**内蒙古科技大学信息工程学院计算机系**

**《Java Web编程A》实验报告**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 姓名 | | 刘鹏 | 学号 | 1667159125 |
| 班级 | | 软件工程16-1 | 实验日期 |  |
| 项目号、实验名称 | | **Java Web第三次实验上交** | | |
| 实  验  要  求 | **实验5 数据库连接池的实验**  本次上机完成的任务  掌握如何进行数据库连接池的连接。  训练的技能点   1. 在Tomcat中设置数据源 2. 在Servlet中进行数据库连接池的连接   **实验6 利用数据库连接池进行数据库查询的实验**  本次上机完成的任务  掌握如何利用数据库连接池技术进行查询。  训练的技能点  (1) 在Servlet中进行数据库连接池的连接  (2) 进行数据库的查询操作显示一个表中的数据 | | | |
| 实  验  内  容 | **实验5 数据库连接池的实验**   1. 在Tomcat中以自己名字设置数据源。 2. 新建以自己名字命名的项目Web Project； 3. 新建以自己名字命名Servlet中进行数据库连接池的连接 4. 测试数据库连接池的程序   **实验6 利用数据库连接池进行数据库查询的实验**   1. 新建以自己名字命名的项目Web Project； 2. 新建以自己名字命名的数据库，数据库命名如：XXX\_DB 3. 新建以自己名字命名的表，表名命名如：XXX\_TBL。 4. 创建表的各个字段，字段名以自己的姓氏的首字母开头，如：c\_id,c\_name。 5. 新建以自己名字命名的Servlet文件，用来进行数据库连接池的连接。 6. 进行数据库的查询操作显示一个表中的数据。 7. 部署Web应用程序；测试运行程序。 | | | |
| 实  验  过  程  及  结  果 | **实验5 数据库连接池的实验**  关于Tomcat中的自带的数据库连接池  Apache 有自家的数据库连接池DBCP  第一部，倒入包  包名   1. Commons DBCP 2. Commons Collections 3. Commons Pool   地址  Tomcat/lib/下  第二步， 修改配置文件  路径  tomcat/conf/context.xml  修改方式  添加  内容  <Resource  name="database\_for\_java\_web"  scope="Shareable"  type="javax.sql.DataSource"  factory="org.apache.tomcat.dbcp.dbcp2.BasicDataSourceFactory"  url="jdbc:mysql://localhost:3306/database\_for\_java\_web"  driverClassName ="com.mysql.jdbc.Driver"  username="root"  password="lp184126"  />  <!--name: 映射名-->  <!--url: 数据库地址-->  路径  Java Web项目/web/WEB-INF/web.xml  修改方式  添加  内容  <!-- JDBC DataSources (java:comp/env/jdbc) -->  <resource-ref>  <description>The default DS</description>  <res-ref-name>database\_for\_java\_web</res-ref-name>  <!--该名称必须与下面的描述文件中的一致-->  <res-type>javax.sql.DataSource</res-type>  <!--看看你的配置-->  <res-auth>Container</res-auth>  </resource-ref>  第三部， 编写测试servlet  类名  DataSourceTestServlet  路径  src.ServletPackage.DataSourceTestServlet  代码  package ServletPackage;    import javax.naming.Context;  import javax.naming.InitialContext;  import javax.naming.NamingException;  import javax.servlet.ServletException;  import javax.servlet.annotation.WebServlet;  import javax.servlet.http.HttpServlet;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  import javax.sql.DataSource;  import java.io.IOException;  import java.io.PrintWriter;  import java.sql.Connection;  import java.sql.SQLException;    @WebServlet(name = "DataSourceTestServlet")  public class DataSourceTestServlet extends HttpServlet {  protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  PrintWriter out = response.getWriter();    try {  Context context = new InitialContext();  DataSource ds = (DataSource) context.lookup("java:/comp/env/database\_for\_java\_web");  Connection connection = ds.getConnection();  if (connection != null) {  out.print("get connection success!");  }    } catch (NamingException e) {  // TODO Auto-generated catch block  e.printStackTrace();  } catch (SQLException e) {  // TODO Auto-generated catch block  e.printStackTrace();  }  }    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  doPost(request, response);  }  }  在xml文件中修改  路径  Java Web项目/web/WEB-INF/web.xml  修改方式  添加  代码  <servlet>  <servlet-name>DataSourceTestServlet</servlet-name>  <servlet-class>ServletPackage.DataSourceTestServlet</servlet-class>  </servlet>  <servlet-mapping>  <servlet-name>DataSourceTestServlet</servlet-name>  <url-pattern>/ServletPackage/DataSourceTestServlet</url-pattern>  </servlet-mapping>  运行结果  **/Users/liupeng/Desktop/屏幕快照 2018-10-12 上午11.18.52.png**  **实验6 利用数据库连接池进行数据库查询的实验**   1. 