

Java + MyEclipse + Tomcat (二)配置Servlet及简单实现表单提交

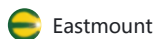
原创 Eastmount 2015-05-07 04:42:12 12428 收藏 2

展开



Python+TensorFlow人工智能

该专栏为人工智能入门专栏，采用Python3和TensorFlow实现人工智能相关算法。前期介绍安装流程、基础语法、



¥9.90

订阅

在Java EE应用编程中Servlet是基础，JSP是建立在Servlet基础之上的，其他Web框架如Struts、WebWork和Spring MVC都是基于Servlet的。本文主要讲述MyEclipse和Tomcat配置Servlet的过程和简单实现表单提交功能。

希望对大家有所帮助，参考书籍《Java EE企业级应用开发实例教程》。

[Java+MyEclipse+Tomcat \(一\)配置过程及jsp网站开发入门](#)

一. Servlet编程

1.Web基础知识

在讲述Servlet之前，先了解下Web基础知识。Web(World Wide Web，万维网)本质就是Internet所有文档的集合，Web文档主要类型有HTML网页、CSS、JavaScript、各种动态网页、图片、声音和视频等。

Web文档保存在Web站点(Site)上，Web站点驻留在Web服务器上。Web服务器是一种软件系统，提供Web文档的管理和请求服务，常见的Web服务器软件有Apache、IIS、WebLogic和Tomcat等。每个服务器都有一个唯一的IP地址，Web服务器对外都有一个服务端口，默认是80或8080端口。

Web文档都有一个唯一的地址，通过URL格式来进行定位，其格式为：

协议://IP地址:端口/站点名/目录/文件名

其中协议主要有HTTP、HTTPS和FTP。根据不同的协议，默认端口可以省略，HTTP/HTTPS为80端口，FTP为21端口。例：

http://210.30.108.30:8080/test/admin/login.jsp

Web服务器接收到请求后，根据URL定位到相应文档，根据文档类型进行对应的处理，将文档通过网络发送到客户端，一般是浏览器，用户即可查看或下载请求的文档。Web通过请求/响应(Request/Response)模式进行工作，即由浏览器使用URL对Web文档进行请求，Web服务器接收并处理请求，处理结束后将响应内容发送到浏览器。

Web请求方式主要包括GET、POST、PUT、DELETE和HEAD。其中GET请求直接返回请求的文档，同时传递参数在URL中；POST请求将传递到Web服务器的数据保存到数据流中，只有通过表单提交形式才能实现。如下：

```
Get请求：
http://localhost:8080/web01/main.do?id=1&password=123456
POST请求：
<form action = "add.do" method = "post">
    <input type="text" name="username">
    <input type="submit" value="提交">
</form>
```

这些知识大家应该都非常熟悉了，我就不再介绍了，引出表单就开始叙述Servlet。

2.什么是Servlet

在Sun公司制定Java EE规范初期，为实现动态Web而引入了Servlet，用于替代笨重的CGI(通用网关接口)，实现了Java语言编程的动态Web技术，奠定了Java EE的基础。后来为进一步简化动态Web网页的生成，并且在微软公司推出了ASP(Active X服务系统页面)技术的竞争下，Sun推出了JSP规范，进一步简化了Web网页的编程。但JSP在进行HTTP请求处理方面不如Servlet方便和规范，Servlet在当今的MVC模式Web开发中牢牢占据一地。并且现在流行的Web框架基本基于Servlet技术，如Struts、WebWork和Spring MVC等。只有掌握了Servlet才能真正掌握Java Web编程的核心和精髓。

那么，究竟Servlet是什么东东呢？

Servlet是运行在Web容器的类，能处理Web客户的HTTP请求，并产生HTTP响应。

Servlet是Java EE规范定义的Web组件，运行在Web容器中。由Web容器负责管理Servlet的声明周期，包括创建和销毁Servlet对象。客户不能直接创建Servlet对象和调用Servlet的方法，只能通过向Web服务器发出HTTP请求，间接调

