

**By Vitor Freitas**

I'm a passionate software developer and researcher from Brazil, currently living in Finland. I write about Python, Django and Web Development on a weekly basis.

[Read more.](#)



Adobe Creative Cloud for Teams starting at \$29.99 per month.

ads via Carbon

**Advertise  
Here**

Reach 90k+ Django developers every month. Learn more about sponsorship of the blog.

Sponsored

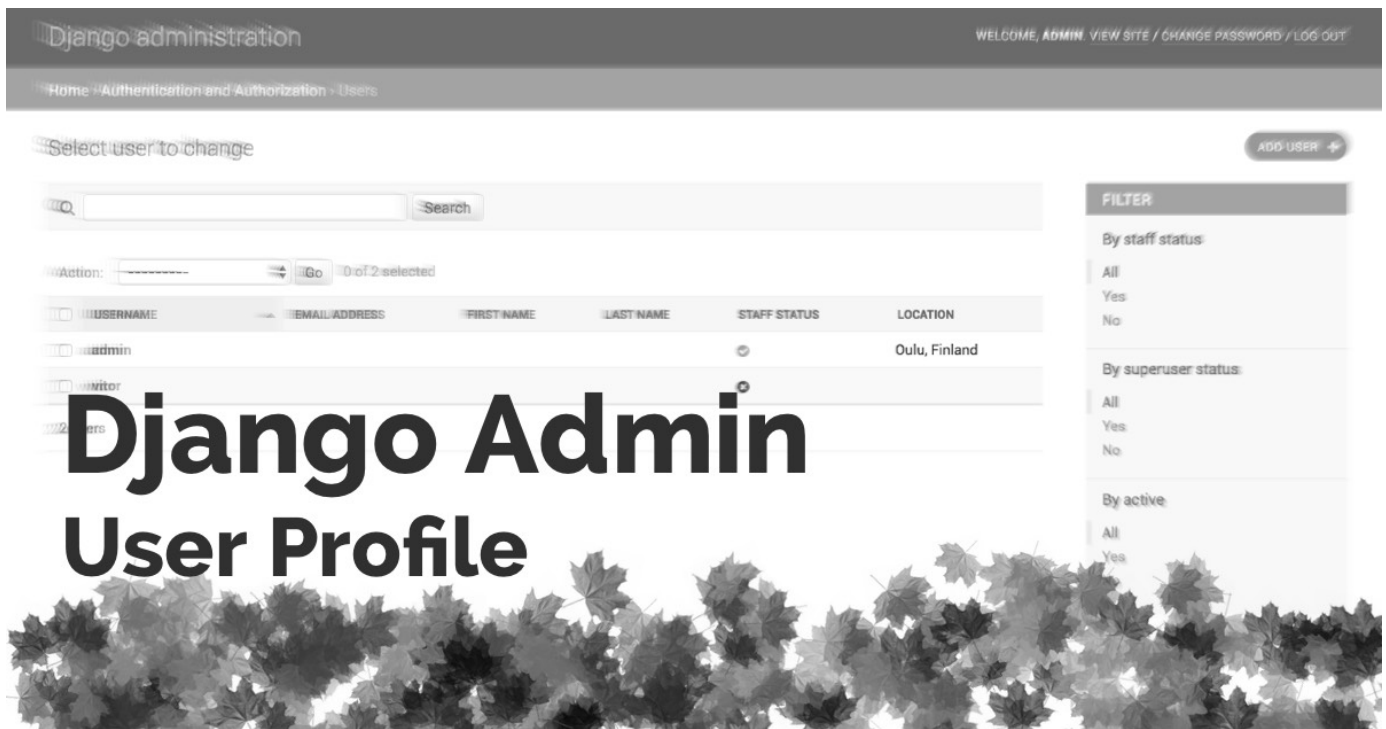
---

## TUTORIAL

---

# How to Add User Profile To Django Admin

📅 Nov 23, 2016 ⌚ 6 minutes read 💬 57 comments 👁 41,610 views



There are several ways to extend the the default Django User model. Perhaps one of the most common way (and also less intrusive) is to extend the User model using a one-to-one link. This strategy is also known as User Profile. One of the challenges of this particular strategy, if you are using Django Admin, is how to display the profile data in the User edit page. And that's what this tutorial is about.

