5	Important note on model managers . . . . .	3
1.6	Format localization . . . . .	4
1.7	Testing . . . . .	5
1.8	Working with Exchange Rates . . . . .	5
1.9	Usage with Django REST Framework . . . . .	5
1.10	Known Issues . . . . .	6
<b>2</b>	<b>Contents</b>	<b>7</b>
2.1	Changelog . . . . .	7
<b>3</b>	<b>Indices and tables</b>	<b>17</b>



A little Django app that uses `py-moneyed` to add support for Money fields in your models and forms.

Fork of the Django support that was in <http://code.google.com/p/python-money/>

This version adds tests, and comes with several critical bugfixes.

Django versions supported: 1.8, 1.10, 1.11

Python versions supported: 2.7, 3.3, 3.4, 3.5, 3.6

PyPy versions supported: PyPy 2.6, PyPy3 2.4

If you need support for older versions of Django and Python you can use the latest version in 0.11.x branch.

Via `py-moneyed`, `django-money` gets:

- Support for proper Money value handling (using the standard Money design pattern)
- A currency class and definitions for all currencies in circulation
- Formatting of most currencies with correct currency sign

## 1.1 Installation

Using `pip`:

```
pip install django-money
```

This automatically installs `py-moneyed` v0.7 (or later).

Add `djmoney` to your `INSTALLED_APPS`. This is required so that money field are displayed correctly in the admin.

```
INSTALLED_APPS = [  
    ...,  
    'djmoney',  
    ...  
]
```

## 1.2 Model usage

Use as normal model fields

```
from djmoney.models.fields import MoneyField
from django.db import models

class BankAccount (models.Model):
    balance = MoneyField(max_digits=10, decimal_places=2, default_currency='USD')
```

Searching for models with money fields:

```
from djmoney.money import Money

account = BankAccount.objects.create(balance=Money(10, 'USD'))
swissAccount = BankAccount.objects.create(balance=Money(10, 'CHF'))

BankAccount.objects.filter(balance__gt=Money(1, 'USD'))
# Returns the "account" object
```

## 1.3 Field validation

There are 3 different possibilities for field validation:

- by numeric part of money despite on currency;
- by single money amount;
- by multiple money amounts.

All of them could be used in a combination as is shown below:

```
from django.db import models
from djmoney.models.fields import MoneyField
from djmoney.money import Money
from djmoney.models.validators import MaxMoneyValidator, MinMoneyValidator

class BankAccount (models.Model):
    balance = MoneyField(
        max_digits=10,
        decimal_places=2,
        validators=[
            MinMoneyValidator(10),
            MaxMoneyValidator(1500),
            MinMoneyValidator(Money(500, 'NOK')),
            MaxMoneyValidator(Money(900, 'NOK')),
            MinMoneyValidator({'EUR': 100, 'USD': 50}),
            MaxMoneyValidator({'EUR': 1000, 'USD': 500}),
        ]
    )
```

The balance field from the model above has the following validation:

- All input values should be between 10 and 1500 despite on currency;

- Norwegian Crowns amount (NOK) should be between 500 and 900;
- Euros should be between 100 and 1000;
- US Dollars should be between 50 and 500;

## 1.4 Adding a new Currency

Currencies are listed on moneyed, and this modules use this to provide a choice list on the admin, also for validation.

To add a new currency available on all the project, you can simple add this two lines on your `settings.py` file

```
import moneyed
from moneyed.localization import _FORMATTER
from decimal import ROUND_HALF_EVEN

BOB = moneyed.add_currency(
    code='BOB',
    numeric='068',
    name='Peso boliviano',
    countries=('BOLIVIA', )
)

# Currency Formatter will output 2.000,00 Bs.
_FORMATTER.add_sign_definition(
    'default',
    BOB,
    prefix=u'Bs. '
)

_FORMATTER.add_formatting_definition(
    'es_BO',
    group_size=3, group_separator=".", decimal_point="," ,
    positive_sign="", trailing_positive_sign="",
    negative_sign="-", trailing_negative_sign="",
    rounding_method=ROUND_HALF_EVEN
)
```

To restrict the currencies listed on the project set a `CURRENCIES` variable with a list of Currency codes on `settings.py`

```
CURRENCIES = ('USD', 'BOB')
```

### The list has to contain valid Currency codes

Additionally there is an ability to specify currency choices directly:

```
CURRENCIES = ('USD', 'EUR')
CURRENCY_CHOICES = (('USD', 'USD $'), ('EUR', 'EUR €'))
```

## 1.5 Important note on model managers

Django-money leaves you to use any custom model managers you like for your models, but it needs to wrap some of the methods to allow searching for models with money values.

