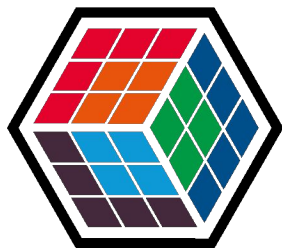


THE DEVELOPER'S CONFERENCE

Trilha – Python

Kamila Santos Oliveira
Software Developer



THE DEVELOPER'S CONFERENCE

Jinja 2

mecanismo para templates no Flask

Kamila Santos Oliveira

21 anos,

Dev na Cognizant,

~ 3 anos na área,

Graduanda em ciência da computação



THE
DEVELOPER'S
CONFERENCE

Agenda



THE
DEVELOPER'S
CONFERENCE

- ❖ O que é Flask
- ❖ Por que micro framework?
- ❖ Hello world com flask
- ❖ Rotas
- ❖ Url building

Agenda



THE
DEVELOPER'S
CONFERENCE

- ❖ HTTP Methods
- ❖ Pymongo
- ❖ De onde saiu esse Jinja?
- ❖ Exemplo



THE
DEVELOPER'S
CONFERENCE

O que é Flask





THE
DEVELOPER'S
CONFERENCE

Flask é um microframework para aplicações web utilizando Python. Ele é classificado como microframework porque precisa do mínimo de configuração possível – mantém o core da aplicação simples e deixa você livre para escolher o que usar em sua aplicação (por exemplo, seu banco de dados).

O que é Flask



THE
DEVELOPER'S
CONFERENCE

Porque microframework



Por default ele não tem camada de abstração de banco de dados, mecanismo de validação de formulário e afins, mas suporta extensões que realizam integrações com banco de dados, requisições, validações de formulários, autenticação, upload de arquivos, entre outras.



THE
DEVELOPER'S
CONFERENCE

Porque microframework

Resumindo: você pode utilizar do Flask tudo o que precisa e dispensar tudo o que não precisa no momento.



THE
DEVELOPER'S
CONFERENCE

O que é Flask



THE
DEVELOPER'S
CONFERENCE

Hello world



```
1  from flask import Flask
2  app = Flask(__name__)
3
4  @app.route('/')
5  def hello_world():
6  |     return 'Hello, World!'
```



THE
DEVELOPER'S
CONFERENCE

Rotas



```
1 @app.route('/')
2 def index():
3     return 'Index Page'
4
5 @app.route('/hello')
6 def hello():
7     return 'Hello, World'
```



THE
DEVELOPER'S
CONFERENCE

URL Building



THE
DEVELOPER'S
CONFERENCE

```
1  from flask import Flask, escape, url_for
2
3  app = Flask(__name__)
4
5  @app.route('/')
6  def index():
7      return 'index'
8
9  @app.route('/login')
10 def login():
11     return 'login'
12
13 @app.route('/user/<username>')
14 def profile(username):
15     return '{}\s profile'.format(escape(username))
16
17 with app.test_request_context():
18     print(url_for('index'))
19     print(url_for('login'))
20     print(url_for('login', next='/'))
21     print(url_for('profile', username='John Doe'))
```




THE
DEVELOPER'S
CONFERENCE

HTTP Methods



```
from flask import request

@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        return do_the_login()
    else:
        return show_the_login_form()
```



THE
DEVELOPER'S
CONFERENCE

Pymongo



```
1 from flask import Flask
2 from flask_pymongo import PyMongo
3
4 app = Flask(__name__)
5 app.config["MONGO_URI"] = "mongodb://localhost:27017/myDatabase"
6 mongo = PyMongo(app)
```



THE
DEVELOPER'S
CONFERENCE

De onde saiu esse
Jinja?





THE
DEVELOPER'S
CONFERENCE

Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks.

Flask offers suggestions, but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy.



THE
DEVELOPER'S
CONFERENCE

WSGI



THE
DEVELOPER'S
CONFERENCE

WSGI Web Server Gateway Interface

Interface o servidor da web, descreve com um servidor web se comunica através de aplicativos web e como estes aplicativos podem ser encadeados para processar uma solicitação.

É um padrão python descrito detalhadamente na PEP333 (Python Web Server Gateway Interface)

*Python Enhancement Proposals (PEPs):

WSGI

Werkzeug

werkzeug substantivo alemão: "ferramenta".

