Django do Zero

Como Criar um Aplicativo Web

*Canal **youtube.com/@VamosCodar** - Wanderson Reis*

08 - Novos Campos no Registro de Usuários

Requisitos

1) Aula 01 - Introdução a Frameworks Web e Django: https://youtu.be/LpFNMn6Uw5s
2) Aula 02 - Configurações do Ambiente: https://youtu.be/7zxJi3_HRuw
3) Aula 03 - Primeira Aplicação - Olá Django: https://youtu.be/5MoZZttT6UA
4) Aula 04 - Introdução aos Templates: https://youtu.be/nYnZ-J8PrxA
5) Aula 05 - Introdução aos Models e ORM: https://youtu.be/-0PnRa29iuU
6) Aula 06 - Registro e Login de Usuários: https://youtu.be/fhtML8L9HMc
7) Aula 07 - Mensagens Flash: https://youtu.be/Gh-mVCyPHUc

Introdução

O model User padrão do Django fornece o mínimo para gerenciamento de usuários, contudo diversas vezes é necessário personalizar campos / informações sobre o usuário. Existem pelo menos três estratégias para implementar informações adicionais ao cadastro do usuário, vamos utilizar a abordagem de criar um novo model User personalizado com novos campos.

Nesta aula:

1) Reverter as migrações dos modelos de dados
2) Criar um Model User personalizado
3) Adaptar o form de registro para coletar os novos campos

Passo 1: Reverter as migrações dos modelos de dados

Para estender do model **User** padrão da forma vamos aplicar, envolve refazer migração dos models do zero para se ajustar ao novo model **User** personalizado.

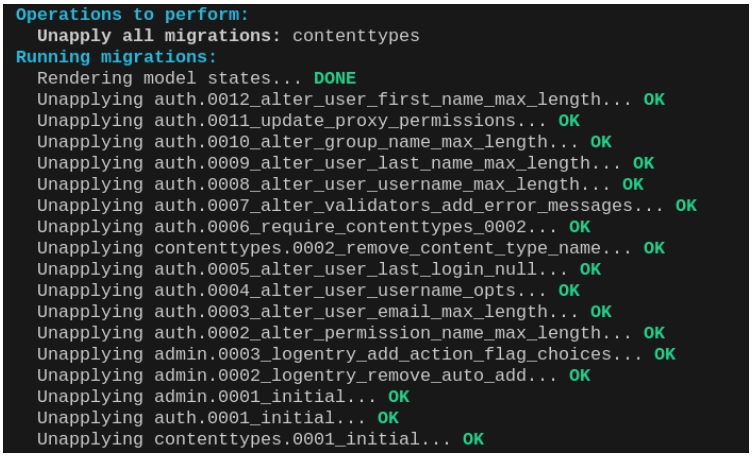
No Django a camada que faz a gestão dos dados usando migrations usa versionamento para todos os models. Por isso existe um encadeamento de dependência entre eles. Desta forma conseguimos reverter de forma prática os principais models antes de aplicar a nossa própria alteração.

Vamos reverter todos os models de app contenttypes (base do Django) e assim reverter automaticamente todas as dependências relacionadas a ele. **É um processo destrutivo que exclui as tabelas do banco de dados.**

Com o venv ativado execute o seguinte comando no terminal:

python manage.py migrate contenttypes zero (ENTER)

Como resultado vai exibir as operações de reversão de cada app e suas versões (Unapplying…), semelhante a imagem abaixo.



Passo 2: Criar um Model User personalizado

A criação do Model User de fato não será totalmente personalizada, pois vamos herdar do Model User padrão do Django. Isso significa que os campos pré-definidos se mantêm e toda a estrutura para a gestão dos objetos User. É uma abordagem mais simples, onde apenas se adiciona os novos campos desejados.

Editar o arquivo **accounts/models.py** que passa a ter o seguinte conteúdo (com dois novos campos de exemplo).

```
# (...manter tudo o que já existe…)
from django.contrib.auth.models import AbstractUser

class CustomUser(AbstractUser): # herda o model User base padrão do Django
    data_nascimento = models.DateField("Data de Nascimento", null=True, blank=True)
    cpf = models.CharField("CPF", max_length=11, null=True, blank=True)
```

(**salve o arquivo**).

Agora precisamos indicar ao Django que este novo model **CustomUser** será o nosso model padrão para gestão de usuário.

Editar o arquivo **django-do-zero/webapp/settings.py** e incluir a seguinte chave de configuração:

```
# na parte inferior do arquivo
AUTH_USER_MODEL = 'accounts.CustomUser'
```

Finalizamos este passo gerando a migração do novo model e aplicando tudo novamente que será criado no banco de dados.

Com o venv ativado execute os seguintes comandos no terminal:

python manage.py makemigrations (ENTER)
python manage.py migrate (ENTER)

Passo 3: Adaptar o form de registro para coletar os novos campos

Editar o arquivo **django-do-zero/accounts/forms.py** e acrescentar as alterações a seguir (**atenção apenas ao conteúdo novo**):

```
# (...manter tudo o que já existe…)

class AccountSignupForm(forms.ModelForm):
    # (...manter tudo o que já existe…)

    class Meta:
        model = User
        fields = ('username',  'email', 'data_nascimento',  'cpf',  'password', )
        widgets = { # data personalizada a nível de formulário para exibição
            'data_nascimento': forms.DateInput(
                attrs={'type': 'date', 'required': 'required'}
            ),
        }
```

(**salve o arquivo**).

Finalizada a personalização dos campos do form para o novo model **CustomUser**, execute o servidor de desenvolvimento e acesse a rota de registro de usuários para testar.

python manage.py runserver (ENTER)

No navegador acesse http://127.0.0.1:8000/accounts/signup e experimente criar novos usuários com o campo novo.

Observação: A partir de agora os objetos do tipo User no sistema já disponibilizam os novos atributos (campos) do usuário. Em qualquer contexto que existir um objeto **user** você poderá usar/acessar as informações da seguinte forma: **user.data_nascimento e user.cpf**