

Sequelize

- Sequelize est un des ORM proposés en Javascript pour les BDR comme MySQL, Postgres,

Installing

Sequelize is available via [npm](#) (or [yarn](#)).

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

You'll also have to manually install the driver for your database of choice:

```
# One of the following:
$ 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
```



<https://sequelize.org/v5/>

<https://numa-bord.com/miniblog/nodejs-exemples-dutilisation-avec-lorm-sequelize/>

Setting up a connection

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

// Option 1: Passing parameters separately
const sequelize = new Sequelize('database', 'username', 'password', {
  host: 'localhost',
  dialect: /* one of 'mysql' | 'mariadb' | 'postgres' | 'mssql' */
});

// Option 2: Passing a connection URI
const sequelize = new Sequelize('postgres://user:pass@example.com:5432/dbname');
```

Sequelize

- Mise en œuvre
 - Création d'un projet node express à base de vue ejs


```
express sequelize_mysql
```
 - Installation des dépendances NPM dans le dossier sequelize_mysql

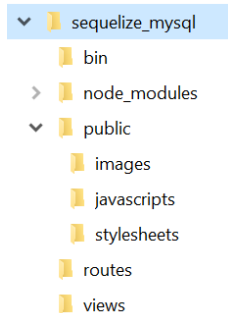
```
npm install
```

```
create : sequelize_mysql\bin\
create : sequelize_mysql\bin\www

change directory:
> cd sequelize_mysql

install dependencies:
> npm install

run the app:
> SET DEBUG=sequelize-mysql:* & npm start
```



Sequelize



Mise en œuvre

- Installation de Mysql

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

- Installation de Sequelize

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

- Installation de Sequelize-cli permettant l'initialisation de sequelize pour générer, entre autre le fichier de config Mysql

```
npm install --save mysql sequelize-cli
```

```
node_modules\.bin\sequelize init
```

```
D:\workspace\Formation\Node.js\exples\sequelize_mysql> node_modules\.bin\sequelize init
sequelize CLI [Node: 10.15.1, CLI: 5.5.1, ORM: 5.21.2]
Created "config/config.json"
Successfully created models folder at "D:\workspace\Formation\Node.js\exples\sequelize_mysql\models".
Successfully created migrations folder at "D:\workspace\Formation\Node.js\exples\sequelize_mysql\migrations".
Successfully created seeders folder at "D:\workspace\Formation\Node.js\exples\sequelize_mysql\seeders".
```

L'application se lancera avec :

<http://localhost:3000/>

```
sequelize_mysql
├── bin
├── config
├── migrations
├── models
├── node_modules
├── public
├── images
├── javascripts
├── stylesheets
├── routes
├── seeders
└── views
```

Sequelize



Mise en œuvre

- Corriger config/config.json

```
"username": "root",
"password": null,
"database": "formation",
"host": "127.0.0.1",
"dialect": "mysql",
"operatorsAliases": false
```

- Génération des modèles depuis la base de données

- Installation de sequelize-auto (nécessite mysql)

```
npm install --save sequelize-auto
```

- Génération des entités depuis les tables

```
npx sequelize-auto -o './models' -d formation -h localhost -u root -p 3306
```

Il est possible de générer le modèle pour une table :
-t [tableName]

La version de sequelize-auto est détectée vulnérable,
désinstaller la aussitôt

```
sequelize_mysql > models
Nom
├── activite.js
├── categorie.js
├── client.js
├── commande.js
├── commande_produit.js
├── droit.js
├── facture.js
├── fournisseur.js
├── index.js
├── livraison.js
├── maire.js
├── paiement.js
├── pays.js
├── produit.js
├── produit_categorie.js
├── responsable.js
├── service.js
├── societe.js
├── user.js
├── utilisateur.js
└── ville.js
```

Sequelize



- Mise en œuvre

- Exemple de code généré depuis la table user

#	Nom	Type
1	id_user	int(11)
2	login	varchar(255)
3	nom	varchar(255)
4	password	varchar(255)
5	prenom	varchar(255)
6	email	varchar(255)
7	profil	bit(1)

```
module.exports = function(sequelize, DataTypes) {
  return sequelize.define('user', {
    id_user: {
      type: DataTypes.INTEGER(11),
      allowNull: false,
      primaryKey: true,
      autoIncrement: true
    },
    login: {
      type: DataTypes.STRING(255),
      allowNull: true
    },
    nom: {
      type: DataTypes.STRING(255),
      allowNull: true
    },
    password: {
```

+ Options

	id_user	login	nom	password	prenom	email	profil
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	1	dp	Dupond	abc	Paul	dp@gmail.com	1
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	2	nouveau login	Un user	azerty	david	pl@yahoo.fr	1
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	3	lanrand	Le Grand	unMntrdePass	Pierre	lanrand@free.fr	0

Sequelize



- Mise en œuvre

- L'auto-génération a également créé un fichier index.js

```
'use strict';
const fs = require('fs');
const path = require('path');
const Sequelize = require('sequelize');
const basename = path.basename(__filename);
const env = process.env.NODE_ENV || 'development';
const config = require(__dirname + '/../config/config.json')[env];
const db = {};

let sequelize;
if (config.use_env_variable) {
  ...

db.sequelize = sequelize;
db.Sequelize = Sequelize;

module.exports = db;
```

Sequelize



- Mise en œuvre
 - Utilisation
 - Modifier users.js dans le dossier routes

```
const express = require('express');
const db = require('../models/index');
var User = require('../models/user')(db.sequelize, db.Sequelize.DataTypes);
const router = express.Router();
/* GET users listing. */
router.get('/', function (req, res) {
  return User.findAll().then(result => {res.send(result);})
});
router.get('/:id', function (req, res) {
  let id = parseInt(req.params.id);
  return User.findByPk(id).then(result => {res.send(result);})
});
module.exports = router;
```

Sequelize



- Mise en œuvre
 - Utilisation
 - Résoudre l'erreur «Unhandled rejection SequelizeDatabaseError: Champ 'createdAt' inconnu dans field list»

en ajoutant l'option timestamps:false à la configuration

```
"development": {
  "username": "root",
  "password": null,
  "database": "formation",
  "host": "127.0.0.1",
  "dialect": "mysql",
  "operatorsAliases": 0,
  "define": { "timestamps": false }
},
```

config.json

Utiliser 0 à la place de false pour éviter le warning

npm

```
(node:18484) [SEQUELIZE0004] DeprecationWarning: A boolean value was passed to options.operatorsAliases. This is a no-op with v5 and should be removed.
```

Sequelize



- **Mise en œuvre**

- **Tester**

```
\sequelize_mysql>npm start
```

localhost:3000/users

JSON Données brutes En-têtes

Enregistrer Copier Tout réduire Tout développer Filtrer le JSON

▼ 0:

```
{
  "id_user": 1,
  "login": "dp",
  "nom": "Dunond"
}
```

localhost:3000/users/3

Données brutes En-têtes

Copier Tout réduire Tout développer Filtrer le JSON

```
{
  "id_user": 3,
  "login": "legrand",
  "nom": "Le Grand",
  "password": "unMotdePass",
  "prenom": "Pierre",
  "email": "legrand@free.fr",
  "profil": false
}
```

- **Remarque : exécuter en mode p**

- **Désactiver les logs**

```
"operatorsAliases": 0,
"logging": false,
"define":{"timestamps": false}
```

```
\sequelize_mysql>set NODE_ENV=production
\sequelize_mysql>npm start
```