Biblioteca para aplicativos web do WSGI, inicialmente era uma coleção simples de vários aplicativos WSGI e passou a ser uma das bibliotecas WSGI mais avançadas.

O Flask utiliza ele para lidar com os detalhes do WSGI fornecendo estruturas e padrões para aplicações



THE
DEVELOPER'S
CONFERENCE

Werkzeug



THE
DEVELOPER'S
CONFERENCE

Jinja2

Jinja

Jinja2 is a full-featured template engine for Python. It has full unicode support, an optional integrated sandboxed execution environment, widely used and BSD licensed.

Jinja is Beautiful



```
{% extends "layout.html" %}
{% block body %}
<ul>
  {% for user in users %}
    <li><a href="{{ user.url }}">{{ user.username }}</a></li>
  {% endfor %}
</ul>
{% endblock %}
```

And Powerful



Jinja2 is one of the most used template engines for Python. It is inspired by Django's templating system but extends it with an expressive language that gives template authors a more powerful set of tools. On top of that it adds sandboxed

Jinja2



THE
DEVELOPER'S
CONFERENCE

Jinja2



Jinja2

Mecanismo para templates no Python com suporte unicode completo com integração sandboxed opcional para o ambiente de execução.



THE
DEVELOPER'S
CONFERENCE

Jinja2



THE
DEVELOPER'S
CONFERENCE

Jinja2

Jinja is Beautiful

```
{% extends "layout.html" %}
{% block body %}
  <ul>
    {% for user in users %}
      <li><a href="{{ user.url }}">{{ user.username }}</a></li>
    {% endfor %}
  </ul>
{% endblock %}
```

Jinja2

Jinja é poderoso

É um dos mecanismos de templates mais utilizados no Python, foi inspirado nos mecanismos de template do Django.

Recursos:

Modelo de execução em sandbox: todos os aspectos de execução do modelo são monitorados e colocados em whitelisted ou blacklisted



THE
DEVELOPER'S
CONFERENCE

Jinja2

Jinja é poderoso

Sistema automático de escape HTML que previne scripts entre os sites .



THE
DEVELOPER'S
CONFERENCE

Jinja2

Jinja é poderoso

Herança entre os modelos permite utilizar um mesmo layout ou algumas características dele para vários modelos



THE
DEVELOPER'S
CONFERENCE

Jinja2

Jinja é poderoso

Alta performance com compilação just in time para bytecode Python.

Ele traduz o código fonte dos seus templates no primeiro carregamento para bytecode Python tendo um melhor desempenho de tempo de execução.

Opcionalmente pode implementar compilação ahead-of-time



THE
DEVELOPER'S
CONFERENCE

Jinja2

Jinja é poderoso

Fácil de debugar com um sistema de report que integra erros de compilação de de runtime, no sistema padrão de erros de Python.



THE
DEVELOPER'S
CONFERENCE

Jinja2

Jinja é poderoso

Sintaxe configurável: podemos configurar o Jinja2 para ajustar melhores formatos de saída como LaTeX ou JS.



THE
DEVELOPER'S
CONFERENCE

Jinja2

Jinja é poderoso

Vem com vários helpers que ajudam a resolver tarefas comuns a modelos, como divisão de conteúdo em colunas, etc.



THE
DEVELOPER'S
CONFERENCE

Jinja2



THE
DEVELOPER'S
CONFERENCE

High Level API



```
class jinja2.Environment([options])
```

Seu principal componente é o Environment, ela contém informações como as variáveis compartilhadas, configuração, filtros, testes.



THE
DEVELOPER'S
CONFERENCE

High Level API



THE
DEVELOPER'S
CONFERENCE

`block_start_string`

Iniciamos um bloco com '{%'

High Level API

`block_end_string`

Encerramos um bloco com '%}'



THE
DEVELOPER'S
CONFERENCE

High Level API

variable_start_string

Para chamar e imprimir variáveis utilizamos '{{'



THE
DEVELOPER'S
CONFERENCE

High Level API



THE
DEVELOPER'S
CONFERENCE

variable_end_string

Para encerrar um bloco de variáveis utilizamos '}}

High Level API



THE
DEVELOPER'S
CONFERENCE

`comment_start_string`

Para iniciar um comentário utilizamos '#{#'

High Level API



THE
DEVELOPER'S
CONFERENCE

`comment_end_string`

Para iniciar um comentário utilizamos '#}'

High Level API

class jinja2.Template

Objeto central dos templates, através deles os templates são compilados e renderizados.