This is done automatically for the “objects” attribute in any model that uses MoneyField. However, if you assign managers to some other attribute, you have to wrap your manager manually, like so:

```
from djmoney.models.managers import money_manager

class BankAccount (models.Model):
    balance = MoneyField(max_digits=10, decimal_places=2, default_currency='USD')
    accounts = money_manager(MyCustomManager())
```

Also, the money\_manager wrapper only wraps the standard QuerySet methods. If you define custom QuerySet methods, that do not end up using any of the standard ones (like “get”, “filter” and so on), then you also need to manually decorate those custom methods, like so:

```
from djmoney.models.managers import understands_money

class MyCustomQuerySet (QuerySet):

    @understands_money
    def my_custom_method(*args, **kwargs):
        # Awesome stuff
```

## 1.6 Format localization

The formatting is turned on if you have set `USE_L10N = True` in the your settings file.

If formatting is disabled in the configuration, then in the templates will be used default formatting.

In the templates you can use a special tag to format the money.

In the file `settings.py` add to `INSTALLED_APPS` entry from the library `djmoney`:

```
INSTALLED_APPS += ('djmoney', )
```

In the template, add:

```
{% load djmoney %}
...
{% money_localize money %}
```

and that is all.

Instructions to the tag `money_localize`:

```
{% money_localize <money_object> [ on(default) | off ] [as var_name] %}
{% money_localize <amount> <currency> [ on(default) | off ] [as var_name] %}
```

Examples:

The same effect:

```
{% money_localize money_object %}
{% money_localize money_object on %}
```

Assignment to a variable:

```
{% money_localize money_object on as NEW_MONEY_OBJECT %}
```

Formatting the number with currency:

```
{% money_localize '4.5' 'USD' %}
```

Return::

```
Money object
```

## 1.7 Testing

Install the required packages:

```
git clone https://github.com/django-money/django-money
cd ./django-money/
pip install -e .[tests] # installation with required packages for testing
```

Recommended way to run the tests:

```
tox
```

Testing the application in the current environment python:

```
make test
```

## 1.8 Working with Exchange Rates

To work with exchange rates, check out this repo that builds off of django-money: <https://github.com/evonove/django-money-rates>

django-money can be configured to automatically use this app for currency conversions by settings `AUTO_CONVERT_MONEY = True` in your Django settings. Note that currency conversion is a lossy process, so automatic conversion is usually a good strategy only for very simple use cases. For most use cases you will need to be clear about exactly when currency conversion occurs, and automatic conversion can hide bugs. Also, with automatic conversion you lose some properties like commutativity ( $A + B == B + A$ ) due to conversions happening in different directions.

## 1.9 Usage with Django REST Framework

Make sure that `djmoney` is in the `INSTALLED_APPS` of your `settings.py` and `MoneyFields` to automatically work with Django REST Framework.

Built-in serializer works in the following way:

```
class Expenses(models.Model):
    amount = MoneyField(max_digits=10, decimal_places=2)
```

```
class Serializer(serializers.ModelSerializer):
    class Meta:
        model = Expenses
        fields = '__all__'

>>> instance = Expenses.objects.create(amount=Money(10, 'EUR'))
>>> serializer = Serializer(instance=instance)
>>> serializer.data
ReturnDict([
    ('id', 1),
    ('amount_currency', 'EUR'),
    ('amount', '10.000'),
])
```

## 1.10 Known Issues

Updates to a model form will not save in Django 1.10.1. They will save in 1.10.0 and is expected to be fixed in Django 1.10.2.

