

Make User email unique django

Asked 1 year, 2 months ago Active yesterday Viewed 3k times



How do I make a Django User email unique when a user is signing up please?

3

forms.py



```
class SignUpForm(UserCreationForm):
    email = forms.EmailField(required=True)

    class Meta:
        model = User
        fields = ("username", "email", "password1", "password2")

    def save(self, commit=True):
        user = super(SignUpForm, self).save(commit=False)
        user.email = self.cleaned_data["email"]
        if commit:
            user.save()
        return user
```



3



Im using the from django.contrib.auth.models User. Do I need to override the User in the model. Currently the model doesnt make a reference to User.

views.py

```
class SignUp(generic.CreateView):
    form_class = SignUpForm
    success_url = reverse_lazy('login')
    template_name = 'signup.html'
```

Thanks for your thoughts.

django

edited Nov 24 '18 at 19:01



[ruddra](#)

25.9k 5 37 64

asked Nov 24 '18 at 18:58



[mogoli](#)

791 2 10 29

3 Answers



The best answer is to use `CustomUser` by subclassing the `AbstractUser` and put the unique email address there. For example:

12

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



```
class CustomUser(AbstractUser):
    email = models.EmailField(unique=True)
```



and update the settings with `AUTH_USER_MODEL="app.CustomUser"` .

But if its not necessary for you to store the unique email in Database or maybe not use it as username field, then you can update the form's `clean` method to put a validation. For example:

```
class YourForm(UserCreationForm):

    def clean(self):
        email = self.cleaned_data.get('email')
        if User.objects.filter(email=email).exists():
            raise ValidationError("Email exists")
        return self.cleaned_data
```

Update

If you are in mid project, then you can follow the [documentation](#) on how to change migration, in short which is to:

1. Backup you DB
2. Create a custom user model identical to `auth.User`, call it `User` (so many-to-many tables keep the same name) and set `db_table='auth_user'` (so it uses the same table)
3. Delete all Migrations File(except for `__init__.py`)
4. Delete all entry from table `django_migrations`
5. Create all migrations file using `python manage.py makemigrations`
6. Run fake migrations by `python manage.py migrate --fake`
7. Unset `db_table` , make other changes to the custom model, generate migrations, apply them

But if you are just starting, then delete the DB and migrations files in migration directory except for `__init__.py` . Then create a new DB, create new set of migrations by `python manage.py makemigrations` and apply migrations by `python manage.py migrate` .

And for references in other models, you can reference them to `settings.AUTH_USER_MODEL` to avoid any future problems. For example:

```
user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.DO_NOTHING)
```

It will automatically reference to the current User Model.

edited Jul 24 '19 at 6:57

answered Nov 24 '18 at 19:52



ruddra

25.9k 5 37 64

So when I try to add the CustomUser I get an error as there are other models that use the `django.contrib.auth.User`. Trying to figure out what i need to do to extend the User model. Do I need to change all references from User to CustomUser? – [mogoli](#) Nov 25 '18 at 1:07

@mogoli See my comment above. You need to remove db, migrations, etc. and then start the project over. – [joshlsullivan](#) Nov 25 '18 at 2:18
