1. 使用PreparedStatement

PreparedStatement叫预编译声明！

PreparedStatement是Statement的子接口，你可以使用PreparedStatement来替换Statement。

PreparedStatement的好处：

* 防止SQL攻击；
* 提高代码的可读性，以可维护性；
* 提高效率。

|  |
| --- |
| String sql = “select \* from tab\_student where s\_number=?”;  PreparedStatement pstmt = con.prepareStatement(sql);  pstmt.setString(1, “S\_1001”);  ResultSet rs = pstmt.executeQuery();  rs.close();  pstmt.clearParameters();  pstmt.setString(1, “S\_1002”);  rs = pstmt.executeQuery(); |

1. 解决JDBC繁琐的问题

使用连接池（c3p0和dbcp）+DBUtils，推荐使用c3p0，其具有空闲链接自动回收的功能，Commons DbUtils是Apache组织提供的一个对JDBC进行简单封装的开源工具类库，使用它能够简化JDBC应用程序的开发，同时也不会影响程序的性能。

教程地址：https://www.cnblogs.com/CQY1183344265/p/5854418.html

1. 使用JSP+EL+JSTL

### jsp内置/隐式对象（9个）

jsp被翻译成servlet之后，service方法中有9个对象定义并初始化完毕，我们在jsp 脚本中可以直接使用这9个对象

|  |  |  |
| --- | --- | --- |
| **名称** | **类型** | **描述** |
| out | javax.servlet.jsp.JspWriter | 用于页面输出 |
| request | javax.servlet.http.HttpServletRequest | 得到用户请求信息， |
| response | javax.servlet.http.HttpServletResponse | 服务器向客户端的回应信息 |
| config | javax.servlet.ServletConfig | 服务器配置，可以取得初始化参数 |
| session | javax.servlet.http.HttpSession | 用来保存用户的信息 |
| application | javax.servlet.ServletContext | 所有用户的共享信息 |
| page | java.lang.Object | 指当前页面转换后的Servlet类的实例 |
| **pageContext** | javax.servlet.jsp.PageContext | JSP的页面容器 |
| exception | java.lang.Throwable | 表示JSP页面所发生的异常，在错误页中才起作用 |

#### out对象

out的类型：JspWriter

out作用就是想客户端输出内容----out.write()

out缓冲区默认8kb 可以设置成0 代表关闭out缓冲区 内容直接写到respons缓冲器

#### pageContext对象

jsp页面的上下文对象，作用如下：

page对象与pageContext对象不是一回事

1. pageContext是一个域对象

setAttribute(String name,Object obj)

getAttribute(String name)

removeAttrbute(String name)

pageContext可以向指定的其他域中存取数据

setAttribute(String name,Object obj,int scope)

getAttribute(String name,int scope)

removeAttrbute(String name,int scope)

**findAttribute(String name)**

**---依次从pageContext域，request域，session域，application域中获 取属性，在某个域中获取后将不在向后寻找**

**四大作用域的总结：**

**page域：当前jsp页面范围**

**request域：一次请求**

**session域：一次会话**

**application域：整个web应用**

1. 可以获得其他8大隐式对象

例如： pageContext.getRequest()

pageContext.getSession()

### EL表达式

EL（Express Lanuage）表达式可以嵌入在jsp页面内部，减少jsp脚本的编写，E出现的目的是要替代jsp页面中脚本的编写。

EL最主要的作用是获得四大域中的数据，格式**${EL表达式}**

EL获得pageContext域中的值：${pageScope.key};

EL获得request域中的值：${requestScope.key};

EL获得session域中的值：${sessionScope.key};

EL获得application域中的值：${applicationScope.key};