用Servlet的方法。这是Servlet与普通Java类的重要区别。

Sun在如下两个包中提供了Servlet的全部接口和类：

1. `javax.servlet` 包含支持所有协议的通用Web组件接口和类
2. `javax.servlet.http` 包含支持HTTP协议的接口和类

Servlet可以完成Web组件具有的所有功能，如下：

1. 接收HTTP请求
2. 取得请求信息，包括请求头和请求参数数据
3. 调用其他Java类方法完成具体的业务功能
4. 生成HTTP响应，包括HTML和非HTML响应
5. 实现到其他Web组件的跳转，包括重定向和转发

编写Servlet需要引入的两个包和Java I/O包：

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
```

编写接收HTTP请求并进行HTTP响应的Servlet要继承`javax.servlet.http.HttpServlet`。Servlet类的定义如下：

```
public class LoginAction extends HttpServlet {}
```

每个Servlet一般都需要重写`doGet`方法和`doPost`方法。当用户使用GET方式请求Servlet时，Web容器调用`doGet`方法处理请求；当用户使用POST方法请求Servlet时，Web容器调用`doPost`方法。

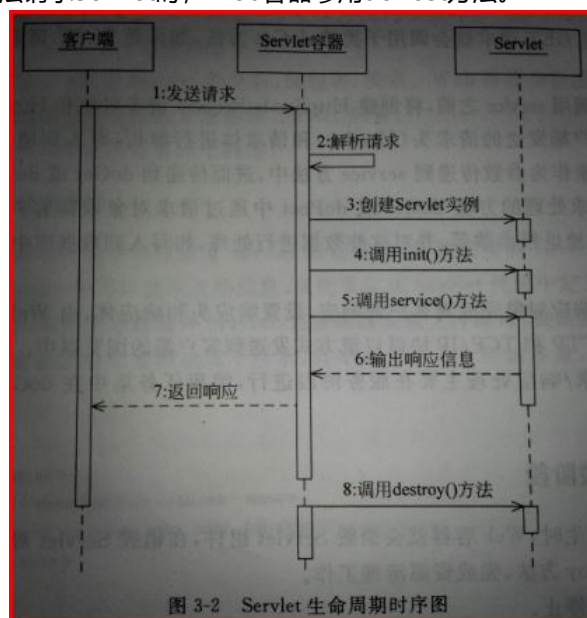


图 3-2 Servlet 生命周期时序图

Servlet生命周期时序图如上图所示，包括加载类和实例化阶段、初始化阶段、处理请求阶段和销毁阶段。

二. 配置Servlet

Servlet作为Web组件可以处理HTTP请求/响应，因而对外要求一个唯一的URL地址。但由于Servlet是一个Java类文件，不像JSP那样直接存放在Web目录下就能获得URL请求访问地址。Servlet必须在Web的配置文件`/WEB-INF/web.xml`中进行配置和映射才能响应HTTP请求。

1. Servlet声明

它的任务是通知Web容器Servlet的存在，声明语法如下：

```
<servlet>
<servlet-name>HomePageAction</servlet-name>
<servlet-class>servlet.HomePageAction</servlet-class>
</servlet>
```

其中<servlet-name>声明Servlet的名字，一般与Servlet的类名相同即可，要求在一个web.xml文件内名字唯一。<servlet-class>指定Servlet的全名，即包名.类型。Web容器会根据此定义载入类文件到内容中，进而调用默认构造方法创建Servlet对象。

Servlet初始参数如数据库Driver、URL、账号和密码等信息，在Servlet中可以读取这些信息，如下定义了一个初始参数，即数据库的JDBC驱动。

```
<servlet>
<servlet-name>HomePageAction</servlet-name>
<servlet-class>servlet.HomePageAction</servlet-class>
<init-param>
  <param-name>driveName</param-name>
  <param-value>sun.jdbc.odbc.JdbcOdbcDriver</param-value>
</init-param>
</servlet>
```

在Servlet中可以通过ServletConfig取得定义的初始化参数，取得以上定义的参数的代码如下：

