

# Sequelize e sua configuração

DigitalHouse>

# Índice

1. [Instalação](#)
2. [Rotas e diretórios](#)
3. [Configuração](#)
4. [Objeto DB](#)

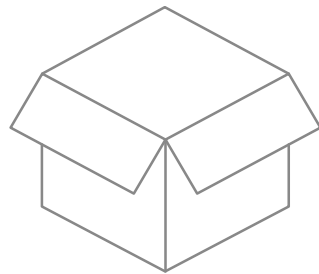


Sequelize é um ORM que nos ajuda a conectar e interagir com bancos de dados como Postgres, MySQL, MariaDB, SQLite, Microsoft SQL Server e muito mais.



# Preparando o projeto

Antes de instalar o **Sequelize**, devemos considerar que, como se trata de um pacote utilizado pelo **Node.js**, teremos que utilizar **npm**.



# 1 | Instalação

# Instalação

Dentro da pasta do projeto Node.js, execute os seguintes comandos:

```
npm install sequelize-cli -g  
  
>_ npm install sequelize  
  
npm install mysql2
```

# 2 | Rotas e diretórios

# Rotas e diretórios

Uma vez instalados os pacotes de que precisamos, é preciso estabelecer as rotas/caminhos e diretórios. Para fazer isso, precisamos criar um arquivo chamado `.sequelizerc` na raiz do projeto, e dentro dele, escrever o seguinte:

```
const path = require('path')

module.exports = {
  config: path.resolve('./database/config', 'config.js'),
  'models-path': path.resolve('./database/models'),
  'seeders-path': path.resolve('./database/seeders'),
  'migrations-path': path.resolve('./database/migrations'),
}
```



# Inicie o **Sequelize** no projeto

Para que o Sequelize possa criar todas as pastas e arquivos necessários para começar a trabalhar, precisamos executar o seguinte comando:

```
>_ sequelize init
```

# 3 | Configuração

# Configuração

Finalmente, precisamos configurar a conexão com o banco de dados. Nas pastas criadas pela Sequelize, encontraremos o arquivo **config.js** no caminho **/database/config/config.js**. Dentro dele, encontramos um JSON com credenciais padrão que devemos substituir por nossas próprias credenciais.

```
{
  "development": {
    "username": "root",
    "password": "Monito123!",
    "database": "movies_db",
    "host": "127.0.0.1",
    "dialect": "mysql",
    "operatorsAliases": false
  }
}
```

# ATENÇÃO!



Precisamos acrescentar um detalhe para não nos depararmos com um problema ao exigir o Sequelize. Para fazer isso, no arquivo config.js que editamos anteriormente, precisamos atribuir todo o JSON que modificamos ao **module.exports**.

```
JS config.js ×
database > config > JS config.js > [?] <unknown>
1  module.exports = {
2    "development": {
3      "username": "root",
4      "password": "Monito123!",
5      "database": "movies_db",
6      "host": "127.0.0.1",
7      "dialect": "mysql",
8      "operatorsAliases": false
9    },|
```

# 4 | Objeto DB

# Objeto **DB**

Dizemos que no final do arquivo **index.js**, localizado em **/database/modelos/index.js**, encontramos a exportação do objeto **DB**.

Esta é a que chamaremos sempre que quisermos usar a Sequelize para consultar nosso banco de dados.



# Documentação



**Para saber mais, você pode acessar a documentação oficial de Sequelize clicando no link a seguir:**

<https://sequelize.org/>

DigitalHouse>