THE
DEVELOPER'S
CONFERENCE

High Level API

name

O nome de carregamento do modelo, se for um template String é None.



THE
DEVELOPER'S
CONFERENCE

High Level API



THE
DEVELOPER'S
CONFERENCE

filename

Nome do arquivo do modelo, se não foi carregado se um arquivo é None.

High Level API



THE
DEVELOPER'S
CONFERENCE

render([context])

Aceita um construtor, subclasse ou argumentos com keywords

```
1 template.render(knights='that say nih')
2 template.render({'knights': 'that say nih'})
```

High Level API

`generate([context])`

Quando temos modelos muito grandes, é útil as vezes não renderizar o template todo de uma vez só, mas avaliar cada instrução do nosso modelo e gerar de acordo com a nossa necessidade.

Ele retorna um generator que gera um item após o outro como strings unicode.

Aceita os mesmos argumentos que o `render()`;



THE
DEVELOPER'S
CONFERENCE

High Level API



THE
DEVELOPER'S
CONFERENCE

render_async([context])

Semelhante ao `render()`, porém retorna uma corotina, que quando está em `await`, retorna toda a string do modelo a ser renderizado

```
1 await template.render_async(knights='that say nih; asynchronously')
2 |
```

High Level API



THE
DEVELOPER'S
CONFERENCE

class jinja2.runtime.Context

Contém as variáveis de um modelo, armazena os valores passados para o modelo e os nomes que o modelo exporta.

O Context suporta somente operações read-only (get, keys, values, items, iterkeys, itervalues, iteritems, __getitem__, __contains__)

High Level API



THE
DEVELOPER'S
CONFERENCE

class jinja2.BaseLoader

São responsáveis por carregar os templates de um recurso como o FileSystem.

Basicamente, o ambiente fornece um método `get_template` que chama o método `load` do loader para obter este objeto `template`.

High Level API



```
1 from jinja2 import BaseLoader, TemplateNotFound
2 from os.path import join, exists, getmtime
3
4 class MyLoader(BaseLoader):
5
6     def __init__(self, path):
7         self.path = path
8
9     def get_source(self, environment, template):
10        path = join(self.path, template)
11        if not exists(path):
12            raise TemplateNotFound(template)
13        mtime = getmtime(path)
14        with file(path) as f:
15            source = f.read().decode('utf-8')
16        return source, path, lambda: mtime == getmtime(path)
```

```
get_source(environment, template)
```

Obtém a fonte do modelo, nome do arquivo e carrega o helper para o template. Passa o nome do ambiente e do modelo e precisa retornar uma tupla (source, filename, uptodate) ou gerar um erro `TemplateNotFound` caso não localize esse modelo.



THE
DEVELOPER'S
CONFERENCE

High Level API



THE
DEVELOPER'S
CONFERENCE

```
load(environment, name,  
globals=None)
```

Responsável por carregar os modelos, busca um modelo no cache ou carrega um novo chamando o `get_source()`.

High Level API



THE
DEVELOPER'S
CONFERENCE

Low Level API



Expõe funcionalidades referentes a detalhes de implementação, depuração, extensão.

“Unless you know exactly what you are doing we don’t recommend using any of those” - Documentação do Flask



THE
DEVELOPER'S
CONFERENCE

Low Level API



THE
DEVELOPER'S
CONFERENCE

Exemplo



Pokedex

Nome	Espécie	Tipo
Bulbasaur	Seed	Grass/Poison
Charmander	Lizard	Fire
Clefairy	Fairy	Fairy
Squirtle	Tiny Turtle	Water
Azumarill	Aqua Rabbit	Water

Login

Novo

Logout



THE
DEVELOPER'S
CONFERENCE

Faça seu login

Nome da treinadora:

Senha:

Entrar



THE
DEVELOPER'S
CONFERENCE

Faça seu login

Nome da treinadora:

Senha:

Entrar



Ada Lovelace acesso permitido!

