npm install --save sequelize

Drivers :

$ npm install --save pg pg-hstore # Postgres

$ npm install --save mysql2

$ npm install --save mariadb

$ npm install --save sqlite3

$ npm install --save tedious # Microsoft SQL Server

Setup

const { Sequelize } = require('sequelize');

const sequelize = new Sequelize('database', 'username', 'password', {

host: 'localhost',

dialect: /\* one of 'mysql' | 'mariadb' | 'postgres' | 'mssql' \*/

});

Test Connection

try {

await sequelize.authenticate();

console.log('Connection has been established successfully.');

} catch (error) {

console.error('Unable to connect to the database:', error);

}

Close Connection

sequelize.close()

async & return Promise

const { Sequelize, Op, Model, DataTypes } = require("sequelize");

config/database.js

const Sequelize = require("sequelize");

const db = new Sequelize("learn\_sqlize", "hbstudent", "hbstudent", {

    dialect: "mysql",

    host: "localhost",

});

module.exports = db;

models/Course.js

const { DataTypes, Model } = require("sequelize");

const db = global.import("./config/database");

const Course = db.define("course", {

    id: {

        type: DataTypes.INTEGER,

        autoIncrement: true,

        allowNull: false,

        primaryKey: true,

    },

    title: {

        type: DataTypes.STRING,

        allowNull: false,

        unique: true,

    },

});

module.exports = Course;

app.js

const db = require("./config/database");

db.sync({

    // force: true,

})

    .then((rs) => {

        console.log("Đã kết nối thành công với DB");

        app.listen(PORT, () => {

            console.log(`Server is running on PORT: ${PORT}`);

        });

    })

    .catch((err) => {

        console.error(err);

    });