**EL从四个域中获得某个值${key};**

---同样是依次从pageContext域，request域，session域，application域中 获取属性，在某个域中获取后将不在向后寻找

1. 获得普通字符串
2. 获得User对象的值

3）获得List<User>的值

### jstl表达式

需要导入两个jar包，然后使用jsp的taglib命令导入jstl的核心标签库：

其常用的两组标签为

1. <c:if test=””>标签

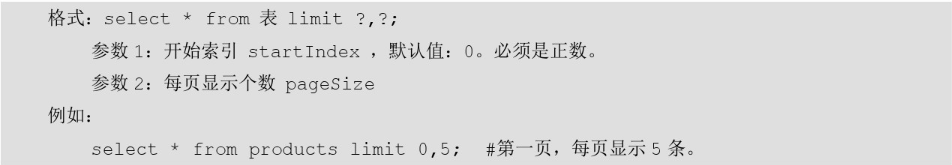
其中test是返回boolean的条件

1. <c:forEach>标签，可以通过这个标签结合el表达式遍历数组中的元素或对象及对象的属性。

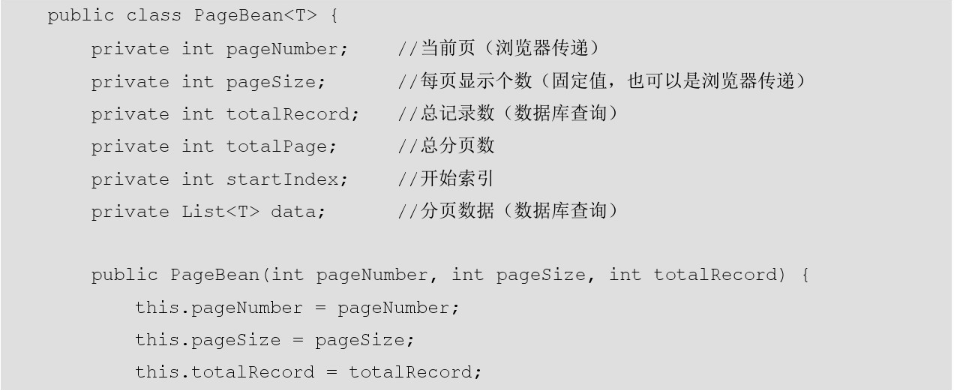


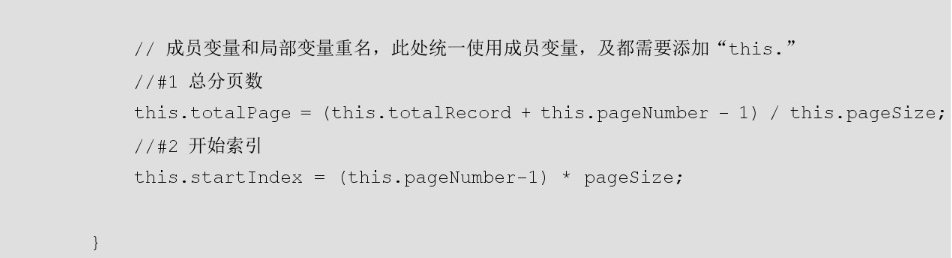


1. 最好利用sql的limit实现数据分页
2. sql查询语句的limit介绍



1. 实现分页的7个步骤
2. 编写pageBean以便包装分页方法





1. 确定入口

http://localhost:8080/project\_name/page.jsp?pageNumber=2

