Role Details

RoleDAO.java

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

public class RoleDAO {

public ArrayList<Role> getAllRoles() {

ArrayList<Role> roleList = new ArrayList<Role>();

// get connection

Connection connection = DbConnection.getConnection();

// execute query;

Statement statement;

try {

statement = connection.createStatement();

ResultSet rs = statement.executeQuery("SELECT \* FROM role");

while (rs.next()) {

// create Role object

Role role = new Role(rs.getInt("id"), rs.getString("name"));

roleList.add(role);

}

} catch (SQLException e) {

e.printStackTrace();

}

return roleList;

}

}

Main.java

import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

public class Main {

public static void main(String[] args) {

// create object to DAO

RoleDAO roleDao = new RoleDAO();

ArrayList<Role> roleList = roleDao.getAllRoles();

// sort

Collections.sort(roleList, new Comparator<Role>() {

@Override

public int compare(Role o1, Role o2) {

return o1.getName().compareTo(o2.getName());

}

});

// display

System.out.println("Role Details:");

for (Role role :roleList) {

System.out.println(role.getName());

}

}

}

Role.java

public class Role {

private Integer id;

private String name;

public Role(Integer id, String name) {

this.id = id;

this.name = name;

}

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

DbConnection.java

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.util.ResourceBundle;

public class DbConnection {

public static Connection getConnection() {

ResourceBundle rb = ResourceBundle.getBundle("mysql");

String url = rb.getString("db.url");

String username = rb.getString("db.username");

String password = rb.getString("db.password");

Connection connection = null;

try {

Class.forName("com.mysql.jdbc.Driver");

connection = DriverManager.getConnection(url, username, password);

} catch (ClassNotFoundException e) {

e.printStackTrace();

} catch (SQLException e) {

e.printStackTrace();

}

return connection;

}

}

===============================================

JDBC User Role

Role.java

public class Role {

private Integer id;

private String name;

public Role(Integer id, String name) {

this.id = id;

this.name = name;

}

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

User.java

public class User {

private Integer id;

private String name;

private Role role;

private Contact contact;

public User(Integer id, String name,Role role,Contact contact) {

this.id = id;

this.name = name;

this.role = role;

this.contact = contact;

}

public Contact getContact() {

return contact;

}

public void setContact(Contact contact) {

this.contact = contact;

}

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public Role getRole() {

return role;

}

public void setRole(Role role) {

this.role = role;

}

}

Main.java

import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

public class Main {

public static void main(String[] args) {

UserDAO userDao = new UserDAO();

ArrayList<User> userList = userDao.getAllUsers();

Collections.sort(userList, new Comparator<User>() {

@Override

public int compare(User o1, User o2) {

return o1.getName().compareTo(o2.getName());

}

});

// display

System.out.format("%-10s %-20s %-25s %-10s %-10s \n", "User", "Role", "Street", "City", "State");

for (User user : userList) {

System.out.format("%-10s %-20s %-25s %-10s %-10s \n", user.getName(), user.getRole().getName(),

user.getContact().getStreet(), user.getContact().getCity(), user.getContact().getState());

}

}

}

UserDAO.java

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

public class UserDAO {

public ArrayList<User> getAllUsers() {

ArrayList<User> userList = new ArrayList<User>();

try {

// get connection

Connection connection;

connection = DbConnection.getConnection();

// execute query;

Statement statement = connection.createStatement();

ResultSet userResultSet = statement.executeQuery("SELECT \* FROM user");

while (userResultSet.next()) {

// get values

int id = userResultSet.getInt("id");

String name = userResultSet.getString("name");

int contactId = userResultSet.getInt("contact\_id");

int roleId = userResultSet.getInt("role\_id");

Role role = null;

// execute query;

Statement statement1 = connection.createStatement();

// get resultset of role table

ResultSet roleResultSet = statement1.executeQuery("SELECT \* FROM role");

while (roleResultSet.next()) {

int id1 = roleResultSet.getInt("id");

// get values

if (roleId == id1) {

role = new Role(id1, roleResultSet.getString("name"));

break;

}

}

statement1.close();

