django-oscar-mpesa Documentation

Release 0.0.1

Craig Loftus, Samir Shah, Victor Munene

October 27, 2014

Contents

This package provides integration between django-oscar and M-Pesa's Instant Payment Notification (IPN) service.

CHAPTER 1

Installation

Install the package using either:

```
pip install django-oscar-mpesa
```

or:

```
pip install https://bitbucket.org/regulusweb/django-oscar-mpesa/get/master.zip
```

Add `mpesa` to your `INSTALLED_APPS`, and add mpesa's url configuration to your urlpatterns:

```
urlpatterns = patterns('',
    # your other url patterns go here

    (r'', include(mpesa.urls))
)
```

Finally run:

```
python manage.py syncdb
```

Contents:

1.1 How it works

Safaricom's Instant Payment Notification (IPN) service allows organisations to receive real-time notification of M-Pesa payments made to their Pay Bill number.

An IPN is a notification that is sent every time a payment is made to your to your Pay Bill account. Safaricom sends this notification in the form of a HTTP GET request.

This module provides integration between the IPN service and Oscar, making it it easy to receive and allocate M-Pesa payments for online purchases on your Oscar site. The process is, broadly, as follows:

1. A 'Pay by M-Pesa' option is made available to the customer at checkout.
2. The customer is instructed to send the order amount to the Pay Bill number, and enter the transaction reference provided by Safaricom.
3. When an IPN notification matching that transaction reference is received, the payment is allocated to the order.

1.2 Getting Started

Safaricom's Instant Payment Notification (IPN) service allows organisations to receive rapid notification of M-Pesa payments made to their Pay Bill number.

When signing up for IPN, you will be asked to provide an endpoint URL to Safaricom. You will need to create an endpoint in your project's urls.py file. The view used for this url is *mpesa.views.IPNReceiverView*. An example would look like this:

```
urlpatterns += pattern('',
    url(r"^ipn/?$", mpesa.views.IPNReceiverView.as_view(), name="ipn-receiver"),
)
```

Now IPN notifications can be received at */ipn*. This is the URL you will supply to Safaricom.

The IPN service provides basic password authentication for notifications. You will be asked to supply a username and password when applying for IPN. These parameters will be sent with every IPN notification.

Add your chosen username and password to your settings file as follows:

```
MPESA_IPN_USER = 'username'
MPESA_IPN_PASS = 'password'
```

Warning: If the MPESA_IPN_USER and MPESA_IPN_PASS settings do not match the username and password used in your IPNs, the IPNs will be ignored.

Add your Pay Bill number to the settings:

```
MPESA_PAYBILL_NUMBER = '123456'
```

This will be used in the payment instructions displayed to buyers.

If you're using oscar's dashboard app, extend the dashboard navigation:

```
from oscar.defaults import *
from django.utils.translation import ugettext_lazy as _
OSCAR_DASHBOARD_NAVIGATION.append(
        {
                'label': _('M-Pesa'),
                'icon': 'icon-globe',
                'children': [
                        {
                                'label': _('M-Pesa transactions'),
                                'url_name': 'mpesa-payments-list',
                        },
                ]
        })
```

1.3 Order Handling

This module works in conjunction with oscar's `payment` app to track payments. Note that this module does not do any order processing for you.

When an order is created a `Source` of type *M-Pesa* is created for the order. Every time a payment is made to your Pay Bill account, a `Transaction` indicating the amount paid, will be added to this Source and a signal is sent containing the M-Pesa payment and the transaction.

Your application should listen for this signal, and use it to check if the order has been fully paid and take the appropriate steps thereafter.

The sandbox contains a sample implementation of order processing. See sandbox/apps/order/processing.py.

1.4 Fake IPN Generator

For testing purposes, there is an IPN generator page. This can be used to test your order handling code.

By default the IPN generator can be accessed at */en-gb/ipn-gen*.

1.5 Limitations of the IPN service

The IPN service has some limitations which developers should be aware of. All of these are undocumented "features" and have only been learnt of through experience.

- **Responses are not sent back to the customer.** The IPN documentation provided by Safaricom suggests that you can respond to a payment with either a "success" or "failure" message, which should determine whether they payment is accepted, and send the message to th cutomer. For example a response like this:

  ```
  "OK|Custom Message"
  ```

  ... would send an SMS to the customer with the message `Custom Message`. This could be useful, for example, for letting a customer that they have underpaid/overpaid for an order.

  Unfortunately, this feature hasn't been implemented by Safaricom. IPN is at present a notification-only service and the response has no effect on the status of the payment. It isn't relayed to the customer, either.
- **Failed notifications are not re-sent.** If a notification is sent while your server is experiencing down-time, Safaricom will not make an attempt to send it again in the future.
- **Notifications might be sent more than once for the same payment.** These duplicate IPNs are ignored.
- **Passwords longer than 20 characters are truncated.** Keep your authentication password shorter than 20 characters.
- **An IPN is sent when the Paybill account holder withdraws money from their account.** From the perspective of this module, this is not a payment and will therefore be ignored.

CHAPTER 2

Indices and tables

- *genindex*
- *modindex*
- *search*