## Background

I've published a while ago an article about [How to Extend Django User Model](#), describing in great detail all the different strategies. If you are still not familiar with it, I strongly suggest that you have a look in this article.

This tutorial is about the User Profile strategy. So, consider we have an app named **core** with the following model definition:

### models.py

```
from django.contrib.auth.models import User
from django.db import models
from django.db.models.signals import post_save
from django.dispatch import receiver

class Profile(models.Model):
```

```

STUDENT = 1
TEACHER = 2
SUPERVISOR = 3
ROLE_CHOICES = (
    (STUDENT, 'Student'),
    (TEACHER, 'Teacher'),
    (SUPERVISOR, 'Supervisor'),
)
user = models.OneToOneField(User, on_delete=models.CASCADE)
location = models.CharField(max_length=30, blank=True)
birthdate = models.DateField(null=True, blank=True)
role = models.PositiveIntegerField(choices=ROLE_CHOICES, null=True, blank=True)

def __str__(self): # __unicode__ for Python 2
    return self.user.username

@receiver(post_save, sender=User)
def create_or_update_user_profile(sender, instance, created, **kwargs):
    if created:
        Profile.objects.create(user=instance)
    instance.profile.save()

```

What we want to achieve is making the fields **location**, **birthdate** and **role** available to be edited on Django Admin.

## Editing The User Profile

The trick is very simple. We will extend the default `UserAdmin`, add the profile instance as an **inline** and switch the `UserAdmin` Django uses.

### admin.py

```

from django.contrib import admin
from django.contrib.auth.admin import UserAdmin
from django.contrib.auth.models import User

from .models import Profile

class ProfileInline(admin.StackedInline):
    model = Profile
    can_delete = False
    verbose_name_plural = 'Profile'

```

```

fk_name = 'user'

class CustomUserAdmin(UserAdmin):
    inlines = (ProfileInline, )

    def get_inline_instances(self, request, obj=None):
        if not obj:
            return list()
        return super(CustomUserAdmin, self).get_inline_instances(request, obj)

admin.site.unregister(User)
admin.site.register(User, CustomUserAdmin)

```

A few things here: I intentionally defined the **verbose\_name\_plural** as **Profile**, because each user can have only one profile, and as the component we are using is an `StackedInline` (which normally is used for formsets), just to avoid displaying **Profiles** in plural when we only have one profile. It's just a minor thing.

Then a very important thing, we need to override the `get_inline_instances` method, so to display the inlines *only* in the edit form. Otherwise we might get some problems because of how the Signals work. Remember that the Signal is responsible for creating the Profile instance.

Since by default Django only provides **username** and **password** upon user creation, this way we keep the default behavior.

Finally we **unregister** the old User admin and register the new one, the custom with the Profile model.

It will look like this:

Change user | Django site admin x

127.0.0.1:8000/admin/auth/user/1/change/

Vitor

Time: 14:12:09 Now | ⌚

Note: You are 2 hours ahead of server time.

Date joined: Date: 2016-11-23 Today | 📅

Time: 13:34:57 Now | ⌚

Note: You are 2 hours ahead of server time.

PROFILE

Profile: admin

Location: Oulu, Finland

Birthdate: 2016-11-23 Today | 📅

Note: You are 2 hours ahead of server time.