<https://github.com/django/django/pull/7217>

## 2.1 Changelog

### 2.1.1 0.12.1 - 2017-11-20

#### Fixed

- Fixed migrations on SQLite. #139, #338 (Stranger6667)
- Fixed `Field.related_name` usage for Django 2.0. #349 (richardowen)
- Fixed Django REST Framework behaviour for serializers without `*_currency` field in serializer's `Meta.fields`. #351 (elcolie, Stranger6667)

### 2.1.2 0.12 - 2017-10-22

#### Added

- Ability to specify name for currency field. #195 (Stranger6667)
- Validators for `MoneyField`. #308 (Stranger6667)

#### Changed

- Improved Money support. Now `django-money` fully relies on `pymoneyed` localization everywhere, including Django admin. #276 (Stranger6667)
- Implement `__html__` method. If used in Django templates, an `Money` object's amount and currency are now separated with non-breaking space (`&nbsp;`; ) #337 (jonashaag)

### Deprecated

- `djmoney.models.fields.MoneyPatched` and `moneyed.Money` are deprecated. Use `djmoney.money.Money` instead.

### Fixed

- Fixed model field validation. #308 (Stranger6667).
- Fixed managers caching for Django  $\geq$  1.10. #318 (Stranger6667).
- Fixed F expressions support for `in` lookups. #321 (Stranger6667).
- Fixed money comprehension on querysets. #331 (Stranger6667, jaavii1988).
- Fixed errors in Django Admin integration. #334 (Stranger6667, adi-).

### Removed

- Dropped support for Python 2.6 and 3.2. (Stranger6667)
- Dropped support for Django 1.4, 1.5, 1.6, 1.7 and 1.9. (Stranger6667)

## 2.1.3 0.11.4 - 2017-06-26

### Fixed

- Fixed money parameters processing in update queries. #309 (Stranger6667)

## 2.1.4 0.11.3 - 2017-06-19

### Fixed

- Restored support for Django 1.4, 1.5, 1.6, and 1.7 & Python 2.6 #304 (Stranger6667)

## 2.1.5 0.11.2 - 2017-05-31

### Fixed

- Fixed field lookup regression. #300 (lmdsp, Stranger6667)

## 2.1.6 0.11.1 - 2017-05-26

### Fixed

- Fixed access to models properties. #297 (mithrilstar, Stranger6667)

## Removed

- Dropped support for Python 2.6. (Stranger6667)
- Dropped support for Django < 1.8. (Stranger6667)

## 2.1.7 0.11 - 2017-05-19

### Added

- An ability to set custom currency choices via `CURRENCY_CHOICES` settings option. #211 (Stranger6667, ChessSpider)

### Fixed

- Fixed `AttributeError` in `get_or_create` when the model have no default. #268 (Stranger6667, lobziik)
- Fixed `UnicodeEncodeError` in string representation of `MoneyPatched` on Python 2. #272 (Stranger6667)
- Fixed various displaying errors in Django Admin . #232, #220, #196, #102, #90 (Stranger6667, arthurk, mstarostik, eriktelepovsky, jplehmann, graik, benjaoming, k8n, yellow-sky)
- Fixed non-Money values support for `in` lookup. #278 (Stranger6667)
- Fixed available lookups with removing of needless lookup check. #277 (Stranger6667)
- Fixed compatibility with `py-moneyed`. (Stranger6667)
- Fixed ignored currency value in Django REST Framework integration. #292 (gonzalobf)

## 2.1.8 0.10.2 - 2017-02-18

### Added

- Added ability to configure decimal places output. #154, #251 (ivanchenkodmitry)

### Fixed

- Fixed handling of `defaults` keyword argument in `get_or_create` method. #257 (kjagiello)
- Fixed handling of currency fields lookups in `get_or_create` method. #258 (Stranger6667)
- Fixed `PendingDeprecationWarning` during form initialization. #262 (Stranger6667, spookylukey)
- Fixed handling of `F` expressions which involve non-Money fields. #265 (Stranger6667)

## 2.1.9 0.10.1 - 2016-12-26

### Fixed

- Fixed default value for `djmoney.forms.fields.MoneyField`. #249 (tsouvarev)

