Joshua McCann  
DAO Lab 1

|  |
| --- |
| package astontech.dao.mysql;  import astontech.dao.ClientDAO; import com.astontech.bo.Client;  import java.sql.CallableStatement; import java.sql.ResultSet; import java.sql.SQLException; import java.util.ArrayList; import java.util.List;  */\*\*  \* Created by Joshua.McCann on 6/28/2017.  \*/* public class ClientDAOImpl extends MySQL implements ClientDAO {   @Override  public Client getClientById(int clientId) {  *Connect*();  Client client = null;  try{  String sp = "{ call usp\_GetClient(?,?)}";  CallableStatement cStmnt = *conn*.prepareCall(sp);   cStmnt.setInt(1, *GET\_BY\_ID*);  cStmnt.setInt(2, clientId);   ResultSet rs = cStmnt.executeQuery();   while(rs.next()){  client = new Client();  client = *HydrateClient*(rs);  }  } catch (SQLException SqlEx){  *logger*.error(SqlEx);  }   return client;  }   @Override  public List<Client> getClientList() {  *Connect*();  List<Client> clientList = new ArrayList<Client>();  try {  String sp = "{ call usp\_GetClient(?,?)}";  CallableStatement cStmnt = *conn*.prepareCall(sp);   cStmnt.setInt(1, *GET\_COLLECTION*);  cStmnt.setInt(2, 0);   ResultSet rs = cStmnt.executeQuery();   while(rs.next()){  clientList.add(*HydrateClient*(rs));  }   } catch (SQLException SqlEx){  *logger*.error(SqlEx);  }   return clientList;  }   @Override  public void insertClient(Client client) {   }   @Override  public void updateClient(Client client) {   }   @Override  public void deleteClient(int clientId) {   }   private static Client HydrateClient(ResultSet rs) throws SQLException{  /\*  \* ClientId int index 1  \* ClientName varchar index 2  \* CreateDate date index 3  \*/   Client client = new Client();  client.setClientId(rs.getInt(1));  client.setClientName(rs.getString(2));  client.setCreateDate(rs.getDate(3));   return client;  } } |
| package astontech.dao.mysql;  import astontech.dao.EmailDAO; import com.astontech.bo.Email; import com.astontech.bo.EntityType;  import java.sql.CallableStatement; import java.sql.ResultSet; import java.sql.SQLException; import java.util.ArrayList; import java.util.List; import java.util.concurrent.Callable;  */\*\*  \* Created by Joshua.McCann on 6/28/2017.  \*/* public class EmailDAOImpl extends MySQL implements EmailDAO {   @Override  public Email getEmailById(int emailId) {  *Connect*();  Email email = null;  try{  String sp = "{ call usp\_GetEmail(?,?) }";  CallableStatement cStmnt = *conn*.prepareCall(sp);   cStmnt.setInt(1, *GET\_BY\_ID*);  cStmnt.setInt(2, emailId);   ResultSet rs = cStmnt.executeQuery();   if(rs.next()){  email = new Email();  email = HydrateEmail(rs);  }   } catch (SQLException SqlEx){  *logger*.error(SqlEx);  }  return email;   }   @Override  public List<Email> getEmailList() {  *Connect*();  List<Email> emailList = new ArrayList<Email>();   try{  String sp = "{ call usp\_GetEmail(?,?) }";  CallableStatement cStmnt = *conn*.prepareCall(sp);   cStmnt.setInt(1, *GET\_COLLECTION*);  cStmnt.setInt(2, 0);   ResultSet rs = cStmnt.executeQuery();   while(rs.next()){  emailList.add(HydrateEmail(rs));  }   } catch (SQLException SqlEx){  *logger*.error(SqlEx);  }   return emailList;  }   @Override  public void insertEmail(Email email) {   }   @Override  public void updateEmail(Email email) {   }   @Override  public void deleteEmail(int emailId) {   }   public Email HydrateEmail(ResultSet rs) throws SQLException{  /\*  \* EmailId int index 1  \* EmailAddress varchar index 2  \* EmployeeId int index 3  \* EntityTypeId int index 4  \*/   Email email = new Email();  EntityType entityType = new EntityType();  entityType.setEntityTypeId(rs.getInt(4));   email.setEmailId(rs.getInt(1));  email.setEmailAddress(rs.getString(2));  email.setEmailEmployee(new EmployeeDAOImpl().getEmployeeById(rs.getInt(3)));  email.setEmailType(entityType);   return email;  } } |