Role: Teacher ▼

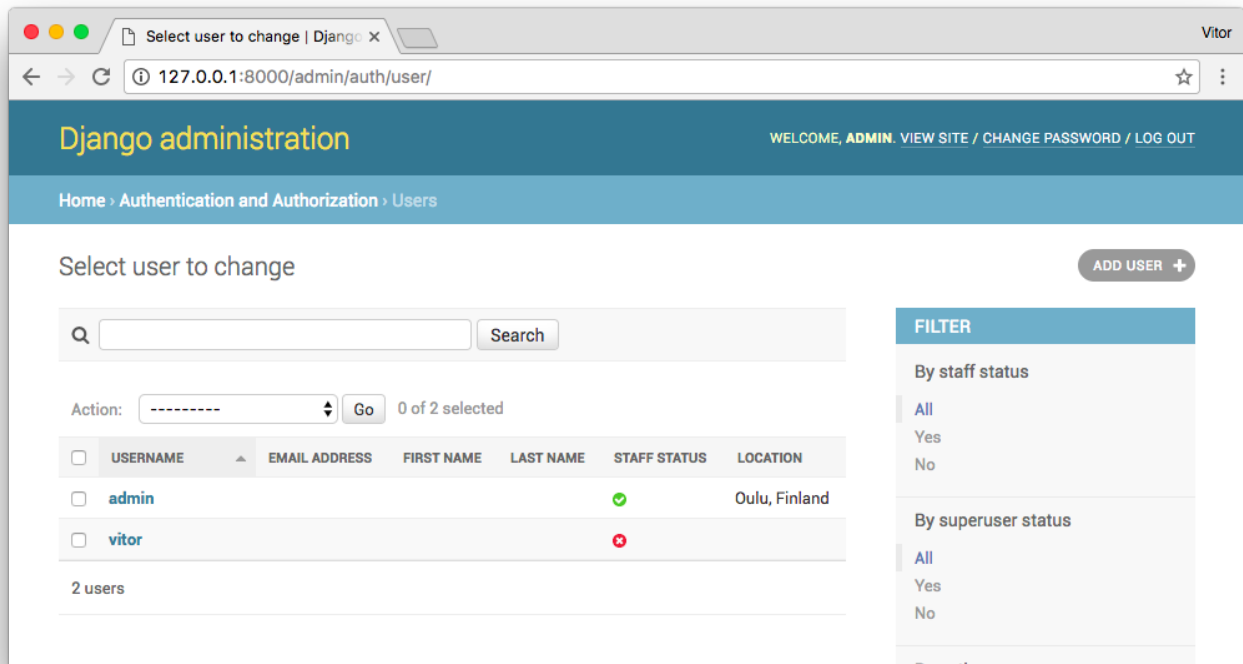
Delete Save and add another Save and continue editing SAVE

Cool, right? It just works.

---

## Adding Profile Fields to List View

Now what if we want to display the User location (which is defined inside the profile model), in the list view, like this:



Since we are already extending the `UserAdmin`, it's just a matter of overriding the `list_display` attribute.

```
class CustomUserAdmin(UserAdmin):
    inlines = (ProfileInline, )
    list_display = ('username', 'email', 'first_name', 'last_name', 'is_staff', 'get_location')
    list_select_related = ('profile', )

    def get_location(self, instance):
        return instance.profile.location
    get_location.short_description = 'Location'

    def get_inline_instances(self, request, obj=None):
        if not obj:
            return list()
        return super(CustomUserAdmin, self).get_inline_instances(request, obj)
```

We have to define an extra method, `get_location` and append it to the `list_display`, because the field is defined in an external model (the relationship is defined in the Profile model, not in the User model). So for the Django Admin to understand how to display the location attribute, we have to play it this way.

The `get_location.short_description` is just to display it prettier in the table header. Otherwise Django Admin would display it as “Get Location” (which is the name of the method).

A very important detail: note the `list_select_related`. We are adding the **profile** relationship there, so to avoid firing hundreds of unnecessary queries in the database. For more on that subject, check this post: [Django Tips #3 Optimize Database Queries](#).

