

This is Google's cache of <http://apress.com/book/errata/1278>. It is a snapshot of the page as it appeared on 3 Apr 2011 02:32:27 GMT. The [current page](#) could have changed in the meantime. [Learn more](#)

These search terms are highlighted: **beginning django e commerce errata**

[Full version](#)

[Go To Home](#)

- [Books](#)
- [eBookshop](#)
- [Community](#)
- [About Apress](#)
- [Support](#)

- [Member Login](#)
- [Sales & Distribution](#)
- [Source Code](#)
- [Academic Resources](#)
- [Trade Site](#)

Book Details

[Beginning Django E-Commerce book cover](#)

- By James McGaw
- ISBN13: 978-1-4302-2535-5
- ISBN10: 1-4302-2535-1
- 300 pp.
- Published Oct 2009
- Print Book Price: \$44.99
- eBook Price: \$31.49

[Buy eBook](#)

[Buy Print Book](#)

[Tweet](#)

Book Resources

- [Source Code](#)
- [Submit/View Errata](#)

My eBook

- [My Account](#)
- [Download eBook\(s\)](#)

[Home](#) [Book Catalog](#) [Beginning Django E-Commerce](#) [Errata](#)

Errata Submission

If you think that you've found an error in **Beginning Django E-Commerce**, please let us know about it. You will find any confirmed erratum [below](#), so you can check if your concern has already been addressed.

Submit **Errata** Your Name: (optional)

Your Email: (optional)

Book Title:

Beginning Django E-Commerce (978-1-4302-2535-5)

Error: (Please include chapter number, page number, and a brief description of the issue you've encountered.)

To prove you are an actual person, and not an automated script, please provide the solution to this simple arithmetic problem: $6 + 2 = ?$

Errata

Issue Author's Response

Chapter 5, page 139

```
from ecomstore.checkout import authnet # should be:
from ecomstore.checkout import authnet
```

On page 61, I run the server. But I got this message:

Request Method: GET

Request URL: http://localhost:8000/admin/

Django Version: 1.2.3

Exception Type: ImproperlyConfigured

Exception Value:

'CategoryAdmin.exclude' refers to field 'created_at' that is missing from the form.

After, googling around, I found that matter may come from:

- 'auto_now_add=True' in field 'created_at';
- 'auto_now=True' from field 'update_at'.

It seems that it is not possible to view in an admin form, not editable fields.

Indeed, I removed those variables in models.py and got possibility to log in.

Latter I changed admin.py list_display variables by:

- list_display = ('name',);
- list_display = ('name', 'price', 'old_price',).

It was possible to log in too.

Is there a solution to view those fields in administration?

Thanks

Chapter 13, page 285,

```
from ecomstore.settings import SESSION_AGE_DAYS
```

There is no SESSION_AGE_DAYS constant in settings, it should be SESSION_COOKIE_DAYS

ch.5, p.136

Should the clean_credit_card_number return a value such as self.cleaned_data['credit_card_number']?

also, where are these 2 clean_ functions called?

page 65 in book shows:

```
(r'^static/(?P<path>.*$)', 'django.views.static.serve',
```

but I think it's suppose to be:

```
(r'^static/(?P<path>.*$)', 'django.views.static.serve',
```

Chapter 4 page 100

```
in def show_cart,
```

```
cart_subtotal = cart.cart_subtotal(request)
```

should be placed in a new line.

It looks like on page 74 the /utils app is under the /templates app. I

Good catch. Thanks!

Unfortunately, I've never encountered this error. I've always heard that auto_now_add and auto_now can be problematic, but I've never encountered any problems, and I've used them both pretty extensively. Troubleshooting might require some back and forth to nail down the exact problem, so here's a link to a resource that I hope helps you with your issue.

<http://stackoverflow.com/questions/1737017/django-auto-now-and-auto-now-add>

Excellent catch. Thanks.

You're right, the first clean_ function doesn't have a return line. In this case, it being missing won't cause any errors because the form is just used to validate the data on the form, but the order processing never uses the request object to get the data (instead of a form instance.) To fix this, add this to the end of clean_credit_card_number():

```
return cc_number
or (as you proposed)
self.cleaned_data['credit_card_number']
```

The two 'clean_' functions are called when you call form.is_valid() on the CheckoutForm instance. This in turn calls a method called full_clean(), which validates the form as a whole and any individual fields.