Contact contact = null;

// execute query;

Statement statement2 = connection.createStatement();

// get resultset of contact table

ResultSet contactResultSet = statement2.executeQuery("SELECT \* FROM contact");

while (contactResultSet.next()) {

int id2 = contactResultSet.getInt("id");

if (contactId == id2) {

contact = new Contact(id2, contactResultSet.getString("street"),

contactResultSet.getString("city"), contactResultSet.getString("state"));

break;

}

}

statement2.close();

// create user object

User user = new User(id, name, role, contact);

userList.add(user);

}

statement.close();

connection.close();

} catch (SQLException e) {

e.printStackTrace();

} catch (InstantiationException | IllegalAccessException | ClassNotFoundException e) {

e.printStackTrace();

}

Collections.sort(userList, new Comparator<User>() {

@Override

public int compare(User o1, User o2) {

return o1.getName().compareTo(o2.getName());

}

});

return userList;

}

}

DbConnection.java

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.util.ResourceBundle;

public class DbConnection {

public static Connection getConnection() throws SQLException, InstantiationException, IllegalAccessException, ClassNotFoundException{

ResourceBundle rb = ResourceBundle.getBundle("mysql");

String url = rb.getString("db.url");

String username = rb.getString("db.username");

String password = rb.getString("db.password");

Connection connect = null;

//fill your code

Class.forName("com.mysql.jdbc.Driver");

connect = DriverManager.getConnection(url, username, password);

return connect;

}

}

Contact.java

public class Contact {

private Integer id;

private String street;

private String city;

private String state;

Contact(Integer id,String street,String city,String state){

this.id = id;

this.street = street;

this.city = city;

this.state = state;

}

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getStreet() {

return street;

}

public void setStreet(String street) {

this.street = street;

}

public String getCity() {

return city;

}

public void setCity(String city) {

this.city = city;

}

public String getState() {

return state;

}

public void setState(String state) {

this.state = state;

}

}

===============================================

JDBC – Assign privileges to roles

Main.java

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.Iterator;

import java.util.List;

public class Main {

public static void main(String[] args)

throws NumberFormatException, IOException, ClassNotFoundException, SQLException {

RoleDAO roleDAO = new RoleDAO();

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

PrivilegeDAO privilegeDAO = new PrivilegeDAO();

List<Privilege> privileges = privilegeDAO.getAllPrivileges();

System.out.println("List of privileges :");

System.out.format("%-15s %s\n", "Privilege ID", "Privilege Name");

for (int i = 0; i < privileges.size(); i++) {

System.out.format("%-15s %s\n", privileges.get(i).getId(), privileges.get(i).getName());

}

System.out.println("Enter number of new Roles to be created :");

Integer n = Integer.parseInt(br.readLine());

System.out.println("Enter the role and privileges :");

for (int i = 0; i < n; i++) {

ArrayList<Privilege> privilegeList = new ArrayList<Privilege>();

String[] values = br.readLine().split(",");

String role = values[0];

for (int j = 1; j < values.length; j++) {

privilegeList.add(new Privilege(values[j]));

}

roleDAO.createRole(new Role(role), privilegeList);

}

System.out.println("Enter the Role :");

String rol = br.readLine();

System.out.println("Privileges for " + rol + " :");

for (Privilege privilege : roleDAO.getPreviligeByRole(rol)) {

System.out.println(privilege.getName());

}

}

}

RoleDAO.java

import java.sql.\*;

import java.util.ArrayList;

import java.util.Iterator;

import java.util.List;

public class RoleDAO {

public void createRole(Role roleIns, ArrayList<Privilege> privilegeList)

throws SQLException, ClassNotFoundException {

Connection connection = DbConnection.getConnection();

String roleName = roleIns.getName();

Statement statement = connection.createStatement();

statement.executeUpdate("INSERT INTO role (name) VALUES('" + roleName + "')");

// assign provileges to role

// getting role id

Statement statementRole = connection.createStatement();

ResultSet rsRole = statementRole.executeQuery("SELECT \* FROM role where name='" + roleName + "'");

int roleId = rsRole.next() ? roleId = rsRole.getInt("id") : 0;

