# day14 编码实战演练

## 案例：用户注册登录

　　要求：3层框架，使用验证码

###### 功能分析

* 注册
* 登录

1.1　JSP页面

* regist.jsp
* 注册表单：用户输入注册信息；
* 回显错误信息：当注册失败时，显示错误信息；
* login.jsp
* 登录表单：用户输入登录信息；
* 回显错误便利店：当登录失败时，显示错误信息；
* index.jsp
* 用户已登录：显示当前用户名，以及“退出”链接；
* 用户未登录：显示“您还没有登录”；

1.2　实体类

User：

* String username；
* String password；

1.3　Servlet

* VerifyCodeServlet
* 生成验证码；
* 在session中保存验证码文本；
* 把图片输出到页面
* RegistServlet
* 获取用户名、密码，封装到User对象中；
* 获取验证码、获取确认密码；
* 校验用户名、密码、验证码不能为空，校验失败，向request中保存错误信息，转发回regist.jsp显示错误信息；
* 比较两次输入的错误是否一致，如果不一致，向request中保存错误信息，转发回regist.jsp显示错误信息；
* 获取session中的验证码，与表单输入的验证码比较，如果不一致，向request中保存错误信息，转发回regist.jsp显示错误信息；
* 使用UserService的regist()方法完成注册，如果注册失败，向request中保存错误信息，转发回regist.jsp显示错误信息，如果注册成功，转发到login.jsp页面，表示注册成功；
* LoginServlet
* 获取用户名、密码、验证码；
* 校验用户名、密码、验证码是否为空，校验失败，向request中保存错误信息，转发回login.jsp显示错误信息；
* 获取session中的验证码，与表单中的验证码比较，如果不同，向request中保存错误信息，转发回login.jsp显示错误信息；
* 删除session中的验证码；
* 通过UserService的login()方法完成登录，如果抛出异常，获取异常信息，保存到request中，转发到login.jsp显示错误信息；
* 向session中保存当前用户对象；
* 转发到index.jsp页面，表示登录成功！
* QuitServlet
* 获取session，销毁之；
* 重定向到index.jsp；

1.4　Service

　　UserException：为UserService使用的异常类；

UserService：

* void regist(User user)：
* 使用UserDao的findByUsername()方法查询名为user.getUsername()的用户，如果用户存在，说明用户名已经被注册，抛出异常；
* 使用UserDao的add(User)方法保存用户信息；
* User login(String username, String password)：
* 使用UserDao的findByUsername()方法查询名为user.getUsername()的用户，如果用户不存在，说明用户名错误，抛出异常；
* 如果查询到了User，那么比较参数password与user.getPassword()是否相等，如果不等，说明密码错误，抛出异常；
* 如果一致，表示登录成功，返回User对象；