为了方便起见，我创建了数据库连接类，内部的实现通过数据库连接池 2. 代码如下：   **package** JavaBean.linkDatabase;  **import** javax.naming.Context; **import** javax.naming.InitialContext; **import** javax.naming.NamingException; **import** javax.sql.DataSource; **import** java.sql.Connection; **import** java.sql.ResultSet; **import** java.sql.SQLException; **import** java.sql.Statement;  **public class** linkDBByDBCP {   **public** linkDBByDBCP() **throws** NamingException, SQLException {  Context context = **new** InitialContext();  DataSource ds = (DataSource) context.lookup(**"java:/comp/env/database\_for\_java\_web"**);  **this**.**connection** = ds.getConnection();  **this**.**statement** = **connection**.createStatement();  }   **public boolean** saveData(String sql) **throws** SQLException {  **boolean** key = **false**;   String string = sql;  *//string = "INSERT INTO user (lp\_id, lp\_name, lp\_password) VALUES (1809120006,1234,1234);";* key = **this**.**statement**.execute(string);   **return** key;  }   **public** ResultSet getData(String sql) **throws** SQLException {  ResultSet resultSet = **null**;  resultSet = **this**.**statement**.executeQuery(sql);  **return** resultSet;  }   **public** Connection getConnection() {  **return connection**;  }   **public void** setConnection(Connection connection) {  **this**.**connection** = connection;  }   **public** Statement getStatement() {  **return statement**;  }   **public void** setStatement(Statement statement) {  **this**.**statement** = statement;  }   Connection **connection** = **null**;   Statement **statement** = **null**; }   1. 然后就是对数据库中数据的添加与查看 2. 添加代码：   **package** ServletPackage;  **import** JavaBean.linkDatabase.getSQLString; **import** JavaBean.linkDatabase.linkDBByDBCP; **import** JavaBean.linkDatabase.linkDatabases;  **import** javax.naming.NamingException; **import** javax.servlet.ServletException; **import** javax.servlet.annotation.WebServlet; **import** javax.servlet.http.HttpServlet; **import** javax.servlet.http.HttpServletRequest; **import** javax.servlet.http.HttpServletResponse; **import** java.io.IOException; **import** java.io.PrintWriter; **import** java.sql.ResultSet; **import** java.sql.SQLException;  @WebServlet(name = **"work2getInformationServlet"**) **public class** work2getInformationServlet **extends** HttpServlet {  **protected void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  request.setCharacterEncoding(**"utf-8"**);  response.setContentType(**"text/html;charset=utf-8"**);  PrintWriter out = response.getWriter();   getSQLString lpGetSQLString = **new** getSQLString();  lpGetSQLString.setSqlForWord2\_2();  String sql = lpGetSQLString.getSqlForWord2\_2();  **try** {  out.print(**"<table>"**);  *//linkDatabases lpLinkDatabases = new linkDatabases();* linkDBByDBCP lpLinkDatabases = **new** linkDBByDBCP();  