Introduction to Sequelize ORM for Node.js

Patrick Schroeder
SOFTWARE ENGINEER

Overview

2 sections

Core features

- New project

- Models

- Associations

Sample app

- Build out backend

Prerequisites

Prerequisites

JavaScript

- Arrays, variables, objects

Node.js

Visual Studio code

Helpful to know

- SQL

- Angular

What Is Sequelize?

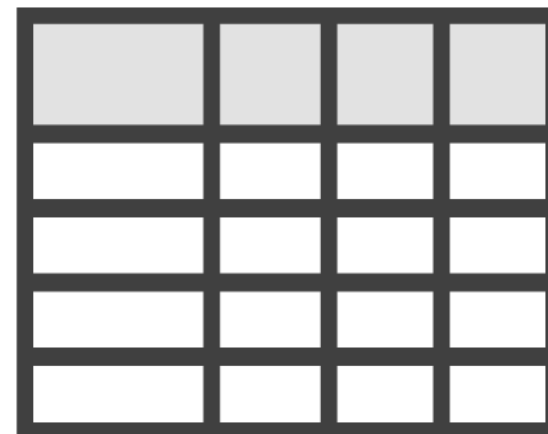Sequelize is a promise-based Object Relational Mapper.

Sequelize ORM

Model

Transform into Table

Table

SQL Database Explained

SQL Database

Stores data in tables

Tables have columns and rows

Each row contains a new entry

```
[
  {
    "id": 1,
    "title": 'First post',
    "userId": 1,
    "User": {
       "name": 'Buckminster'
       "email": '123@email.com'
    }
  }
]
```

◄ Results of join

SQL Databases

Tables

Columns and rows
- Primary key

JOIN table

Sequelize Is Promise Based

Promise

Represents the eventual completion (or failure) of an asynchronous operation, and its resulting value.

Then / Catch

Returns the result of a promise. 'Then' for success and 'catch' for error.

```
User.findById(1)                      ◄ Find user

    .then((user) => {                 ◄ Respond with user

        res.json(user);

    })

    .catch((err) => {                 ◄ Catch any error

        console.log(err);

    })
```

```
app.get('/async', async (req, res)

    => {

 let users;

 try {

    users = await User.findById(1);

 } catch(err) {

    console.log(err);

 }

 return res.json(users);

});
```
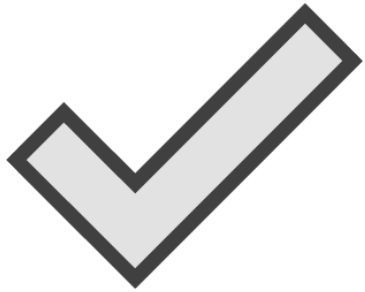
◄ Async/Await example

Benefits of Sequelize

Sequelize Benefits

Simplifies code

JavaScript, not SQL

Robust API

Security

Community

CLI tool

Summary

Sequelize fundamentals

SQL databases

Promises