```
private Connection cn=null;    // 定义数据库连接对象
private String driverName=null; // 数据库驱动器
//取得Servlet定义的初始参数
driverName=config.getInitParameter("driverName");
//根据Servlet初始参数连接数据库
Class.forName(driverName);
cn=DriverManager.getConnection("jdbc:odbc:cityoa");
```

其中config是在Servlet中定义的ServletConfig类型的属性变量，由init方法取得它的实例。可见连接不同数据库，直接修改配置文件即可，不需要代码的修改和重新编译。

2.Servlet映射

任何Web文档在Internet上都要有一个URL地址才能被请求访问，Servlet不能像JSP一样直接放在Web的发布目录下，因此Servlet需要单独映射URL地址。在WEB-INF/web.xml中进行Servlet的URL映射。

```
<servlet-mapping>
  <servlet-name>HomePageAction</servlet-name>
  <url-pattern>/loginAction.do</url-pattern>
</servlet-mapping>
```

上面代码是绝对地址方式映射，只能映射到一个地址。其中Servlet-name名称和Servlet声明中要一致，URL的格式如下：/目录/目录/文件名.扩展名。而匹配目录模式映射方式如下，可以请求多个URL。

```
<servlet-mapping>
  <servlet-name>MainAction</servlet-name>
  <url-pattern>/main/*</url-pattern>
</servlet-mapping>
```

只要是以/main为开头的任何URL都能请求此Servlet。如下：

http://localhost:8080/web01/main/login.jsp

http://localhost:8080/web01/main/info/add.do

同样能你还能使用匹配扩展名模式映射方式，也可以响应多地址的请求。

```

<servlet-mapping>
    <servlet-name>MainAction</servlet-name>
    <url-pattern>*.action</url-pattern>
</servlet-mapping>

```

以上配置中扩展名为action的任何请求均能被Servlet响应。如：

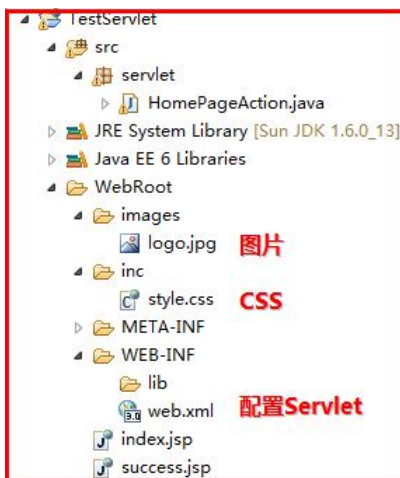
http://localhost:8080/web01/login.action

http://localhost:8080/web01/main/info/add.action

注意：你不能混合使用两种匹配模式，否则会映射出错，如/main/*.action这种是错误的混合模式。

三. 简单实现表单提交功能

创建Web Project项目，项目名称为TestServlet。项目结构如下图所示：



在WebRoot文件夹下创建images文件夹，并添加图片logo.jpg。添加inc文件夹，新建style.css文件，代码如下：

```

.main {
    width: 1024px;
    text-align:left;
}
.font {
    font-family: "Trebuchet MS";
    font-size: 14px;
    font-weight: bold;
    color: #FFFFFFF;
}
.div {
    margin: 0px;
    padding: 0px;
    width: 1014px;
}
.tdBgColor{
    background-color:#6b1101;
}
a{
    font-family: "Trebuchet MS";
    font-size: 14px;
    font-weight: bold;
    color: #FFFFFFF;
    line-height:50px;
    text-decoration:none;
}

```

```

a.hover{
    font-family: "Trebuchet MS";
    font-size:14px;
    font-weight: bold;
    color:#0000FF;
    line-height:50px;
    text-decoration:underline;
    padding-bottom:1px;

