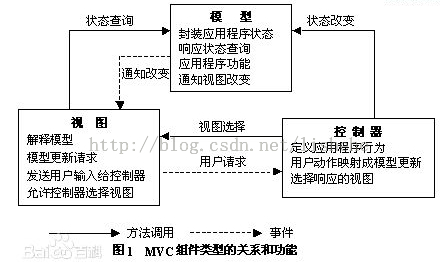
[深入浅出Java MVC(Model View Controller) ---- （JSP + servlet + javabean实例）](http://blog.csdn.net/lishehe/article/details/23125111)

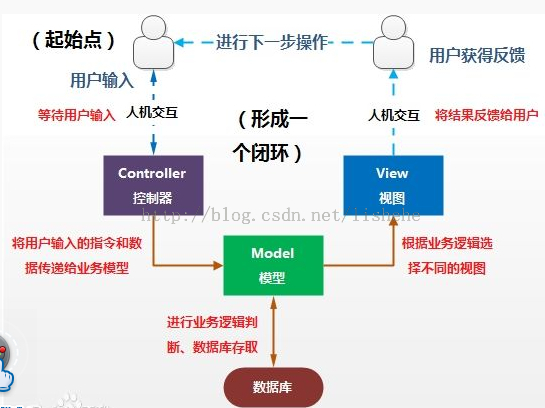
    在DRP中终于接触到了MVC,感触是确实这样的架构系统灵活性不少，现在感触最深的就是使用tomcat作为服务器发布比IIS好多了，起码发布很简单，使用起来方便。

首先来简单的学习一下MVC的基础知识，MVC全名是Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑和数据显式分离的方法组织代码，将业务逻辑被聚集到一个部件里面，在界面和用户围绕数据的交互能被改进和个性化定制的同时而不需要重新编写业务逻辑。

**概览**



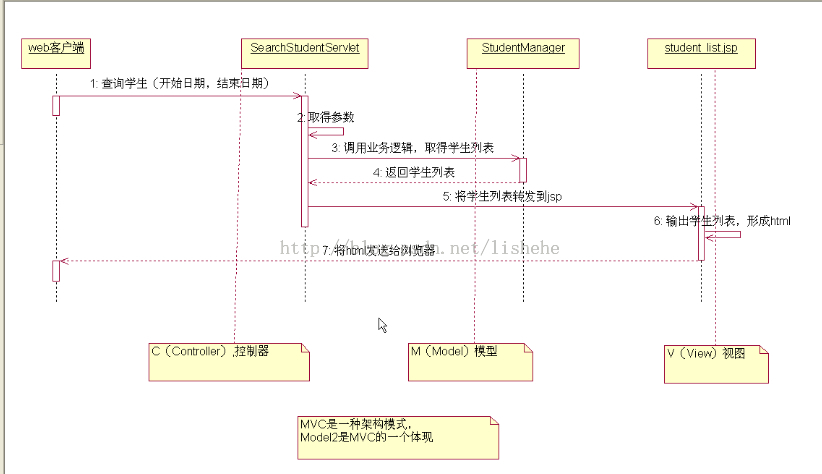
    MVC被独特的发展起来用于映射传统的输入、处理和输出功能在一个逻辑的图形化用户界面的结构中，MVC是分层的思想的体现，但是区别于三层设计模式（区别以后再和大家分享）。



   MVC是一个框架模式，它强制性的使应用程序的输入、处理和输出分开。使用MVC应用程序被分成三个核心部件：模型、视图、控制器。它们各自处理自己的任务。最典型的MVC就是JSP + servlet + javabean的模式。

**实例解析**

**UML图：大家熟悉MVC的调用流程逻辑**



**第一：JSP：由页面指令和HTML组成的查询界面query\_condention.jsp，也就是咱们现在的html页和asp页面类似。**

**[html]** [view plain](http://blog.csdn.net/lishehe/article/details/23125111) [copy](http://blog.csdn.net/lishehe/article/details/23125111)

