

NodeJs

Sequelize

O sequelize é um ORM que abstrai toda a camada de banco de dados.

Terminal na pasta do projeto:

```
npm install --save sequelize  
npm install mysql2
```

Importar sequelize:

```
const Sequelize = require('sequelize');
```

Para realizar a conexão com banco, O sequelize recebe como parâmetros o nome do banco de dados, o usuário, a senha e um **Json** entre as chaves.

No caso abaixo os valores se encontram armazenados em variáveis, caso deseje digitar direto no parâmetro é necessário estar entre aspas.

O host representa o local onde o banco está rodando, o dialect é basicamente qual banco, no caso o mysql:

```
const sequelize = new Sequelize(nomeDb, user, password, {  
  host:"localhost",  
  dialect:'mysql'  
});
```

A função authenticate basicamente verifica a conexão, o then e o catch recebem funções como parâmetros que são executadas de acordo se a conexão foi bem sucedida ou não.

```
sequelize.authenticate().then(function() {  
  console.log("conexão realizada com sucesso")  
}).catch(function(erro) {  
  console.log("Falha ao se conectar "+ erro);  
});
```

código completo:

```
const Sequelize = require('sequelize');  
  
var nomeDb= "Teste";  
var user = "root";  
var password = ""  
  
const sequelize = new Sequelize(nomeDb, user, password, {  
  host:"localhost",
```

```

dialect: 'mysql'
});

sequelize.authenticate().then(function() {
  console.log("conexão realizada com sucesso")
}).catch(function(erro) {
  console.log("Falha ao se conectar "+ erro);
});

```

Models

O model basicamente é uma abstração que representa uma tabela SQL.

definindo um Model:

```

const usuario = sequelize.define('usuarios', {
  nome: {
    type: Sequelize.STRING
  },
  sobrenome: {
    type: Sequelize.STRING
  },
  idade: {
    type: Sequelize.INTEGER
  },
  email: {
    type: Sequelize.STRING
  }
})

```

Tabela criada com o model referente:

#	Field	Type	Null	Key	Default	Extra
1	id	int	NO	PRI	NULL	auto_increment
2	nome	varchar(255)	YES		NULL	
3	sobrenome	varchar(255)	YES		NULL	
4	idade	int	YES		NULL	
5	email	varchar(255)	YES		NULL	
6	createdAt	datetime	NO		NULL	
7	updatedAt	datetime	NO		NULL	

Realizando INSERT na tabela criada:

```
usuario.create({  
  nome: "Nathan",  
  sobrenome: "Azevedo",  
  idade: "19",  
  email: "nathan.azevedo28@gmail.com"  
})
```