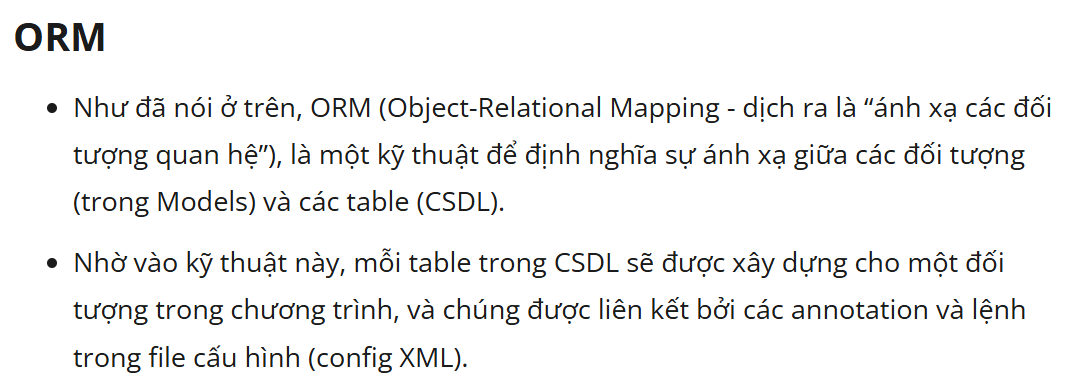
Sequelize

# 1, Giới thiệu qua

* Sequelize là ORM (Object Relational Mapping) dùng cho server Nodejs.
* Nó giống chức năng của Hibernate bên Java.



(trích)

* Đọc về Hibernate ở đây: <https://viblo.asia/p/nhat-ky-hoc-hibernate-tu-so-0-7ymJXGQa4kq>
* Hiểu ngắn gọn hơn, ORM hay ở đây là sequelize, là một cách thức để liên kết database với server nodejs.

# 2, Cài đặt

* Đảm bảo đã cài Nodejs.
* Cài với npm/yarn: $ npm install –save sequelize
* Cài thêm mysql2: $ npm install –save mysql2
* Sau đó nếu yêu cầu tải thêm cái gì thì npm install nốt cái đó.

**Các bước tiếp theo: bây giờ cần thiết lập và test kết nối với DB, sau đó định nghĩa các Models và mối quan hệ (association), đồng bộ hóa models và database (sync), sau đó thực hiện CRUD.**

# 3, Kết nối với Database

* Vì ở đây đang trong trường hợp có Database sẵn nên sẽ làm theo hướng có sẵn.

Các bước:

* Import thư viện
* Tạo instance Sequelize
* Thử kết nối. (tùy chọn)

**Phần kết nối này sẽ trả về một instance của đối tượng Sequelize (đại diện cho kết nối tới database)**

Phần kết nối với database sẽ được viết ở config/dbconnect.js.

|  |
| --- |
| /\* Import thư viện\*/  const { Sequelize } = require('sequelize');  /\*tạo instance Sequelize\*/  // Cách 1: Kết nối bằng URI  const sequelize1 = new Sequelize('mysql://username:password@localhost:3306/databasename');  // Cách 2: Truyền tham số riêng lẻ  const sequelize2 = new Sequelize('databasename', 'username', 'password', {  host: 'localhost', // Địa chỉ máy chủ MySQL  dialect: 'mysql', // Sử dụng MySQL làm cơ sở dữ liệu  logging: false, // Tắt log câu lệnh SQL (tuỳ chọn)  pool: {  max: 5, // Số kết nối tối đa trong pool  min: 0, // Số kết nối tối thiểu trong pool  acquire: 30000, // Thời gian tối đa để lấy một kết nối (ms)  idle: 10000 // Thời gian tối đa một kết nối có thể ở trạng thái nhàn rỗi (ms)  }  });  // Kiểm tra kết nối  (async () => {  try {  await sequelize2.authenticate();  console.log('Kết nối MySQL thành công!');  } catch (error) {  console.error('Kết nối MySQL thất bại:', error);  }  })(); |

