作业号：JW\_32 得分： of 100

题目要求(作业题干)

|  |
| --- |
| JW\_32 登录验证  访问任何一个资源之前，都要进行登录验证。如果发现访问者没有请求登录功能，则返回“您没有登录，请登录”  登录功能本身，不能进行登录验证。  提示：  使用Filter和Session |

源程序(打印版本不得含有注释，不能修改字体字号)

|  |
| --- |
| //201802104029陈义乾  01 package bysj.Controller.Login;  02  03 import bysj.Daomain.User;  04 import bysj.Service.UserService;  05 import com.alibaba.fastjson.JSONObject;  06  07 import javax.servlet.ServletException;  08 import javax.servlet.annotation.WebServlet;  09 import javax.servlet.http.HttpServlet;  10 import javax.servlet.http.HttpServletRequest;  11 import javax.servlet.http.HttpServletResponse;  12 import javax.servlet.http.HttpSession;  13 import java.io.IOException;  14 import java.sql.SQLException;  15  16 @WebServlet("/login")  17 public class LoginController extends HttpServlet {  18 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  19 String username=request.getParameter("username");  20 String password=request.getParameter("password");  21  22 JSONObject message = new JSONObject();  23 try{  24 User loggedUser = UserService.getInstance().login(username,password);  25 if(loggedUser!=null){  26 message.put("message","登陆成功");  27 HttpSession session =request.getSession();  28  29 session.setMaxInactiveInterval(10\*60);  30 session.setAttribute("currentUser",loggedUser);  31 response.getWriter().println(message);  32  33 return;  34 }else {  35 message.put("message","用户名或密码错误");  36 }  37 }catch (SQLException e){  38 message.put("message","数据库操作异常");  39 e.printStackTrace();  40 }catch (Exception e){  41 message.put("message","网络异常");  42 e.printStackTrace();  43 }  44 response.getWriter().println(message);  45 }  46 } |
| 01 package bysj.Controller.logout;  02  03 import com.alibaba.fastjson.JSONObject;  04  05 import javax.servlet.ServletException;  06 import javax.servlet.annotation.WebServlet;  07 import javax.servlet.http.HttpServlet;  08 import javax.servlet.http.HttpServletRequest;  09 import javax.servlet.http.HttpServletResponse;  10 import javax.servlet.http.HttpSession;  11 import java.io.IOException;  12  13 @WebServlet("/logout")  14 public class LogoutController extends HttpServlet {  15 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  16 HttpSession session=request.getSession();  17 if(session != null) {  18 session.invalidate();  19 }  20 response.getWriter().println("已退出");  21 }  22 } |
| 01 import com.alibaba.fastjson.JSONObject;  02  03 import javax.servlet.\*;  04 import javax.servlet.annotation.WebFilter;  05 import javax.servlet.http.HttpServletRequest;  06 import javax.servlet.http.HttpServletResponse;  07 import javax.servlet.http.HttpSession;  08 import java.io.IOException;  09  10  11 @WebFilter(filterName = "Filter0",urlPatterns = "/\*")  12 public class Filter0 implements Filter {  13 public void destroy() {  14 }  15  16 public void doFilter(ServletRequest req, ServletResponse resp, FilterChain chain) throws ServletException, IOException {  17 System.out.println("Filter0 - encodimg begins");  18  19 HttpServletRequest request=(HttpServletRequest)req;  20  21 HttpServletResponse response=(HttpServletResponse) resp;  22  23 String path=request.getRequestURI();  24 boolean u1=path.contains("/login");  25 boolean u2=path.contains("/mysystem");  26  27 if(!u1 && !u2){  28 String method=request.getMethod();  29 if(method.equals("PUT")||method.equals("POST")){  30 request.setCharacterEncoding("UTF-8");  31 }else {  32 response.setContentType("text/html;charset=utf-8");  33 }  34 JSONObject message=new JSONObject();  35 HttpSession session=request.getSession(false);  36 if(session==null || session.getAttribute("currentUser") ==null){  37 message.put("message","请登录或重新登录");  38 response.getWriter().println(message);  39 return;  40 }  41 }  42 response.setContentType("text/html;charset=utf-8");  43  44 chain.doFilter(req, resp);  45 System.out.println("Filter0 - encoding ends");  46 }  47 } |
