



Curso completo de Python!





Instrutor: Vitor Mazuco

<http://facebook.com/vitormazuco>

Email: vitor.mazuco@gmail.com

WebSite:

<http://vmzsolutions.com.br>



Flask

web development,
one drop at a time



Dividindo a aplicação em Blueprints

Nas aulas passadas, fizemos algumas views e rotas, porém precisamos gerenciar os grupos também. Você pode criar a sua aplicação tudo em um único arquivo, mas podemos fazer as divisões de arquivos também com o uso do Blueprints.



Dividindo a aplicação em Blueprints

As Blueprints são pequenas aplicações em Flask e que podem ser integradas a sua aplicação para assim modular o seu projeto. A própria **PEP20** diz que é melhor trabalhar com arquivos de modo separado do que tudo junto.



Dividindo a aplicação em Blueprints

Para podermos usar esse recurso, basta seguir esse exemplo logo abaixo ao criar um arquivo em Python:

```
from flask import Blueprint

grupos = Blueprint("grupos",__name__)

@grupos.route("/grupos/")
def index_route():
    return "Index do gerenciamento de grupos"
```



Dividindo a aplicação em Blueprints

Agora, você tem que registrar essa Blueprint em sua aplicação principal. Para isso, basta você dar o import do blueprint e registrar em sua aplicação principal. Ex:

```
from grupos import grupos  
# a linha seguinte deverá estar abaixo de app = Flask(__name__)  
app.register_blueprint(grupos)
```

Se você for usar o python 2 não se esqueça de ter um arquivo `__init__.py` dentro desse diretório!



Dividindo a aplicação em Blueprints

Agora, podemos replicar o mesmo conteúdo do nosso arquivo principal para outros arquivos e remover as rotas referentes aos usuários e criar uma Blueprint separada para poder gerenciar esses recursos. Lembrando que não existe um certo limite na quantidade de Blueprints que podemos ter em nosso projeto!



Dividindo a aplicação em Blueprints

Agora vamos testar o nosso projeto com a pasta blueprint de arquivo complementar e abra o seu RestClient para ver os resultados!