---

## Conclusions

That's it! Plain simple. This also open an opportunity to explore more the User model in Django Admin. You can override several attributes and very easily add new list actions for example.

Django's documentation is awesome. But even better is to explore its the source code. For example, to understand more on the definitions of the `UserAdmin`, don't be afraid to explore the source code and see the class' attributes and methods: [django.contrib.auth.admin](https://docs.djangoproject.com/en/1.11/contrib/auth/admin/).

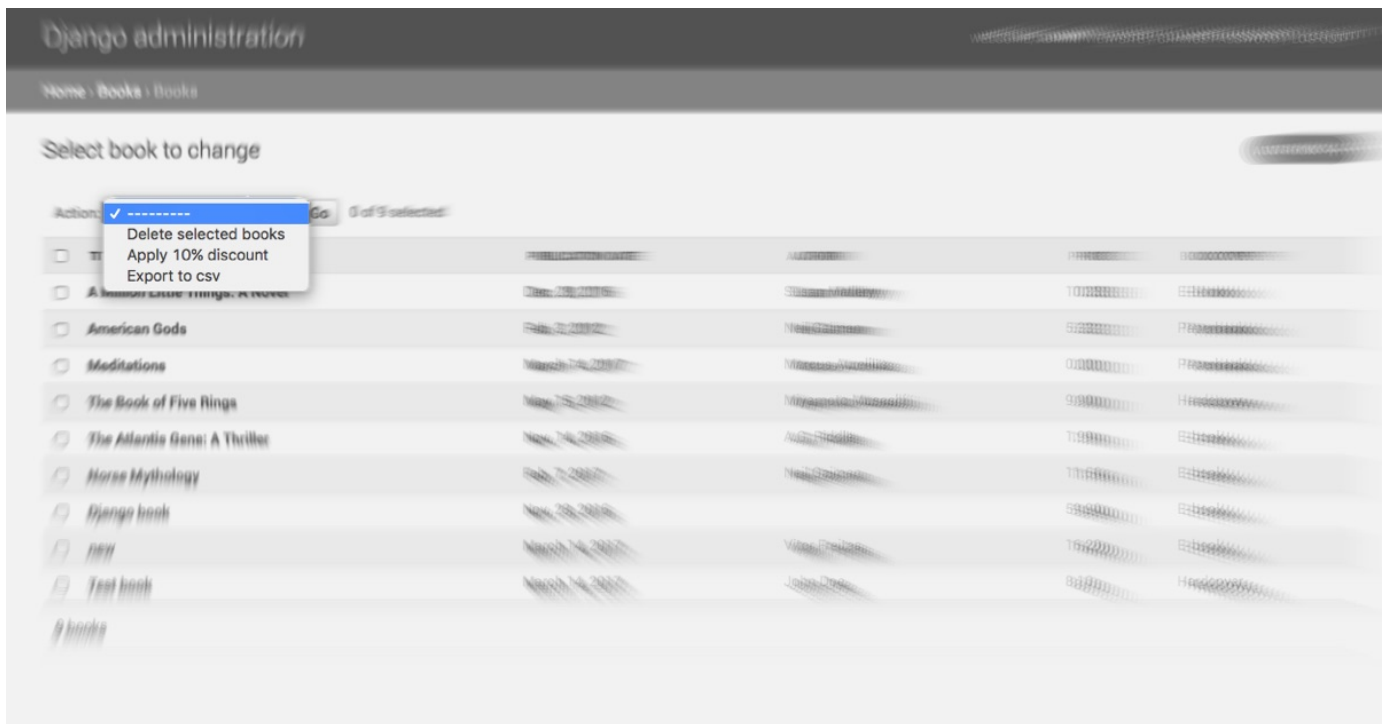
Also the source code I used in this tutorial is available on GitHub: [sibtc/django-admin-user-profile](#).