// for each privilege iterate and get privilege id and assign

for (Privilege privilege : privilegeList) {

Statement statement1 = connection.createStatement();

ResultSet rsPrivilege = statement1

.executeQuery("SELECT id FROM privilege WHERE name='" + privilege.getName() + "'");

int privId = rsPrivilege.next() ? privId = rsPrivilege.getInt("id") : 0;

Statement statementInsert = connection.createStatement();

statementInsert.executeUpdate(

"INSERT INTO role\_privilege (role\_id,privilege\_id) VALUES(" + roleId + "," + privId + ")");

}

}

public List<Privilege> getPreviligeByRole(String role) throws ClassNotFoundException, SQLException {

List<Privilege> privilegeList = new ArrayList<Privilege>();

Connection connection = DbConnection.getConnection();

// getting role id

Statement statementRole = connection.createStatement();

ResultSet rsRole = statementRole.executeQuery("SELECT \* FROM role where name='" + role + "'");

int roleId = rsRole.next() ? roleId = rsRole.getInt("id") : 0;

// get all the privilege id's from role\_privilege table using roleId

Statement privStatement = connection.createStatement();

ResultSet rsPriv = privStatement.executeQuery("SELECT \* FROM role\_privilege where role\_id=" + roleId);

while (rsPriv.next()) {

Statement privelegeStatement = connection.createStatement();

ResultSet rs = privelegeStatement

.executeQuery("SELECT name FROM privilege WHERE id=" + rsPriv.getInt("privilege\_id"));

if (rs.next())

privilegeList.add(new Privilege(rs.getString("name")));

}

return privilegeList;

}

}

PrivilegeDAO.java

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.List;

public class PrivilegeDAO {

public List<Privilege> getAllPrivileges() throws SQLException, ClassNotFoundException {

List<Privilege> privilegeList = new ArrayList<Privilege>();

Connection connection = DbConnection.getConnection();

Statement statement = connection.createStatement();

ResultSet rs = statement.executeQuery("SELECT \* FROM privilege");

while (rs.next()) {

Privilege privilege = new Privilege(rs.getInt("id"), rs.getString("name"));

privilegeList.add(privilege);

}

return privilegeList;

}

}

privilege.java

public class Privilege {

private Integer id;

private String name;

public Privilege(){}

public Privilege(String name){

this.name = name;

}

public Privilege(Integer id, String name) {

this.id = id;

this.name = name;

}

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

DbConnection.java

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.util.ResourceBundle;

public class DbConnection {

public static Connection getConnection() throws ClassNotFoundException, SQLException {

ResourceBundle rb= ResourceBundle.getBundle("mysql");

String url = rb.getString("db.url");

String username = rb.getString("db.username");

String password = rb.getString("db.password");

Class.forName("com.mysql.jdbc.Driver");

Connection connect = DriverManager.getConnection(url, username, password);

return connect;

}

}

Role.java

import java.util.ArrayList;

public class Role {

private Integer id;

private String name;

private ArrayList<Privilege> privilegeList;

public Role(){}

public Role(String name) {

this.name = name;

}

public Role(Integer id, String name) {

this.id = id;

this.name = name;

}

public Role(Integer id, String name, ArrayList<Privilege> privilegeList) {

this.id = id;

this.name = name;

this.privilegeList = privilegeList;

}

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public ArrayList<Privilege> getPrivilegeList() {

return privilegeList;

}

public void setPrivilegeList(ArrayList<Privilege> privilegeList) {

this.privilegeList = privilegeList;

}

}

===============================================

Assign role from database to user using JDBC

RoleDAO.java

import java.sql.\*;

import java.util.\*;

public class RoleDAO {

public List<Role> getAllRoles() throws Exception {

ArrayList<Role> roleList = new ArrayList<Role>();

Connection connection = DbConnection.getConnection();

// execute query;

Statement statement = connection.createStatement();

ResultSet rs = statement.executeQuery("SELECT \* FROM role");

while (rs.next()) {

// create Role object

Role role = new Role(rs.getInt("id"), rs.getString("name"), rs.getString("description"));

roleList.add(role);

