

Sequelize

Node.js ORM

Sequelize is a promise-based Node.js ORM
for Postgres, MySQL, MariaDB, SQLite and
Microsoft SQL Server.

Getting Started

First, you might need install MySQL/MariaDB

Ubuntu MySQL

```
sudo apt install mysql
```

Ubuntu MariaDB

```
sudo apt install mysql
```

macOS MariaDB

```
brew install mysql
```

macOS MariaDB

```
brew install mariadb
```

Install PostgreSQL

Ubuntu

```
sudo apt install postgresql postgresql-contrib  
sudo su - postgres  
psql
```

macOS

```
brew install postgresql  
psql postgres  
create database [username]
```

```
npm i sequelize  
npm i sequelize-cli
```

```
npm i pg pg-hstore
```

```
npm i mysql2
```

```
npm i mariadb
```

```
npm i sqlite3
```

```
npm i tedious
```

Migrations


```
npx sequelize-cli init
```

Environment Variable

DotEnv

npm i dotenv

Rename config.json to config.js

```
require('dotenv').config()
```

```
module.exports = {  
  "development": {  
    "username": process.env.DB_USERNAME,  
    "password": process.env.DB_PASSWORD,  
    "database": process.env.DB_DATABASE,  
    "host": process.env.DB_HOST,  
    "dialect": "postgres",  
  },  
}
```

Create new file ".env" to root project folder

```
DB_USERNAME=root
```

```
DB_PASSWORD=
```

```
DB_DATABASE=my_app
```

```
DB_HOST=127.0.01
```

You may want to change
sequelize's folders

Add .sequelizerc to root project folder

// Optional!

```
const path = require('path');
```

```
module.exports = {  
  'config': path.resolve('config', 'database.json'),  
  'models-path': path.resolve('db', 'models'),  
  'seeders-path': path.resolve('db', 'seeders'),  
  'migrations-path': path.resolve('db', 'migrations')  
}
```


Generate model

```
npx sequelize model:generate \  
--name user \  
--attributes name:string,email:string,is_active:boolean,gender:integer \  
--underscored
```

```
npx sequelize model:generate \  
--name Article \  
--attributes title:string,body:text,user_id:integer \  
--underscored
```

Generate seeder

```
npx sequelize seed:generate --name user_seeder
```

FindAll

```
const User = require('../models').user

router.get('/', async (req, res, next) => {

  const result = await User.findAll()

  res.json(result)

});
```

FindAll where

```
const User = require('../models').user

router.get('/', async (req, res, next) => {

  const result = await User.findAll({
    where: { is_active: true }
  })

  res.json(result)

});
```

FindOne

```
const User = require('../models').user

router.get('/', async (req, res, next) => {

  const result = await User.findOne({
    where: { id: 190 }
  })

  res.json(result)

});
```

Find By Primary Key

```
const User = require('../models').user

router.get('/', async (req, res, next) => {

  const result = await User.findByPk(190)

  res.json(result)

});
```

Raw Queries

```
const sequelize = require('..../models').sequelize

router.get('/', async (req, res, next) => {

  const sql = 'select * from users'
  const [result, meta] = await sequelize.query(sql)

  res.json(result)

});
```

Raw Queries with prepared statement

```
const sequelize = require('../models').sequelize

router.get('/', async (req, res, next) => {
  const sql = 'select * from users where id = ?'

  const [result, meta] = await sequelize.query(sql, {
    replacements: [ 190 ],
    type: sequelize.QueryTypes.SELECT
  })

  res.json(result)
});
```


Raw Queries with multi-prepared statement

```
const sequelize = require('../models').sequelize

router.get('/', async (req, res, next) => {
  const sql = 'select * from users where email = ? and is_active = ?'

  const [result, meta] = await sequelize.query(sql, {
    replacements: [ mulia@nasution.dev, true ],
    type: sequelize.QueryTypes.SELECT
  })

  res.json(result)
});
```

OR operator

```
// SELECT * FROM users WHERE gender = 'male' AND status = true
```

```
const User = require('../models').user  
const Op = require('Sequelize').Op
```

```
const result = await User.findAll({  
  where: {  
    [Op.or]: { gender: 'male', status: true }  
  }  
})
```

LIKE operator

```
// SELECT * FROM users WHERE name LIKE %mul%
```

```
const User = require('../models').user
const Op = require('Sequelize').Op
```

```
const result = await User.findAll({
  where: {
    name: {
      [Op.iLike]: '%mul%'
    }
  }
})
```

Relationship/Associate

Define relationship in a model

```
// models/user.js  
// ...
```

```
user.associate = function(models) {  
  user.hasMany(models.article)  
};
```

Define relationship in a model

```
// models/article.js
// ...

article.associate = function(models) {
  article.belongsTo(models.user)
};
```

Usage

```
const user = require('../models').user  
const article = require('../models').article
```

```
user.findAll({  
  include: article  
})
```

Usage

```
const user = require('../models').user  
const article = require('../models').article
```

```
article.findAll({  
  include: user  
})
```


<https://sequelize.org/v5/manual/models-usage.html>

Questions?