}

a.visited{
    font-family: "Trebuchet MS";
    font-size:14px;
    font-weight: bold;
    color:#000066;
    line-height:50px;
    text-decoration:none;

}

a.active{
    font-family: "Trebuchet MS";
    font-size:14px;
    font-weight: bold;
    color:#0000FF;
    line-height:50px;
    text-decoration:none;
    padding-bottom:1px;

}

```

然后修改index.jsp主页，代码如下：

```

<%@ page language="java" import="java.util.*" pageEncoding="UTF-8"%>
<%
String path = request.getContextPath();
String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";
%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <base href="<%=basePath%>">

    <title>火车票订票系统</title>
    <meta http-equiv="pragma" content="no-cache">
    <meta http-equiv="cache-control" content="no-cache">
    <meta http-equiv="expires" content="0">
    <meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
    <meta http-equiv="description" content="This is my page">
    <!--
    <link rel="stylesheet" type="text/css" href="styles.css">
    -->
  </head>

  <body>
    <!-- 界面布局 -->
    <div align="center">
      <div class="style">
        <table width="1024" height="150" cellpadding="0" cellspacing="0" >
          <!-- 动态首页订票界面 同时获取年月日和动态滚动消息 -->
          <tr>
            <td colspan="2">
              

```

```

        </td>
    </tr>
    <tr>
        <td width="205" bgcolor="#b7d7ec">
            <p align="center">
                <!-- 注意: 方法today.getYear()已经过时, 需要添加1900 -->
                <script language=JavaScript>
                    today = new Date();
                    function initArray(){
                        this.length=initArray.arguments.length;
                        for(var i=0;i<this.length;i++)
                            this[i+1]=initArray.arguments[i]; }
                    var d=new initArray(
                        "星期日","星期一","星期二","星期三","星期四", "星期五","星期六");
                    document.write("<font color=##ff0000 style='font-size:12pt;font-
family: 宋体'> ",
                    today.getYear()+1900,"年",today.getMonth()+1,"月",today.getDate(),"日
",d[today.getDay()+1],
                    "</font>" );
                </script>
            </p>
        </td>
        <td width="819" bgcolor="#b7d7ec">
            <marquee direction="left" onmouseover=this.stop()
onmouseout=this.start() scrollAmount=3 scrollDelay=100>
            <FONT style="FONT-SIZE: 18px"color=blue>欢迎使用火车票订票系统管理系统 , 如有不足, 敬请指导!
        </FONT>
    </marquee>
    </td>
    </tr>
</table>
</div>

<table width="100%" height="75%" border="0" align="center">
    <tr>
        <td width="40%" height="20%"> </td>
        <td width="20%">
            <!-- 中间登录部分布局 提交表单 配置web.xml -->
            <form action="loginAction.do">
                <table width="500" height="80%" border="2" bordercolor="#12A0F5" bgcolor="#dfeaf1">
                    <tr>
                        <td></td>
                        <td align="center"><br />系统管理员登录 <br/><br/>
                            用户名: <input type="text" name="userid" size="19" maxlength="19" /><br/><br/>
                            密 码: <input type="password" name="password" size="20" maxlength="19" /><br
                        </td>
                    </tr>
                    <tr>
                        <td>
                            <input type="submit" name="Submit" value="登录"/>
                            <input type="reset" name="Submit" value="重置"
                        </td>
                        <td></td>
                    </tr>
                </table>
            </form>
        </td>
    </tr>
    <td width="40%"></td>
    </tr>
    <tr><td height="67" colspan="3">
        <!-- 底部版权所有界面 -->
        <TABLE class=wrapper border=0 cellSpacing=0 cellPadding=0 width="100%" align=center>
            <TBODY>
                <TR>
                    <TD style="LINE-HEIGHT: 150%" align="center">
                        <HR style="WIDTH: 96%; HEIGHT: 4px; COLOR: #02457c">
                        <font size="2"> <SCRIPT language=javascript src=""></SCRIPT>
                        <!-- target=_blank开启新的网页 -->
                        <BR><A href="aboutme.jsp" target=_blank>关于我们</A> | <A href="wzsm.jsp">网站声明
                    </TD>
                </TR>
            </TBODY>
        </TABLE>
    </td>
    </tr>
</table>
</A>
    <BR>版权所有&copy;2014-2015 北京理工大学 Eastmount <BR>京ICP备10009636号 </font>

```

