**实验报告**

**学院（系）名称：**计算机科学与工程学院

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **姓名** | | 韩金泷 | | **学号** | 20175228 | | **专业** | 计算机科学与技术 |
| **班级** | | 计算机2班 | | **实验**  **项目** | 实验3：扩展的JavaScript框架应用 | | | |
| **课程名称** | | | | Web应用程序设计与开发 | | | **课程**  **代码** | 0662813 |
| **实验时间** | | | | 2020年4月23日 3、4节  2020年4月27日 5、6节 | | | **实验**  **地点** |  |
| 考核标 准 | 实验准备（实验目的/工具熟悉情况）10分 | | 实验过程（实验方案可行性及步骤完整性）40分 | 实验报告(实验内容丰富度与格式清晰度) 30分 | | 实验结果（结论正确性以及分析合理性）20分 | **成绩** |  |
|  |  | |  |  | |  | **教师签字：** | |
| 考核内容 | 评价实验目的是否明确，实验工具是否清晰了解以及熟悉情况 | | ○可行，完整  ○可行，不完整  ○不可行，不完整 | ○丰富，清晰  ○较丰富，较清晰  ○丰富，不清晰  ○不丰富，不清晰 | | ○结论正确，分析合理  ○结论正确，分析不充分  ○结论不正确，分析不合理 |
| **一、实验目的**  （1）掌握利用扩展的JavaScript框架构架Web应用程序。  （2）掌握EasyUI中常用组件的使用方法。  （3）掌握jQuery常用方法$.post()的使用。  （4）JSON格式串和Java集合工具类的转换工具的使用。  **二、实验环境**  Windows操作系统，Tomcat，MyEclipse，HBuilder，记事本。  **三、 实验要求**  1、构建基于Java的JDBC访问数据库的类。具体要求：  （1）类的实现上应具有执行查询SQL语句的返回结果集的方法；  （2）能够实现执行insert into、update及delete方法；  （3）类应进行相关资源的释放。  （4）构建时使用数据库连接池技术实现。  2、EasyUI组件ComboBox的属性和使用，要求：  （1）构建表，将表中的数据初始化到ComboBox中；  （2）将ComboBox组件的值提交到Servlet中；  （3）提交数据利用jQuery中的$. post ()方法。  3、EasyUI组件数据表格组件：DataGrid的属性和使用，具体要求：  （1）初始化DataGrid组件：将表中的数据显示在DataGrid组件中；  （2）获得选中DataGrid组件中某一行中的数据；  （3）DataGrid控件中应带有分页功能；  （4）当对数据库表中的数据进行修改时，能够自动刷新DataGrid中的数据。  **四、实验过程记录（源程序、测试用例、测试结果及心得体会等）**  数据库连接池配置：  （1）在【META-INF】目录下创建Context.xml文件，并写入如下内容：  <Context>  <Resource  name=*"jdbc/DBPool"*  auth=*"Container"*  type=*"javax.sql.DataSource"*  factory=*"org.apache.commons.dbcp2.BasicDataSourceFactory"*  username=*"sa"*  password=*"111"*  driverClassName=*"com.microsoft.sqlserver.jdbc.SQLServerDriver"*  url=*"jdbc:sqlserver://localhost:1433;DatabaseName=webdb"*  maxTotal=*"1000"*  maxIdle=*"100"*  maxWaitMillis=*"5000"* />  </Context>  （2）修改web目录中web.xml文件，加入如下内容：  <resource-ref>  <res-ref-name>jdbc/DBPool</res-ref-name>  <res-type>javax.sql.DataSource</res-type>  <res-auth>Container</res-auth>  </resource-ref>  **DBHelper.java**  此为数据库封装类（老师给的，不是我自己写的，膜拜老师）  //连接池型数据库连接语块  public static Connection getConnection() {  Connection conn = null;  try {  Context ctx = new InitialContext();  DataSource ds = (DataSource) ctx.lookup("java:comp/env/jdbc/DBPool");  conn = ds.getConnection();  } catch (Exception ex) {  ex.printStackTrace();  }  return conn;  }  使用ArratList定义List  public static boolean executeBatch(ArrayList<String> list) {  boolean flag = true;//  try {  conn = getConn();//  conn.setAutoCommit(false);  st = conn.createStatement();  for (int i = 0; i < list.size(); i++) {  st.addBatch(list.get(i));  }  st.executeBatch();  conn.commit();// 执行事务  conn.setAutoCommit(true);  } catch (Exception ex) {  try {  conn.rollback();// 事务回滚  } catch (SQLException e) {  e.printStackTrace();  }  flag = false;//  ex.printStackTrace();  } finally {  finallyHandle(conn, st, rs);  }  return flag;  }  对数据库进行增删改的操作（带事务）：  public static boolean executeBatch(String sql) {  boolean flag = true;//  try {  conn = getConn();//  conn.setAutoCommit(false);  st = conn.createStatement();  st.addBatch(sql);  st.executeBatch();  