Follow mbrochh

# MARTIN BROCHHAUS

ENTREPRENEUR, SOFTWARE ENGINEER, FREELANCE WEBDESIGNER, HOUSE DJ, PHOTOGRAPHER NOOB, WORLD TRAVELER

ASK ARCHIVE RANDOM RSS SEARCH

- 19TH MAY 2011 | 3 NOTES -

# DJANGO-SHOP ADVENTURES: SHIPPING AND PAYMENT

After we have implemented the **model and admin** for our shop's products in the last post on this series, we are going to deal with shipping and payment next.

#### Adding FlatRateShipping and PayOnDelivery

This is the first time where we have to leave the **getting started** document which is a pity, because I feel that instructions for a very simple shipping backend and a very simple payments processor should be in there as well.

The **plugins** document gives us some hints on how to proceed. While the instructions are quite clear, I don't seem to get the big picture. Browsing the code a bit reveals that an example implementation for a hipping backend comes with the shop: The **FlatRateShipping** backend.

Let's just press that fancy big red button and see what happens. So we will add the shipping backend to our settings.py:

I also added the shipping fee. Notes on the available settings can be found in the **settings** 

**document**. And when we are at it I also added the setting for the payment backend.

#### Enjoying our newly available views

Now we can proceed to checkout. We will be presented a page where we can chose our shipping backend:

## **Shipping**

Please select a shipping option:

Flat rate

After chosing "Flat rate shipping" we see a page that tells us how much the shipping will be. This is mainly a demonstration to show that the shipping backend can have it's own views and URLs and therefore the backend maintainer is able to create a little application within the shop application.

You will be charged -30 for shipping

Proceed

When clicking at "Proceed" we get redirected to the payment selection page:

### **Payment**

Please select a payment option:

Pay On Delivery

After choosing "Pay On Delivery" the order is complete and we see a thank-you-page. Sweet!

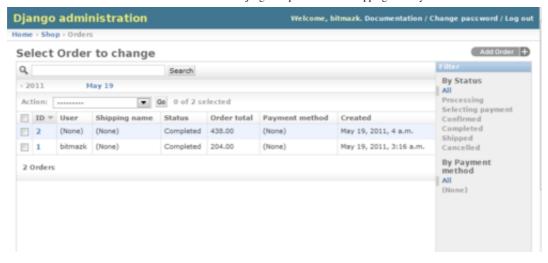
# Thank you for your order!

Your order has been placed

Back to the main page

So far setting up the shop was simple and straightforward. There still is no magic involved and once there are enough shipping backends and payment processors available, it will be relatively easy to quickly setup a new shop and start templating right away which should be awesome for agile projects.

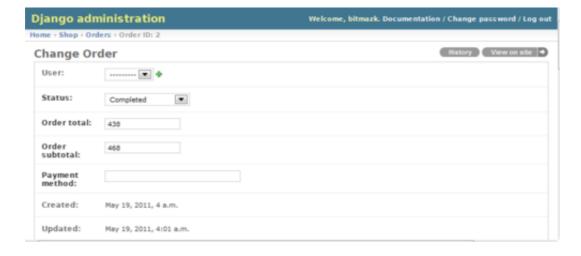
Let's have a look what we got in the admin so far:



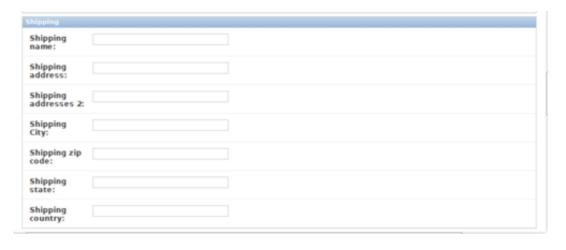
The list view for orders does show the orders in our system which is great but some things look wrong here:

- Why is "Shipping name" = "(None)"? Shouldn't it be "Flat rate"?
- Why is "Payment method" = "(None)"? Shouldn't it be "Pay on delivery"?

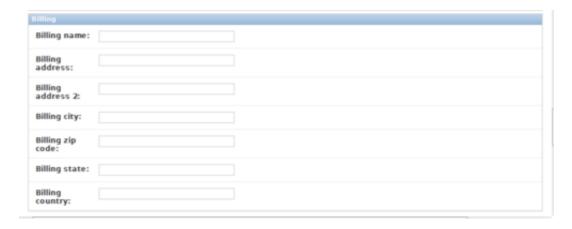
Let's have a look at a single order. First we get general information about the order. Again I wonder why "Payment method" is empty.



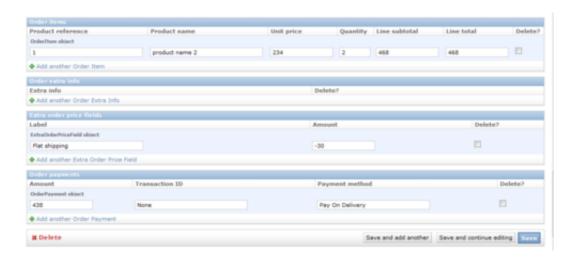
After that we have a fieldset for Shipping information. How can we gather these information from the user? I guess we would need to extend the views of our shipping backend and add some nice forms. This will probably be the next important step for setting up our shop and will be covered in the next post.



Shipping address and billing address often differ, therefore we have another address fieldset here. Again, how to fill this should be covered in my next post.



Finally we get an overview of the items ordered. Interesting is the last part about "Order payments". This seems to tell me that a payment of 438 € has been made through "Pay on delivery". However, the user has just ordered the item and certainly has not paid it already. This is due to the simplicity of the example backends that ship with the shop. We will have to invent something smarter here, soon.



So far, so good. I think in the next post I will create our first own backends and make sure that we can gather the customer's address information.

**Lacodelpaco** likes this **mbrochh** posted this

« Previous Next »

The Minimalist Theme designed by  ${\bf Pixel\ Union}\mid {\bf Powered\ by\ Tumblr}$