[print](http://blog.csdn.net/lishehe/article/details/23125111)[?](http://blog.csdn.net/lishehe/article/details/23125111)[在CODE上查看代码片](https://code.csdn.net/snippets/279991)

1. **<span** style="font-family:FangSong\_GB2312;font-size:18px;"**><**%@ page language="java" contentType="text/html;charset=GBK"%**>**
2. **<html>**
3. **<head>**
4. **<title>**学生信息**</title>**
5. **</head>**
6. **<body>**
7. **<form** action="SearchStudentServlet" method="post"**>**
8. 出生日期：**<input** type="text" name="beginDate"**>**至**<input** type="text" name="endDate"**>**
9. **<input** type="submit" value="查询学生"**>**
10. **</form>**
11. **</body>**
12. **</html></span>**

**第二：控制层 SearchStudentServlet用来接受客户的请求，来处理流程，调用Model（StudentManager.java）,转发到要请求的后台服务器的student\_list.jsp页面**

**[html]** [view plain](http://blog.csdn.net/lishehe/article/details/23125111) [copy](http://blog.csdn.net/lishehe/article/details/23125111)

[print](http://blog.csdn.net/lishehe/article/details/23125111)[?](http://blog.csdn.net/lishehe/article/details/23125111)[在CODE上查看代码片](https://code.csdn.net/snippets/279991)

1. **<span** style="font-family:FangSong\_GB2312;font-size:18px;"**>**import java.text.\*;
2. import java.util.\*;
3. import java.io.\*;
4. import javax.servlet.http.\*;
5. import javax.servlet.\*;
7. import com.bjpowernode.exam.model.\*;
8. import com.bjpowernode.exam.manager.\*;
10. public class SearchStudentServlet extends HttpServlet {
12. public void doGet(HttpServletRequest request, HttpServletResponse response)
13. throws ServletException, IOException {
14. doPost(request, response);
15. }
17. public void doPost(HttpServletRequest request, HttpServletResponse response)
18. throws ServletException, IOException {
20. String sBeginDate = request.getParameter("beginDate");
21. String sEndDate = request.getParameter("endDate");
23. Date beginDate = new Date();
24. Date endDate = new Date();
25. try {
26. beginDate = new SimpleDateFormat("yyyy-MM-dd").parse(sBeginDate);
27. endDate = new SimpleDateFormat("yyyy-MM-dd").parse(sEndDate);
28. }catch(Exception e) {
29. e.printStackTrace();
30. }
32. StudentManager studentManager = new StudentManagerImpl();
33. List**<Student>** studentList = studentManager.findStudentList(beginDate, endDate);
35. //将学生列表设置到requet范围中
36. //request.setAttribute("student\_list", studentList);
38. //转发,转发是在服务器端转发的，客户端是不知道的
39. //request.getRequestDispatcher("/student\_list.jsp").forward(request, response);

42. //将studentList放到session中
43. HttpSession session = request.getSession();
44. session.setAttribute("student\_list", studentList);
46. //重定向，不会共享request
47. //以下写法错误,该 "/"代表了8080端口
48. //response.sendRedirect("/student\_list.jsp");
49. response.sendRedirect(request.getContextPath() + "/student\_list.jsp");
50. }
51. }**</span>**

**第三 ：student\_list.jsp页面接收数据形成html,返回到浏览器，渲染在界面上**

**[java]** [view plain](http://blog.csdn.net/lishehe/article/details/23125111) [copy](http://blog.csdn.net/lishehe/article/details/23125111)

[print](http://blog.csdn.net/lishehe/article/details/23125111)[?](http://blog.csdn.net/lishehe/article/details/23125111)[在CODE上查看代码片](https://code.csdn.net/snippets/279991)

