WordPress to Django Users

Cinco de Mayo, 2014 - Python Web Houston

https://github.com/jmoswalt/wp-to-django-users

The Problem

WordPress Users -> Django Users

Move without disruption

The Steps to Convert

1. Export the data from WordPress
2. Import the data into Django
3. Update staff and superusers
4. Install python requirements, settings
5. Update passwords with Management Command

1. Export the data from WordPress

```
SELECT id,
  user_login as username,
  user_pass as password,
  display_name as first_name,
  user_registered as date_joined,
  "1" as is_active,
  user_registered as last_login,
  user_email as email,
  "0" as is_staff, "0" as is_superuser
FROM wp_users
```

2. Import the data into Django

```
mysql –u username –p  dbname < path/to/
export.sql
```

Use the right tool for your data backend.

3. Update staff and superusers

```
UPDATE auth_user SET
    is_staff = True,
    is_superuser = True
WHERE id in (1, 2, 3, 26, 553)
```

Take note of the user id's of your WP admins. Might also be able to use `wp_user_level` from `wp_usermeta`

4. Install python requirements, settings

requirements.txt:

```
pip install django-hashers-passlib==0.1
```

settings.py:

```
PASSWORD_HASHERS = (
    ...
    'hashers_passlib.phpass',
)
```

https://github.com/mathiasertl/django-hashers-passlib

5. Update passwords with Management Command

```
from django.contrib.auth.hashers import get_hasher
hasher = get_hasher('phpass')
user.password = hasher.from_orig(user.password)
```

Old:

```
   `$P$B8Wa4IPrveTlsVAIPhT5WIot8qfc67/`
```

New:

```
   `phpass$$P$B8Wa4IPrveTlsVAIPhT5WIot8qfc67/`
```

Stop Here

If you are no longer using WordPress, then you are done!

If you still use WP or other apps in your service, keep listening

Replacing a custom WP user API

Previously had an XML API to read user data out of WordPress from other apps.

Now needed to recreate that in Django.

Also useful to replace built-in WP user APIs

Replacement Steps

1. Set an apex domain cookie
2. Create a View to display user info
3. Return XML
4. Read XML from other apps

1. Set an apex domain cookie

settings.py:

```
SESSION_COOKIE_DOMAIN = “.example.com"

SESSION_COOKIE_NAME = “mysite_sessionid"
```

Note the leading ‘.’   Cookie Name optional

2. Add a URL and View for user info

```
cookie_value = kwargs['cookie_value']
try:
   session = Session.objects.get(
      session_key=cookie_value
   )
   user = User.objects.get(
      id=session.get_decoded()[‘_auth_user_id']
   )
except Session.DoesNotExist:
   user = None

context[‘requested_user’] = user
```

3. Return XML

```
<?xml version="1.0" encoding="UTF-8"?>
  <user
    status="{{ requested_user.status }}"
    username="{{ requested_user.username }}"
    email="{{ requested_user.email }}”
  />
```

4. Read XML from other apps

Pick the best method for your language to read and digest the new XML api in order to validate your users across your domain.