

Criando nossa primeira aplicação no Sequelize

Model

Agora que criamos o banco de dados, o próximo passo é criar nosso model/modelo, um módulo que vai conter as informações do nosso mapeamento objeto-relacional (ORM), ou seja, o código JavaScript que vai representar uma tabela do banco de dados.

Assim como o banco de dados, esta tabela não vai existir no SQLite, ela será criada pelo Sequelize quando necessário.

Passo 1: Abra o repl.it no projeto que começamos na aula anterior. Crie um arquivo Node chamado **cliente.js** para inserir o código de definição do nosso modelo.

A recomendação é um nome no singular pois representa o schema para um funcionario, que usaremos várias vezes.

Passo 2: Vamos criar nosso Model cliente. O Model cliente sem acento, terá os atributos para criação da nossa tabela.

A codificação completa do Model é mostrada abaixo:

```
const Sequelize = require('sequelize');
const database = require('./db');

const Cliente = database.define('cliente', {
  id: {
    type: Sequelize.INTEGER,
    autoIncrement: true,
    allowNull: false,
    primaryKey: true
```

```
},  
  
  nome: {  
    type: Sequelize.STRING,  
    allowNull: false  
  },  
  nascimento: {  
    type: Sequelize.DATE,  
    allowNull: false  
  },  
  cidade: {  
    type: Sequelize.STRING,  
    allowNull: false  
  },  
  telefone: {  
    type: Sequelize.STRING,  
    allowNull: false  
  }  
})  
module.exports = Cliente;
```

Entendendo passo a passo o código:

Criamos duas constantes informando os arquivos que utilizaremos.

```
const Sequelize = require('sequelize');  
const database = require('./db');
```

Logo após, criamos a constante Cliente e definimos o nome do schema que iremos criar, no caso cliente, lembre-se o sequelize coloca os nomes das tabelas no plural, ou seja, a tabela será chamada de **clientes**.

```
const Cliente = database.define('cliente', {
```

O código a seguir é igual ao que já fizemos em aulas anteriores, ele possui o schema dos campos da tabela com suas restrições e tipos.

```
id: {  
  type: Sequelize.INTEGER,  
  autoIncrement: true,  
  allowNull: false,  
  primaryKey: true  
},  
nome: {  
  type: Sequelize.STRING,  
  allowNull: false  
},  
nascimento: {  
  type: Sequelize.DATE,  
  allowNull: false  
},  
cidade:{  
  type: Sequelize.STRING,  
  allowNull:false  
},  
telefone:{  
  type: Sequelize.STRING,  
  allowNull:false  
}  
})
```

Para finalizar, exportamos o módulo Cliente para ser usado na nossa aplicação.

```
module.exports = Cliente;
```

Finalizamos nosso model Cliente.

Até a próxima aula...