
djreservation Documentation

Release 0.2.2

Luis Zárate Montero

Dec 17, 2018

Contents

1	Installation	3
1.1	In settings.py	3
1.2	Configure database	4
1.3	In urls.py	4
2	Basic usage	5
2.1	ProductReservationView:	5
2.2	SimpleProductReservation:	6
2.3	Templates	6
3	Settings configuration	7
4	Contrib CRUD	9
4.1	ObjectView	9
4.2	UserObjectView	9
5	Indices and tables	11

Reserve object

Information

Code: v4
Type: Material
Quantity: 20.000 Milimeters

Description

dasdadassdaada

Amount:	10
Available:	20.0
Measurement unit	Milimeters

Reserve

Contents:

Install with pip

```
$ pip install django-reservation
```

1.1 In settings.py

Set “djreservation” in your INSTALLED_APPS.

```
INSTALLED_APPS = [  
    ...  
    'djreservation'  
]
```

Set ‘djreservation.middleware.ReservationMiddleware’ in MIDDLEWARE

```
MIDDLEWARE = [  
    ...  
    'djreservation.middleware.ReservationMiddleware'  
]
```

Note: Middleware is not necessary if you just use *SimpleProductReservation*

Configure your email settings

```
DEFAULT_FROM_EMAIL = "mail@example.com"  
EMAIL_HOST = "localhost"  
EMAIL_PORT = "1025"
```

1.2 Configure database

Run migrations

```
python manage.py migrate
```

1.3 In urls.py

Append django reservation to urlpatterns

```
from djreservation import urls as djreservation_urls

urlpatterns = [
    ...
] + djreservation_urls.urlpatterns
```


django-reservation have two different approaches:

- **ProductReservationView:** You have a shop cart for a reservation in determinate period of time.
- **SimpleProductReservation:** You reserve one product in determinate period of time.

Both views inherit from `CreateView`. so all verifications and functions are included and of course you can overwrite.

The main diferent is the way how the user make a reservation

2.1 ProductReservationView:

In this approach exist a button that start the reservation process, suppose your are the user, when start you need to set the period of time you want the reservation, then you reserve all products you want and as final step you finish the reservation and an email is send you.

How to implement that behaviour:

Create a view for reserve a product

```
from django.views import ProductReservationView

class MyObjectReservation(ProductReservationView):
    base_model = MyObject # required
    amount_field = 'quantity' # required
    extra_display_field = ['measurement_unit'] # not required
```

Set the urlpatterns in your urls.py

```
urlpatterns = [
    ...
    url(r"^reservation/create$", MyObjectReservation.as_view())
]
```

2.2 SimpleProductReservation:

In this approach you can reserve a product for a period of time, like a room in a hotel.

```
from django.views import SimpleProductReservation
from .models import MyModel

class RoomReservation(SimpleProductReservation):
    base_model = MyModel # required
    amount_field = 'quantity' # required
    max_amount_field = 'max_amount' # required
    extra_display_field = [] # not required
```

Set the urlpatterns in your urls.py

```
urlpatterns = [
    ...
    url(r'^reservationroom/create/(?<pk>\d+)$', RoomReservation.as_view())
]
```

2.3 Templates

django-reservation require a template 'base.html' with the follow blocks

```
{% block css%} {%endblock%}

{% block content %}
{%endblock%}

{% block js %}
{%endblock%}
```

you can overwrite whatever template you want base in app templates

Settings configuration

- **DJRESERVATION_TOKENIZE**: Reservation can be updated by non-registered user using a token send by mail. Token only work one time. default False
- **DJRESERVATION_START_RESERVATION_DATETIME**: initialize start reservation datetime widget with this date. default format ‘%d/%m/%Y %H:%M’
- **DJRESERVATION_END_RESERVATION_DATETIME**: initialize end reservation datetime widget with this date. default format ‘%d/%m/%Y %H:%M’

Create a CRUD (Create, Remove, Update, Detail) using one class. It Work if user is loggin.

4.1 ObjectView

Create simple models, than not need set request.user

Example:

```
from djreservation.contrib.CRUD import ObjectView
class TShirt(UserObjectView):
    model = TShirtmodel # requiered
    template_name_base = "tshirt/tshirt" # not required but recomendable
    namespace = "tshirt" # required
    fields = [ ... ] # not required

tshirts = TShirt()
```

set urlpatterns in urls.py

```
urlpatterns = [
    ...
    url(r'^tshirts/', include(tshirts.get_urls(), namespace="tshirts")),
]
```

4.2 UserObjectView

If your models have a user field and want to set automáticamente the user them

```
from djreservation.contrib.CRUD import UserObjectView
```

```
from djreservation.contrib.CRUD import ObjectView
class SpeakproposeCRUD(UserObjectView):
    model = Speakpropose # requiered
    template_name_base = "tshirt/tshirt" # not required but recomandable
    namespace = "proppose" # required
    fields = [ fields with out user ] # not required

propposes = SpeakproposeCRUD()
```

set urlpatterns in urls.py

```
urlpatterns = [
    ...
    url(r'^propposes/', include(propposes.get_urls(), namespace="propposes")),
]
```

CHAPTER 5

Indices and tables

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