Recursos avançados do Flask

Módulo 3 - Experiência Criativa: Criando Soluções Computacionais

ANTÔNIO DAVID VINISKI
antonio.david@pucpr.br
PUCPR

Agenda

- Arquivos estáticos na aplicação
- Inclusão de templates
- Mensagens Flash.
- Manipulação de Erros.
- Filtros no Jinja2.
- Testes no Jinja2
- Comentários.
- Macros.
- Upload Files.

Arquivos estáticos

- O diretório padrão do **flask** para os arquivos estáticos é o `static/`, localizado na raiz do projeto.
- Dentre os arquivos estáticos tempo
  - `css/, js/, img/, fonts/, json/`
- Além disso, assim como em uma aplicação web comum, podemos utilizar nossos arquivos estáticos diretamente passando um link para algum diretório na web.
- O ideal é manter a importação dos arquivos estáticos no template base da aplicação (ou módulo) e ele será automaticamente utilizado nas demais páginas que por meio da herança de template

Exemplo do restaurante

- O diretório padrão do **flask** para os arquivos estáticos é o `static/`, localizado na raiz do projeto.
  - Neste exemplo estamos importando os arquivos do framework bootstrap para ajustar o desing das páginas da aplicação

```
<html>
<head>
    {% block title %}<title>Meu Restaurante</title>{% endblock %}
    <link rel="stylesheet"
        href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css">
    <link rel="stylesheet" href="static/css/style.css" />
</head>
<body>
    {% include "menu.html" %}
    <div id="content">
        {% block content %} {% endblock %}
    </div>
</body>
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.min.js"></script>
</html>
```

Inclusão de template

- A inclusão permite a criação de conteúdos da página em arquivos html separados.
  - Assim, cada conteúdo pode ser incluído em várias páginas posteriormente.
  - No exemplo anterior, incluímos o menu no template base da aplicação do restaurante.

```
<html>
<head>
    {% block title %}<title>Meu Restaurante</title>{% endblock %}
    <link rel="stylesheet"
        href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css">
    <link rel="stylesheet" href="static/css/style.css" />
</head>
<body>
    {% include "menu.html" %}
    <div id="content">
        {% block content %} {% endblock %}
    </div>
</body>
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.min.js"></script>
</html>
```

Inclusão de template

- Assim, o arquivo menu.html precisa estar acessível no diretório dos templates da aplicação para que possa ser importado

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <a class="navbar-brand" href="#">
        <img src="static/img/170781.png" width="30" height="30" alt="">
    </a>
    <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
        <div class="navbar-nav">
            <a class="nav-item nav-link active" href="/">Home</a>
            <a class="nav-item nav-link" href="/iot">IOT</a>
            <a class="nav-item nav-link" href="/billing">Cobrança</a>
            <a class="nav-item nav-link" href="/payment">Pagamento</a>
            <a class="nav-item nav-link" href="/people">Pessoas</a>
            <a class="nav-item nav-link" href="/product">Produtos</a>
            <a class="nav-item nav-link" href="/ticket">Comandas</a>
            <a class="nav-item nav-link" href="/auth">Login</a>
        </div>
    </div>
</nav>
```

Mensagens Flash

Arquivo Python

```
flash('A senha informada é inválida', 'error')
```

Precisamos importar o método **flash** do pacote flask

Página HTML

```
{% with messages = get_flashed_messages() %}
    {% if messages %}
        <div class="notification is-danger">
            {{ messages[0] }}
        </div>
    {% endif %}
{% endwith %}
```

Utilizamos o declaração **with** para criar a variável **messages** que estará disponível apenas no respectivo escopo

Manipulação de Erros

```python
@app.errorhandler(400)
def error_400(error):
    return render_template("error_pages/400.html"), 400

@app.errorhandler(404)
def error_404(error):
    return render_template("error_pages/404.html"), 404

@app.errorhandler(405)
def error_405(error):
    return render_template("error_pages/405.html"), 405