---

## Related Posts

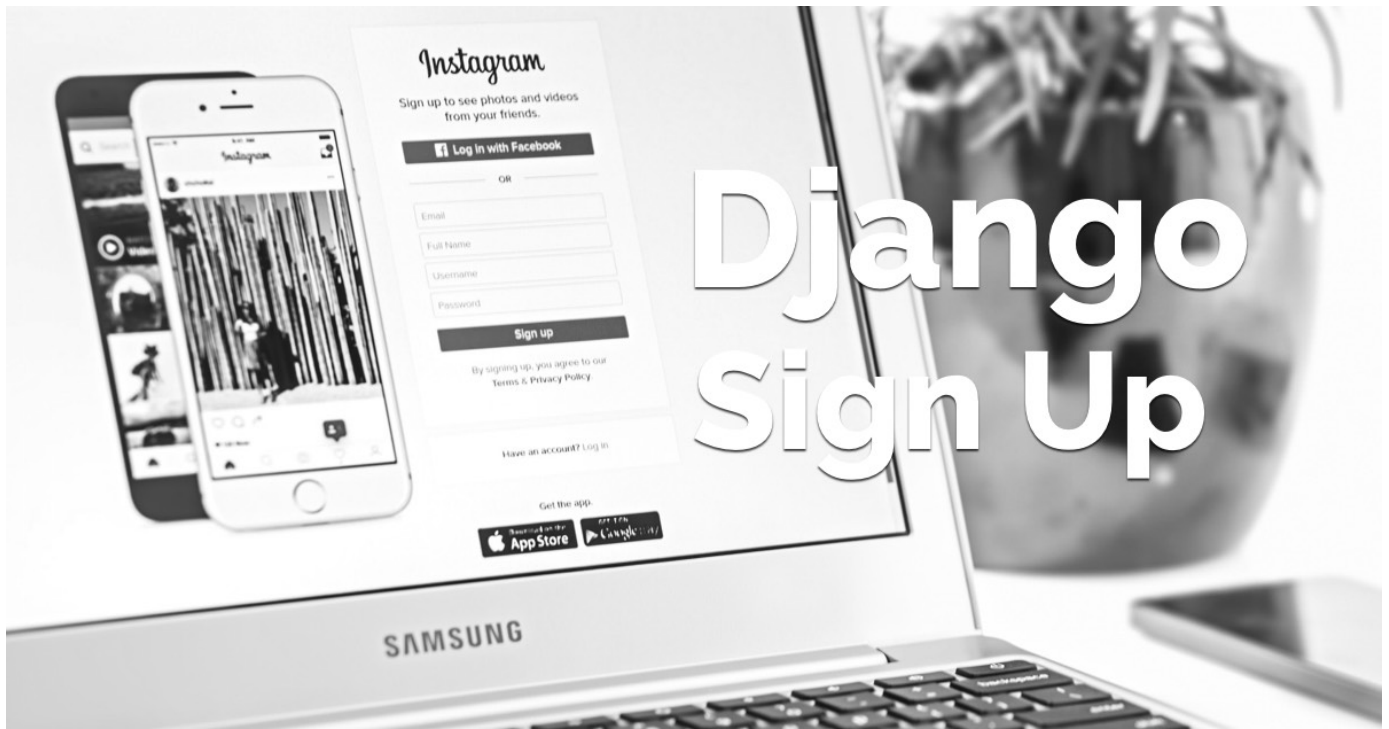


## [How to Implement Multiple User Types with Django](#)



## [How to Create Django Admin List Actions](#)





[How to Create User Sign Up View](#)


[django](#) [django admin](#) [user](#)

Like 13

331

Share this post





Search and buy domains  
from Namecheap. Lowest prices.

