

10-3-2023

Investigación: librerías para conectar una aplicación Web con MongoDB Atlas

Bases de datos para cómputo en la nube

Alumno: Rodriguez Rodriguez Cristian

Cuatrimestre: 5

Grupo: C

Carrera: DSM

Profesor: Hugo Omar Alejandro Sánchez

MongoDB es una base de datos NoSQL muy popular para aplicaciones web. Para conectarse a MongoDB Atlas, que es un servicio de base de datos en la nube de MongoDB, existen varias librerías que proporcionan una interfaz para interactuar con la base de datos. En esta investigación, se describen las librerías más utilizadas para conectar una aplicación web con MongoDB Atlas.

1. Mongoose

Mongoose es una librería de modelado de objetos para MongoDB que proporciona una API sencilla para interactuar con la base de datos. Permite definir esquemas y modelos, realizar consultas y actualizaciones de forma sencilla, y también incluye funciones para validación y transformación de datos. Mongoose es compatible con Node.js y se puede utilizar en aplicaciones web basadas en este lenguaje.

Para conectarse a MongoDB Atlas con Mongoose, se debe proporcionar la URL de conexión a la base de datos, que se puede obtener en la consola de MongoDB Atlas. Por ejemplo, para conectarse a una base de datos llamada "mydatabase" con el usuario "myuser" y la contraseña "mypassword", se puede hacer lo siguiente:

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

mongoose.connect('mongodb+srv://myuser:mypassword@mycluster.mongodb.net/mydatabase?retryWrites=true&w=majority', {
  useNewUrlParser: true,
  useUnifiedTopology: true
})
.then(() => console.log('Conexión exitosa'))
.catch(error => console.error(error));
```

2. MongoDB Node.js Driver

MongoDB Node.js Driver es la librería oficial de MongoDB para Node.js. Proporciona una API de bajo nivel para interactuar con la base de datos, lo que permite un mayor control y flexibilidad. También es compatible con MongoDB Atlas y proporciona funciones para realizar consultas, actualizaciones y operaciones en la base de datos.

Para conectarse a MongoDB Atlas con MongoDB Node.js Driver, se debe proporcionar la URL de conexión a la base de datos, al igual que con Mongoose. Por ejemplo, para conectarse a una base de datos llamada "mydatabase" con el usuario "myuser" y la contraseña "mypassword", se puede hacer lo siguiente:

```
const { MongoClient } = require('mongodb');

const uri =
'mongodb+srv://myuser:mypassword@mycluster.mongodb.net/mydatabase?retryWrites=true&w=majority';

const client = new MongoClient(uri, {
  useNewUrlParser: true,
  useUnifiedTopology: true
});

client.connect()
  .then(() => console.log('Conexión exitosa'))
  .catch(error => console.error(error));
```

3. Monk

Monk es una librería de MongoDB para Node.js que proporciona una API simple y fácil de usar. Utiliza la librería oficial MongoDB Node.js Driver y proporciona una interfaz más sencilla para interactuar con la base de datos. También es compatible con MongoDB Atlas y proporciona funciones para realizar consultas, actualizaciones y operaciones en la base de datos.

Para conectarse a MongoDB Atlas con Monk, se debe proporcionar la URL de conexión a la base de datos, al igual que con las otras librerías. Por ejemplo, para conectarse a una base de datos llamada "mydatabase" con el usuario "myuser" y la contraseña "mypassword", se puede hacer lo siguiente:

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

const uri =
'mongodb+srv://myuser:mypassword@mycluster.mongodb.net/mydatabase?retryWrites=true&w=majority';

const db = monk(uri);

db.then(() => console.log('Connected to database'))
  .catch((err) => console.log('Error connecting to database:', err));
```

4. Mongodb

Mongodb es otra librería de bajo nivel para Node.js que proporciona una API para interactuar con MongoDB. Es similar a MongoDB Node.js Driver, pero proporciona una interfaz más fácil de usar y está más enfocada en la experiencia del desarrollador. También es compatible con MongoDB Atlas y proporciona funciones para realizar consultas, actualizaciones y operaciones en la base de datos.

Para conectarse a MongoDB Atlas con Mongodb, se debe proporcionar la URL de conexión a la base de datos, al igual que con las otras librerías. Por ejemplo, para conectarse a una base de datos llamada "mydatabase" con el usuario "myuser" y la contraseña "mypassword", se puede hacer lo siguiente:

```
const { MongoClient } = require('mongodb');

const uri =
'mongodb+srv://myuser:mypassword@mycluster.mongodb.net/mydatabase?retryWrites=true&w=majority';

const client = new MongoClient(uri, {
  useNewUrlParser: true,
  useUnifiedTopology: true
});

client.connect()
  .then(() => console.log('Conexión exitosa'))
  .catch(error => console.error(error));
```

5. Node-mongodb-native

Node-mongodb-native es otra librería de bajo nivel para Node.js que proporciona una API para interactuar con MongoDB. Es similar a MongoDB Node.js Driver y Mongodb, pero proporciona una interfaz más enfocada en la simplicidad y el rendimiento. También es compatible con MongoDB Atlas y proporciona funciones para realizar consultas, actualizaciones y operaciones en la base de datos.

