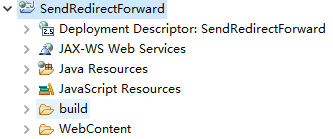
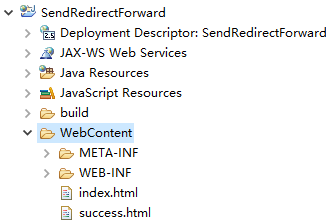
一.重定向和转发工程

1.新建一个SendRedirectForward的Web工程



2.在WebContent下新建index.html和success.html



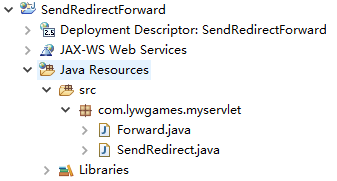
3.编写index.html

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <title>重定向和转发</title>  </head>  <body>  <a href="SendRedirect.action">重定向</a><br/>  <a href="Forward.action">转发</a>  </body>  </html> |

4.编写success.html

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <title>success</title>  </head>  <body>  <h1>success</h1>  </body>  </html> |

5.新建SendRedirect.java和Forward.java



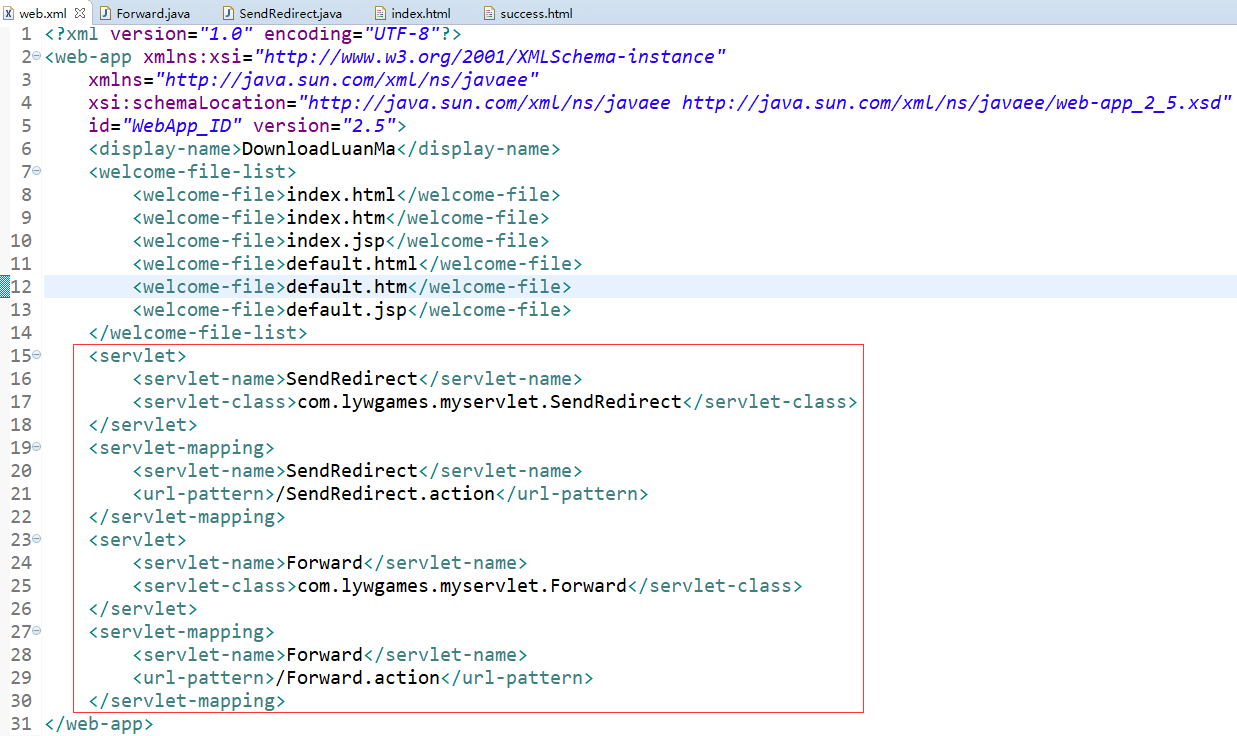
6.编写SendRedirect.java

|  |
| --- |
| package com.lywgames.myservlet;  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 SendRedirect extends HttpServlet {  private static final long serialVersionUID = 1L;  @Override  protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  resp.sendRedirect("success.html");  }  @Override  protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  doGet(req, resp);  }  } |