1.5　DAO

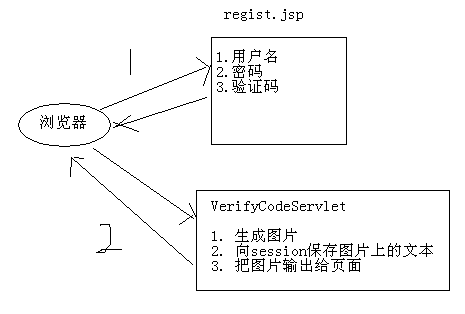
UserDao：

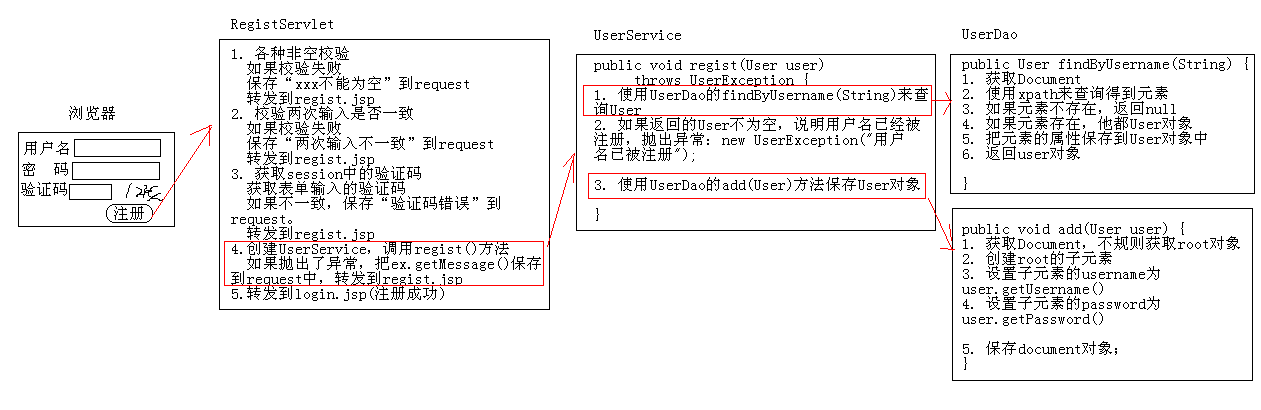
* void add(User)：
* 创建SAXReader对象，获取Document对象，再获取根元素；
* 给root元素添加子元素；
* 给子元素设置username属性，值为user.getUsername()；
* 给子元素设置password属性，值为user.getPassword()；
* 创建OutputFormat对象，指定缩进为“\t”，指定添加换行；
* 设置OutputFormat清空原有空白；
* 使用FileWrtier和OutputFormat创建XMLWriter对象；
* 使用XMLWriter对象的write()保存Document；
* 关闭XMLWriter对象；
* User findByUsername(String username)：
* 创建SAXReader对象，获取Document对象；
* 使用Xpath（//user[username=’xxx’]）来查询元素；
* 如果元素没有查询到，返回null；
* 如果元素查询到了，那么创建User对象；
* 把元素的username属性赋给User的username属性；
* 把元素的password属性赋给User的password属性；
* 返回user对象；

###### 2　流程图

2.1　注册

* 用户在浏览器地址栏中请求regist.jsp；
* 服务器发送html给浏览器；
* 浏览器收到html，开始解析，并显示；
* 解析到<img>时，请求VerifyCodeServlet；
* VerifyCodeServlet生成验证码图片，保存验证码文本，把图片响应给浏览器；
* 浏览器显示在页面中显示图片。





2.2　登录

　　此处省略10000字

###### 3　代码

login.jsp

|  |
| --- |
| <body>  <h1>登录</h1>  <hr/>  <p style="font-weight: *900*; color: *red*;">${msg }</p>  <form action=*"*<c:url value=*'/LoginServlet'*/>*"* method=*"post"*>  用户名：<input type=*"text"* name=*"username"* value=*"*${user.username }*"* /><br/>  密　码：<input type=*"password"* name=*"password"*/><br/>  验证码：<input type=*"text"* name=*"loginCode"* size=*"2"*/>  <img id=*"vCode"* src=*"*<c:url value=*'/VerifyCodeServlet?name=loginCode'*/>*"* border=*"2"*/>  <a href=*"javascript:\_change()"* style="font-size: *12*;">看不清，换一张</a><br/>  <input type=*"submit"* value=*"登录"*/>  </form>  </body> |
| <script type=*"text/javascript"*>  **function** \_change() {  **var** img = document.getElementById("vCode");  img.src = "<c:url value='/VerifyCodeServlet?name=loginCode&'/>" + **new** Date().getTime();  }  </script> |

regist.jsp

|  |
| --- |
| <body>  <h1>注册</h1>  <hr/>  <p style="font-weight: *900*; color: *red*;">${msg }</p>  <form action=*"*<c:url value=*'/RegistServlet'*/>*"* method=*"post"*>  用户名：<input type=*"text"* name=*"username"* value=*"*${user.username }*"* /><br/>  密　码：<input type=*"password"* name=*"password"*/><br/>  确认密码：<input type=*"password"* name=*"repassword"*/><br/>  验证码：<input type=*"text"* name=*"registCode"* size=*"2"*/>  <img id=*"vCode"* src=*"*<c:url value=*'/VerifyCodeServlet?name=registCode'*/>*"* border=*"2"*/>  <a href=*"javascript:\_change()"* style="font-size: *12*;">看不清，换一张</a><br/>  <input type=*"submit"* value=*"注册"*/>  </form>  </body> |
| <script type=*"text/javascript"*>  **function** \_change() {  **var** img = document.getElementById("vCode");  img.src = "<c:url value='/VerifyCodeServlet?name=registCode&'/>" + **new** Date().getTime();  }  </script> |