### 2.1.10 0.10 - 2016-12-19

#### Changed

- Do not fail comparisons because of different currency. Just return `False` #225 (benjaoming and ivirabyan)

#### Fixed

- Fixed `understands_money` behaviour. Now it can be used as a decorator #215 (Stranger6667)
- Fixed: Not possible to revert `MoneyField` currency back to default #221 (benjaoming)
- Fixed invalid `creation_counter` handling. #235 (msgre and Stranger6667)
- Fixed broken field resolving. #241 (Stranger6667)

### 2.1.11 0.9.1 - 2016-08-01

#### Fixed

- Fixed packaging.

### 2.1.12 0.9.0 - 2016-07-31

NB! If you are using custom model managers **not** named *objects* and you expect them to still work, please read below.

#### Added

- Support for `Value` and `Func` expressions in queries. (Stranger6667)
- Support for `in` lookup. (Stranger6667)
- Django REST Framework support. #179 (Stranger6667)
- Django 1.10 support. #198 (Stranger6667)
- Improved South support. (Stranger6667)

#### Changed

- Changed auto conversion of currencies using `djmoney_rates` (added in 0.7.3) to be off by default. You must now add `AUTO_CONVERT_MONEY = True` in your `settings.py` if you want this feature. #199 (spookylukey)
- Only make *objects* a `MoneyManager` instance automatically. #194 and #201 (inureyes)

#### Fixed

- Fixed default currency value for nullable fields in forms. #138 (Stranger6667)
- Fixed `_has_changed` deprecation warnings. #206 (Stranger6667)
- Fixed `get_or_create` crash, when `defaults` is passed. #213 (Stranger6667, spookylukey)

## Note about automatic model manager patches

In 0.8, Django-money automatically patches every model managers with `MoneyManager`. This causes migration problems if two or more managers are used in the same model.

As a side effect, other managers are also finally wrapped with `MoneyManager`. This effect leads Django migration to point to fields with other managers to `MoneyManager`, and raises `ValueError` (`MoneyManager` only exists as a return of `money_manager`, not a class-form. However migration procedure tries to find `MoneyManager` to patch other managers.)

From 0.9, Django-money only patches objects with `MoneyManager` by default (as documented). To patch other managers (e.g. custom managers), patch them by wrapping with `money_manager`.

```
from djmoney.models.managers import money_manager

class BankAccount (models.Model) :
    balance = MoneyField(max_digits=10, decimal_places=2, default_currency='USD')
    accounts = money_manager(MyCustomManager())
```

## 2.1.13 0.8 - 2016-04-23

### Added

- Support for serialization of `MoneyPatched` instances in migrations. (AlexRiina)
- Improved django-money-rates support. #173 (Stranger6667)
- Extended F expressions support. (Stranger6667)
- Pre-commit hooks support. (benjaoming)
- Isort integration. (Stranger6667)
- Makefile for common commands. (Stranger6667)
- Codecov.io integration. (Stranger6667)
- Python 3.5 builds to tox.ini and travis.yml. (Stranger6667)
- Django master support. (Stranger6667)
- Python 3.2 compatibility. (Stranger6667)

### Changed

- Refactored test suite (Stranger6667)

### Fixed

- Fixed fields caching. #186 (Stranger6667)
- Fixed m2m fields data loss on Django < 1.8. #184 (Stranger6667)
- Fixed managers access via instances. #86 (Stranger6667)
- Fixed currency handling behaviour. #172 (Stranger6667)
- Many PEP8 & flake8 fixes. (benjaoming)

- Fixed filtration with F expressions. #174 (Stranger6667)
- Fixed querying on Django 1.8+. #166 (Stranger6667)

### 2.1.14 0.7.6 - 2016-01-08

#### Added

- Added correct paths for py.test discovery. (benjaoming)
- Mention Django 1.9 in tox.ini. (benjaoming)

#### Fixed

- Fix for `get_or_create / create` manager methods not respecting currency code. (toudi)
- Fix unit tests. (toudi)
- Fix for using `MoneyField` with F expressions when using Django  $\geq$  1.8. (toudi)

