

Elias Santos  
Aula: Session

---

## O que é uma sessão (session):

Uma sessão é um mecanismo fundamental em aplicações web que permite armazenar informações temporárias do usuário entre várias requisições HTTP. Em outras palavras, é uma maneira de manter o estado do usuário durante sua interação com a aplicação, mesmo quando ele navega entre diferentes páginas ou realiza ações.

## A importância da sessão:

**Persistência de dados do usuário:** As sessões permitem que você armazene informações específicas do usuário, como preferências, carrinhos de compras, dados de login e outras variáveis que precisam ser mantidas enquanto o usuário está ativo no site.

**Autenticação e segurança:** As sessões são frequentemente usadas para rastrear a autenticação do usuário. Ao manter um token de sessão seguro, você pode garantir que apenas usuários autenticados tenham acesso a determinadas partes do seu aplicativo.

**Personalização da experiência do usuário:** As sessões podem ser usadas para personalizar a experiência do usuário com base em suas interações anteriores. Por exemplo, você pode exibir recomendações com base no histórico de navegação do usuário.

**Rastreamento de atividades:** As sessões ajudam a rastrear a atividade do usuário, permitindo análises de comportamento e aprimoramento contínuo da aplicação com base nas preferências e ações dos usuários.

## Habilitando as Sessões

As sessões foram ativadas automaticamente quando criamos o esqueleto do site.

### PYTHON

```
INSTALLED_APPS = [
    ...
    'django.contrib.sessions',
    ....

MIDDLEWARE = [
    ...
    'django.contrib.sessions.middleware.SessionMiddleware',
    ....
```

A configuração é feita nas seções `INSTALLED_APPS` e `MIDDLEWARE` do arquivo `(projeto/projeto/settings.py)`, exibidas ao lado:

## Usando Sessões

### Exemplo 1:

Neste exemplo, vamos criar um formulário para receber o nome do usuário enviar para uma view, esta view irá armazenar o nome do usuário em uma requisição de sessão e mostrar em outro arquivo .html  
⇒ Não iremos inserir no menu, usaremos somente como um modelo de exemplo para entender a session.  
⇒ Aproveitamos o app home para criar nossas interfaces

Vamos aos passos:

1. Crie em **templates/home** o arquivo **sessao.html** com o seguinte conteúdo:

```
{% extends 'global/base.html' %}

{% block formulario %}

<div class="container">
    <h1>Usuário</h1>
    <form action="#" method="post">
        {% csrf_token %} <!-- para evitar ataques -->
        <label for="usuario">Usuário</label>
        <input type="text" class="form-control" name="usuario">
        <br>
        <button class="btn btn-primary">Enviar</button>
        <a href="{% url 'exibir_valor' %}" class="btn btn-primary">Valor da
Seção</a>
    </form>
</div>
</body>
</html>

{% endblock formulario %}
```

2. Crie em **templates/home** o arquivo **exibir\_valor.html** com o seguinte conteúdo:

```
{% extends 'global/base.html' %}

{% block info %}
    <div class="container">
        <h1>Valor da Sessão</h1>
        <p>O valor da sessão é: {{ login }}</p>
        <a href="{% url 'sessao' %}" class="btn btn-primary">Voltar</a>
    </div>
{% endblock info %}
</body></html>
```

3. Vamos duas funções em view para carregar esses arquivos. O código está a seguir:

```
def sessao(request):  
  
    if request.method == 'POST':  
        # Obtém o valor do campo de formulário  
        secao = request.POST.get('usuario')  
  
        # Define a chave na sessão com o valor obtido  
        request.session['login'] = secao  
  
    return render(  
        request,  
        'home/sessao.html',  
    )  
  
def exibir_valor(request):  
  
    # Obtenha o valor da sessão  
    # Defina um valor padrão se a sessão estiver vazia  
    login = request.session.get('login', 'Nenhum valor definido')  
  
    contexto = {'login': login}  
  
    return render(  
        request,  
        'home/exibir_valor.html',  
        contexto,  
    )
```

4. Por último, faça a ligação das funções criadas na views, com os arquivos .html:

```
from django.contrib import admin  
from django.urls import path  
from . import views  
urlpatterns = [  
    path('', views.home, name='principal'),  
    path('sessao/', views.sessao, name='sessao'),  
    path('exibir_valor/', views.exibir_valor, name='exibir_valor'),  
]
```

Elias Santos  
Aula: Session

---

#### Testando o exemplo:

1. Chame no seu browser a url secao: [127.0.0.1:8000/home/secao/](http://127.0.0.1:8000/home/secao/)
2. Depois de inserir um valor, clique no botão valor da sessão e o nome digitado lá estará. Se você voltar, não preenchendo nenhum outro valor e clicar novamente no valor da sessão, ele continuará aparecendo.

#### Acrescentando ao exemplo:

4. Criamos abaixo uma função para, enfim, destruir a session. Veja o código:

```
def encerrar_sessao(request):  
    # Use o método clear() para limpar todos os dados da sessão  
    request.session.clear()  
  
    return exibir_valor(request)
```

5. Não esqueça de inserir a url para este fim:

```
path('encerrar/', views.encerrar_sessao, name='encerrar'),
```

#### Testando o exemplo:

Refaça os testes anteriores, agora chamando também a url encerrar/

#### Desafio:

Teste se o login da sessão é igual ao seu nome.

Se for igual ao seu nome, exiba na tela ao invés de seu nome a palavra “Administrador”

#### Exemplo 2: obtendo a contagem de visitas

Como um exemplo simples do mundo real, atualizaremos nossa biblioteca para informar ao usuário atual quantas vezes ele visitou o site.

## 1. Alteração na função home:

```
def home(request):  
  
    # Número de visitas a esta visualização, conforme contado na variável de  
    sessão.  
    num_visitas = request.session.get('num_visits', 0)  
    request.session['num_visits'] = num_visitas + 1  
  
    contexto = {  
        'title' : 'Home | By Elias',  
        'num_visitas': num_visitas,  
    }  
    return render(  
        request,  
        'home/index.html',  
        contexto  
    )
```

## 2. Alteração no arquivo index.html:

```
{% extends 'global/base.html' %}  
  
{% block texto %} Mudando o texto com a herança {% endblock texto %}  
  
{% block info %}  
<p>  
    Você visitou esta página {{ num_visitas }}  
    {% if num_visitas == 1 %} vez  
    {% else %} vezes  
    {% endif %}.  
</p>  
  
{% endblock info %}
```