

1.2 新建常用工具类

数据库连接类 DB

文件名: DB.java

所属包: cn.network.shop.util

方法:

`public static Connection getConnection()`

说明: 返回 Connection 对象

提示:

```
//5.x
//Class.forName("com.mysql.jdbc.Driver");
//connection= DriverManager.getConnection("jdbc:mysql://localhost:3306/shop", "root",
"12345678");

//8.x
Class.forName("com.mysql.cj.jdbc.Driver");
connection=
DriverManager.getConnection("jdbc:mysql://localhost:3306/shop?useSSL=false&serverTi
mezone=Asia/Shanghai&useUnicode=true&characterEncoding=utf-8", "root", "12345678");
```

DB.java

```
public class DB {
    public static Connection getConnection() {
        Connection connection = null;
        try {
            Class.forName("com.mysql.jdbc.Driver");
            connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/shop", "root",
"root");
            System.out.println("已连接数据库...");
        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace();
        }
        return connection;
    }
}
```

1.3 新建用户实体类 User

文件名: User.java

所属包: cn.network.shop.user

成员变量: id,name,password,role,realName,address //和 user 表的字段一致

方法: 所有成员变量对应的 getter 和 setter 方法, 利用 IDEA 自动生成

User.java

1.4 新建用户服务类 UserService

文件名: UserService.java

所属包: cn.network.shop.user

方法:

//登录检查

public User loginCheck(String name,String password)

说明: 根据传入的用户名 name 和密码 password 检索数据库, 如果匹配, 返回该用户的完整数据 User 对象; 否则, 返回 null。

UserService.java

```
public class UserService {  
    /**  
     * 登录检查  
     * @param name: 账号  
     * @param password: 密码  
     * @return User 对象  
     */  
    public static User loginCheck(String name,String password){  
        User user=null;  
        try {  
            //1. 连接数据库  
            Connection connection = DB.getConnection();  
            //2. 创建预编译对象  
            String sql = "select * from user where name=? and password=md5(?)";  
            PreparedStatement ps = connection.prepareStatement(sql);  
            ps.setString(1,name);  
            ps.setString(2,password);  
            //3. 执行sql  
            ResultSet rs = ps.executeQuery();  
            //4. 读取数据  
            if(rs.next()){  
                user=new User();  
                user.setId(rs.getInt("id"));  
                user.setName(rs.getString("name"));  
                user.setPassword(rs.getString("password"));  
                user.setRole(rs.getInt("role"));  
            }  
        }  
    }  
}
```

```

        user.setRealName(rs.getString("realname"));
        user.setAddress(rs.getString("address"));
    }

    //5. 释放资源
    rs.close();
    ps.close();
    connection.close();
} catch (SQLException e) {
    e.printStackTrace();
}
return user;
}

}

```

1.5 新建用户登录控制器 ServletLogin

文件名: ServletLogin.java

所属包: cn.network.shop.user

处理的请求路径: /login

重写方法: doPost()

//1. 获得登录表单传入的参数 name,password

//2. 调用 UserService 的登录检查方法 loginCheck

//3. 根据返回值判断登录是否成功, 如果返回的 User 对象不为空, 则登录成功, 将该 User 对象以键名"user"存入 session, 跳转至首页; 否则, 登录失败, 将登录失败的信息以键名"loginErrorText"存入 request 中, 转发至首页。

ServletLogin.java

```

@WebServlet(name = "ServletLogin",value = "/login")
public class ServletLogin extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        //1 获得表单数据
        String name = request.getParameter("name");
        String password = request.getParameter("password");
        String rememberMe = request.getParameter("rememberMe");

        //2 调用业务层的方法
        User user = UserService.LoginCheck(name, password);
        if(user!=null){
            ....
        }else {

```

```
        ....
    }
}

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

    }
}
```