```

        </TD>
    </TR>
</TBODY>
</TABLE>
</td>
</tr>
</table>
</div>
</body>
</html>

```

此时运行Web网站，点击Run As=》MyEclipse Server Application。如图所示：



此时，输入用户名和密码登录效果如下图所示：



然后在src中新建文件夹Servlet，新建HomePageAction.java文件，代码如下：

```

package servlet;

import java.io.IOException;
import java.sql.*; // 导入数据库处理所有库
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.swing.JOptionPane;

// 用户登录处理Servlet 系统登录主页处理表单
public class HomePageAction extends HttpServlet {

    private Connection cn=null; // 定义数据库连接对象
    private String driverName=null; // 数据库驱动器
    private String url=null; // 数据库地址URL

    // 初始化方法，取得数据库连接对象

```



```

public void init(ServletConfig config) throws ServletException
{
    super.init(config);
    driverName=config.getInitParameter("driverName");
    url=config.getInitParameter("url");

    try {
        Class.forName(driverName);
        cn=DriverManager.getConnection(url);
    } catch(Exception e) {
        System.out.println("取得数据库连接错误: "+e.getMessage());
    }
}

//处理GET请求方法
public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException
{
    //取得用户注册表单提交的数据
    String userid=request.getParameter("userid");
    String password=request.getParameter("password");
    //判断登录账号为空, 则自动跳转到注册页面
    if(userid==null||userid.trim().length()==0) {
        response.sendRedirect("index.jsp");
        JOptionPane.showMessageDialog(null, "User name or password can't be empty!");
    }
    //判断登录密码为空
    if(password==null||password.trim().length()==0) {
        response.sendRedirect("index.jsp");
        JOptionPane.showMessageDialog(null, "User name or password can't be empty!");
    }
    //查询数据库和跳转到登录主界面
    try {
        //查询数据库操作

        //跳转到主界面
        response.sendRedirect("success.jsp");
    } catch(Exception e) {
        System.out.println("错误: "+e.getMessage());
        response.sendRedirect("index.jsp");
    }
}

//处理POST请求方法
public void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException
{
    doGet(request,response);
}

//销毁方法
public void destroy() {
    super.destroy();
    try {
        cn.close();
    }catch(Exception e) {
        System.out.println("关闭数据库错误: "+e.getMessage());
    }
}
}

```


同时在WEB-INF文件夹web.xml文件中配置Servlet，代码如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.0"
    xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
        http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd">
    <display-name></display-name>
    <welcome-file-list>
        <welcome-file>index.jsp</welcome-file>
    </welcome-file-list>

    <!-- 配置Servlet -->
    <servlet>
        <servlet-name>HomePageAction</servlet-name>
        <servlet-class>servlet.HomePageAction</servlet-class>
        <init-param>
            <param-name>driveName</param-name>
            <param-value>sun.jdbc.odbc.JdbcOdbcDriver</param-value>
        </init-param>
        <init-param>
            <param-name>url</param-name>
            <param-value>jdbc:odbc:cityoa</param-value>
        </init-param>
    </servlet>
    <servlet-mapping>
        <servlet-name>HomePageAction</servlet-name>
        <url-pattern>/loginAction.do</url-pattern>
    </servlet-mapping>
</web-app>
```

