

Conectando-se ao PostgreSQL com Node.js

Mentor: Carlos Júnior

1 - Conexão com o PostgreSQL

- Para que serve?
- Como fazer isso?

2 - Executando consultas SQL

- Para que serve?
- Como fazer isso?

3 - Variáveis de Ambiente

- O que são?
- Para que servem?

Conexão com o PostgreSQL

Até então vimos como criar uma base de dados com Postgres e como criar uma API com Node.js e Express.js. Agora veremos como conectar ambos. Para isso, instale o pacote [pg](#) antes de prosseguir.

Conexão com o banco de dados:

```
const { Pool } = require('pg')

const pool = new Pool({
  host: 'localhost',
  user: 'postgres',
  port: 5432,
  password: 'admin',
  database: 'teste'
})
```

Executando consultas SQL

Consultas simples:

```
const result = await pool.query('SELECT * FROM pessoas;')
```

Consultas parametrizadas:

```
const query = 'INSERT INTO pessoas (nome, idade) VALUES ($1, $2);'  
const params = ['Iude Pleisteixon', 33]
```

```
const result = await pool.query(query, params)
```

```
const query = 'SELECT * FROM pessoas WHERE nome ILIKE $1;'  
const params = ['%silv%']
```

```
const result = await pool.query(query, params)
```

Executando consultas SQL

Múltiplas consultas:

```
const client = await pool.connect()

const idPerson = await client.query(
  'insert into pessoas (nome, idade) values ($1, $2) returning id;',
  ['Luizenaro Linssias', 77]
)
const result = await client.query(
  'insert into telefones (telefone, id_pessoa) values ($1, $2);',
  [08007700, idPerson]
)

client.release()
```

Obs: use essa forma para efetuar transações (transactions).

Executando consultas SQL

Consultas dinâmicas:

```
const result = await pool.query(  
  'insert into telefones (telefone, id_pessoa) values ($1, $2), ($3, $4);',  
  [99999999, 1, 88888888, 1]  
)
```

Para essas situações o pacote **pg**, até a versão 8.11.3, não tem alternativas além de formatar a string da consulta. Você também pode utilizar pacotes como **pg-format** ou **sql** para criar a string com a consulta. Abaixo temos um exemplo utilizando o **pg-format**.

```
const persons = [[99999999, 1], [88888888, 1]]  
const sqlQuery = format('insert into telefones (telefone, id_pessoa) values %L;', persons)  
  
const result = await pool.query(sqlQuery)
```

Variáveis de Ambiente

Ao inserir as credenciais do banco de dados no próprio código (hard coded), você está expondo dados sensíveis. Uma alternativa para contornar essa vulnerabilidade é utilizar variáveis de ambiente.

Listando variáveis de ambiente: (Prompt de comando: Windows)

```
set
```

Criando variáveis de ambiente: (Prompt de comando: Windows)

```
setx <var_name> <var_value> [<var_name> <var_value>]
```

Deletando variáveis de ambiente: (Prompt de comando: Windows)

```
setx <var_name> "" [&& setx <var_name> ""]
```

Variáveis de Ambiente

Resgatando variáveis de ambiente utilizando Node.js

```
process.env
```

A manipulação de variáveis de ambiente pode variar de Sistema Operacional (SO) para SO. Sendo assim, existem diversas formas de automatizar a manipulação delas com Node.js. Segue duas delas abaixo e ambas permite a declaração das variáveis em um arquivo .env:

```
// .env
```

```
var1_name = var1_value  
var2_name = var2_value
```


Variáveis de Ambiente

Pacote dotenv:

```
require('dotenv').config() // dessa maneira o pacote considera que o .env está na raiz do projeto
```

Node.js:

```
node --env-file <.env_address> <file_address>
```

Sua vez!

1. Crie a base de dados do [mini projeto de JavaScript](#) utilizando o PostgreSQL, mas com uma modificação de regra de negócio: uma pessoa agora pode possuir nenhum ou vários números e um número pertence apenas a uma pessoa.
2. Refatore a API do [mini projeto de JavaScript](#) que você criou para persistir os dados na base de dados criada na questão 1. Lembre-se que a regra de negócio mudou e isso pode afetar o funcionamento das rotas. Por exemplo, ao listar as pessoas a resposta deve ser um JSON no seguinte formato

2. ○

caso tenha pessoas:

```
[  
  {  
    "name": "Fulano",  
    "email": "fulano@gmail.com",  
    "phones": [ 40028922, 40038933 ]  
  }  
]
```

caso não tenha pessoas:

```
[]
```