

Lista 3

Gerado com IA

Exercício: Criando um Formulário com Flask para Processar Requisições GET e POST

Objetivo: Aprender a usar formulários em templates Flask, processar dados usando métodos **GET** e **POST**, e manipular objetos de requisição com **request**.

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: Criando o Arquivo de Aplicação

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

```
from flask import Flask, request, render_template

app = Flask(__name__)

@app.route('/')
```

```
def home():  
    return "Página Inicial"
```

2. **Objetivo deste passo:** Configurar a estrutura básica da aplicação Flask.

Passo 3: Criando o Template com Formulário

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

```
<!DOCTYPE html>  
<html lang="pt-br">  
<head>  
    <meta charset="UTF-8">  
    <title>Formulário de Feedback</title>  
</head>  
<body>  
    <h1>Deixe seu Feedback</h1>  
    <form action="{{ url_for('feedback') }}" method="post">  
        <label for="nome">Nome:</label>  
        <input type="text" id="nome" name="nome"><br><br>  
        <label for="comentario">Comentário:</label><br>  
        <textarea id="comentario" name="comentario" rows="4" cols="50">  
</textarea><br><br>  
        <button type="submit">Enviar</button>  
    </form>  
</body>  
</html>
```

3. **Objetivo deste passo:** Criar um formulário HTML simples que envia dados usando o método `POST`.

Passo 4: Definindo a Rota para o Formulário

1. Adicione uma rota para a página do formulário no arquivo `app.py`:

```
@app.route('/formulario', methods=['GET', 'POST'])  
def feedback():  
    if request.method == 'POST':  
        nome = request.form.get('nome')  
        comentario = request.form.get('comentario')  
        return f"Obrigado pelo feedback, {nome}! Comentário recebido:  
{comentario}"  
    return render_template('formulario.html')
```

2. **Objetivo deste passo:** Definir uma rota que processa requisições `GET` e `POST`:

- Se a requisição for `GET`, renderiza o formulário.

- Se a requisição for **POST**, exibe uma mensagem de agradecimento usando os dados enviados.

Passo 5: Testando o Formulário

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

```
python app.py
```

2. Acesse **http://127.0.0.1:5000/formulario** no navegador e preencha o formulário com seu nome e um comentário.
3. Ao enviar, o navegador deve exibir uma mensagem de agradecimento com o nome e comentário fornecidos.
4. **Objetivo deste passo:** Testar o funcionamento do formulário e a manipulação de dados usando o método **POST**.

Passo 6: Explorando o Método GET

1. Modifique a rota **/formulario** para exibir uma mensagem quando acessada pelo método **GET**:

```
@app.route('/formulario', methods=['GET', 'POST'])
def feedback():
    if request.method == 'POST':
        nome = request.form.get('nome')
        comentario = request.form.get('comentario')
        return f"Obrigado pelo feedback, {nome}! Comentário recebido: {comentario}"
    return "Bem-vindo ao formulário! Por favor, envie seu feedback." +
    render_template('formulario.html')
```

2. Reinicie a aplicação e acesse o formulário novamente. Observe a mensagem que aparece na página ao usar o método **GET**.
3. **Objetivo deste passo:** Compreender a diferença entre os métodos **GET** e **POST** e observar como o **GET** pode ser usado para instruções e o **POST** para processar dados do usuário.

Conclusão do Exercício

Este exercício mostrou como:

1. Criar e renderizar formulários em um template HTML.
2. Processar requisições **GET** e **POST** em uma rota.
3. Utilizar **request** para obter dados enviados pelo usuário.

Desafio Extra: Adicione um campo "Email" no formulário e modifique o código para exibir o email recebido após o envio do formulário.