1. <span style="font-family:FangSong\_GB2312;font-size:18px;"><%@ page language="java" contentType="text/html;charset=GBK"%>
2. <%@ page **import**="java.util.\*"%>
3. <%@ page **import**="java.text.\*"%>
4. <%@ page **import**="com.bjpowernode.exam.model.\*"%>
5. <%@ page **import**="com.bjpowernode.exam.manager.\*"%>
6. <html>
7. <head>
8. <title>学生信息</title>
9. <style type="text/css">
10. /\*表格宽度为1px，实线，黑色\*/
11. table{
12. border:1px solid black;
13. border-collapse:collapse;
14. }
16. td {
17. border:1px solid black;
18. border-collapse:collapse;
19. }
21. </style>
22. </head>
23. <body>
24. <table border="1">
25. <tr>
26. <td>学生代码</td>
27. <td>姓名</td>
28. <td>性别</td>
29. <td>出生日期</td>
30. <td>联系电话</td>
31. <td>家庭住址</td>
32. <td>班级名称</td>
33. <td>年龄</td>
34. </tr>
35. <%
36. //List<Student>  studentList = (List)request.getAttribute("student\_list");
37. List<Student>  studentList = (List)session.getAttribute("student\_list");
38. **for** (Iterator<Student> iter=studentList.iterator(); iter.hasNext();) {
39. Student student = iter.next();
40. %>
41. <tr>
42. <td><%=student.getStudentId()%></td>
43. <td><%=student.getStudentName()%></td>
44. <td><%=student.getSex()%></td>
45. <td><%=**new** SimpleDateFormat("yyyy-MM-dd").format(student.getBirthday())%></td>
46. <td><%=student.getContactTel()%></td>
47. <td><%=student.getAddress()%></td>
48. <td><%=student.getClasses().getClassesName()%></td>
49. <%
50. **long** b = 1000L\*60L\*60L\*24L\*365L;
51. **long** a = System.currentTimeMillis() - student.getBirthday().getTime();
52. %>
53. <td><%=a/b%></td>
54. </tr>
55. <%
56. }
57. %>
58. </table>
59. </body>
60. </html></span>

在View的student\_list.jsp页面中是大量的html和java代码的混合，在查询条件界面query\_condention.jsp主要是html，因为不涉及后台数据的交互.

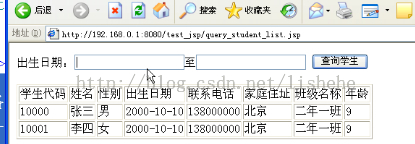
**第四：xml配置Servlet:**

**[html]** [view plain](http://blog.csdn.net/lishehe/article/details/23125111) [copy](http://blog.csdn.net/lishehe/article/details/23125111)

[print](http://blog.csdn.net/lishehe/article/details/23125111)[?](http://blog.csdn.net/lishehe/article/details/23125111)[在CODE上查看代码片](https://code.csdn.net/snippets/279991)

1. **<span** style="font-family:FangSong\_GB2312;font-size:18px;"**><?xml** version="1.0" encoding="ISO-8859-1"**?>**
2. **<web-app** xmlns="http://java.sun.com/xml/ns/j2ee"
3. xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4. xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app\_2\_4.xsd"
5. version="2.4"**>**
6. **<servlet>**
7. **<servlet-name>**SearchStudentServlet**</servlet-name>**
8. **<servlet-class>**SearchStudentServlet**</servlet-class>**
9. **</servlet>**
10. **<servlet-mapping>**
11. **<servlet-name>**SearchStudentServlet**</servlet-name>**
12. **<url-pattern>**/SearchStudentServlet**</url-pattern>**
13. **</servlet-mapping>**
15. **</web-app></span>**

**第五：显示查询结果**



**总结**

    以上query\_condention.jsp（输入查询条件）、SearchStudentServlet.java（请求控制Control分）+student\_list.jsp（界面显示）的组合相当于三层中的U层，都与界面的显示相关，而StudentManager.java才是进入业务处理相当于三层的B层。故，我们可以简单理解成，MVC就是java基于U层的又一个细化，将界面显示和请求处理做了进一步细化分工。

    MVC被独特的发展起来用于映射传统的输入、处理和输出功能在一个逻辑的图形化用户界面的结构中，MVC是分层的思想的体现，但是区别于三层设计模式（区别以后再和大家分享）。