

Módulo de autenticação e Role Based Access Control



DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

AGENDA

- Continuando a lógica da autenticação;
- O que é o RBAC;
- Preparando Middleware para o RBAC;

O JSON Web Token é um padrão da Internet para a criação de dados com assinatura opcional e/ou criptografia cujo payload contém o JSON que afirma algum número de declarações.

Criando o módulo de autenticação

Agora, iremos criar uma pasta chama `src/app/middlewares` para utilizarmos nosso método de autenticação, dentro da pasta, um arquivo `auth.py`

Criando o módulo de autenticação

Primeiramente, adicionaremos uma nova configuração na nossa config, chamado `SECRET_KEY` para utilizarmos no nosso JWT como garantia de validação.

Feito o passo anterior, iremos criar uma pasta chamado `src/app/middlewares` para utilizarmos nosso método de autenticação, dentro da pasta, um arquivo `auth.py`

Criando o módulo de autenticação

```
from functools import wraps
from jwt import decode
from flask import request, jsonify, current_app
from src.app.models.user import User

def jwt_required(function_current):
    @wraps(function_current)
    def wrapper(*args, **kwargs):
        token = None

        if 'authorization' in request.headers:
            token = request.headers['authorization']
```

Criando o módulo de autenticação

```
if not token:
```

```
    return jsonify({ "error": "Você não tem permissão para acessar essa rota" }),
```

```
403
```

```
if not "Bearer" in token:
```

```
    return jsonify({ "error": "Você não tem permissão para acessar essa rota" }),
```

```
401
```

```
try:
```

```
    token_pure = token.replace("Bearer ", "")
```

```
    decoded = decode(token_pure, current_app.config['SECRET_KEY'], 'HS256')
```

```
    current_user = User.query.get(decoded['user_id'])
```

```
except:
```

```
    return jsonify({"error": "O token é inválido"}), 403
```

```
    return function_current(current_user=current_user, *args, **kwargs)
```

```
    return wrapper
```

Criando o módulo de autenticação

```
if not token:
```

```
    return jsonify({ "error": "Você não tem permissão para acessar essa rota" }),
```

```
403
```

```
if not "Bearer" in token:
```

```
    return jsonify({ "error": "Você não tem permissão para acessar essa rota" }),
```

```
401
```

```
try:
```

```
    token_pure = token.replace("Bearer ", "")
```

```
    decoded = decode(token_pure, current_app.config['SECRET_KEY'], 'HS256')
```

```
    current_user = User.query.get(decoded['user_id'])
```

```
except:
```

```
    return jsonify({"error": "O token é inválido"}), 403
```

```
    return function_current(current_user=current_user, *args, **kwargs)
```

```
    return wrapper
```


O que é o RBAC?

O controle de acesso baseado na função (RBAC – Role-based access control) é um método de restringir o acesso à rede com base nas funções de usuários individuais em uma empresa.



DEVinHouse

Parcerias para desenvolver a sua carreira

OBRIGADO!



<LAB365>