



Django do Zero

Como Criar um Aplicativo Web

Canal youtube.com/@VamosCodar - Wanderson Reis

07 - Flash Messages

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>

Introdução

Durante a interação de um usuário com o sistema é recomendado manter uma comunicação clara sobre o status das ações executadas. Todas as operações, incluindo criar, editar, excluir registros podem gerar resultados diversos (por exemplo: sucesso ou fracasso). Por este motivo, implementamos as mensagens de alerta (ou flash messages). Como exemplo vamos considerar a view `index()` para mostrar uma mensagem de aviso permanente.

Nesta aula:

- 1) Configurar as classes CSS para as mensagens
- 2) Incluir o suporte padrão de mensagens na view
- 3) Incluir a exibição das mensagens em nosso template

Passo 1: Configurar as classes CSS para as mensagens

Como estamos usando o Bootstrap em nossos exemplos vamos utilizar classes CSS próprias para formatação de mensagens. Edite o arquivo `django-do-zero/webapp/settings.py` e acrescente o conteúdo abaixo (**atenção apenas ao conteúdo novo**):

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

```
# na parte superior do arquivo
```

```
import os
```

```
from django.contrib.messages import constants as messages
```

```
# na parte inferior do arquivo
```

```
# define os nomes das classes CSS a utilizar
```

```
MESSAGE_TAGS = {  
    messages.DEBUG: 'secondary',  
    messages.INFO: 'info',
```

```
    messages.SUCCESS: 'success',
    messages.WARNING: 'warning',
    messages.ERROR: 'danger',
}
```

(salve o arquivo).

Passo 2: Incluir o suporte padrão de mensagens na view

Edite o arquivo `django-do-zero/polls/views.py` e modifique a view baseada em classe da seguinte forma (**atenção apenas ao conteúdo novo e as modificações**):

```
# (...manter tudo o que já existe...)
# na parte superior do arquivo
from django.contrib import messages

# modificar a view abaixo
def index(request):
    aviso = 'Aviso importante: esta página não exige login.'
    messages.waring(request, aviso)
    return render(request, 'index.html', {'titulo': 'Últimas Enquetes'})
```

(salve o arquivo).

DICA: Para incluir mensagens de sucesso nas views genéricas baseada em classes siga o mesmo exemplo do **CreateView** como fizemos no App accounts para a view **AccountCreateView**,

Outras tipos de mensagens que podem ser utilizadas conforme a necessidade:

`messages.success()`, `messages.debug()`, `messages.info()`, `messages.error()` ou `messages.warning()`

Passo 3: Incluir a exibição das mensagens em nosso template

É necessário editar o template `_messages.html` dentro da pasta de `templates/` com o seguinte conteúdo. (**atenção apenas ao conteúdo novo**):

```
{# habilita a exibição de todas as mensagens quando necessário #}
{% for message in messages %}
    <div class="container-fluid p-0">
        <div class="alert alert-{{ message.tags }} alert-dismissible" role="alert">
            <button type="button" class="btn-close" data-bs-dismiss="alert" arial-label="Close">
            </button>
            {{ message }}
        </div>
    </div>
{% endfor %}
```

(salve o arquivo).

Agora vamos fazer com que todos os nossos templates utilizem (por herança) o template `_messages.html`. Neste caso o melhor é incluir o template de mensagens diretamente no template `_layout1.html`.

Edite o arquivo `django-do-zero/templates/_layout1.html` (localizado conforme o seu ambiente) e acrescente o conteúdo abaixo (**atenção apenas ao conteúdo novo**):

```
{% block main %}
<div class="container">
    {% include '_messages.html' %}
{% block content %}
```

Em seguida suba o servidor de desenvolvimento e acesse a rota que exibirá o paginador.

`python manage.py runserver` (ENTER)

No navegador acesse <http://127.0.0.1:8000/index> e verifique se a mensagem de alerta foi exibida corretamente.

Outra rota que gera uma flash message é a de cadastro de usuário acesse <http://127.0.0.1:8000/accounts/signup> e crie um novo cadastro para testar.