Novo Pokemon

Nome

Espécie

Tipo

Registrar na pokedex

Login

Novo

Logout



Novo Pokemon

Nome

Espécie

Tipo

Registrar na pokedex

Login

Novo

Logout



Pokedex

Nome	Espécie	Tipo
Bulbasaur	Seed	Grass/Poison
Charmander	Lizard	Fire
Clefairy	Fairy	Fairy
Squirtle	Tiny Turtle	Water
Azumarill	Aqua Rabbit	Water
Psyduck	Duck	Water

[Login](#)[Novo](#)[Logout](#)



THE
DEVELOPER'S
CONFERENCE

Treinadora, logue novamente para cadastrar os pokemons que encontrar!

Pokedex

Nome	Espécie	Tipo
Bulbasaur	Seed	Grass/Poison
Charmander	Lizard	Fire
Clefairy	Fairy	Fairy
Squirtle	Tiny Turtle	Water
Azumarill	Aqua Rabbit	Water
Psyduck	Duck	Water

Login

Novo

Logout



THE
DEVELOPER'S
CONFERENCE



localhost:5000/login?proxima=%2Fnovo



Faça seu login

Nome da treinadora:

Senha:

Entrar



THE
DEVELOPER'S
CONFERENCE

▾ static

bootstrap.css

▾ templates

◇ lista.html

◇ login.html

◇ novo.html

◇ template.html

🔗 pokedex.py



```
pokedex.py ▶ ...
1  from flask import Flask, render_template, request, redirect, session, flash, url_for
2
3  app = Flask(__name__)
4  app.secret_key = 'flask'
5
6  class Pokemon:
7      def __init__(self, nome, especie, tipo):
8          self.nome = nome
9          self.especie = especie
10         self.tipo = tipo
11
12     class Treinadora:
13         def __init__(self, id, nome, senha):
14             self.id = id
15             self.nome = nome
16             self.senha = senha
17
```




```
37 @app.route('/')
38 def index():
39     return render_template('lista.html', titulo='Pokedex', pokemons=lista)
40
41 @app.route('/novo')
42 def novo():
43     if 'usuario_logado' not in session or session['usuario_logado'] == None:
44         return redirect(url_for('login', proxima=url_for('novo')))
45     return render_template('novo.html', titulo='Novo Pokemon')
46
47 @app.route('/criar', methods=['POST',])
48 def criar():
49     nome = request.form['nome']
50     especie = request.form['especie']
51     tipo = request.form['tipo']
52     pokemon = Pokemon(nome, especie, tipo)
53     lista.append(pokemon)
54     return redirect(url_for('index'))
55
```



```
57 @app.route('/login')
58 def login():
59     proxima = request.args.get('proxima')
60     return render_template('login.html', proxima=proxima)
61
62
63 @app.route('/autenticar', methods=['POST', ])
64 def autenticar():
65     if request.form['treinadora'] in treinadoras:
66         treinadora = treinadoras[request.form['treinadora']]
67         if treinadora.senha == request.form['senha']:
68             session['usuario_logado'] = treinadora.id
69             flash(treinadora.nome + ' acesso permitido!')
70             proxima_pagina = request.form['proxima']
71             return redirect(proxima_pagina)
72     else:
73         flash('Acesso negado, digite novamente!')
74         return redirect(url_for('login'))
75
```



```
76
77 @app.route('/logout')
78 def logout():
79     session['usuario_logado'] = None
80     flash('Treinadora, logue novamente para cadastrar os pokemons que encontrar!')
81     return redirect(url_for('index'))
82
83
84 app.run(debug=True)
85 |
```




```
<> lista.html x
templates ▾ <> lista.html ▾ table.table.table-striped.table-responsive.table-bordered ▾ tbody
1  {% extends "template.html" %}
2  {% block conteudo %}
3      <table class="table table-striped table-responsive table-bordered">
4          <thead>
5              <tr>
6                  <th>Nome</th>
7                  <th>Espécie</th>
8                  <th>Tipo</th>
9              </tr>
10         </thead>
11         <tbody>
12             {% for pokemon in pokemons %}
13                 <tr>
14                     <td>{{ pokemon.nome }}</td>
15                     <td>{{ pokemon.especie }}</td>
16                     <td>{{ pokemon.tipo }}</td>
17                 </tr>
18             {% endfor %}
19         </tbody>
20     </table>
21 {% endblock %}
```