| package astontech.dao.mysql;  import astontech.dao.EmailDAO; import astontech.dao.EmployeeDAO; import astontech.dao.PersonDAO; import com.astontech.bo.Employee; import com.astontech.bo.Person;  import java.sql.CallableStatement; import java.sql.ResultSet; import java.sql.SQLException; import java.util.ArrayList; import java.util.List;  */\*\*  \* Created by Joshua.McCann on 6/28/2017.  \*/* public class EmployeeDAOImpl extends MySQL implements EmployeeDAO {   @Override  public Employee getEmployeeById(int employeeId) {  *Connect*();  Employee employee = null;   try{  String sp = "{ call usp\_GetEmployee(?,?,?) }";  CallableStatement cStmnt = *conn*.prepareCall(sp);   cStmnt.setInt(1, *GET\_BY\_ID*);  cStmnt.setInt(2, employeeId);  cStmnt.setInt(3, 0);   ResultSet rs = cStmnt.executeQuery();   if(rs.next()){  employee = new Employee();  employee = HydrateEmployee(rs);  }   } catch (SQLException SqlEx){  *logger*.error(SqlEx);  }   return employee;  }   @Override  public List<Employee> getEmployeeList() {  *Connect*();  List<Employee> employeeList = new ArrayList<Employee>();   try{  String sp = "{ call usp\_GetEmployee(?,?,?) }";  CallableStatement cStmnt = *conn*.prepareCall(sp);   cStmnt.setInt(1, *GET\_COLLECTION*);  cStmnt.setInt(2, 0);  cStmnt.setInt(3, 0);   ResultSet rs = cStmnt.executeQuery();   while(rs.next()){  employeeList.add(HydrateEmployee(rs));  }   } catch (SQLException SqlEx){  *logger*.error(SqlEx);  }   return employeeList;  }   @Override  public void insertEmployee(Employee employee) {   }   @Override  public void updateEmployee(Employee employee) {   }   @Override  public void deleteEmployee(int employeeId) {   }   public Employee HydrateEmployee(ResultSet rs) throws SQLException {  /\*  \* EmployeeId int index 1  \* HireDate Date index 2  \* TermDate Date index 3  \* BirthDate Date index 4  \* PersonId int index 5  \* CreateDate Date index 6  \*/   Person person = new PersonDAOImpl().getPersonById(rs.getInt(5));   Employee employee = new Employee();  employee.setEmployeeId(rs.getInt(1));  employee.setHireDate(rs.getDate(2));  employee.setTermDate(rs.getDate(3));  employee.setBirthDate(rs.getDate(4));  employee.setPersonId(rs.getInt(5));  employee.setFirstName(person.getFirstName());  employee.setLastName(person.getLastName());  employee.setDisplayFirstName(person.getDisplayFirstName());  employee.setCreateDate(rs.getDate(6));   return employee;  } } |
| package astontech.dao.mysql;  import astontech.dao.ClientDAO; import astontech.dao.PersonDAO; import astontech.dao.PhoneDAO; import com.astontech.bo.Client; import com.astontech.bo.EntityType; import com.astontech.bo.Person; import com.astontech.bo.Phone;  import java.sql.CallableStatement; import java.sql.ResultSet; import java.sql.SQLException; import java.util.ArrayList; import java.util.List;  */\*\*  \* Created by Joshua.McCann on 6/28/2017.  \*/* public class PhoneDAOImpl extends MySQL implements PhoneDAO{   @Override  public Phone getPhoneById(int phoneId) {  *Connect*();  Phone phone = null;  try {  String sp = "{ call usp\_GetPhone(?,?) }";  CallableStatement cStmnt = *conn*.prepareCall(sp);   cStmnt.setInt(1,*GET\_BY\_ID*);  cStmnt.setInt(2,phoneId);   ResultSet rs = cStmnt.executeQuery();   if(rs.next()){  phone = new Phone();  phone = *HydratePhone*(rs);  }   } catch (SQLException SqlEx) {  *logger*.error(SqlEx);  }  return phone;  }   @Override  public List<Phone> getPhoneList() {  *Connect*();  List<Phone> phoneList = new ArrayList<Phone>();  try {  String sp = "{ call usp\_GetPhone(?,?) }";  CallableStatement cStmnt = *conn*.prepareCall(sp);   cStmnt.setInt(1,*GET\_COLLECTION*);  cStmnt.setInt(2,0);   ResultSet rs = cStmnt.executeQuery();   while(rs.next()){  phoneList.add(*HydratePhone*(rs));  }   } catch (SQLException SqlEx) {  *logger*.error(SqlEx);  }  return phoneList;  }   @Override  public void insertPhone(Phone phone) {   }   @Override  public void updatePhone(Phone phone) {   }   @Override  public void deletePhone(int phoneId) {   }   private static Phone HydratePhone(ResultSet rs) throws SQLException{  /\*  \* PhoneId int index 1  \* EntityTypeId int index 2  \* ClientId int index 3  \* PersonId int index 4  \* AreaCode int index 5  \* PhoneNumber int index 6  \* PhoneNumberPost int index 7  \*/   EntityType entityType = new EntityType();  /\*entityType.setEntityTypeId(rs.getInt(2));  ClientDAO clientDAO = new ClientDAOImpl();  PersonDAO personDAO = new PersonDAOImpl();  Client client = clientDAO.getClientById(rs.getInt(3));  Person person = personDAO.getPersonById(rs.getInt(4));\*/   Phone phone = new Phone();  phone.setPhoneId(rs.getInt(1));  phone.setPhoneType(entityType);  phone.setPhoneClient(new ClientDAOImpl().getClientById(rs.getInt(3)));  phone.setPhonePerson(new PersonDAOImpl().getPersonById(rs.getInt(4)));  phone.setAreaCode(rs.getInt(5));  phone.setPhoneNumber(rs.getInt(6));  phone.setPhoneNumberPost(rs.getInt(7));   return phone;  } } |