Buy Now

Comments

Community

 Login ▾

 Recommend 6

 Tweet

 Share

Sort by Best ▾

Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS 

Name



**Natiq Vahabov** • 2 years ago

and I am following your code, after editing `admin.py` it gives error. I am using Oracle db

DatabaseError at /admin/auth/user/2/change/  
ORA-00904: "PROFILES\_PROFILE"."USER\_ID": invalid identifier

27 ^ | v • Reply • Share ›



**Vitor Freitas** Mod → Natiq Vahabov • 2 years ago

Hi Natiq,

in the Profile model, do you have a field named `user`, like this:

```
user = models.OneToOneField(User, on_delete=models.CASCADE)
```

The relation is a `OneToOneField` or is it a `ForeignKey`?

In the `admin.py`, did you define the `fk_name`?

Another possible problem, are you sure that every User instance have its own Profile instance? You can check it by running the following query in via command line:

```
python manage.py shell
from django.contrib.auth.models import User
User.objects.filter(profile__isnull=True).count()
```

If the count returns a number greater than 0, it means there are User instances without Profile instance, which might be causing the exception...

For now that's what came to my mind.. also there might be a case that the Profile model wasn't created in the database.. did you run `makemigrations` & `migrate`?

^ | v • Reply • Share ›



**Pulkit Kumar** • 2 years ago

You seriously have a great blog!

4 ^ | v • Reply • Share ›



**Daniel** • 9 months ago

Concept is explained very clearly. Keep posting more of these!

What should be done if I have different types of Profiles for various types of Users

Can I have a Profile for a User and a Profile for a Group?

for eg. User can be Teacher and Student. I want different fields to be stored for Teacher and Student

1 ^ | v • Reply • Share ›



**Alejandro Hurtado** • a year ago

I would like for example when a user registers, his profile is generated but at the same time that the mail of opening of his account is activated ..! Thank you

1 ^ | v • Reply • Share ›



**Ibo** • a year ago

If we use "Add User" of django admin, the extended fields will not show up and in case that we have made some of them required, the in the admin site practically Add User will not function correctly. To solve this problem we should make the fields from the profile class visible in "Add User" page of admin site, how can we do that?

1 ^ | v • Reply • Share ›



**Aldo Vazquez** ➔ Ibo • 17 days ago

Oh, someone else noticed too

^ | v • Reply • Share ›



**Aldo Vazquez** • 17 days ago

Hi Vitor!

How do we add the profile form to " /admin/auth/user/add/" ? otherwise it throws RelatedObjectDoesNotExist when trying to create a new user on the admin site

^ | v • Reply • Share ›



**Annie Thiessen** • 2 months ago

Love the blog! You have saved me so many headaches.

Would you recommend using User or Profile as the FK in other models. For example, in class Payment(models.Model), would it be better to go with user = models.ForeignKey(get\_user\_model()) or user = models.ForeignKey(Profile)

^ | v • Reply • Share ›



**aqui\_c** • 5 months ago

Why do you use get\_location and then set the name, instead of defining location as a property of the class? Wouldn't it be quicker and more transparent?

^ | v • Reply • Share ›



**许向** • 5 months ago

not working on django 2.0 ...

^ | v • Reply • Share ›



[instance.supplier.save\(\)](#)

I do have a similar profile for distributor that has a foreign key to supplier. Could this be the problem?

^ | v • Reply • Share ›



**Prithviraj Mitra** • a year ago

Great article and it worked. Many thanks. I have one query. The profile stack appears at the very end of the page that means after 'Permissions' and 'Important dates'. Is it possible to bring up the 'Profile' before those 2 stacks.

^ | v • Reply • Share ›



**Heriyansyah Mirsuma** • a year ago

i have tried this code but there is an error message,.  
create\_or\_update\_user\_profile() missing 1 required positional argument: 'selfsender'

^ | v • Reply • Share ›



**Heriyansyah Mirsuma** → Heriyansyah Mirsuma  
• a year ago

this happen affter i update the user information insde  
django default admin backend

^ | v • Reply • Share ›



**gamesbook** • a year ago

Hi Vitor - great article, thanks.

