



Django Framework do Zero

Como Criar um Aplicativo Web

Canal [youtube.com/@VamosCodar](https://www.youtube.com/@VamosCodar) - Wanderson Reis

25 - Importação de Dados por Upload

Requisitos

- 1) Aulas 01 a 24 - Django Framework do Zero - Playlist:

<https://www.youtube.com/playlist?list=PLFOqHo8oljzwcT23HCxJV0xWO451CTJe>

Introdução

Nesta aula vamos aprender a importar dados através do upload de um arquivo CSV. Serão geradas novas perguntas conforme o arquivo enviado.

Nesta aula:

- 1) Criar um form para envio do arquivo de dados CSV
- 2) Implementar a view para exibir o formulário
- 3) Criar um template para renderizar o formulário
- 4) Definir uma nova rota para a página de importação

Observação: Lembre-se de estar com o ambiente virtual ativado (venv).

Passo 1: Criar um form para envio do arquivo de dados CSV

Editar o arquivo **django-do-zero/polls/forms.py** e modifique conforme indicado na imagem abaixo:

```
polls > forms.py > ...
# importa a classe form do django
from django import forms
from django.core.files.temp import NamedTemporaryFile

# import Question de models.py
from polls.models import Question

# cria o ModelForm
class QuestionForm(forms.ModelForm): ...

class QuestionImportForm(forms.Form):
    import_file = forms.FileField(label='Arquivo para Importação')

    def clean(self):
        cleaned_data = super().clean()
        import_data = cleaned_data.get("import_file")

        if not self.has_error('import_file'):

            # Cria um arquivo temporário
            csv_temp_file = NamedTemporaryFile(delete=False)

            # Salva o arquivo no disco
            with csv_temp_file as uploaded_file:
                for chunk in import_data.chunks():
                    uploaded_file.write(chunk)

            cleaned_data['tmp_file_name'] = csv_temp_file.name

        return cleaned_data
```

(salve o arquivo).

Passo 2: Implementar a view para exibir o formulário

Editar o arquivo `django-do-zero/accounts/views.py` e modifique conforme indicado nas próximas imagens:

Na parte superior do arquivo

```
polls > views.py > ...
import csv
from datetime import datetime

from django.shortcuts import render, redirect, get_object_or_404, Http404
from django.views.generic.edit import CreateView, UpdateView, DeleteView
from django.urls import reverse_lazy
from django.contrib.auth.mixins import LoginRequiredMixin
from django.views.generic import DetailView, ListView, TemplateView, FormView
from django.core.exceptions import ValidationError

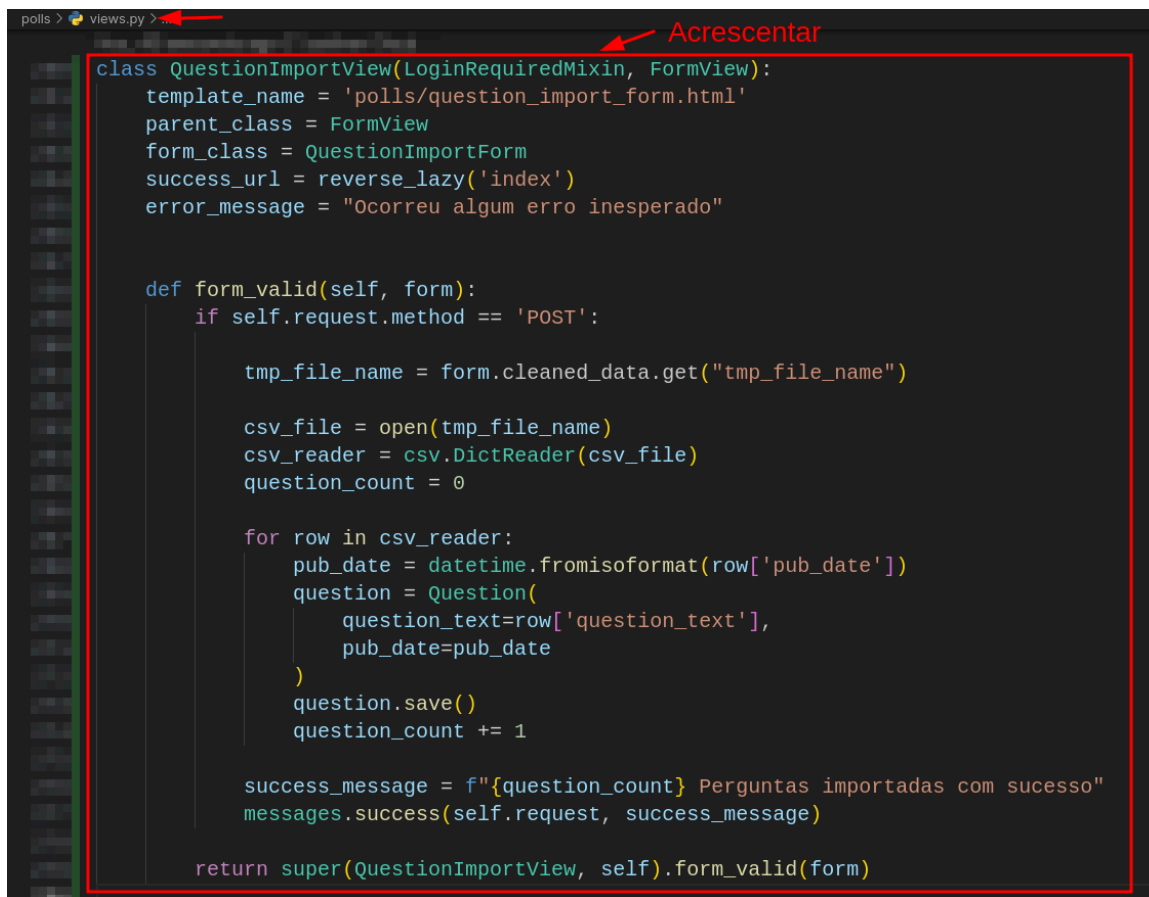
from django.contrib.auth import get_user_model
User = get_user_model()

# Create your views here.

from django.http import HttpResponse
from django.contrib.auth.decorators import login_required
from django.contrib import messages

from polls.models import Question, Choice
from polls.forms import QuestionForm, QuestionImportForm
```

Acrescentar a nova view, na parte inferior do arquivo



```
polls > views.py > ...
class QuestionImportView(LoginRequiredMixin, FormView):
    template_name = 'polls/question_import_form.html'
    parent_class = FormView
    form_class = QuestionImportForm
    success_url = reverse_lazy('index')
    error_message = "Ocorreu algum erro inesperado"

    def form_valid(self, form):
        if self.request.method == 'POST':

            tmp_file_name = form.cleaned_data.get("tmp_file_name")

            csv_file = open(tmp_file_name)
            csv_reader = csv.DictReader(csv_file)
            question_count = 0

            for row in csv_reader:
                pub_date = datetime.fromisoformat(row['pub_date'])
                question = Question(
                    question_text=row['question_text'],
                    pub_date=pub_date
                )
                question.save()
                question_count += 1

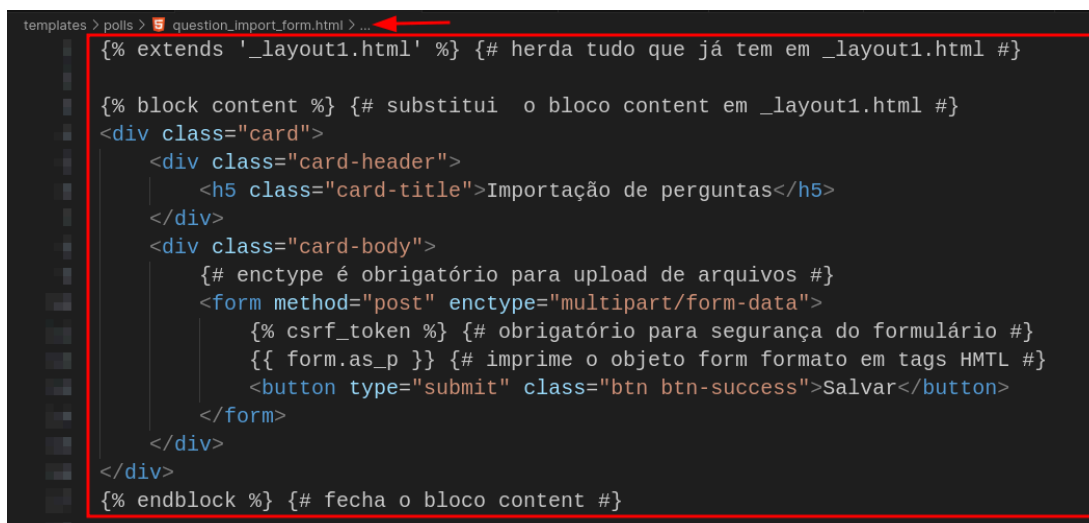
            success_message = f"{question_count} Perguntas importadas com sucesso"
            messages.success(self.request, success_message)

        return super(QuestionImportView, self).form_valid(form)
```

