

[Follow mbrochh](#)

MARTIN BROCHHAUS

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

[ASK](#) [ARCHIVE](#) [RANDOM](#) [RSS](#) [SEARCH](#)

8TH JUN 2011 | 2 NOTES

DJANGO-SHOP ADVENTURES: SIMPLE CATEGORIES

In my **last post** I described how to handle shipping and payment backends in django-shop. This post was planned to deal with how to simplify the workflow so that the user does not have to chose any backends if there only exists one.

I got all the code ready and this post was saved as a draft half way when **Chris Glass** pushed a new update to **django-shop** that does exactly this. Just in a much better way than I did it :)

Great! Additionally they got rid of the client model, as it turned out that most developers just want to use the User model built into Django. Because we still need to save billing and shipping addresses at the User, Chris added an **addressmodel** which resides in its own app. This ensures easy extendability: If you need an address model with more complex fields, you can just create your own app. Just make sure that your model has two foreign keys to the User model named `user_shipping` and `_user_billing`.

Although these were some rather big changes to the shop design, all we need to do to get our django-shop-adventures code back on track is to add `shop.addressmodel` to our `INSTALLED_APPS`. I take this as a good sign that the highly modular implementation of django-shop was a good design decision.

Adding django-shop-simplecategories

In my last posts I added my own implementation of categories to my product model. However Chris has done a great job on this (very common) use case: **django-shop-simplecategories**.

The instructions found in the Github repository leave no room for interpretation and so I was able to replace my own category system by this better one in just a couple of minutes. All we need to do is really add 'shop_simplecategories' to `INSTALLED_APPS` and add the 'ProductWithCategoryForm' to the product's admin. The latter is optional but I highly

recommend it as it makes categorizing of products much simpler. See it yourself in the screenshots:

As the name suggest, the shop_simplecategories app gives us a simple category model. We got a slug, we can even define relationships between categories and we can add products to a category. This of course can quickly become cumbersome as there are more and more products in your database.

The screenshot shows the Django administration interface for the 'shop_simplecategories' app. The page title is 'Django administration' with a welcome message for 'bitmazzk'. The breadcrumb trail is 'Home > Shop_simplecategories > Categories > Röhre'. The main heading is 'Change category'. The form includes fields for 'Name' (Röhre), 'Slug' (roehre), 'Parent category' (a dropdown menu), 'Ordering' (0), and 'Products' (a list box containing '[101001] LED Leuchtsystem Röhre, beidseitig'). Below the list box is a note: 'Hold down "Control", or "Command" on a Mac, to select more than one.' At the bottom, there are buttons for 'Delete', 'Save and add another', and 'Save and continue editing'.

For this reason **Pavel Zhukov** added a very cool form that can be used to enhance your own product's admin. Now you can add your product to categories immediately when you create the product itself. Sweet!

The screenshot shows a product admin form. The 'Weight' field is set to '123.0'. The 'Categories' section features two side-by-side list boxes: 'Available categories' on the left and 'Chosen categories' on the right. The 'Chosen categories' list contains 'Röhre'. Between the lists are '+' and '-' buttons. Below the 'Available categories' list is a 'Choose all' button, and below the 'Chosen categories' list is a 'Clear all' button. At the bottom, there are buttons for 'Delete', 'Save and add another', and 'Save and continue editing'.

django-simple-categories has some minor glitches, for example the slug field does not get auto populated, but this should be fixed easily. Time for submitting a patch :)

As usual the code for this project can be found on **Github**.

UPDATE: Giving credit to Pavel Zhukov



pythoner likes this



mbrochh posted this

[« Previous](#) [Next »](#)

The Minimalist Theme designed by **Pixel Union** | Powered by **Tumblr**