

实例 7-1 利用 JDBC 显示宠物分类

源程序：pom.xml 配置文件

```
<!-- mysql 数据库驱动程序包-->
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.28</version>
</dependency>
<!-- jstl 程序包-->
<dependency>
    <groupId>org.apache.taglibs</groupId>
    <artifactId>taglibs-standard-spec</artifactId>
    <version>1.2.1</version>
</dependency>
<dependency>
    <groupId>org.apache.taglibs</groupId>
    <artifactId>taglibs-standard-impl</artifactId>
    <version>1.2.1</version>
</dependency>
```

源程序：Category.java 文件

```
public class Category {
    private int id;           //编号
    private String name;      //名称
    public Category() {}
    public int getId() { return id; }
    public void setId(int id) { this.id = id; }
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }
}
```

源程序：GetCategoryServlet.java 文件

```
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.annotation.*;
import java.io.IOException;
import java.sql.*;
import java.util.ArrayList;
@WebServlet("/GetCategoryServlet")
public class GetCategoryServlet extends HttpServlet {
```

```

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
    Connection conn = null;
    Statement stmt = null;
    ResultSet rs = null;
    ArrayList<Category> categoryList = null;
    //1. 加载并注册数据库驱动
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
    try {
        //2 . 获取数据库连接
        conn= DriverManager.getConnection("jdbc:mysql://localhost:3306/petstore","root","root");
        //3 . 创建 Statement 对象, 准备 SQL 语句
        stmt = conn.createStatement();
        String sql = "select id,name from category";
        //4 . 使用 Statement 对象执行 SQL 语句
        rs = stmt.executeQuery(sql);
        //5 . 处理 ResultSet 结果集
        // 将结果数据封装为 Category 对象, 加入到
        ArrayList<Category> categoryList = new ArrayList<Category>();
        while(rs.next()){
            Category category = new Category();
            category.setId(rs.getInt("id"));
            category.setName(rs.getString("name"));
            categoryList.add(category);
        }
        //6 . 关闭连接, 释放资源
        rs.close();
        stmt.close();
        conn.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    HttpSession session = request.getSession();
    //分类对象列表写入 Session 域, JSP 页面通过 EL 表达式读取并展示
    session.setAttribute("categoryList",categoryList);
    response.sendRedirect(request.getContextPath()+ "/categoryList.jsp");
}
}

```

源程序：categoryList.jsp 文件

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<html>
<head>
<title>利用 JDBC 显示宠物分类</title>
</head>
<body>
<h3>宠物分类列表</h3>
<table>
<tr>
<th>编号</th><th>名称</th><th>宠物列表</th>
</tr>
<%-- category 为每一次迭代时从集合中取得的对象变量， --%>
<%-- categoryList 类型为列表，来自 Session 域 --%>
<c:forEach var="category" items="${categoryList}" >
<tr>
<td>${category.id}</td><td>${category.name}</td>
<td><a href="${pageContext.request.contextPath}/
GetPetServlet?category_id=${category.id}">查看</a></td>
</tr>
</c:forEach>
</table>
<hr>
<a href="${pageContext.request.contextPath}
/AddCategoryServlet?category_name=昆虫">新增分类</a>
</body>
</html>
```

程序说明：

- 本案例测试时 URL 地址栏应输入 GetCategoryServlet，因为该 Servlet 连接数据库、执行 SQL 语句、获取数据并写入 Session，再重定向至 JSP 页面显示数据信息。
- 本案例中数据库的用户名为 root，密码为 root，读者需要根据实际情况做相应的修改。
- 若页面显示文字信息出现乱码，需在数据库连接字符串中添加编码参数，示例如下：
jdbc:mysql://localhost:3306/petstore?characterEncoding=utf-8