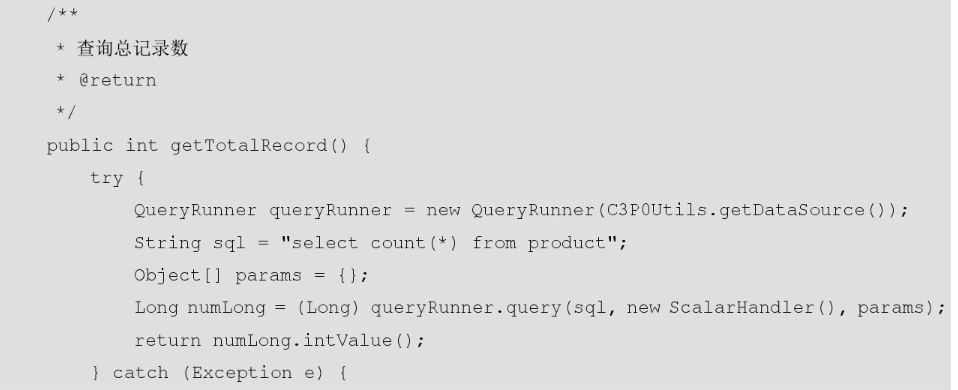
1. 编写servlet获得请求参数，并调用service进行分页

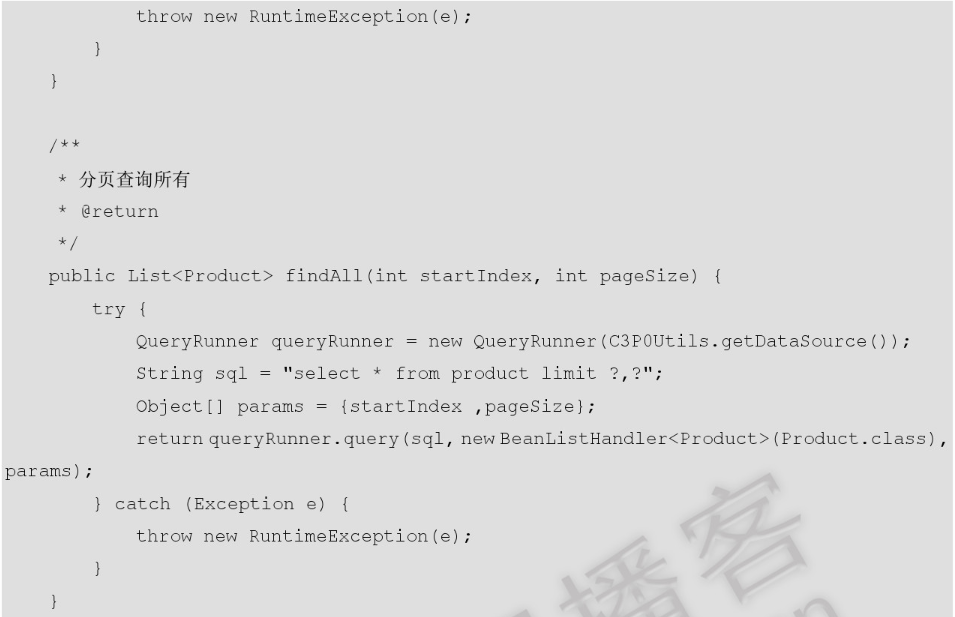


1. 编写service进行分页查询，并将数据封装到pageBean

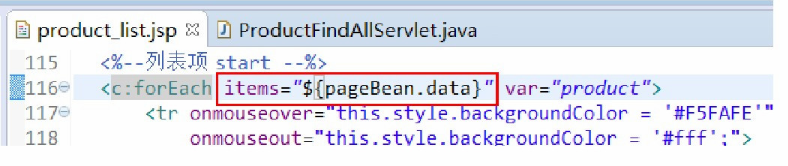


1. 编写dao，查询总记录数和分页数





1. 渲染jsp页面，显示分页数据



1. 关于${pageContext.request.contextPath}

${pageContext.request.contextPath}用于解决使用相对路径时出现的问题，它的作用是取出所部署项目的名字，定位项目中资源位置时使用这个可以有效防止404错误出现。

1. 关于request.getRequestDispatcher

getRequestDispatcher是服务器内部跳转，地址栏信息不变，只能跳转到web应用内的网页。   
sendRedirect是页面重定向，地址栏信息改变，可以跳转到任意网页。

1. 关于@WebServlet("/")

在servlet3.0以后，我们可以不用再web.xml里面配置servlet，只需要加上@WebServlet注解就可以修改该servlet的属性了。