index.jsp

|  |
| --- |
| <body>  <h1>主页</h1>  <hr/>  <c:choose>  <c:when test="${**empty** sessionScope.user }">  您还没有登录  </c:when>  <c:otherwise>  用户名：${sessionScope.user.username }  <a href=*"*<c:url value=*'/QuitServlet'*/>*"*>退出</a>  </c:otherwise>  </c:choose>  </body> |

VerifyCodeServlet

|  |
| --- |
| **public** **class** VerifyCodeServlet **extends** HttpServlet {  **public** **void** doGet(HttpServletRequest request, HttpServletResponse response)  **throws** ServletException, IOException {  String name = request.getParameter("name");    VerifyCode vc = **new** VerifyCode();//创建验证码类  BufferedImage image = vc.getImage();//创建验证码图片  request.getSession().setAttribute(name, vc.getText());//获取验证码文本  System.*out*.println(vc.getText());  VerifyCode.*output*(image, response.getOutputStream());//输出图片到页面  }  } |

RegistServlet

|  |
| --- |
| **public** **class** RegistServlet **extends** HttpServlet {  **public** **void** doPost(HttpServletRequest request, HttpServletResponse response)  **throws** ServletException, IOException {  request.setCharacterEncoding("utf-8");  response.setContentType("text/html;charset=utf-8");    User user = **new** User();  **try** {  BeanUtils.*populate*(user, request.getParameterMap());  } **catch** (Exception e) {  }  String loginCode = request.getParameter("registCode");  String repassword = request.getParameter("repassword");    **if**(user.getUsername() == **null** || user.getUsername().trim().isEmpty()) {  request.setAttribute("msg", "用户名不能为空!");  request.setAttribute("user", user);  request.getRequestDispatcher("/regist.jsp").forward(request, response);  **return**;  }  **if**(user.getPassword() == **null** || user.getPassword().trim().isEmpty()) {  request.setAttribute("msg", "密码不能为空!");  request.setAttribute("user", user);  request.getRequestDispatcher("/regist.jsp").forward(request, response);  **return**;  }  **if**(!user.getPassword().equals(repassword)) {  request.setAttribute("msg", "两次输入不一致！");  request.setAttribute("user", user);  request.getRequestDispatcher("/regist.jsp").forward(request, response);  **return**;  }  **if**(loginCode == **null** || loginCode.trim().isEmpty()) {  request.setAttribute("msg", "验证码不能为空！");  request.setAttribute("user", user);  request.getRequestDispatcher("/regist.jsp").forward(request, response);  **return**;  }    String vCode = (String)request.getSession().getAttribute("registCode");  request.getSession().removeAttribute("registCode");  **if**(!vCode.equalsIgnoreCase(loginCode)) {  request.setAttribute("msg", "验证码错误！");  request.setAttribute("user", user);  request.getRequestDispatcher("/regist.jsp").forward(request, response);  **return**;  }    UserService userService = **new** UserService();  **try** {  userService.regist(user);  request.getRequestDispatcher("/login.jsp").forward(request, response);  } **catch** (UserException e) {  request.setAttribute("msg", e.getMessage());  request.setAttribute("user", user);  request.getRequestDispatcher("/regist.jsp").forward(request, response);  **return**;  }  }  } |

