

Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free, no registration required.

[Tell me more](#)


Best way to do register a user in Django

CAREERS 2.0
by stackoverflow



Easily apply for your dream job
No formatting needed!

I'm trying to implement User Registration for my Django app. The book I read mentions `UserCreationForm` but I need more than just name, password, and email address. I could implement my own user object and use `ModelForm` I think, but then I lose out on some of the django conveniences for authentication. Finally, I read something about `UserProfiles` which supplement `Users` I guess, so I think I need some combination of all of these things. Here's what I have so far.

View - this is simple enough. All I want to do here is create my form and save the user

```
def main(request):
    rform1 = forms.RegisterForm1()
    rform2 = forms.RegisterForm2()
    return render_to_response("authentication/index.html", {'form1': rform1, 'form2': rform2})

def register(request):
    if request.method == 'POST':
        rform1 = forms.RegisterForm1(request.POST)
        rform2 = forms.RegisterForm2(request.POST)
        if rform1.is_valid() and rform2.is_valid():
            new_user = rform1.save()
            return HttpResponseRedirect("/register-success/")
    return render_to_response("authentication/index.html", {'form1': rform1, 'form2': rform2})
```

Form - this is my form for creating a user. Or the start of it at least

```
class RegisterForm1(forms.ModelForm):
    class Meta:
        model = User

class RegisterForm2(forms.ModelForm):
    class Meta:
        model = UserProfile
```

Model - this is my `UserProfile` that I want to supplement `User` with.

```
class UserProfile(models.Model):
    user = models.OneToOneField(User)
    phonenumber = PhoneNumberField()
```

Is it clear what I'm trying to do? Am I on the right track

[django](#) [django-models](#) [django-forms](#)

edited Aug 19 '10 at 16:19

asked Aug 19 '10 at 16:05



JPC

1,788 1 20 64

3 Answers

Make *two* model forms, one for the `User` model and one for the `UserProfile` model.

```
class UserForm(django.forms.ModelForm):
    class Meta:
        model = User
```

```
class UserProfileForm(django.forms.ModelForm):
    class Meta:
        model = UserProfile
        exclude = ['user']
```

Give them prefixes and display them together in the same HTML form element. In the view, check that both are valid before you save the User and then the UserProfile.

```
def register(request):
    if request.method == 'POST':
        uf = UserForm(request.POST, prefix='user')
        upf = UserProfileForm(request.POST, prefix='userprofile')
        if uf.is_valid() and upf.is_valid():
            user = uf.save()
            userprofile = upf.save(commit=False)
            userprofile.user = user
            userprofile.save()
            return django.http.HttpResponseRedirect(...something...)
        else:
            uf = UserForm(prefix='user')
            upf = UserProfileForm(prefix='userprofile')
    return django.shortcuts.render_to_response('register.html',
                                              dict(userform=uf,
                                                  userprofileform=upf),
                                              context_instance=django.template.Re
```

In register.html :

```
<form method="POST" action="">
    {% csrf_token %}
    {{userform}}
    {{userprofileform}}
    <input type="submit">
</form>
```

edited Aug 19 '10 at 16:50

answered Aug 19 '10 at 16:11



Vebjorn Ljosa

4,803 2 27 51

I edited my code. I think I'm getting closer. Thanks! – JPC Aug 19 '10 at 16:21

CAREERS 2.0
by stackoverflow



Put all that writing to good use!
Update your profile today!

Your best bet is to use James Bennett's [django-registration](#) and [django-profiles](#) apps - they take care of a lot of this sort of thing for you.

answered Aug 19 '10 at 16:14



Daniel Roseman

148k 8 140 208

True, but the general problem of having a form that represents two models is something one encounters also outside the context of registration. – Vebjorn Ljosa Aug 19 '10 at 16:18

I stumbled upon those at some point, but never looked into them. I'll look at it. – JPC Aug 19 '10 at 16:20

I was just looking into this, and found out that Django has a very simple built-in new user creation form, which you can use if you want simple registration -- there's no option to, eg, save an email address and validate it, so it won't work for the OP's use case, but am posting it here in case other come across this question as I did, and are looking for something simple.

If you just want people to be able to choose a username and password, then **the built-ins will do everything for you**, including validation and saving the new user:

```
from django.views.generic.edit import CreateView
from django.contrib.auth.forms import UserCreationForm

urlpatterns = patterns('',
    url('^register/', CreateView.as_view(
        template_name='register.html',
        form_class=UserCreationForm,
        success_url='/'
    )),
    url('^accounts/', include('django.contrib.auth.urls')),

    # rest of your URLs as normal
)
```

So the only thing you have to do yourself is create *register.html*...

More info here: <http://www.obeythetestinggoat.com/using-the-built-in-views-and-forms-for-new-user-registration-in-django.html>

answered Jun 9 at 1:13



hwjp

2,161 1 12 24

Not the answer you're looking for? Browse other questions tagged [django](#)

[django-models](#) [django-forms](#) or [ask your own question](#).