| 001 package bysj.Dao;  002  003 import bysj.Daomain.\*;  004 import bysj.Service.\*;  005 import bysj.Util.JdbcHelper;  006  007 import java.sql.\*;  008 import java.util.Collection;  009 import java.util.HashSet;  010 import java.util.Set;  011 //https://github.com/chen-yiqian/JW\_201802104029\_31.git  012 public final class TeacherDao {  013 private static TeacherDao teacherDao=new TeacherDao();  014 public static TeacherDao getInstance(){  015 return teacherDao;  016 }  017  018 public Collection<Teacher> findAll() throws SQLException {  019  020 Set teachers=new HashSet<Teacher>();  021  022 Connection connection = JdbcHelper.getConn();  023  024 Statement statement = connection.createStatement();  025  026 ResultSet resultSet= statement.executeQuery("select \* from teacher");  027  028 while (resultSet.next()){  029 Proftitle proftitle= ProftitleDao.getInstance().find(resultSet.getInt("proftitle\_id"));  030 Degree degree=DegreeDao.getInstance().find(resultSet.getInt("degree\_id"));  031 Department department=DepartmentDao.getInstance().find(resultSet.getInt("department\_id"));  032  033 Teacher teacher = new Teacher(resultSet.getInt("id"),  034 resultSet.getString("no"),  035 resultSet.getString("name"),  036 proftitle,  037 degree,  038 department);  039  040 teachers.add(teacher);  041 }  042 return teachers;  043 }  044  045 public Teacher find(Integer id)throws SQLException{  046 Teacher teacher = null;  047  048 Connection connection=JdbcHelper.getConn();  049  050 String findteacher="select \* from teacher where id=?";  051  052 PreparedStatement preparedStatement=connection.prepareStatement(findteacher);  053  054 preparedStatement.setInt(1,id);  055  056 ResultSet resultSet=preparedStatement.executeQuery();  057  058  059  060 if(resultSet.next()){  061 Proftitle proftitle= ProftitleService.getInstance().find(resultSet.getInt("proftitle\_id"));  062 Degree degree= DegreeService.getInstance().find(resultSet.getInt("degree\_id"));  063 Department department=DepartmentService.getInstance().find(resultSet.getInt("department\_id"));  064  065 teacher=new Teacher(  066 resultSet.getInt("id"),  067 resultSet.getString("no"),  068 resultSet.getString("name"),  069 proftitle,  070 degree,  071 department  072 );  073 }  074  075 JdbcHelper.close(preparedStatement,connection);  076 return teacher;  077 }  078  079  080 public boolean update(Teacher teacher)throws SQLException{  081  082 Connection connection=JdbcHelper.getConn();  083  084 String updateteacher="update teacher set no=?,name=?,proftitle\_id=?,degree\_id=?,department\_id=? where id=?";  085  086 PreparedStatement preparedStatement=connection.prepareStatement(updateteacher);  087  088 preparedStatement.setString(1,teacher.getNo());  089 preparedStatement.setString(2,teacher.getName());  090 preparedStatement.setInt(3,teacher.getProftitle().getId());  091 preparedStatement.setInt(4,teacher.getDegree().getId());  092 preparedStatement.setInt(5,teacher.getDepartment().getId());  093 preparedStatement.setInt(6,teacher.getId());  094  095 int affected=preparedStatement.executeUpdate();  