Http请求

一、新建动态web项目

1、新建动态web

2、新建一个Servlet

3、添加Servlet架包

4、新建一个用于登录的HTML文件，该文件必须置于web目录下，Tomcat默认WEB-INF目录下的文件不可访问

4、在WEB-INF目录下配置web.xml，实现HTML文件到Servlet的映射

```
<servlet>
    <servlet-name>LoginServlet</servlet-name>
    <servlet-class>LoginServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>LoginServlet</servlet-name>
    <url-pattern>/login</url-pattern>
  </servlet-mapping>
```

二、部署项目

1、选中整个项目方式

①、配置Tomcat服务器

②、单击项目，以服务器方式运行

③、跳转到页面后显示404（因为此时URL为：localhost：/项目名称/,如:http://localhost/j2ee/），要手动添加请求页面在访问

2、只选中请求页面方式

①、单击对应的请求页面（一般是HTML文件），以服务器方式运行

②、可见跳转到浏览器之后，自动显示请求页面:http://localhost/j2ee/login.html

三、浏览器请求

1、请求的URL为：localhost：/项目名称/请求页面

2、若出现页面不可正常显示，排查：

①、查看服务器端口

②、web.xml中地址映射是否正确

③、是否有具体要跳转的页面

3、注意请求方法

四、服务器获取请求参数

1、根据Http请求方式获取请求参数

```
protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
        String name=req.getParameter("name");
        String password=req.getParameter("password");
        System.out.println(name);
        System.out.println(password);
    }
```

2、注意传参与form表单的name属性值相同

```
<form action="login" method="post">
        账号: <input type="text" name="name"> <br>
        密码: <input type="password" name="password"> <br>
        <input type="submit" value="登录">
</form>
```

五、实现HTML到Servlet的两种映射方式

1、上述1.4所讲的web.xml

2、使用注解（简单描述）

> 实现向浏览器response登录情况
>
> HTML仍然使用4.2的form表单，在服务器端进行验证。

```
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet({ "/LoginServlet", "/login" })//此处使用注解实现HTML到Servlet的映射路径
public class LoginServlet extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
            throws ServletException, IOException {
        String name=req.getParameter("name");
        String password=req.getParameter("password");
        System.out.println(name);
        System.out.println(password);

        String html;
        if(name.equals("AlexanderBai")&&password.equals("123456")) {
            html="<div style='color:green'>success</div>";
```

```
        }else {
            html="<div style='color:red'>fail</div>";
        }
        PrintWriter out=resp.getWriter();
        out.println(html);
    }
}
```

六、Servlet中文问题

1、HTML的head标签中注明向服务器提交的编码方式

```
<meta http-equiv="Content-Type" content="text/html" charset="UTF-8">
```

2、服务器进行相应的解码（接受参数）和编码（response）

在请求方法相对应的服务器端处理方法中添加如下代码

```
req.setCharacterEncoding="UTF-8";//告诉服务器已utf-8解码
resp.setCharacterEncoding="UTF-8";//以utf-8编码方式响应
```

七、Servlet跳转

1、服务器端跳转（请求转发）

Http请求到服务器，服务器发现匹配的Servlet，当Servlet执行相应的步骤之后去执行**request**的**getRequestDispatcher()**方法，把同一个请求转发到另一个对象（HTML、JSP、Servlet等）

```
req.getRequestDispatcher("success.html").forward(req,resp);
```

2、客服端跳转（重定向）

> Http请求到服务器，服务器发现匹配的Servlet，当Servlet执行相应的步骤之后去执行**sendRedirect()**方法（注意这里用的是**response**的方法），立即向客户端返回这个响应（在响应头中包含新请求所对应的URL），并告诉客户端必须再发送一个新的请求。
>
> 紧接着客户端收到响应后再发送一个请求，服务器端收到新的请求后做出相应的处理。以上两个请求互不干扰、相互独立。

```
resp.sendRedirect("fail.html");
```

3、案例

> **LonginServlet.java**
>
> ```
> import java.io.IOException;
> import java.io.PrintWriter;
>
> import javax.servlet.ServletException;
> import javax.servlet.annotation.WebServlet;
> ```

```java
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet({ "/LoginServlet", "/login" })
public class LoginServlet extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
            throws ServletException, IOException {

        req.setCharacterEncoding("utf-8");//告诉服务器已utf-8解码
        resp.setCharacterEncoding("UTF-8");//以utf-8编码方式响应

        String name=req.getParameter("name");
        String password=req.getParameter("password");
        System.out.println(name);
        System.out.println(password);

        String html;
        if(name.equals("AlexanderBai")&&password.equals("123456")) {
            req.getRequestDispatcher("success.html").forward(req, resp);
        }else {
            resp.sendRedirect("fail.html");
        }
    }
}
```

longin.html

```html
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>登录页面</title>
</head>
<body>
    <form action="login" method="post">
        账号: <input type="text" name="name"> <br>
        密码: <input type="password" name="password"> <br>
        <input type="submit" value="登录">
    </form>
</body>
</html>
```

success.html

```html
<div style='color:green'>login success</div>
```

fail.html

```html
<div style='color:green'>login fail</div>
```