Introdução ao Flask: Criando Sua Primeira Aplicação Web em Python

Flask é um micro framework web escrito em Python que facilita a criação de aplicações web. Com sua simplicidade e flexibilidade, é uma ótima escolha tanto para iniciantes quanto para desenvolvedores experientes. Neste artigo, abordaremos os principais comandos do Flask em quatro tópicos, com exemplos de código para ajudar você a começar.

1. Instalação e Configuração Inicial

Para começar a usar Flask, primeiro é necessário instalá-lo. Você pode fazer isso utilizando o pip, o gerenciador de pacotes do Python.

```
pip install Flask
```

Após a instalação, crie um arquivo chamado `app.py` e importe o Flask:

```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, World!'

if __name__ == '__main__':
    app.run(debug=True)
```

No exemplo acima, criamos uma aplicação Flask simples que retorna "Hello, World!" quando acessamos a raiz (`/`) do site. O `app.run(debug=True)` ativa o modo de debug, facilitando o desenvolvimento ao fornecer mensagens de erro detalhadas.

2. Rotas e Métodos HTTP

No Flask, as rotas definem os URLs que a aplicação responde e quais funções devem ser chamadas. Você pode especificar diferentes métodos HTTP (GET, POST, etc.) para cada rota.

```python
@app.route('/hello/<name>')
def hello(name):
    return f'Hello, {name}!'

@app.route('/data', methods=['POST'])
def receive_data():
    data = request.get_json()
    return jsonify(data), 201
```

No exemplo acima, criamos uma rota dinâmica (/hello/<name>) que aceita um parâmetro e uma rota /data que aceita apenas requisições POST, retornando os dados recebidos em formato JSON.

3. Templates e Renderização

Flask utiliza o Jinja2 como mecanismo de templates para gerar HTML dinâmico. Crie uma pasta chamada templates e adicione um arquivo index.html.

```html
<!DOCTYPE html>
<html>
<head>
    <title>{{ title }}</title>
</head>
<body>
    <h1>{{ message }}</h1>
</body>
</html>
```

Modifique app.py para renderizar este template:

```python
from flask import render_template

@app.route('/template')
def template_example():
    return render_template('index.html', title='Flask Template',
message='Olá, Flask!')
```

No exemplo acima, usamos a função render_template para renderizar o arquivo index.html, passando variáveis para serem usadas no template.

4. Formulários e Processamento de Dados

Para lidar com formulários HTML, Flask facilita o processamento dos dados enviados pelo usuário.

```
<!DOCTYPE html>
<html>
<head>
    <title>Formulário</title>
</head>
<body>
    <form action="/submit" method="post">
        <input type="text" name="name">
        <input type="submit">
    </form>
</body>
</html>
```

Modifique app.py para processar o formulário:

```
from flask import request

@app.route('/form')
def form():
    return render_template('form.html')

@app.route('/submit', methods=['POST'])
def submit():
    name = request.form['name']
    return f'Nome recebido: {name}'
```

No exemplo acima, criamos um formulário simples e uma rota para processar os dados enviados, retornando o nome recebido.

5. Conclusão

Esses são os comandos básicos para começar a desenvolver com Flask. Com esses exemplos, você pode criar rotas, renderizar templates, processar formulários e trabalhar com métodos HTTP. Flask é uma ferramenta poderosa e flexível, ideal para construir aplicações web de forma rápida e eficiente. Explore a documentação oficial para aprofundar seu conhecimento e descobrir mais funcionalidades avançadas!