

package ec.edu.espe.dpexsystem.controller;

import com.google.gson.Gson;

import com.mongodb.ConnectionString;

import com.mongodb.MongoClientSettings;

import com.mongodb.client.FindIterable;

import com.mongodb.client.MongoClient;

import com.mongodb.client.MongoClients;

import com.mongodb.client.MongoCollection;

import com.mongodb.client.MongoDatabase;

import ec.edu.espe.dpexsystem.model.User;

import ec.edu.espe.dpexsystem.model.User.UserType;

import java.io.FileWriter;

import java.io.IOException;

import java.util.Scanner;

import org.bson.Document;

import java.util.UUID;

/\*\*

\*

\* @author mate-

\*/

public class UserMongoDBConection {

public static void mongoConection() {

System.out.println("User's Menu");

String connectionString = "mongodb+srv://materoge32:Fpztsb3yVp417t3Q@cluster0.znbumcs.mongodb.net/?retryWrites=true&w=majority";

MongoClientSettings settings = MongoClientSettings.builder()

.applyConnectionString(new ConnectionString(connectionString))

.build();

try (MongoClient mongoClient = MongoClients.create(settings)) {

try {

MongoDatabase database = mongoClient.getDatabase("DPEXUsers");

MongoCollection<Document> collection = database.getCollection("User");

} catch (Exception e) {

e.printStackTrace();

}

}

}

private static void addUserToMongoDB(MongoCollection<Document> collection, Scanner scanner) {

System.out.print("Ingresa un ususario: ");

String username = scanner.next();

System.out.print("Ingresa una contraseña: ");

String password = scanner.next();

System.out.print("Enter name: ");

String name = scanner.next();

System.out.print("Enter lastname: ");

String lastName = scanner.next();

User newUser = new User(username, password,UserType.EMPLOYEE,name,lastName);

saveUserToMongoDB(newUser, collection);

System.out.println("User added successfully!");

}

private static void readUsersFromMongoDB(MongoCollection<Document> collection) {

System.out.println("Users in MongoDB:");

// Retrieve all documents from the collection

FindIterable<Document> documents = collection.find();

for (Document document : documents) {

User user = convertDocumentToUser(document);

System.out.println(user);

}

}

private static User convertDocumentToUser(Document document) {

String username = document.getString("username");

String password = document.getString("password");

String firstName = document.getString("First name");

String lastName = document.getString("Last name");

return new User(username, password,UserType.EMPLOYEE,firstName,lastName);

}

private static void updateUserInMongoDB(MongoCollection<Document> collection, Scanner scanner) {

System.out.println("Buffer content: " + scanner.nextLine());

System.out.print("Enter username of the user to update: ");

String usernameToUpdate = scanner.nextLine();

System.out.println("Enter updated information:");

System.out.print("New username: ");

String newUsername = scanner.nextLine();

System.out.print("New password: ");

String newPassword = scanner.nextLine();

// Update the user in MongoDB

updateExistingUser(collection, usernameToUpdate, newUsername, newPassword);

System.out.println("User updated successfully!");

}

private static void updateExistingUser(MongoCollection<Document> collection, String usernameToUpdate, String newUsername, String newPassword) {

// Find the user by the old username

Document query = new Document("username", usernameToUpdate);

Document existingUser = collection.find(query).first();

if (existingUser != null) {

// Update the user's information

existingUser.put("username", newUsername);

existingUser.put("password", newPassword);

// Save the updated document back to MongoDB

collection.replaceOne(query, existingUser);

} else {

System.out.println("User not found!");

}

}

private static void deleteUserFromMongoDB(MongoCollection<Document> collection, Scanner scanner) {

System.out.println("Buffer content: " + scanner.nextLine());

System.out.print("Enter username of the user to delete: ");

String usernameToDelete = scanner.nextLine();

// Delete the user from MongoDB

deleteExistingUser(collection, usernameToDelete);

System.out.println("User deleted successfully!");

}

private static void deleteExistingUser(MongoCollection<Document> collection, String usernameToDelete) {

// Find the user by the username

Document query = new Document("username", usernameToDelete);

collection.deleteOne(query);

}

private static void saveUserToJson(User user) {

Gson gson = new Gson();

try (FileWriter writer = new FileWriter("User.json")) {

gson.toJson(user, writer);

System.out.println("User saved to JSON file.");

} catch (IOException e) {

e.printStackTrace();

System.out.println("Error al guardar el usuario en archivo JSON.");

}

}

private static void saveUserToMongoDB(User user, MongoCollection<Document> collection) {

// Convert User object to Document for MongoDB

Document userDocument = new Document("id", user.getId().toString())

.append("username", user.getUsername())

.append("password", user.getPassword());

// Insert the Document into the 'users' collection

collection.insertOne(userDocument);

System.out.println("User saved to MongoDB.");

}

}