LoginServlet

|  |
| --- |
| **public** **class** LoginServlet **extends** HttpServlet {  **public** **void** doPost(HttpServletRequest request, HttpServletResponse response)  **throws** ServletException, IOException {  request.setCharacterEncoding("utf-8");  response.setContentType("text/html;charset=utf-8");    String username = request.getParameter("username");  String password = request.getParameter("password");  String loginCode = request.getParameter("loginCode");    **if**(username == **null** || username.trim().isEmpty()) {  request.setAttribute("msg", "用户名不能为空!");  request.setAttribute("username", username);  request.getRequestDispatcher("/login.jsp").forward(request, response);  **return**;  }  **if**(password == **null** || password.trim().isEmpty()) {  request.setAttribute("msg", "密码不能为空!");  request.setAttribute("username", username);  request.getRequestDispatcher("/login.jsp").forward(request, response);  **return**;  }  **if**(loginCode == **null** || loginCode.trim().isEmpty()) {  request.setAttribute("msg", "验证码不能为空！");  request.setAttribute("username", username);  request.getRequestDispatcher("/login.jsp").forward(request, response);  **return**;  }    String vCode = (String)request.getSession().getAttribute("loginCode");  request.getSession().removeAttribute("loginCode");  **if**(!vCode.equalsIgnoreCase(loginCode)) {  request.setAttribute("msg", "验证码错误！");  request.setAttribute("username", username);  request.getRequestDispatcher("/login.jsp").forward(request, response);  **return**;  }    UserService userService = **new** UserService();  User user;  **try** {  user = userService.login(username, password);  } **catch** (UserException e) {  request.setAttribute("msg", e.getMessage());  request.setAttribute("username", username);  request.getRequestDispatcher("/login.jsp").forward(request, response);  **return**;  }    request.getSession().setAttribute("user", user);  request.getRequestDispatcher("/index.jsp").forward(request, response);  }  } |

QuitServlet

|  |
| --- |
| **public** **class** QuitServlet **extends** HttpServlet {  **public** **void** doGet(HttpServletRequest request, HttpServletResponse response)  **throws** ServletException, IOException {  // 使session失效  request.getSession().invalidate();  response.sendRedirect(request.getContextPath() + "/index.jsp");  }  } |

UserException

|  |
| --- |
| **public** **class** UserException **extends** Exception {  **public** UserException() {}  **public** UserException(String message, Throwable cause) {  **super**(message, cause);  }  **public** UserException(String message) {  **super**(message);  }  **public** UserException(Throwable cause) {  **super**(cause);  }  } |

UserService

|  |
| --- |
| **public** **class** UserService {  **private** UserDao userDao = **new** UserDao();    **public** User login(String username, String password) **throws** UserException {  User user = userDao.findByUsername(username);  **if**(user == **null**) {  **throw** **new** UserException("用户名错误！");  }  **if**(!user.getPassword().equals(password)) {  **throw** **new** UserException("密码错误！");  }  **return** user;  }    **public** **void** regist(User user) **throws** UserException {  User \_user = userDao.findByUsername(user.getUsername());  **if**(\_user != **null**) {  **throw** **new** UserException("用户名已注册！");  }  userDao.add(user);  }  } |

UserDao

|  |
| --- |
| **public** **class** UserDao {  **private** String path;    **public** UserDao() {  path = **this**.getClass().getResource("/users.xml").getPath();  }    **public** **void** add(User user) {  **try** {  SAXReader reader = **new** SAXReader();  Document doc = reader.read(path);    Element root = doc.getRootElement();  Element userEle = root.addElement("user");  userEle.addAttribute("username", user.getUsername());  userEle.addAttribute("password", user.getPassword());    // 创建格式化器，使用\t缩进，添加换行  OutputFormat format = **new** OutputFormat("\t", **true**);  // 清空数据中原有的换行  format.setTrimText(**true**);  // 创建XML输出流对象  XMLWriter writer = **new** XMLWriter(**new** FileWriter(path), format);  // 输出Document  writer.write(doc);  // 关闭流  writer.close();  } **catch** (Exception e) {  **throw** **new** RuntimeException(e);  }  }  **public** User findByUsername(String username) {  **try** {  SAXReader reader = **new** SAXReader();  Document doc = reader.read(path);  Element ele = (Element) doc.selectSingleNode("//user[@username='" + username + "']");  **if**(ele == **null**) {  **return** **null**;  }  User user = **new** User();  user.setUsername(ele.attributeValue("username"));  user.setPassword(ele.attributeValue("password"));  **return** user;  } **catch** (Exception e) {  **throw** **new** RuntimeException(e);  }  }  } |