Yes, that missing single quote at the end of the regular expression is a mistake, and it would cause a syntax error. Thank you.

Yep, that's an "oops". Thanks for submitting.

Yep, 'utils/' should be in the root directory. Nice catch.

think /utils is suppose to be under the root directory.

Chapter 2, page 28. The book says the line is suppose to be: `urlpatterns += patterns`

but I'm thinking it's a typo. it should be `urlpatterns = patterns`
ch 4 p 88 `def show_product(request, product_slug)`

should be `request`

ch 4, pg 94. `css background-color:Red;color:White;`
error message extends across the whole form.

pg 37, code should read
`(r'^catalog/$', 'preview.views.home')`,

rather what what it does read:
`(r'^catalog/$', 'views.home')`,

Chapter 2 page 30
in the `css.css` file the `#sidebar` tag has a field with two semi-colons and it should only be one.
`#sidebar{`
`width: 200px;`
`float: left;;`
`}`

Chapter 6, page 155

On this page you can read the following lines:
`(r'^accounts/', include('accounts:url'))`,
`(r'^accounts/', include('django.contrib.auth.urls'))`,

This I don't really understand. Both entries are pointing to the same url. Doesn't this mean that the second entry will never be used?

And the I take it that `'accounts:url'` is a typo and should read `'accounts.url'`

Chapter 2, page 19.

I don't know how to create the MySQL database and the book does not explain it. This is a book of WHAT and HOW, but here the HOW has been omitted. I'm on a Windows machine as most readers are, simply displaying something that was typed on a "Unix variant" doesn't help me at all.

You advised to download and install Workbench, well, how about explaining how to do it in Workbench instead of skipping it?

Chapter 4, page 85:

The `_cart_id()` function on page 85 is not the same as the one on page 90. Looks like the text on page 85 references the version on page 90.

related package: `django_db_log-1.4.0`

This is not an error on your side, but I just want to point it out that the

You're right, at this point in the text, it should be `urlpatterns += patterns()`. The `+=` operator should only be used if adding to the already-existing `urlpatterns`, as shown on page 32. But on page 28, it's a typo. Thanks!

"should be request". Good catch.

Hard to find a solution to this problem without seeing your HTML and CSS. My guess: sounds like something is amiss in the DOM. I'd check to see if your UL element is closed properly; use Firebug (<http://getfirebug.com>) to see which exact element is red. If all else fails, email me about this. Thanks!

Good catch. Thank you.

Whoops. Yep, that's a typo. Thanks.

`'accounts:url'` should be `'accounts.url'`. It is a typo.

Both of those entries are used. I opted to place the built-in **Django** auth items at `/accounts/[path]`. So the logout URL resides at `/accounts/logout/`, as one example. If we add any of our own URLs to `accounts/urls.py`, they will also appear under `/accounts/`. If we create a URL in our own `accounts/urls.py` that overrides any of the built-in **Django** URLs in the auth app, ours will come first and be used.

The MySQL Workbench is new to me. I used the MySQL Administrator under Window, but it appears to have been deprecated in favor of the Workbench. This was part of my reasoning for omitting specific instructions from the book: things change pretty quickly. In any event, I'm sorry you were left frustrated. I'll try to help.

Here are some useful links, for you or anyone else in this situation:

Download: <http://dev.mysql.com/downloads/workbench/>

Installation Instructions (Windows):
<http://dev.mysql.com/doc/refman/5.1/en/windows-installation.html>

Getting started instructions: <http://wb.mysql.com/?p=406>

In the last link, notice the SQL Editor window where you can execute raw SQL. The commands included in the text will work on Windows just as well.

If you need more specific instructions or help, please feel free to email me directly at jim@django-e-commerce.com and I'll do my best to walk you through it. Thanks!

Yep, that's a typo. The code on page 90 will still work, but to be consistent with the code example on page 85, it should read:

```
def _cart_id(request):
    if CART_ID_SESSION_KEY in request.session:
        # etc
```

Thanks very much for your report and extensive analysis. It's much appreciated. A lot of readers have written to me and expressed similar problems with the **django-db-log** app. This is by far the most serious

author of `django_db_log` might have introduced a bug in 1.4.0 that might cause your readers some grief. The author seems to assumed that everyone uses latin1 character set.

Also, since he moved the project to github, your installation instructions on p.22 are unfortunately not quite correct now. C'est la vie...