(salve o arquivo).

Passo 3: Criar um template para renderizar o formulário

Criar o novo arquivo **question_import_form.html** que deve estar dentro de **django-do-zero/templates/polls** conforme indicado na imagem:



```
templates > polls > question_import_form.html > ...
{% extends '_layout1.html' %} {# herda tudo que já tem em _layout1.html #}

{% block content %} {# substitui o bloco content em _layout1.html #}
<div class="card">
    <div class="card-header">
        <h5 class="card-title">Importação de perguntas</h5>
    </div>
    <div class="card-body">
        {# enctype é obrigatório para upload de arquivos #}
        <form method="post" enctype="multipart/form-data">
            {% csrf_token %} {# obrigatório para segurança do formulário #}
            {{ form.as_p }} {# imprime o objeto form formato em tags HTML #}
            <button type="submit" class="btn btn-success">Salvar</button>
        </form>
    </div>
</div>
{% endblock %} {# fecha o bloco content #}
```

(salve o arquivo).

Passo 4: Definir uma nova rota para a página de importação

Editar o arquivo **django-do-zero/polls/urls.py** e acrescente a nova rota conforme indicado na imagem abaixo:

```
polls > urls.py > ...  
    name="choice_edit"  
),  
path('enquete/<int:pk>/alternativa/delete',  
     views.ChoiceDeleteView.as_view(),  
     name="choice_delete"  
),  
path('enquete/import',  
     views.QuestionImportView.as_view(),  
     name="question_import"  
),  
]  
]
```

(salve o arquivo).

Para complementar vamos editar o template `django-do-zero/templates/polls/question_list.html` e acrescentar o link para a nova rota que acessa o formulário de importação:

```
<div class="card">  
    <div class="card-header">  
        <h5 class="card-title">Listagem de Perguntas</h5>  
    </div>  
    <div class="card-body">  
        <a href="{% url 'poll_add' %}" class="btn btn-primary mb-3">  
            Cadastrar nova pergunta  
  
            &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
        <a href="{% url 'question_import' %}" class="btn btn-primary mb-3">  
            Importar perguntas  
        </a>  
    </div>  
</div>
```

{% for question in questions %}

-

 {{ question.question_text }}

Em seguida suba o servidor de desenvolvimento e acesse a nova rota.

```
python manage.py runserver (ENTER)
```

Para testar, execute o servidor de desenvolvimento, acesse a rota <http://127.0.0.1:8000/enquete/list> e tente realizar a importação de um arquivo. Lembre-se que o formato do arquivo é o mesmo da Aula 18.