EL表达式

目标：理解EL表达式语言在jsp 中的使用

掌握EL表达式语言在MVC设计模式中的主要作用

什么EL表达式？

为了让输出存储在标志位置的java对象的值，jsp2.0引入了一种简单的语言，表达式语言（Expression Language）,简称EL，是jsp2.0最主要的特征之一，但是不支持更早的版本

标志位置：对应的是属性的保存范围（page(pageContext) request session application）

EL 的主要功能：

精确的访问存储对象

Bean属性的简略记法

空值取代错误信息

EL的语法和调用：

语法：${ Expression }

jsp中有四种属性保存范围，要输出保存的属性，使用表达式语言只要使用属性名字${名字}

|  |
| --- |
| <%@page contentType=*"text/html; charset=GBK"*%>  <%  pageContext.setAttribute("pagecon","WANCZY");  %>  <%  String str = "";  **if**(**null**!=pageContext.getAttribute("pagecon")){  str = (String)pageContext.getAttribute("pagecon");  }  %>  <h1><%=str %></h1> |

以上是我们之前去的属性的形式，而且发现，如果属性没有设置，取得则会出现为null，这样子合理吗？

下面使用El表达式：

|  |
| --- |
| <%@page contentType=*"text/html; charset=GBK"*%>  <%  pageContext.setAttribute("pagecon","JJM");  %>  <h1>属性内容为：${pagecon}</h1> |

如果在page(pageContext) request session application属性范围里面都保存同一个名字的属性，怎会如何？

|  |
| --- |
| <%@page contentType=*"text/html; charset=GBK"*%>  <%  request.setAttribute("pagecon","WANCZY");  session.setAttribute("pagecon","www.wanczy.com");  application.setAttribute("pagecon","Jiangjianmin");  pageContext.setAttribute("pagecon","JJM");  %>  <h1>属性内容为：${pagecon}</h1> |

设置不同范围的同一个名称的属性，使用EL表达式的取值的顺序为：PageContext,HttpServletRequest、HttpSession、ServletContext

当然根据隐式对象进行取值：

page——pageScope

request——requestScope

session——sessionScope

application——applicationScope

在EL表达式中使用隐式对象取得相应属性范围的值

|  |
| --- |
| <%@page contentType=*"text/html; charset=GBK"*%>  <%  request.setAttribute("pagecon","WANCZY");  session.setAttribute("pagecon","www.wanczy.com");  application.setAttribute("pagecon","Jiangjianmin");  pageContext.setAttribute("pagecon","JJM");  %>  <h1>属性内容为：${requestScope.pagecon}</h1> |

El表达式主要用于MVC设计模式中：

|  |
| --- |
| package com.jjm.servlet;  import java.io.IOException;  import javax.servlet.ServletException;  import javax.servlet.http.HttpServlet;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  public class ElServletDemo01 extends HttpServlet {  public void doGet(HttpServletRequest request, HttpServletResponse response)  throws ServletException, IOException {  request.setAttribute("name", "WANCZY");  request.getRequestDispatcher("Eldemo/demo04.jsp").forward(request,response);    }  public void doPost(HttpServletRequest request, HttpServletResponse response)  throws ServletException, IOException {  doGet(request, response);  }  } |
| <%@page contentType=*"text/html; charset=GBK"*%>  <h1>属性内容为：${name}</h1> |

以上程序是在Servlet中设置了一个request范围属性，然后在jsp中使用el表达式接收属性值

Bean属性的简略记法：

现在新建一个Person类，有姓名和年龄的属性，然后在Servlet端new一个Person对象，然后通过request范围设置一个属性，属性值为此对象，然后在jsp中使用el表达式输出此对象的姓名和年龄属性：

|  |
| --- |
| package com.jjm.servlet;  import java.io.IOException;  import javax.servlet.ServletException;  import javax.servlet.http.HttpServlet;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  import comjjm.classdemo.Person;  public class ElServletDemo02 extends HttpServlet {  public void doGet(HttpServletRequest request, HttpServletResponse response)  throws ServletException, IOException {  Person p = new Person("张三",30);  request.setAttribute("pseron", p);  request.getRequestDispatcher("Eldemo/demo05.jsp").forward(request, response);    }  public void doPost(HttpServletRequest request, HttpServletResponse response)  throws ServletException, IOException {  doGet(request, response);  }    } |
| <%@page contentType=*"text/html; charset=GBK"*%>  <h1>姓名为：${pseron.name}</h1>  <h1>年龄为：${pseron.age}</h1> |

以上是一个简单的对象传递，并且使用el输出对象的属性

现在有一个Person类，同样还是有姓名和年龄的属性，还有一个员工类Employee，有工资和岗位的属性，还有一个公司类Company，公司类有Person和Employee属性，现在通过Servlet设置属性，值为公司类的一个对象，然后要在jsp中使用el表达式输出此对象的属性Person中的姓名和年龄，输出此对象属性中的Employee的工资和岗位

|  |
| --- |
| package com.jjm.servlet;  import java.io.IOException;  import javax.servlet.ServletException;  import javax.servlet.http.HttpServlet;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  import comjjm.classdemo.Company;  import comjjm.classdemo.Employee;  import comjjm.classdemo.Person;  public class ElServletDemo03 extends HttpServlet {  public void doGet(HttpServletRequest request, HttpServletResponse response)  throws ServletException, IOException {  Person p = new Person("王五",30);  Employee e = new Employee("软件工程师",5000.00);  Company c = new Company(p,e);  request.setAttribute("com", c);  request.getRequestDispatcher("Eldemo/demo06.jsp").forward(request, response);  }  public void doPost(HttpServletRequest request, HttpServletResponse response)  throws ServletException, IOException {  doGet(request, response);  }  } |
| <%@page contentType=*"text/html; charset=GBK"*%>  <h1>Person对象姓名为：${com.person.name}</h1>  <h1>Person对象年龄为：${com.person.age}</h1>  <h1>Employee对象岗位为：${com.employee.post}</h1>  <h1>Employee对象薪水为：${com.employee.salary}</h1> |

访问集合：

|  |
| --- |
| package com.jjm.servlet;  import java.io.IOException;  import java.util.ArrayList;  import java.util.List;  import javax.servlet.ServletException;  import javax.servlet.http.HttpServlet;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  import comjjm.classdemo.Person;  public class ElServletDemo02 extends HttpServlet {  public void doGet(HttpServletRequest request, HttpServletResponse response)  throws ServletException, IOException {  Person p = new Person("张三",31);  Person p2 = new Person("李四",32);  Person p3 = new Person("王五",33);  Person p4 = new Person("赵六",34);  Person p5 = new Person("田七",35);  List list = new ArrayList();  list.add(p);  list.add(p2);  list.add(p3);  list.add(p4);  list.add(p5);    request.setAttribute("list", list);  request.getRequestDispatcher("Eldemo/demo05.jsp").forward(request, response);    }  public void doPost(HttpServletRequest request, HttpServletResponse response)  throws ServletException, IOException {  doGet(request, response);  }    } |
| <%@page contentType=*"text/html; charset=GBK"*%>  <%@page import=*"java.util.\*"* %>  <!-- <h1>姓名为：${pseron.name}</h1>  <h1>年龄为：${pseron.age}</h1> -->  <%  List list = (List)request.getAttribute("list");  Iterator it = list.iterator();  **while** (it.hasNext()){  pageContext.setAttribute("per",it.next());  %>  <h4>${per.name}</h4>  <h4>${per.age}</h4>  <%  }  %> |

总结：

掌握El表达式的基本语法

掌握el表达式在MVC设计模式中的使用