-----[START]-----

When I tried to run "python manage.py syncdb" on p.23, I got a page of error message ending up with

```
_mysql_exceptions.OperationalError: (1071, 'Specified key was too long; max key length is 1000 bytes')
```

After googling around and manually loading the tables, I found the culprits:

The MySQL command on p.19:

```
mysql> CREATE DATABASE ecomstore CHARACTER SET utf8;
```

And the line in the `djangoblog_errorbatch` table:

```
UNIQUE (logger, class_name, server_name, checksum)
```

Each UTF-8 character is 3-byte-long, and `logger + class_name + server_name + checksum = 64 + 128 + 128 + 32 = 352 characters = 1,056 bytes`. I was able to load the table after removing this line. Of course, this was not the right way to solve this, so instead I dropped the "CHARACTER SET utf8" part in the MySQL query.

After re-running "python manage.py syncdb", it went smoothly (except this annoying warning "manager.py:44: SyntaxWarning: assertion is always true, perhaps remove parentheses?") as described.

-----[END]-----

On your **Errata** web page, (<http://www.apress.com/book/errata/1278>) My **errata** has **errata**. (Woot!) Thanks for pointing it out. your **errata** for Chapter 5, Page 115 has a misspelling:

... The `GOOGLE_CHECKOUT_RUL` on page 115 of the book should be:

... The `GOOGLE_CHECKOUT_URL` on page 115 of the book should be:

May be an error...

```
On page 140 the method returned by is_empty should be
return cart_distinct_item_count(request) == 0
```

```
not
```

```
return cart_item_count(request) == 0
```

This relates to the earlier error on p89.

If I did not change it here too I got "global name 'cart_item_count' is not defined".

Chapter 3, page 59

...all fields cannot be empty unless you set `blank=False`.

I believe this should say `..blank=True`

Chapter 2, page 36

Ensuring that `DEBUG` is set to `True` is good for security,...

This should be `False`, rather than `True`

In Chapter 5 Site Checkout & Orders:
page 118 in `google_checkout.py` file, line

```
if node.tagName == 'redirect-url':
```

I kept getting this error:

(and disappointing) one.

For other readers, if you have to choose between using `django-db-log` and making your database use the UTF-8 character set by default, I'd go with the latter. Getting the character set correct is pretty important, especially for long-term planning of larger sites.

Excellent catch.

Another error. (I was having some trouble with my Boolean math when I wrote this chapter.) If you want to allow someone to leave values on a model empty when they use the admin interface, you have to make those fields with `blank=True`. By default, the blank attribute to set to `False`, so it cannot be blank.

Good catch! `DEBUG` should certainly always be set to `False` in production. `True` is definitely not good for security!

That's going to be a tough one to troubleshoot remotely. Instead of trying to find exactly that problem, I'll try to describe, in general, what I think the problem is. If you just tuned in, here are the lines:

```
root = xml_doc.documentElement
node = root.childNodes[1]
```

Exception Type: AttributeError
 Exception Value:
 Text instance has no attribute 'tagName'

Could there be some **Errata** here?

In the demo site <http://www.django-e-commerce.com>
 If you add a product to the cart than use the browser back button and then try to click 'Add to cart' button again you get a message "Cookies must be enabled."

Maybe the test cookie shouldn't be deleted or maybe **django** hold in the session the fact that the cookies are enabled and we can use it?

p57
 # Uncomment the next line to enable the admin:
 (r'^admin/(.*)', admin.site.root), # my default...
 #(r'^admin/(.*)', admin.site.urls), # the book says this...
 Chapter 5, pg 115:

Is GOOGLE_CHECKOUT_URL correct? I couldn't get this to work until trying:

```
https://sandbox.google.com/checkout/api/checkout/v2/request/Merchant
+ GOOGLE_CHECKOUT_MERCHANT_ID
```

on page 89, in the definition of show_cart, should it be "cart_item_count = cart.distinct_item_count(request)"?

in reference to the last two lines on page 90.

In page 91, should the import be "from ecomstore.catalog.forms import ProductAddToCartForm" instead of import from cart app, since in page 87, it says 'go back to catalog ap and find forms.py'.

cart.py (pg. 90)

```
cart_id += random.choice(characters) is probably better than cart_id
+= characters[random.randint(0, len(characters)-1)]
```

also,

```
if cart_item.product.id = p.id:
```

should be

```
if cart_item.product.id == p.id:
```

On Chapter 1 - Page 9, the author shows how to include PyDev Extensions into Eclipse in order to write Python/**Django** code easier.