096  097 JdbcHelper.close(preparedStatement,connection);  098 return affected>0;  099 }  100  101 public boolean add(Teacher teacher)throws SQLException{  102  103 PreparedStatement pstmt=null;  104 Connection connection=null;  105 Boolean affected = false;  106 int teacher\_id = 0;  107 try{  108  109 connection=JdbcHelper.getConn();  110  111 connection.setAutoCommit(false);  112  113 String addTeacher\_sql = "insert into teacher(no,name,proftitle\_id,degree\_id,department\_id) values " + "(?,?,?,?,?)";  114  115 pstmt = connection.prepareStatement(addTeacher\_sql);  116  117 pstmt.setString(1,teacher.getNo());  118 pstmt.setString(2,teacher.getName());  119 pstmt.setInt(3,teacher.getProftitle().getId());  120 pstmt.setInt(4,teacher.getDegree().getId());  121 pstmt.setInt(5,teacher.getDepartment().getId());  122  123 int affectedRowNum=pstmt.executeUpdate();  124 System.out.println("teacher添加了"+affectedRowNum+"行");  125  126 pstmt =connection.prepareStatement("select \* from teacher where id=(select max(id) from teacher)");  127  128 ResultSet resultSet=pstmt.executeQuery();  129 if (resultSet.next()){  130 teacher\_id = resultSet.getInt("id");  131 }  132 pstmt = connection.prepareStatement("insert into user(username,password,teacher\_id) values " + "(?,?,?)");  133  134 pstmt.setString(1,teacher.getNo());  135 pstmt.setString(2,teacher.getNo());  136 pstmt.setInt(3,teacher\_id);  137  138 int affectedRowNum2 =pstmt.executeUpdate();  139  140 connection.commit();  141 affected=affectedRowNum2>0;  142 }catch (Exception e){  143 try{  144  145 if(connection!=null){  146  147 connection.rollback();  148 }  149 }catch (SQLException e1) {  150 e1.printStackTrace();  151 }  152 }finally {  153 try{  154  155 if(connection!=null){  156 connection.setAutoCommit(true);  157 }  158 }catch (SQLException e){  159 e.printStackTrace();  160 }  161  162 JdbcHelper.close(pstmt,connection);  163 }  164 return affected;  165 }  166  167 public boolean delete(Integer id) throws SQLException {  168  169 Connection connection=null;  170 PreparedStatement preparedStatement = null;  171 Boolean affected=false;  172 try{  173  174 connection=JdbcHelper.getConn();  175  176 connection.setAutoCommit(false);  177  178 preparedStatement=connection.prepareStatement("SELECT \* FROM user where teacher\_id = ?");  179 preparedStatement.setInt(1, id);  180 ResultSet resultSet = preparedStatement.executeQuery();  181 if(resultSet.next()){  182 UserDao.getInstance().delete(resultSet.getInt("id"));  183 }  184  185  186 String deleteTeacher\_sql="DELETE FROM teacher WHERE id=?";  187  188 preparedStatement=connection.prepareStatement(deleteTeacher\_sql);  189  190 preparedStatement.setInt(1,id);  191  192 int affectedRowNum=preparedStatement.executeUpdate();  193  194 connection.commit();  195 affected=affectedRowNum>0;  196 } catch (SQLException e) {  197 System.out.println(e.getMessage()+"\n errorCode="+e.getErrorCode());  198 try{  199  200 if(connection!=null){  201  202 connection.rollback();  203 }  204 }catch (SQLException e1){  205 e1.printStackTrace();  206 }  207 }finally {  208 try{  209  210 if(connection!=null){  211 connection.setAutoCommit(true);  212 }  213 }catch (SQLException e){  214 e.printStackTrace();  215 }  216  217 JdbcHelper.close(preparedStatement,connection);  218 }  219 return affected;  220 }  221  222 } |

运行结果

|  |
| --- |
| JW_201802104029_32陈义乾01 |
| JW_201802104029_32陈义乾02 |
| JW_201802104029_32陈义乾03 |
| JW_201802104029_32陈义乾04 |