

# Valid use case for django admin?

Asked 15 years, 10 months ago   Modified 15 years, 10 months ago

Viewed 3k times



11



I want to build a django site where a certain group of trusted users can edit their profile information. Does it make sense to have each trusted user go through the django admin interface? I'd only want them to be able to see and edit their own information (obviously). It doesn't seem like this fits the way the django people define "trust", especially the bolded bit...

From [The Django Book, Chapter 18](#):

The admin is designed to be used by people who you, the developer, trust. This doesn't just mean "people who have been authenticated;" it means that Django assumes that your content editors can be trusted to do the right thing.

This means that there's no "approval" process for editing content — if you trust your users, nobody needs to approve of their edits. It also means that the permission system, while powerful, has no support for limiting access on a per-object basis. **If you trust someone to edit their own stories, you trust them not to edit anyone else's without permission.**

Is this one of those use cases that fits with django's admin module, or is it just a specialized view for a non-trusted user?

python

django

django-admin

Share

Improve this question

Follow

edited Jan 31, 2009 at 17:14



Fabian Steeg

45.6k ● 7 ● 87 ● 113

asked Jan 31, 2009 at 3:56



Jason

6 Answers

Sorted by:

Highest score (default)



17

No, the Django admin is not suited for individual user profiles, each user would be able to see, and edit, all other user profiles. This is suited more to an administrator who has to manage all the users at once.



What you need to build is a user profile page. Django already has a nice login system courtesy of the `django.contrib.auth` module. You can easily integrate this into your pages, and its exactly what the Django admin uses to authenticate users.

Next you'll have to build a simple page that exposes that specific user's profile information based on their User

model. This should be relatively painless as it will only require one view and one template, and the template can take advantage of ModelForms.

Share Improve this answer

answered Jan 31, 2009 at 4:48

Follow



Soviut

91.4k ● 53 ● 205 ● 279

- 
- 2 It is possible to restrict the admin so you could only edit your own profile; but I wouldn't recommend it for this use case. +1  
– [Carl Meyer](#) Jan 31, 2009 at 14:54
- 



5

I would suggest you to create a Person model which contains a OneToOneField to the User model(Admin site User model.). Some what like this..



```
from django.contrib.auth.models import User

class Person(models.Model):
    """The person class FKs to the User class and
    information
    including userImage, country, etc"""

    user = models.OneToOneField(User, related_name=
    url = models.URLField(max_length=255, blank=Tr
    country = models.CharField(max_length=2, blank=
    state = models.CharField(max_length=50, blank=
    zipCode = models.IntegerField(max_length=7, bl
    userImage = models.ImageField(upload_to=genera
    null=True)
```

Share Improve this answer

answered Jan 31, 2009 at 18:40

Follow



aatifh

2,337 ● 4 ● 27 ● 31



Django's model for authorization is a little too simplistic. It is just checks for permission on the Model as a whole.

3



For this kind of thing, you are pretty much forced to write your own view functions that handle the additional check.



After you've written one or two, you'll see the pattern.



Then you can think about writing your own decorator to handle this.

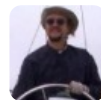
```
def profileView( request, object_id ):
    p= Profile.objects.get( id=int(object_id) )
    if request.session['user'] != p.get_user():
        # respond with a 401 not authorized or a helpf
    # process normally, since the session['user'] is t
```

For the above to work, you'll need to enable sessions, and be sure you record the user in the session when they login successfully. You'll also need to eradicate the session when they logout.

Share Improve this answer

answered Jan 31, 2009 at 16:50

Follow



[S.Lott](#)

391k ● 82 ● 517 ● 788

---

Writing your own views is one option, and often a good one. But you aren't "forced" to. Django's ModelAdmin classes have method hooks (such as "has\_change\_permission") that allow you to modify the admin's behavior in pretty major ways. – [Carl Meyer](#) Feb 1, 2009 at 19:15

---

@Carl Meyer: has\_x\_permission is model-wide. You need to also validate the ownership of each object, which isn't part of standard decorators. Easy to add, but still an extension.  
– [S.Lott](#) Feb 1, 2009 at 22:48

---



3

I wouldn't consider editing my personal profile on a website an administrative task. I think [django-profiles](#) is what you are looking for.



Share Improve this answer  
Follow



edited Feb 5, 2009 at 7:37



[Soviut](#)

91.4k ● 53 ● 205 ● 279

answered Jan 31, 2009 at 4:43



[Jonas](#)

19.6k ● 10 ● 57 ● 67



1

Just for reference, here's a [snippet](#) demonstrating how you can fairly easily achieve this effect (users can only edit their "own" objects) in the Django admin. Caveat: I wouldn't recommend doing this for user profiles, it'll be easier and more flexible to just create your own edit view using a `ModelForm`.



Share Improve this answer  
Follow

answered Feb 1, 2009 at 19:13



[Carl Meyer](#)

126k ● 20 ● 109 ● 117



1



There are a few django pluggable apps that allow row level permissions on your admin model. However, I'd be more inclined to write my own view that allows users to do it from within the application.

I had similar aspirations (using the admin contrib) when designing the app I'm currently working on, but decided that the admin app really is for admin use, and that regular users should be given their own pages to do the work and customisation (if required).

You can easily generate the CRUD views for a particular model using generic views and modelforms, and just apply style sheets for consistent look with the rest of your application.

Share Improve this answer

Follow

answered Feb 4, 2009 at 12:15



[Josh Smeaton](#)

48.7k ● 24 ● 135 ● 165