@app.errorhandler(500)
def error_500(error):
    return render_template("error_pages/500.html"), 500
```

- Utilizamos o método `errorhandler(<código>)` disponível na aplicação flask.
- Podemos manipular e redirecionar para páginas específicas quando um erro ocorrer.

Filtros Jinja2

abs()	float()	lower()	round()	tojson()
attr()	forceescape()	map()	safe()	trim()
batch()	format()	max()	select()	truncate()
capitalize()	groupby()	min()	selectattr()	unique()
center()	indent()	pprint()	slice()	upper()
default()	int()	random()	sort()	urlencode()
dictsort()	join()	reject()	string()	urlize()
escape()	last()	rejectattr()	striptags()	wordcount()
filesizeformat()	length()	replace()	sum()	wordwrap()
first()	list()	reverse()	title()	xmlattr()

https://jinja.palletsprojects.com/en/2.10.x/templates/#builtin-filters

Testes Jinja2

callable()	even()	le()	none()	string()
defined()	ge()	lower()	number()	undefined()
divisibleby()	gt()	lt()	odd()	upper()
eq()	in()	mapping()	sameas()	
escaped()	iterable()	ne()	sequence()	

https://jinja.palletsprojects.com/en/2.10.x/templates/#builtin-tests

Comentários

Comentário inserido no arquivo html.

```
{# note: commented-out template because we no longer use this
    {% for user in users %}
        ...
    {% endfor %}
#}
```

Macros

As macros servem para criar componentes reutilizáveis da aplicação

```
<!-- form.html -->
{% macro input(label, name, id, value='', type='text', box_size = 4) -%}
<div class="col-md-{{ box_size }}">
    <label for="{{ id }}" class="form-label">{{ label }}</label>
    <input class="form-control" type="{{ type }}" value="{{ value|e }}" name="{{ name }}" id = "{{ id }}">
</div>
{%- endmacro %}

{%- macro textarea(label, name, id, value='', rows=4, cols=40, box_size = 12) -%}
<div class="col-md-{{ box_size }}">
    <label class="form-label" for="{{ id }}">{{ label }}</label>
    <textarea class="form-control" name="{{ name }}" id="{{ id }}" rows="{{ rows }}" cols="{{ cols }}" ></textarea>
</div>
{%- endmacro %}
```

Macros

```
<!- register_sensors.html -->
<form class="row g-3 justify-content-center" action="/iot/save_sensor" method="POST">
    {{ forms.input(label = 'Nome', name = 'name', id = 'name', box_size = 8) }}
    {{ forms.input(label = 'Modelo', name = 'model', id = 'model', value='') }}
    {{ forms.input(label = 'Marca', name = 'brand', id = 'brand', value='', type='text') }}
    {{ forms.input(label = 'Tensão de trabalho', name = 'voltage', id = 'voltage') }}
    {{ forms.input(label = 'Unidade de Medida', name = 'measure', id = 'measure') }}
    {{ forms.input(label = 'Url Acesso',name = 'url', id = 'url', box_size = 12) }}
    {{ forms.textarea(label = 'Descrição', name = 'description', id = 'description') }}
    <div class="col-12">
        <button type="submit" class="float-right btn btn-primary">Cadastrar Atuador</button>
    </div>
</form>
```

Upload Files

```
<form enctype="multipart/form-data" action="/save_file" method="POST"></form>
    <label for="image" class="form-label">Imagem</label>
    <input class="form-control" id="image" type=file name="image">
    <button type="submit" >Salvar Imagem</button>
</form>
```

```
from werkzeug.utils import secure_filename

@iot.route("/save_file", methods = ["POST"])
def save_file():
    f = request.files.get('image', None)
    if f.filename is not None and f.filename != "":
        print(f.filename)
        f.save('static/img/'+secure_filename(f.filename))
```

PBL1 – Passo 2

Implementação do módulo sorteado na aplicação do restaurante

Temas

1 – Auth

2 – Billing

3 – Payment

4 – People

5 – Product

6 – Ticket

Referências

- https://flask.palletsprojects.com/en/2.2.x/patterns/flashing/
- https://flask.palletsprojects.com/en/2.2.x/quickstart/#about-responses
- https://flask.palletsprojects.com/en/2.2.x/patterns/fileuploads/