EXPRESS E POSTGRES

SAE COMP

Instalando dependências

Para fazer uma conexão com o banco de dados, utilizaremos uma biblioteca chamada pg.

```
1  npm install pg
2  npm install @types/pg --save-dev
```

Configuração do pg

Para se conectar ao banco, vamos usar uma pool de conexões. Isso faz com que vários requests consigam ser feitos ao mesmo tempo.

```
1  import { Pool } from 'pg';
2
3  const poolConfig = {
4    host: 'localhost',
5    port: 5432,
6    user: 'postgres',
7    password: '5432',
8    database: 'postgres',
9    ssl: false,
10   max: 6
11 };
12
13 const pool = new Pool(poolConfig);
```

Select com um get request

Geralmente quando utilizamos um tipo GET, vamos fazer uma consulta de select no banco de dados para recuperar informações de uma tabela.

```
const dbQueryGet = `
select
nome,
descricao,
preco::float as preco_do_produto

from produtos
where id = $1
`;
```

```
app.get('/produtos/:id', async (req: Request, res: Response) => {
  const id = parseInt(req.params.id);
  const { moeda } = req.query;
  try {

    const { rows } = await pool.query(dbQueryGet, [id]);
    const produto = rows[0];


    if (!produto) {
      res.status(404).json({ message: 'Produto não encontrado' });
      return;
    }
    if (moeda == 'usd') {
      produto.preco_do_produto = produto.preco_do_produto * 6;
    }
    res.json(produto);
  } catch (error) {
    console.log(error)
    res.status(500).json({ message: 'Erro interno do servidor' });
  }
});
```

Insert com um post request

Quando fazemos POST, queremos inserir algo no banco de dados.

```
const dbQueryPost = `
insert into produtos
(nome, preco, descricao, quantidade)
values ($1, $2, $3, $4)
`;
```

```
app.post('/produtos', async (req: Request, res: Response) => {
  const { nome, preco, descricao, quantidade } = req.body;
  try {
    const { rowCount } = await pool.query(dbQueryPost, [nome, preco, descricao, quantidade]);
    if (rowCount === 0) {
      res.status(400).json({ message: 'Erro ao inserir produto' });
      return;
    }
    res.status(201).json({ message: 'Produto inserido com sucesso' });
  } catch (error) {
    console.log(error);
    res.status(500).json({ message: 'Erro interno do servidor' });
  }
});
```

Código completo do exemplo

https://gist.github.com/Pedro-Porto/a71bd9f5d3d3b7b159c18192fe4a14e7

ATIVIDADES

Atividade 1

Fazer um sistema de cadastro de tarefas

Implementar uma API no express que salve todos os seus dados no postgres com as rotas:

- GET /tarefas -> Listar todas as tarefas
- POST /tarefas -> Adicionar uma nova tarefa
- PATCH /tarefas/:id -> Marcar tarefa como concluída
- GET /tarefas/status/:status -> Listar tarefas por status

MUITO OBRIGADO


@saecomp.ec


saecomp@usp.br


saecomp.github.io


Prédio da Engenharia de Computação,
Campus 2, USP São Carlos

SAECOMP