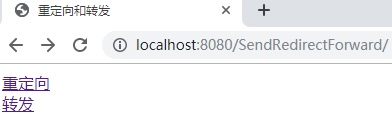
7.编写Forward.java

|  |
| --- |
| package com.lywgames.myservlet;  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 Forward extends HttpServlet {  private static final long serialVersionUID = 1L;  @Override  protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  req.getRequestDispatcher("success.html").forward(req, resp);  }  @Override  protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  doGet(req, resp);  }  } |

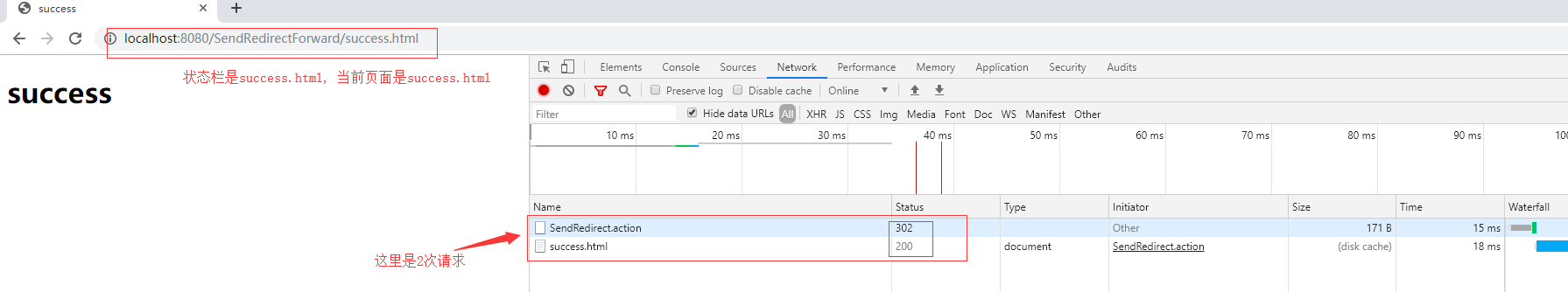
8.编写web.xml



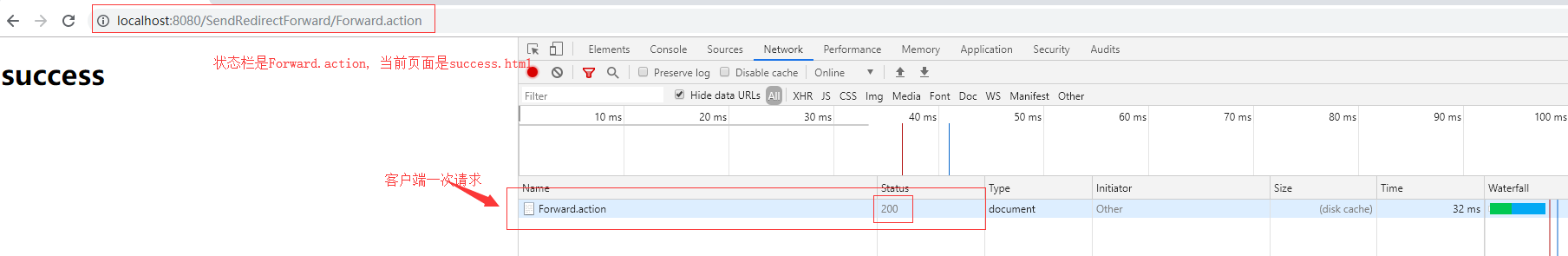
9.部署运行



1. 打开chrome的开发者选项, 点击重定向



11.打开chrome的开发者选项, 点击转发



二.重定向和转发

1.重定向

(1).重定向写法: 使用HttpServletResponse的sendRedirect方法, 参数是重定向的位, 即跳转的位置。response.sendRedirect("success.html")。

(2).地址栏上显示的是最后的那个资源的路径地址。

(3).请求次数最少有两次, 服务器在第一次请求后, 会返回302和一个地址, 浏览器在根据这个地址, 执行第二次访问。

(4).可以跳转到任意路径, 不是自己的工程也可以跳。

(5).效率稍微低一点, 执行两次请求。

(6).后续的请求, 没法使用上一次的request存储的数据, 或者没法使用上一次的request对象, 因为这是两次不同的请求。

2.转发

(1).请求转发的写法,参数即跳转的位置: request.getRequestDispatcher("success.html").forward(request, response);

(2).地址栏上显示的是请求servlet的地址, 返回200(ok)。

(3).请求次数只有一次, 因为是服务器内部帮客户端执行了后续的工作。

(4).只能跳转自己项目的资源路径 。

(5).效率上稍微高一点, 因为只执行一次请求。

(6).可以使用上一次的request对象。