

Lista 4

Gerado com IA

Exercício: Utilizando Sessões e Cookies em Flask

Objetivo: Aprender a gerenciar sessões e cookies para armazenar informações do usuário temporariamente e persistente usando Flask.

Passo 1: Preparando o Ambiente

1. Instalação do Flask (se necessário)

- Caso o ambiente virtual ainda não esteja configurado, crie e ative-o:

```
python -m venv venv
```

- Ative o ambiente virtual:

- No Windows:

```
venv\Scripts\activate
```

- No Mac/Linux:

```
source venv/bin/activate
```

- Instale o Flask:

```
pip install Flask
```

- 2. **Objetivo deste passo:** Preparar o ambiente de desenvolvimento para o Flask.

Passo 2: Configurando o Arquivo de Aplicação para Sessões

- 1. No diretório do projeto, crie o arquivo `app.py` com o seguinte conteúdo:

```
from flask import Flask, request, render_template, session, redirect, url_for, make_response

app = Flask(__name__)
app.secret_key = 'chave-secreta' # Necessário para usar sessões
```

2. **Objetivo deste passo:** Configurar a aplicação para suportar sessões, definindo uma `secret_key`.

Passo 3: Criando um Template com Formulário para Salvar Informações na Sessão

1. Crie uma pasta chamada `templates` no mesmo diretório do arquivo `app.py`.
2. Dentro da pasta `templates`, crie um arquivo chamado `sessao.html` com o seguinte conteúdo:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Formulário de Sessão</title>
</head>
<body>
  <h1>Bem-vindo! Informe seu nome para iniciar a sessão.</h1>
  <form action="{ { url_for('iniciar_sessao') } }" method="post">
    <label for="nome">Nome:</label>
    <input type="text" id="nome" name="nome">
    <button type="submit">Iniciar Sessão</button>
  </form>
  {% if nome %}
    <p>Olá, {{ nome }}! A sessão está ativa.</p>
  {% endif %}
  <a href="{ { url_for('limpar_sessao') } }">Sair da Sessão</a>
</body>
</html>
```

3. **Objetivo deste passo:** Criar um formulário HTML para o usuário inserir seu nome e iniciar uma sessão.

Passo 4: Definindo as Rotas para Manipulação da Sessão

1. No arquivo `app.py`, adicione as rotas para iniciar e limpar a sessão:

```
@app.route('/sessao', methods=['GET', 'POST'])
def iniciar_sessao():
    if request.method == 'POST':
        session['nome'] = request.form['nome'] # Armazena o nome na sessão
        return redirect(url_for('iniciar_sessao'))
    nome = session.get('nome') # Recupera o nome da sessão
    return render_template('sessao.html', nome=nome)

@app.route('/limpar_sessao')
def limpar_sessao():
    session.pop('nome', None) # Remove o nome da sessão
    return redirect(url_for('iniciar_sessao'))
```

2. **Objetivo deste passo:** Definir rotas para:

- **POST**: salvar o nome do usuário na sessão.
- **GET**: exibir o nome do usuário se a sessão estiver ativa.
- Limpar a sessão ao sair.

Passo 5: Trabalhando com Cookies

1. Adicione uma nova rota chamada `/cookie` que salva o nome do usuário em um cookie e exibe a mensagem de boas-vindas. Adicione o código abaixo ao `app.py`:

```
@app.route('/cookie', methods=['GET', 'POST'])
def definir_cookie():
    if request.method == 'POST':
        nome = request.form['nome']
        resposta = make_response(redirect(url_for('definir_cookie')))
        resposta.set_cookie('nome', nome, max_age=60*60*24) # Cookie válido
        por 1 dia
        return resposta
    nome = request.cookies.get('nome')
    return render_template('cookie.html', nome=nome)
```

2. Crie um novo template chamado `cookie.html` em `templates` com o seguinte conteúdo:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>Cookie de Boas-vindas</title>
</head>
<body>
    <h1>Definir Nome no Cookie</h1>
    <form action="{{ url_for('definir_cookie') }}" method="post">
        <label for="nome">Nome:</label>
        <input type="text" id="nome" name="nome">
        <button type="submit">Salvar em Cookie</button>
    </form>
    {% if nome %}
        <p>Olá, {{ nome }}! Este nome foi salvo em um cookie.</p>
    {% endif %}
</body>
</html>
```

3. **Objetivo deste passo:** Salvar e recuperar o nome do usuário em um cookie com validade de um dia.

Passo 6: Testando a Aplicação

1. No terminal, execute o arquivo `app.py` para iniciar a aplicação:

```
python app.py
```

-
2. Acesse <http://127.0.0.1:5000/sessao> para testar a sessão:
 - Insira o nome e observe a sessão sendo criada.
 - Clique no link "Sair da Sessão" para limpar a sessão.
 3. Acesse <http://127.0.0.1:5000/cookie> para testar o cookie:
 - Insira o nome e observe a mensagem de boas-vindas, indicando que o nome foi salvo em um cookie.
 - Atualize a página e veja que o nome permanece, mesmo após fechar o navegador (enquanto o cookie estiver ativo).

Conclusão do Exercício

Este exercício mostrou como:

1. Criar e gerenciar sessões em uma aplicação Flask.
2. Criar e recuperar cookies.
3. Utilizar `request`, `session`, `make_response`, e `set_cookie` para armazenar informações do usuário.

Desafio Extra: Adicione um campo "Idade" ao formulário de cookies e configure um cookie para armazenar a idade. Exiba essa informação ao carregar a página.