### 2.1.15 0.7.5 - 2015-12-22

#### Fixed

- Fallback to `_meta.fields` if `_meta.get_fields` raises `AttributeError` #149 (browniebroke)
- pip instructions updated. (GheloAce)

### 2.1.16 0.7.4 - 2015-11-02

#### Added

- Support for Django 1.9 (kjagiello)

#### Fixed

- Fixed `loaddata`. (jack-cvr)
- Python 2.6 fixes. (jack-cvr)
- Fixed currency choices ordering. (synotna)

### 2.1.17 0.7.3 - 2015-10-16

#### Added

- Sum different currencies. (dnmellen)
- `__eq__` method. (benjaoming)
- Comparison of different currencies. (benjaoming)
- Default currency. (benjaoming)

## Fixed

- Fix using Choices for setting currency choices. (benjaoming)
- Fix tests for Python 2.6. (plumdog)

### 2.1.18 0.7.2 - 2015-09-01

## Fixed

- Better checks on `None` values. (tsouvarev, sjdines)
- Consistency with South declarations and calling `str` function. (sjdines)

### 2.1.19 0.7.1 - 2015-08-11

## Fixed

- Fix bug in printing `MoneyField`. (YAmikep)
- Added fallback value for current locale getter. (sjdines)

### 2.1.20 0.7.0 - 2015-06-14

## Added

- Django 1.8 compatibility. (willhcr)

### 2.1.21 0.6.0 - 2015-05-23

## Added

- Python 3 trove classifier. (dekkers)

## Changed

- Tox cleanup. (edwinlunando)
- Improved README. (glarrain)
- Added/Cleaned up tests. (spookylukey, AlexRiina)

## Fixed

- Append `_currency` to non-money `ExpressionFields`. #101 (alexhayes, AlexRiina, briankung)
- Data truncated for column. #103 (alexhayes)
- Fixed `has_changed` not working. #95 (spookylukey)
- Fixed proxy model with `MoneyField` returns wrong class. #80 (spookylukey)

### 2.1.22 0.5.0 - 2014-12-15

#### Added

- Django 1.7 compatibility. ([w00kie](#))

#### Fixed

- Added `choices=` to instantiation of currency widget. ([davidstockwell](#))
- Nullable `MoneyField` should act as `default=None`. ([jakewins](#))
- Fixed bug where a non-required `MoneyField` threw an exception. ([spookylukey](#))

### 2.1.23 0.4.2 - 2014-07-31

### 2.1.24 0.4.1 - 2013-11-28

### 2.1.25 0.4.0.0 - 2013-11-26

#### Added

- Python 3 compatibility.
- tox tests.
- Format localization.
- Template tag `money_localize`.

### 2.1.26 0.3.4 - 2013-11-25

### 2.1.27 0.3.3.2 - 2013-10-31

### 2.1.28 0.3.3.1 - 2013-10-01

### 2.1.29 0.3.3 - 2013-02-17

#### Added

- South support via implementing the `south_triple_field` method. ([mattions](#))

#### Fixed

- Fixed issues with money widget not passing attrs up to django's render method, caused id attribute to not be set in html for widgets. ([adambregenzer](#))
- Fixed issue of default currency not being passed on to widget. ([snbuchholz](#))
- Return the right default for South. ([mattions](#))
- Django 1.5 compatibility. ([devlocal](#))

### 2.1.30 0.3.2 - 2012-11-30

#### Fixed

- Fixed issues with `display_for_field` not detecting fields correctly. (adambregenzer)
- Added South ignore rule to avoid duplicate currency field when using the frozen ORM. (rach)
- Disallow override of objects manager if not setting it up with an instance. (rach)

### 2.1.31 0.3.1 - 2012-10-11

#### Fixed

- Fix `AttributeError` when Model inherit a manager. (rach)
- Correctly serialize the field. (akumria)

### 2.1.32 0.3 - 2012-09-30

#### Added

- Allow django-money to be specified as read-only in a model. (akumria)
- South support: Declare default attribute values. (pjdelport)

### 2.1.33 0.2 - 2012-04-10

- Initial public release



## CHAPTER 3

---

### Indices and tables

---

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