}

Collections.sort(roleList, new Comparator<Role>() {

@Override

public int compare(Role o1, Role o2) {

return o1.getRoleName().compareTo(o2.getRoleName());

}

});

return roleList;

}

}

User.java

public class User {

private String name;

private String userName;

private String password;

private String mobileNumber;

private Role role;

public User(String name, String userName, String password,String mobileNumber, Role role) {

this.name = name;

this.userName = userName;

this.password = password;

this.role = role;

this.mobileNumber = mobileNumber;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getUserName() {

return userName;

}

public void setUserName(String userName) {

this.userName = userName;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public Role getRole() {

return role;

}

public void setRole(Role role) {

this.role = role;

}

public String getMobileNumber() {

return mobileNumber;

}

public void setMobileNumber(String mobileNumber) {

this.mobileNumber = mobileNumber;

}

}

DbConnection.java

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.util.ResourceBundle;

public class DbConnection {

public static Connection getConnection() throws ClassNotFoundException, SQLException {

ResourceBundle rb= ResourceBundle.getBundle("mysql");

String url = rb.getString("db.url");

String username = rb.getString("db.username");

String password = rb.getString("db.password");

Connection connect = null;

Class.forName("com.mysql.jdbc.Driver");

connect = DriverManager.getConnection(url, username, password);

return connect;

}

}

Role.java

public class Role {

private Integer id;

private String roleName;

private String description;

public Role(Integer id, String roleName, String description) {

this.id = id;

this.roleName = roleName;

this.description = description;

}

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getRoleName() {

return roleName;

}

public void setRoleName(String roleName) {

this.roleName = roleName;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

}

UserDAO.java

import java.sql.\*;

public class UserDAO {

public void createUser(User user) throws Exception {

Connection connection = DbConnection.getConnection();

// execute query;

Statement statement = connection.createStatement();

statement.executeUpdate("INSERT INTO user(name,username,password,`mobile number`,role) VALUES('" + user.getName()

+ "','" + user.getUserName() + "','" + user.getPassword() + "','" + user.getMobileNumber() + "','"

+ user.getRole().getRoleName() + "')");

}

public void displayDetails() throws Exception {

System.out.println("User details:");

System.out.format("%-15s %-15s %-15s %-15s %-15s %s\n", "User id", "Name", "User name", "Password",

"Mobile number", "Role");

Connection connection = DbConnection.getConnection();

// execute query;

Statement statement = connection.createStatement();

ResultSet rs = statement.executeQuery("SELECT \* FROM user");

while (rs.next()) {

System.out.format("%-15s %-15s %-15s %-15s %-15s %s\n", rs.getInt("id"), rs.getString("name"),

rs.getString("username"), rs.getString("password"), rs.getString("mobile number"),

rs.getString("role"));

}

}

}

Main.java

import java.util.\*;

import java.io.\*;

public class Main {

public static void main(String[] args) throws Exception {

int choice, roleId, c = 0;

String userDetails;

Role role = null;

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

User user;

while (true) {

System.out.println("1. Create User\n2. Display Details\n3. Exit\nEnter the choice :");

choice = Integer.parseInt(br.readLine());

switch (choice) {

case 1:

System.out.println("Enter the user details:");

String[] values = br.readLine().split(",");

RoleDAO roleDao = new RoleDAO();

System.out.println("Role details:");

System.out.format("%-15s %-25s %s\n", "Role id", "Role name", "Role description");

for (Role roleVal : roleDao.getAllRoles()) {

System.out.format("%-15s %-25s %s\n", roleVal.getId(), roleVal.getRoleName(),

roleVal.getDescription());

}

System.out.println("Enter the role id:");

roleId = Integer.parseInt(br.readLine());

for (Role roleVal : roleDao.getAllRoles()) {

if (roleVal.getId() == roleId) {

role = roleVal;

break;

}

}

user = new User(values[0], values[1], values[2], values[3], role);

UserDAO userDao = new UserDAO();

userDao.createUser(user);

break;

case 2:

UserDAO userDao1 = new UserDAO();

userDao1.displayDetails();

break;

case 3:

System.exit(0);

}

}

}

}

CC

Contact.java

package jdbc.cc.section1;

public class Contact {

private Integer id;

private String orgName;

private String street;

private String city;

private String state;

Contact(String orgName, String street, String city, String state) {

this.orgName = orgName;

this.street = street;

this.city = city;

this.state = state;

}

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getOrgName() {

return orgName;

}

public void setOrgName(String orgName) {

this.orgName = orgName;

}

public String getStreet() {

return street;

}

public void setStreet(String street) {

this.street = street;

}

public String getCity() {

return city;

}

public void setCity(String city) {

this.city = city;

}

public String getState() {

return state;

}

public void setState(String state) {

this.state = state;

}

}

DBConnection.java

package jdbc.cc.section1;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.util.ResourceBundle;

public class DBConnection {

public static Connection getConnection()

throws SQLException, InstantiationException, IllegalAccessException, ClassNotFoundException {

ResourceBundle rb = ResourceBundle.getBundle("mysql");

String url = rb.getString("db.url");

String username = rb.getString("db.username");

String password = rb.getString("db.password");

Connection connect = null;

Class.forName("com.mysql.jdbc.Driver");

connect = DriverManager.getConnection(url, username, password);

return connect;

}

}

Main.java

package jdbc.cc.section1;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.\*;

public class Main {

public static void main(String ags[]) throws Exception {

DateFormat format = new SimpleDateFormat("yyyy-MM-dd");

UserDAO userDao = new UserDAO();

List<User> userList = userDao.getAllUserSortByLastLogin();

System.out.println("List of all user details based on last login time");

System.out.format("%-15s %-15s %-15s %-25s %-15s %-15s\n", "Name", "LastLogin Time", "Organization", "Street",

"City", "State");

for (User user : userList) {

System.out.format("%-15s %-15s %-15s %-25s %-15s %-15s\n", user.getName(),

format.format(user.getLastLoginTime()), user.getContact().getOrgName(),

user.getContact().getStreet(), user.getContact().getCity(), user.getContact().getState());

}

}

}

User.java

package jdbc.cc.section1;

import java.sql.Date;

public class User {

private Integer id;

private String name;

private Date lastLoginTime;

private Contact contact;

User(String name, Date lastLoginTime, Contact contact) {

this.name = name;

this.lastLoginTime = lastLoginTime;

this.contact = contact;

}

public Contact getContact() {

return contact;

}

public void setContact(Contact contact) {

this.contact = contact;

}

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public Date getLastLoginTime() {

return lastLoginTime;

}

public void setLastLoginTime(Date lastLoginTime) {

this.lastLoginTime = lastLoginTime;

}

}

UserDAO.java

package jdbc.cc.section1;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

import java.util.List;

import java.sql.Date;

public class UserDAO {

public List<User> getAllUserSortByLastLogin()

throws ClassNotFoundException, SQLException, InstantiationException, IllegalAccessException {

List<User> userList = new ArrayList<User>();

Connection connection = DBConnection.getConnection();

// execute query;

Statement statement = connection.createStatement();

ResultSet userRs = statement.executeQuery("SELECT \* FROM user\_details");

while (userRs.next()) {

String userName = userRs.getString("username");

Date date = userRs.getDate("last\_login\_time");

int contactId = userRs.getInt("contact\_id");

String orgName = null;

String street = null;

String city = null;

String state = null;

Statement statement1 = connection.createStatement();

ResultSet contactRs = statement1.executeQuery("SELECT \* FROM contact WHERE id=" + contactId);

if (contactRs.next()) {

orgName = contactRs.getString("org\_name");

street = contactRs.getString("street");

city = contactRs.getString("city");

state = contactRs.getString("state");

}

Contact contact = new Contact(orgName, street, city, state);

User user = new User(userName, date, contact);

userList.add(user);

}

Collections.sort(userList, new Comparator<User>() {

@Override

public int compare(User o1, User o2) {

return o1.getLastLoginTime().compareTo(o2.getLastLoginTime());

}

});

return userList;

}

}