My input is about how to add filter options for the user's profile in the list display. For example, your `get\_location` method allows the location field from the user's profile to be displayed in the list. But to be able to add this location to the list filter, I eventually found I had to use 'profile\_\_location'. Not sure if you want to add this information to your tutorial?

^ | v • Reply • Share ›



Avatar This comment was deleted.



**gamesbook** → Guest • a year ago

Thanks. But I don't understand your response.

^ | v • Reply • Share ›



Avatar This comment was deleted.



**gamesbook** → Guest • a year ago

Thanks, but I don't want to add a  
dependency to another Django app. and  
that is not what my comment was

related to.

^ | v • Reply • Share ›



**Omar McClellan** • 2 years ago

How about doing this in [views.py](#), is it possible ?

^ | v • Reply • Share ›



**Neeraj Gupta** • 2 years ago

Nothing else has worked so beautifully than this. Good job Vitor.

^ | v • Reply • Share ›



**Erick Delfin** • 2 years ago

Dude, I've lurked around many of your blog posts from here and they're always so damn helpful, understandable, and the best part is that it works and it goes along with 'good practices'. Sorry for my language, but I just had to say it. Rock on.

^ | v • Reply • Share ›



**Francisco Marotta** • 2 years ago

Hi Vitor, greetings from Argentina!

Is it possible to make one of the profile fields mandatory? If so, how can I add it to the add user form?

Thanks in advance,

Francisco.

^ | v • Reply • Share ›



**Namju Yoon** • 2 years ago

Hi Vitor!

I am new to Django and I really appreciated to your effort. When I follow your guide, I encountered the following:

SystemCheckError: System check identified some issues:

ERRORS:

```
<class 'photo.admin.userprofileinline'>: (admin.E202)
'photo.UserProfile' has no ForeignKey to 'auth.User'.
photo.UserProfile.user: (fields.E300) Field defines a relation
with model 'User', which is either not installed, or is abstract.
```

please give me a help.

Thanks.

^ | v • Reply • Share ›



**Vitor Freitas** Mod ➔ Namju Yoon • 2 years ago

Hi Namju,

Can you show me the class `photo.UserProfile`?  
From what the error code says, it seems like the  
`UserProfile` class doesn't have a `ForeignKey` or `OneToOne`  
relationship to the default Django User model  
(`auth.User`)  
^ | v • Reply • Share ›



**Daniel Wise** • 2 years ago

Hi Vitor.

First of all, great work on your website!

I am new to Django and i've followed your tutorial about  
extending the user model.

I used the `OneToOneField` solution and everything seems to  
be working perfectly. But I  
am having trouble with my template's form fields.

I use Django-Registration app for handling registration, and  
for that (I believe) the form fields are set up on the template's  
file with `{{ form.username }}` instead of `{{ user.username }}`.

The problem that I'm having is that when I try to add the form  
field in the template for the user to fill out its name, for  
example, writing `{{ user.profile.name }}` doesn't return on the  
template a field. In fact, it renders nothing but the standard  
fields (username, password1, password2 and email).

I have already tried other things, such as `{{ form.profile.name }}`  
or just `{{ profile.name }}` or `{{ myappname.profile.name }}`  
but none of those things seems to work.

Could you give me a hand about that?

P.S: Sorry about my english. It's a bit rusty. :D

Also, don't know if that was the best session to post this. Sorry  
if it wasn't.

^ | v • Reply • Share ›



**Carsten Holst** • 2 years ago

Hi Vitor.

Thanks for your amazing site. It is a big help for a  
Python/Django newbie as myself! :-)

I have encountered a problem with the solution you propose  
above.

My model looks like this:

```
class Profile(User):  
    user = models.OneToOneField(User,
```

```
on_delete=models.CASCADE)
house = models.ForeignKey(House,
on_delete=models.CASCADE, null=True)
```

and my `admin.py` looks like this:

```
class ProfileInline(admin.StackedInline):
    model = Profile
```

[see more](#)

^ | v • Reply • Share ›



**Vitor Freitas** Mod ➔ Carsten Holst • 2 years ago

Hi Carsten!