This information is inaccurate. PyDev Extensions doesn't exist anymore. It has been merged with PyDev, which has been open sourced, and is located at <http://pydev.org/>.

```
if node.tagName == 'redirect-url':
...etc...
```

It's expecting the variable 'node' to be the second XML node in the document, and it's finding some text instead. Not sure why. What I would do, faced with this, is to return the XML document to a calling function and display it somewhere so you can inspect it.

Thanks for sharing. Sorry I couldn't be of more help!

That's an interesting problem! I don't believe there's any way of getting around it except for the suggestion you made: don't delete the test cookie. It's probably a "best practice" kind of thing to do so, but not strictly required, especially if you expect to use the test cookie elsewhere on the site.

For anyone just tuning in, this problem is in the show_product() view on pages 91-92, and in order to correct the error, you just need to comment out (or remove) these two lines in the POST section:

```
if request.session.test_cookie_worked():
request.session.delete_test_cookie()
```

Thanks!

Good catch. On page 57, the default text on the line that you need to uncomment in the middle of the urlpatterns tuple to enable the admin interface does not end with "urls", it ends with "root".

Yep, that's a mistake in the text. I checked the source code and it matches what you submitted. The GOOGLE_CHECKOUT_RUL on page 115 of the book should be:

```
https://sandbox.google.com/checkout/api/checkout/v2/request/Merchant
+ GOOGLE_CHECKOUT_MERCHANT_ID
```

There is a 'checkout' between the domain name and the 'api' part of the URL. Thanks for submitting.

Yes, inside of show_cart, it should call cart_distinct_item_count:

```
def show_cart(request ...):
cart_item_count = cart.cart_distinct_item_count(request)
```

Thanks for catching.

Excellent catch. In the catalog/views.py file, it should be importing from the catalog app and not the cart app:

```
from ecomstore.catalog.forms import ProductAddToCartForm
```

Thanks for sharing.

That's a really good application of the random.choice() function, thanks for offering.

You're right, inside of the add_to_cart() function, the "==" operator should be used here:

```
for cart_item in cart_products:
if cart_item.product.id == p.id:
...etc...
```

Very true, the installation of PyDev has changed quite a bit since I wrote that chapter, and the step-by-step instructions tend to vary depending on the version of Eclipse. I would encourage readers to Google around for instructions about installing.

Aptana is a great suggestion as well, I used for Rails development and found it to be a great free-of-charge IDE. I was unaware of PyDev

PyDev is also installed as part of Aptana, which I would recommend since it is free, supports **Django** & Python, and also has support for HTML, CSS, and Javascript; all of which will likely be used in any **Django** project.

Chapter 2, Page 46

Under MIDDLEWARE_CLASSES

I think.djangoblog.DBLogMiddleware should be
.djangoblog.middleware.DBLogMiddleware

Page 12 & 13 of chapter 4 dont quite read right.

```
if request.session.get('cart_id',) == "":
    request.session['cart_id'] = _generate_cart_id()
return request.session['cart_id']
```

is there a typo here?

I keep getting syntax errors

being part of Aptana, I will look into it.

Thanks!

Actually, both will work. If you look in the `__init__.py` file in the `djangoblog` folder, it imports the middleware classes. However, since I wrote that, the README file has been updated and it now says to add `"djangoblog.middleware.DBLogMiddleware"`.

Since that's what David Cramer is encouraging, it's probably a better option. Good catch!

The syntax here is confusing. There are a couple of ways to clean this up. In the final version of the book, we define the session key for the cart ID at the top of the `cart.py` file, below the import statements but below any code, as follows:

```
CART_ID_SESSION_KEY = 'cart_id'
```

You can then use this in your code instead. Also, since the `request.session` object can be treated like a Python dictionary (in some ways), you can use the following syntax to check for its presence in session:

```
if CART_ID_SESSION_KEY in request.session:
    request.session[CART_ID_SESSION_KEY] = _generate_cart_id()
return request.session[CART_ID_SESSION_KEY]
```

Hope this helps.

[Site Map](#) | [Privacy Policy](#) | [Contact Apress](#)