Đây là cú pháp mẫu, còn code thật thì thêm data vào (từ file env) và xuất modules

|  |
| --- |
| //khai báo thư viện sequelize và dotenv  const { Sequelize } = require('sequelize');  require('dotenv').config();  //tạo instance sequelize  const sequelize = new Sequelize(process.env.DB\_NAME, process.env.DB\_USER, process.env.DB\_PASSWORD, {    host: process.env.DB\_HOST || 'localhost',    dialect: 'mysql',    logging: false,  });  // Kiểm tra thử kết nối  (async () => {    try {      await sequelize.authenticate();      console.log('Kết nối cơ sở dữ liệu thành công!');    } catch (error) {      console.error('Không thể kết nối cơ sở dữ liệu:', error);    }  })();  module.exports = sequelize; |

Với nội dung file /env như sau:

|  |
| --- |
| DB\_NAME=tutorium  DB\_USER=konan  DB\_PASSWORD=1234 |

# 4, Tạo Models

* Một “model” là một đối tượng (trong code) đại diện cho một bảng trong database.
* Một model là một class extend class [Model](https://sequelize.org/api/v6/class/src/model.js~model): là một abstract class xây dựng để đại diện cho một bảng trong cơ sở dữ liệu.
* Tạm thời biết như thế và không cần suy xét kỹ class Model có những gì.
* Bây giờ cần mô phỏng các bảng trong database bằng code js với sequelize.

Đầu tiên phải biết được quy tắc đặt tên các đối tượng.

## 1, đặt tên Model và Table

* Tên bảng: ở dạng số nhiều, dùng chữ thường, dùng snake\_case và tránh những từ khóa sql.
* Tên model: ở dạng số ít và PascalCase, không cần hậu tố "model"

## 2, xây dựng Model

* Có 2 cách để tạo ra một model: một là dùng hàm có sẵn của instance sequelize, 2 là tạo class để extend class tên là Model (như ở trên nói).
* 2 cách này đều cho ra kết quả như nhau nhưng xét về cấu trúc code thì cách 2 phù hợp hơn cho code nhiều Model => chốt cách 2 và không để ý tới cách 1 nữa.

Các bước để code ra một class model:

* Import thư viện và instance của sequelize đã có được sau bước 1.
* Định nghĩa class model (new class)
* Init model đó với những đặc điểm của bảng tương ứng.

**Kết quả của phần tạo model sẽ là một/những class model đại diện cho các bảng trong database (chưa có kết nối giữa các bảng).**

Code mẫu cho một model (nhớ đặt tên cho chuẩn)

|  |
| --- |
| /\* Import thư viện và instance sequellize đã tạo thành công ở bên trên\*/  const { Sequelize, DataTypes, Model } = require('sequelize');  const sequelize = require('../config/db’); // Import kết nối Sequelize từ file db đã code bên trên  /\* Định nghĩa class Model \*/  class ModelName extends Model {}  /\* Khởi tạo Model với các thuộc tính \*/  ModelName.init({  id: {  type: DataTypes.INTEGER,  autoIncrement: true,  primaryKey: true,  },  column1: {  type: DataTypes.STRING,  //dưới này là các thuộc tính  },  /\*các cột khác…\*/  }, {  // phần này là thuộc tính của bảng  sequelize, // Instance của Sequelize  modelName: 'ModelName', // Tên Model (bảng trong database sẽ là 'ModelNames')  tableName: 'custom\_table\_name', // Nếu muốn đặt tên bảng tùy chỉnh  });  module.exports = ModelName; |

Vd cụ thể:

Với table User như sau:

|  |
| --- |
| CREATE TABLE `Users`(  `userid` BIGINT NOT NULL AUTO\_INCREMENT PRIMARY KEY,  `username` VARCHAR(255) NOT NULL UNIQUE,  `password` VARCHAR(255) NOT NULL,  `email` VARCHAR(255) NOT NULL UNIQUE,  `imgurl` TEXT NOT NULL,  `dateofbirth` DATE NOT NULL  ); |