| private static void ClientDAOTest() {  //region CREATE MENU  ClientDAO clientDAO = new ClientDAOImpl();  List<Client> clientList = clientDAO.getClientList();    System.*out*.println("==========================");   for(Client client : clientList){  System.*out*.println(client.getClientId() + " " + client.getClientName());  }   System.*out*.println("==========================");    //endregion   //region PROMPT USER   Scanner scanner = new Scanner(System.*in*);  System.*out*.println("Please Select a Client from list: ");  String clientId = scanner.nextLine();   //endregion   //region GET DETAILS   Client clientDetail = clientDAO.getClientById(Integer.*parseInt*(clientId));   System.*out*.println("------- CLIENT DETAILS ------");  System.*out*.println("Client Name: " + clientDetail.getClientName());  System.*out*.println("Create Date: " + clientDetail.getCreateDate());    //endregion } |
| private static void EmailDAOTest() {  //region CREATE MENU  EmailDAO emailDAO = new EmailDAOImpl();  List<Email> emailList = emailDAO.getEmailList();   System.*out*.println("==========================");   for(Email email : emailList){  System.*out*.println(email.getEmailId() + ": " + email.getEmailAddress());  }   System.*out*.println("==========================");   //endregion   //region PROMPT USER  Scanner scanner = new Scanner(System.*in*);  System.*out*.println("Select an Email");  String emailId = scanner.nextLine();   //endregion   //region GET DETAILS  Email emailDetail = emailDAO.getEmailById(Integer.*parseInt*(emailId));   System.*out*.println("----- EMAIL DETAILS -----");  System.*out*.println("Email Address: " + emailDetail.getEmailAddress());  System.*out*.println("Employee Owner: " + emailDetail.getEmailEmployee().getFullName());  //endregion  } |
| private static void EmployeeDAOTest() {  //region CREATE MENU  EmployeeDAO employeeDAO = new EmployeeDAOImpl();  List<Employee> employeeList = employeeDAO.getEmployeeList();   System.*out*.println("==========================");   for(Employee employee : employeeList){  System.*out*.println(employee.getEmployeeId() + ": " + employee.getLastName() + ", " + employee.getFirstName());  }   System.*out*.println("==========================");   //endregion   //region PROMPT USER  Scanner scanner = new Scanner(System.*in*);  System.*out*.println("Select an Employee:");  String employeeId = scanner.nextLine();   //endregion   //region GET DETAILS  Employee employeeDetail = employeeDAO.getEmployeeById(Integer.*parseInt*(employeeId));  System.*out*.println("----- EMPLOYEE DETAILS -----");  System.*out*.println("Full Name: " + employeeDetail.getFullName());  System.*out*.println("Hire Date: " + employeeDetail.getHireDate());  System.*out*.println("Term Date: " + employeeDetail.getTermDate());  System.*out*.println("Birth Date: " + employeeDetail.getBirthDate());   //endregion  } |
| private static void PhoneDAOTest() {  //region CREATE MENU  PhoneDAO phoneDAO = new PhoneDAOImpl();  List<Phone> phoneList = phoneDAO.getPhoneList();   System.*out*.println("==========================");   for(Phone phone : phoneList){  System.*out*.println(phone.getPhoneId() + ": (" + phone.getAreaCode() + ") " + phone.getPhoneNumber());  }   System.*out*.println("==========================");   //endregion   //region PROMPT USER  Scanner scanner = new Scanner(System.*in*);  System.*out*.println("Select a Phone number");  String phoneId = scanner.nextLine();   //endregion   //region GET DETAILS  Phone phoneDetail = phoneDAO.getPhoneById(Integer.*parseInt*(phoneId));   System.*out*.println("----- PHONE DETAILS -----");  System.*out*.println("Phone Number: " + phoneDetail.getAreaCodeString());  if(phoneDetail.getPhonePerson() != null)  System.*out*.println("Person Name: " + phoneDetail.getPhonePerson().getFullName());  if(phoneDetail.getPhoneClient() != null)  System.*out*.println("Client Name: " + phoneDetail.getPhoneClient().getClientName());   //endregion  } |