Thanks a lot for your comment! :-)

The problem with your Profile model is because you are extending the User model:

```
class Profile(User):
```

Instead, you should create the Profile model from `models.Model`:

```
class Profile(models.Model):
```

Happy holidays!!

Vitor

^ | v • Reply • Share ›



**Carsten Holst** ➔ Vitor Freitas • 2 years ago

Hi Vitor!

Doh! I've should have noticed that... :-)

Thank you very much! :-)

Carsten

^ | v • Reply • Share ›



**Federico** • 2 years ago

Hi Vitor. It's a nice post! Well done!

As you said "Since by default Django only provides username and password upon user creation, this way we keep the default behavior."

When I click on add User button it displays a simple view(1) that only provides username and password. Then if I hit on save a second view(2) is displayed with more data (email, first\_name, etc) and userprofile data too.

I want to display my userprofile form in the first view (1). If I remove method "get\_inline\_instances" I partially achieve that



need.

I have not nullable fields inside my userprofile.

The problem is that:

1 - If I hit on save, I get redirected to my second view (2).

While it's supposed that the form must show me errors.

Something similar happens here but in the second view:

<http://stackoverflow.com/qu...>

2 - If I fill useprofile fields and then hit on save button I get an error "UNIQUE constraint failed: core\_userprofile.user\_id".

Could you provide me some help?

Thanks,

^ | v • Reply • Share ›



**Vitor Freitas** Mod ➔ Federico • 2 years ago

Hi Federico,

If you intend to manage the Users and Profiles directly via Django Admin, adding the Profile data upon User creation, I would suggest you do remove the signals and keep the save/create the profile instances manually outside Django admin.

Like this:

### models

Remove the signals

```
#@receiver(post_save, sender=User)
#def create_or_update_user_profile(sender, instance, created, **kwargs):
#    if created:
#        Profile.objects.create(user=instance)
#    instance.profile.save()
```

Then on the **admin** module:

```
class ProfileInline(admin.StackedInline):
    model = Profile
    can_delete = False
    verbose_name_plural = 'Profile'
    fk_name = 'user'
    extra = 1
    max_num = 1
    min_num = 1

class CustomUserAdmin(UserAdmin):
    inlines = (ProfileInline, )
    list_display = ('username', 'email', 'first_name', 'last_name')
    list_select_related = ('profile', )

    def get_location(self, instance):
        return instance.profile.location
    get_location.short_description = 'Location'

admin.site.unregister(User)
admin.site.register(User, CustomUserAdmin)
```

^ | v • Reply • Share ›

^ | v • Reply • Share ›



**Alejandro Hurtado** → Vitor Freitas • a year ago

ooo very Good extra fields in django1.11.

^ | v • Reply • Share ›



**Federico** → Vitor Freitas • 2 years ago

Okay,  
thanks again Vitor!

^ | v • Reply • Share ›



**BareTech** • 2 years ago

please how do i display this user profile and make it editable  
to the end user (member)

^ | v • Reply • Share ›



**Vitor Freitas** Mod → BareTech • 2 years ago

Hi there! Sorry for the late response  
You want to edit the user Profile out of the Django  
Admin? If that's what you mean, you can simply create  
forms for the user and for the profile instance, and  
process it together in the view. You can find some  
insights here:

<https://simpleisbetterthanc...>

^ | v • Reply • Share ›



**BareTech** → Vitor Freitas • 2 years ago

Thanks Vitor, i read that earlier but somehow  
didn't quite get it. i got it figured out now that i  
studied it again. I'm new to web development  
and your tutorials have really helped. I would  
like it if you could find some time to do a few  
tutorials on how to control template content  
based on user data (something like " if user is

---

## ✈ Subscribe to our Mailing List

Receive updates from the Blog!