Para conectarse a MongoDB Atlas con Node-mongodb-native, se debe proporcionar la URL de conexión a la base de datos, al igual que con las otras librerías. Por ejemplo, para conectarse a una base de datos llamada "mydatabase" con el usuario "myuser" y la contraseña "mypassword", se puede hacer lo siguiente:

```
const { MongoClient } = require('mongodb');

const uri =
'mongodb+srv://myuser:mypassword@mycluster.mongodb.net/mydatabase?retryWrites=true&w=majority';

const client = new MongoClient(uri, {
  useNewUrlParser: true,
  useUnifiedTopology: true
});

client.connect()
  .then(() => console.log('Conexión exitosa'))
  .catch(error => console.error(error));
```

6. PyMongo (Python)

PyMongo es una librería de Python que proporciona una API para interactuar con MongoDB. Es la librería más utilizada para conectarse a MongoDB Atlas desde Python. PyMongo proporciona funciones para realizar consultas, actualizaciones y operaciones en la base de datos.

Para conectarse a MongoDB Atlas con PyMongo, se debe proporcionar la URL de conexión a la base de datos. Por ejemplo, para conectarse a una base de datos llamada "mydatabase" con el usuario "myuser" y la contraseña "mypassword", se puede hacer lo siguiente:

```
from pymongo import MongoClient

uri =
'mongodb+srv://myuser:mypassword@mycluster.mongodb.net/mydatabase?retryWrites=true&w=majority'
client = MongoClient(uri)

db = client.mydatabase
```

7. MongoClient (Java)

MongoClient es una librería de Java que proporciona una API para interactuar con MongoDB. Es la librería más utilizada para conectarse a MongoDB Atlas desde Java. MongoClient proporciona funciones para realizar consultas, actualizaciones y operaciones en la base de datos.

Para conectarse a MongoDB Atlas con MongoClient, se debe proporcionar la URL de conexión a la base de datos. Por ejemplo, para conectarse a una base de datos llamada "mydatabase" con el usuario "myuser" y la contraseña "mypassword", se puede hacer lo siguiente:

```
import com.mongodb.MongoClient;
import com.mongodb.MongoClientURI;
import com.mongodb.client.MongoDatabase;

String uri =
"mongodb+srv://myuser:mypassword@mycluster.mongodb.net/mydatabase?retryWrites=true&w=majority";
MongoClientURI clientURI = new MongoClientURI(uri);
MongoClient client = new MongoClient(clientURI);

MongoDatabase database = client.getDatabase("mydatabase");
```

8. pymongo (Ruby)

pymongo es una librería de Ruby que proporciona una API para interactuar con MongoDB. Es la librería más utilizada para conectarse a MongoDB Atlas desde Ruby. pymongo proporciona funciones para realizar consultas, actualizaciones y operaciones en la base de datos.

Para conectarse a MongoDB Atlas con pymongo, se debe proporcionar la URL de conexión a la base de datos. Por ejemplo, para conectarse a una base de datos llamada "mydatabase" con el usuario "myuser" y la contraseña "mypassword", se puede hacer lo siguiente;

```
require 'mongo'

uri =
"mongodb+srv://myuser:mypassword@mycluster.mongodb.net/mydatabase?retryWrites=true&w=majority"
client = Mongo::Client.new(uri)

db = client.database
```

Fuentes

- MongoDB. (2021). MongoDB drivers. MongoDB, Inc. Recuperado de <https://docs.mongodb.com/drivers/>
- Bradshaw, S., Brazil, E., & Chodorow, K. (2019). MongoDB: the definitive guide: powerful and scalable data storage. O'Reilly Media.
- Monk. (2021). Getting started. Automattic, Inc. Recuperado de https://automattic.github.io/monk/docs/GETTING_STARTED.html
- Mongoose. (2021). Quick start. Recuperado de <https://mongoosejs.com/docs/index.html>
- MongoDB. (2021). Node.js MongoDB driver API reference. Recuperado de <https://mongodb.github.io/node-mongodb-native/4.1/api/>
- Node.js MongoDB Driver Team. (2021). Node.js MongoDB driver documentation. Recuperado de <https://mongodb.github.io/node-mongodb-native/>
- PyMongo. (2021). PyMongo documentation. Recuperado de <https://pymongo.readthedocs.io/en/stable/>
- MongoDB. (2021). MongoDB Java Driver documentation. Recuperado de <https://mongodb.github.io/mongo-java-driver/>
- Ruby (MongoDB Ruby Driver):
MongoDB. (2021). MongoDB Ruby driver documentation. Recuperado de <https://docs.mongodb.com/drivers/ruby/>