conn.commit();// 执行事务  conn.setAutoCommit(true);  } catch (Exception ex) {  try {  conn.rollback();// 事务回滚  } catch (SQLException e) {  e.printStackTrace();  }  flag = false;//  ex.printStackTrace();  } finally {  finallyHandle(conn, st, rs);//  }  return flag;  }  //对数据库进行查询操作  public static ArrayList<HashMap<String, String>> getDataSet(String sql) {  HashMap<String, String> hash = null;  ArrayList<HashMap<String, String>> list = new ArrayList<>();  ResultSetMetaData rsma = null;  int columncount = 0;  try {  conn = DBUtil.getConn();  st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE, ResultSet.CONCUR\_READ\_ONLY);  rs = st.executeQuery(sql);  rsma = rs.getMetaData();  while (rs.next()) {  hash = new HashMap<>();  columncount = rsma.getColumnCount();  for (int i = 1; i <= columncount; i++) {  hash.put(rsma.getColumnName(i), rs.getString(i));  }  list.add(hash);  }  } catch (SQLException e) {  e.printStackTrace();  } finally {  finallyHandle(conn, st, rs);  }  return list;  }  对数据库查询（根据查询结果分页显示查询结果）  public static ArrayList<HashMap<String, String>> getDataSetInfoByCon(String sql, int rowCount, int page) {  Connection conn = null;  ArrayList<HashMap<String, String>> result = null;  Statement st = null;  ResultSet rs = null;  ResultSetMetaData rsmd = null;  try {  conn = getConn();  st = conn.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE, ResultSet.CONCUR\_READ\_ONLY);  if (rowCount > 0)  st.setMaxRows(page \* rowCount);  rs = st.executeQuery(sql);  if (page >= 0 && rowCount > 0)  rs.absolute((page - 1) \* rowCount);  rsmd = rs.getMetaData();  result = new ArrayList<HashMap<String, String>>();  while (rs.next()) {  int columnCount = rsmd.getColumnCount();  HashMap<String, String> record = new HashMap<String, String>();  for (int i = 1; i <= columnCount; i++) {  record.put(rsmd.getColumnName(i), rs.getString(i));  }  result.add(record);  }  } catch (Exception e) {  e.printStackTrace();  } finally {  finallyHandle(conn, st, rs);  }  return result;  }  对数据库结束操作后关闭数据库连接（一定不要忘记）  private static void finallyHandle(Connection conn, Statement st, ResultSet rs) {  try {  if (rs != null) {  rs.close();  rs = null;  }  if (st != null) {  st.close();  st = null;  }  if (conn != null) {  conn.close();  conn = null;  }  } catch (Exception ex) {  ex.printStackTrace();  }  }  2、EasyUI组件和ComboBox组件  <div>  &nbsp;姓名: <input class=*"easyui-combobox"* id=*"div\_name"*  style="width:*150px*"></input> &nbsp;所在学院: <input  class=*"easyui-combobox"* id=*"div\_institute"* style="width:*150px*"></input>  &nbsp; <a id=*"query"* class=*"easyui-linkbutton"* iconCls=*"icon-search"*  plain=*"true"*>查 询</a>  </div>  JavaScript：  $(document).ready(function() {  $("#div\_institute").combobox({  url : "SystemStudentService?op=institute",  valueField : "institute\_id",  textField : "institute\_name",  panelHeight : 'auto'  });  })  Servlet部分：  private void getComboBox(HttpServletResponse response, String type) {  String result = "";  ArrayList<HashMap<String, String>> dt = null;  String sql;  int rowscount = 0;  try {  sql = "SELECT \* FROM " + type;  dt = DBUtil.getDataSet(sql);  result = JSON.toJSONString(dt);  System.out.println(result);  response.setCharacterEncoding("utf-8");  PrintWriter out = response.getWriter();  out.print(result);  out.close();  } catch (Exception ex) {  ex.printStackTrace();  }  }  ComboBox结果展示：    3、DataGrid组件的使用：  <div id=*"dlgStudentInfo"* class=*"easyui-dialog"* style="width:*360px*;height:*450px*;  padding:*10px 10px*" closed=*"true"* buttons=*"#dlg-buttons"*>  <div class=*"ftitle"*>学生信息</div>  <form id=*"fm"* method=*"post"* novalidate>  <table class=*"fitem"*>  <tr>  <td><label class=*"fitem"*>学 号</label></td>  <td><input type=*"text"* id=*"sno"* name=*"sno"*></td>  </tr>  <tr>  <td><label class=*"fitem"*>姓名</label></td>  <td><input type=*"text"* id=*"sname"* name=*"sname"*></td>  </tr>  <tr>  <td><label class=*"fitem"*>年 龄</label></td>  <td><input type=*"text"* id=*"sname"* name=*"sname"*></td>  </tr>  <tr>  <td><label>学院</label></td>  <td><input class=*"easyui-combobox"* type=*"text"*  id=*"dlg\_institute"* name=*"dlg\_institute"* /></td>  </tr>  </table>  </form>  </div>  $(function() {  $('#birthday').datebox({  required : false  });  $("#info").datagrid({  loadMsg : "加载中...",  iconCls : 'icon-issue',  nowrap : false,  striped : true,  collapsible : true,  rownumbers : true,  pagination : true,  singleSelect : true,  autoRowHeight : true,  fitColumns : false,  pageSize : 10,  pageList : [ 10, 20, 30, 40 ],  cache : false,  url : "SystemStudentService?op=init"  });  $("#add").click(function() {  operation = "add";  $('#dlgStudentInfo').dialog('open').dialog('center').dialog('setTitle', '添加学生信息');  $('#fm').form('clear');  $("#dlg\_institute").combobox({  url : "SystemStudentService?op=institute",  valueField : "institute\_id",  textField : "institute\_name",  panelHeight : 'auto'  });  $("#politicalstate").combobox({  url : "SystemStudentService?op=politicalstate",  valueField : "politicalstate\_id",  textField : "politicalstate\_name",  panelHeight : 'auto'  });  });  $("#edit").click(function() {  var row = $('#info').datagrid('getSelected');  if (row) {  sno = row.sno;  $('#fm').form('clear');  operation = "edit";  $('#dlgStudentInfo').dialog('open').dialog('center').dialog('setTitle', '编辑学生信息');  $("#dlg\_institute").combobox({  url : "SystemStudentService?op=institute",  valueField : "institute\_id",  textField : "institute\_name",  panelHeight : 'auto'  });  $("#politicalstate").combobox({  url : "SystemStudentService?op=politicalstate",  valueField : "politicalstate\_id",  textField : "politicalstate\_name",  panelHeight : 'auto'  });  $("#sno").val(row.sno);  $("#sname").val(row.sname);  $("#age").val(row.age);  $("#politicalstate").combobox("setValue", row.politicalstate);  $("#birthday").datebox("setValue", row.birthday);  $("#address").val(row.address);  $("#phone").val(row.phone);  $("#dlg\_institute").combobox("setValue", row.institute);  $("#demo").val(row.demo);  } else {  $.messager.alert("请选择学生信息！", 'info');  return;  }  });  //对话框中的保存按钮操作:完成添加和编辑操作  $("#save").click(function() {  if ($("#sno").val() == "") {  $.messager.alert("学生编码为空！", 'info');  return;  }  if (operation == "add") {  $.post("SystemStudentService?op=add",  $("#fm").serialize(),  function(data) {  var value = eval('(' + data + ')')  if (value.ret == "0") {  $.messager.alert("提示", "添加成功！", 'info');  initstudentinfo();  return;  } else {  $.messager.alert("提示", value.reason, 'info');  return;  }  });  } else if (operation == "edit") {  $.post("SystemStudentService?op=edit&oldsno=" + sno,  $("#fm").serialize(),  function(data) {  var value = eval("(" + data + ")")  if (value.ret == "0") {  $.messager.alert("修改成功！", 'info');  initstudentinfo();  return;  } else {  $.messager.alert(value.reason, 'info');  return;  }  });  }  });  $("#delete").click(function() {  var row = $('#info').datagrid('getSelected');  if (row == "" || row == null) {  $.messager.alert("请选择学生信息！", 'info');  return;  }  $.post("SystemStudentService?op=delete&sno=" + row.sno,  function(data) {  var value = eval("(" + data + ")")  if (value.ret == "1") {  $.messager.alert("删除成功！", 'info');  initstudentinfo();  return;  } else {  $.messager.alert("删除失败！" + value.reason, 'info');  return;  }  });  });  $("#close").click(function() {  $('#dlgStudentInfo').dialog('close');  return;  });  function initstudentinfo() {  $("#studentinfo").datagrid({  loadMsg : "加载中...",  url : 'SystemStudentService?op=init&con='  });  }  })  Servlet部分：  protected void doPost(HttpServletRequest request, HttpServletResponse response)  throws ServletException, IOException {  // TODO Auto-generated method stub  String op = request.getParameter("op");  System.out.println(op);    switch (op) {  case "politicalstate":  getComboBox(response, "politicalstate");  break;    case "institute":  getComboBox(response, "institute");  break;    case "student":  getComboBox(response, "student\_simple");  break;    case "init":  String row = request.getParameter("rows");  String page = request.getParameter("page");  getStudentInfo(response, "", page, row);  break;  default:  break;  }  }  private static String getStudentInfo(HttpServletResponse response, String con, String page, String row) {  String result = "";  Map<String, Object> map = new HashMap<String, Object>();  ArrayList<HashMap<String, String>> dt = null;  String sql;  int rowscount = 0;  if (con == null)  con = "";  if (row == null)  row = "0";  if (page == null)  page = "0";  try {  int r = Integer.parseInt(row);  int p = Integer.parseInt(page);  if (!con.equals("")) {  sql = "select \* from student where " + con;  } else {  sql = "select \* from student";  }  dt = DBUtil.getDataSetInfoByCon(sql, r, p);  rowscount = DBUtil.getRowCount(sql);  map.put("total", rowscount);  map.put("rows", dt);  result = JSON.toJSONString(map);  response.setCharacterEncoding("utf-8");  PrintWriter out = response.getWriter();  out.print(result);  out.close();  } catch (Exception ex) {  ex.printStackTrace();  }  return result;  }  DataGrid结果展示：    心得体会：  通过这次实验，我加深了对扩展的JavaScript框架和EasyUI中常用组件的理解，EasyUI包括ComboBox与DataGrid组件。实验时ComboBox组件多次不能使用，最后在同学与学长的帮助下完成控件的组建，收获很大。 | | | | | | | | |