ResultSet resultSet = lpLinkDatabases.getData(sql);  *//ResultSet resultSet = lpLinkDatabases.getInformation(sql);* **while**(resultSet.next()) {  String name = resultSet.getString(**"name"**);  String phone = resultSet.getString(**"phone"**);  String email = resultSet.getString(**"email"**);  String title = resultSet.getString(**"title"**);  String content = resultSet.getString(**"content"**);  String time = resultSet.getString(**"time"**);   out.print(**"<tr>"**);  out.print(**"<td>姓名</td>"**);  out.print(**"<td>"**+name+**"</td>"**);  out.print(**"<td>电话</td>"**);  out.print(**"<td>"**+phone+**"</td>"**);  out.print(**"<td>Email</td>"**);  out.print(**"<td>"**+email+**"</td>"**);  out.print(**"<td>题目</td>"**);  out.print(**"<td>"**+title+**"</td>"**);  out.print(**"<td>内容</td>"**);  out.print(**"<td>"**+content+**"</td>"**);  out.print(**"<td>时间</td>"**);  out.print(**"<td>"**+time+**"</td>"**);  out.print(**"</tr>"**);  }  out.print(**"</table>"**);  } **catch** (SQLException e) {  e.printStackTrace();  } **catch** (NamingException e) {  e.printStackTrace();  }  }   **protected void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  doPost(request, response);  } }  好处就是：  就该代码特别少，因为代码的耦合度低。    只需要修改俩行就好！   1. 查找代码：   *<%--  Created by IntelliJ IDEA.  User: liupeng  Date: 2018/10/17  Time: 2:32 PM  To change this template use File | Settings | File Templates. --%>* <%@ **page contentType**="**text/html;charset=UTF-8**" **language**="**java**" %> <%@ **page import**="**JavaBean.linkDatabase.linkDBByDBCP**" %> <%@ **page import**="**JavaBean.linkDatabase.getSQLString**" %> <%@ **page import**="**java.sql.ResultSet**" %> <**html**> <**head**>  <**title**>Java Web Work2 Link Database And Show Data</**title**> </**head**> <**body**>  <**table border="1"**>  <**center**>  **<%** linkDBByDBCP lpLinkDBByDBCP = **new** linkDBByDBCP();  getSQLString lpGetSQLString = **new** getSQLString();  lpGetSQLString.setSqlForWord2\_2();  String sql = lpGetSQLString.getSqlForWord2\_2();   ResultSet resultSet = lpLinkDBByDBCP.getData(sql);   **while**(resultSet.next()) {  String name = resultSet.getString(**"name"**);  String phone = resultSet.getString(**"phone"**);  String email = resultSet.getString(**"email"**);  String title = resultSet.getString(**"title"**);  String content = resultSet.getString(**"content"**);  String time = resultSet.getString(**"time"**);   out.print(**"<tr>"**);  out.print(**"<td>姓名</td>"**);  out.print(**"<td>"**+name+**"</td>"**);  out.print(**"<td>电话</td>"**);  out.print(**"<td>"**+phone+**"</td>"**);  out.print(**"<td>Email</td>"**);  out.print(**"<td>"**+email+**"</td>"**);  out.print(**"<td>题目</td>"**);  out.print(**"<td>"**+title+**"</td>"**);  out.print(**"<td>内容</td>"**);  out.print(**"<td>"**+content+**"</td>"**);  out.print(**"<td>时间</td>"**);  out.print(**"<td>"**+time+**"</td>"**);  out.print(**"</tr>"**);  }  **%>** </**center**> </**table**> </**body**> </**html**> | | | |
| 实  验  总  结 | 通过本次实验，我学会了如何使用连接池来连接数据库，并且也了解了使用连接池的优势。数据库连接池的核心理念就是在内存中建立一个虚拟的容器，在这个容器中保存预先建立好的数据库连接对象，把这些连接对象分配给需要的程序使用，当建立对象使用完成之后，不直接关闭连接，而是重新保存到容器中，供其他程序反复使用。这样就减少了建立数据库连接所需要的时间。 | | | |