THE DEVELOPER'S CONFERENCE

login.html x

templates ▸ login.html ▸ html ▸ head ▸ link

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <title>Pokedex</title>
7     <link rel="stylesheet" href="{{ url_for('static', filename='bootstrap.css') }}">
8   </head>
9   <body>
10    <div class="container">
11      {% with messages = get_flashed_messages() %}
12        {% if messages %}
13          <ul id="messages" class="list-unstyled">
14            {% for message in messages %}
15              <li class="alert alert-success">{{ message }}</li>
16            {% endfor %}
17          </ul>
18        {% endif %}
19      {% endwith %}
20
```



```
20
21 <h1>Faça seu login</h1>
22 <form method="POST" action="{{ url_for('autenticar') }}">
23     <input type="hidden" name="proxima" value="{{ proxima or url_for('index') }}">
24     <p><label>Nome da treinadora:</label> <input class="form-control" type="text" name="treinadora" required>
25     <p><label>Senha:</label> <input class="form-control" type="password" name="senha" required></p>
26     <p><button class="btn btn-primary" type="submit">Entrar</button></p>
27 </form>
28 iv>
29
30
```



<> novo.html x

templates ▸ <> novo.html ▸ form ▸ fieldset ▸ div.form-group

```
1  {% extends "template.html" %}
2  {% block conteudo %}
3      <form action="{{ url_for('criar') }}" method="post">
4          <fieldset>
5              <div class="form-group">
6                  <label for="nome">Nome</label>
7                  <input type="text" id="nome" name="nome" class="form-control">
8              </div>
9              <div class="form-group">
10                 <label for="categoria">Espécie</label>
11                 <input type="text" id="especie" name="especie" class="form-control">
12             </div>
13             <div class="form-group">
14                 <label for="console">Tipo</label>
15                 <input type="text" id="tipo" name="tipo" class="form-control">
16             </div>
17             <button type="submit" class="btn btn-primary btn-salvar">Registrar na pokedex</button>
18         </fieldset>
19     </form>
20 {% endblock %}
```



<> template.html x

templates ▸ <> template.html ▸ html ▸ body ▸ a.btn.btn-primary

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <title>Pokedex</title>
7     <link rel="stylesheet" href="{{ url_for('static', filename='bootstrap.css') }}">
8   </head>
9   <body>
10    <div class="container">
11      {% with messages = get_flashed_messages() %}
12      {% if messages %}
13        <ul id="messages" class="list-unstyled">
14          {% for message in messages %}
15            <li class="alert alert-success">{{ message }}</li>
16          {% endfor %}
17        </ul>
18      {% endif %}
19    {% endwith %}
20
```



```
20
21     <div class="page-header">
22         | | <h1>{{ titulo }}</h1>
23     </div>
24     {% block conteudo %}{% endblock %}
25 </div>
26 <a class="btn btn-primary" href="/login" role="button">Login </a>
27 <a class="btn btn-primary" href="/novo" role="button">Novo </a>
28 <a class="btn btn-primary" href="/logout" role="button">Logout </a>
29
30
31 </body>
32 </html>
```

Referências:



THE
DEVELOPER'S
CONFERENCE

<https://palletsprojects.com/p/flask/>

<https://palletsprojects.com/p/jinja/>

<https://palletsprojects.com/p/werkzeug/>

<http://wsgi.tutorial.codepoint.net/intro>

<https://www.python.org/dev/peps/>

<https://www.python.org/dev/peps/pep-0333/>

<https://www.tecmundo.com.br/spyware/1172-o-que-e-sandbox-.htm>

<https://flask.palletsprojects.com/en/1.1.x/>

<https://www.palletsprojects.com/p/werkzeug/>

Referências:



THE
DEVELOPER'S
CONFERENCE

<http://jinja.pocoo.org/docs/2.10/api/#low-level-api>

<http://jinja.pocoo.org/docs/2.10/api/#high-level-api>

<https://flask-pymongo.readthedocs.io/en/latest/>

<https://github.com/Kamilahsantos/Flask--Crud>

<https://www.alura.com.br/curso-online-flask-rotas-templates-autenticacao>

<https://pythonhosted.org/Flask-Session/>

<http://www.devfuria.com.br/python/flask/>

<https://hackernoon.com/top-10-python-web-frameworks-to-learn-in-2018-b2ebab969d1a>

Obrigada <3



THE
DEVELOPER'S
CONFERENCE



@kamilah_santos



in/kamila-santos-oliveira/



@kamila_code



kamilahsantos





THE DEVELOPER'S CONFERENCE