最后新建success.java文件，跳转后显示的页面。

```
<%@ page language="java" import="java.util.*" pageEncoding="UTF-8"%>
<%
String path = request.getContextPath();
String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";
%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
    <head>
        <base href="<%=basePath%>">

        <title>验证成功界面</title>

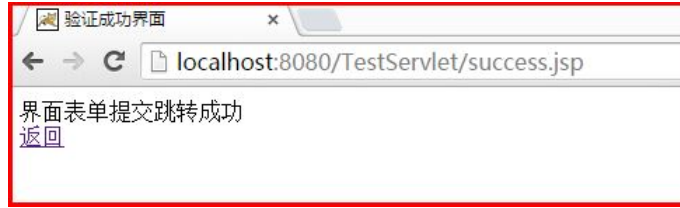
        <meta http-equiv="pragma" content="no-cache">
        <meta http-equiv="cache-control" content="no-cache">
        <meta http-equiv="expires" content="0">
        <meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
        <meta http-equiv="description" content="This is my page">
        <!--
        <link rel="stylesheet" type="text/css" href="styles.css">
        -->

    </head>

    <body>
        界面表单提交跳转成功 <br>
```

```
<a href="index.jsp">返回</a> </body>
</html>
```

显示效果如下图所示：



此时Servlet配置成功，而且表单跳转功能也实现了。你可能遇到如下错误：

```
2015-4-23 16:52:32 org.apache.catalina.core.ContainerBase addChildInternal
P: ContainerBase.addChild: start:
org.apache.catalina.LifecycleException: Failed to start component [StandardEngine[Catalina].StandardHost[localhost]
    at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:154)
    at org.apache.catalina.core.ContainerBase.addChildInternal(ContainerBase.java:901)
    at org.apache.catalina.core.ContainerBase.addChild(ContainerBase.java:877)
    at org.apache.catalina.core.StandardHost.addChild(StandardHost.java:649)
    at org.apache.catalina.startup.HostConfig.deployDirectory(HostConfig.java:1245)
    at org.apache.catalina.startup.HostConfig$DeployDirectory.run(HostConfig.java:1895)
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:441)
    at java.util.concurrent.FutureTask$Sync.innerRun(FutureTask.java:303)
    at java.util.concurrent.FutureTask.run(FutureTask.java:138)
    at java.util.concurrent.ThreadPoolExecutor$Worker.runTask(ThreadPoolExecutor.java:886)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:908)
    at java.lang.Thread.run(Thread.java:619)
Caused by: java.lang.IllegalArgumentException: Servlet mapping specifies an unknown servlet name HomePageAction
    at org.apache.catalina.core.StandardContext.addServletMapping(StandardContext.java:3320)
    at org.apache.catalina.core.StandardContext.addServletMapping(StandardContext.java:3299)
    at org.apache.catalina.deploy.WebXml.configureContext(WebXml.java:1438)
    at org.apache.catalina.startup.ContextConfig.webConfig(ContextConfig.java:1357)
    at org.apache.catalina.startup.ContextConfig.configureStart(ContextConfig.java:889)
    at org.apache.catalina.startup.ContextConfig.lifecycleEvent(ContextConfig.java:386)
    at org.apache.catalina.util.LifecycleSupport.fireLifecycleEvent(LifecycleSupport.java:117)
    at org.apache.catalina.util.LifecycleBase.fireLifecycleEvent(LifecycleBase.java:90)
    at org.apache.catalina.core.StandardContext.startInternal(StandardContext.java:5412)
    at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:150)
    ... 11 more
```

解决方案如下：

解决方案：

注意：

Caused by: java.lang.IllegalArgumentException: Servlet mapping specifies an unknown servlet name scondServlet

这一句的意思就是Servlet映射指定一个未知的servlet名称。

然后检查自己的web.xml中的配置，改正即可！

最后希望文章对你有所帮助，下一篇文章准备讲述JSP网站的一种经典网站模型和MyEclipse数据库配置相关知识。
如果文章有不足或错误的地方，还请海涵！

(By:Eastmount 2015-5-7 清晨4点 <http://blog.csdn.net/eastmount/>)



Eastmount 博客专家

原创文章 459 获赞 6629 访问量 517万+

关注

他的留言板