

# 西南民族大学

## 实验报告

2018 -----2019 学年第 2 学期

课程名称：Web 应用开发（Java）

学 院：计算机科学与技术学院

年级：2016

班级：计科 1602

学号：201631102013

姓名：杜安娜

同组人：无

# 西南民族大学学生实验报告

教学单位：计科学院    实验室名称：计算机应用实验室    实验时间：2019 年 5 月 24 日

实验名称    Servlet 编程

实验编号    9

实验成绩

教师签名

教师评阅：

实验项目报告内容

## 1、实验目的

A. 熟悉 Servlet

## 2、实验内容

### 2.1. 参照练习

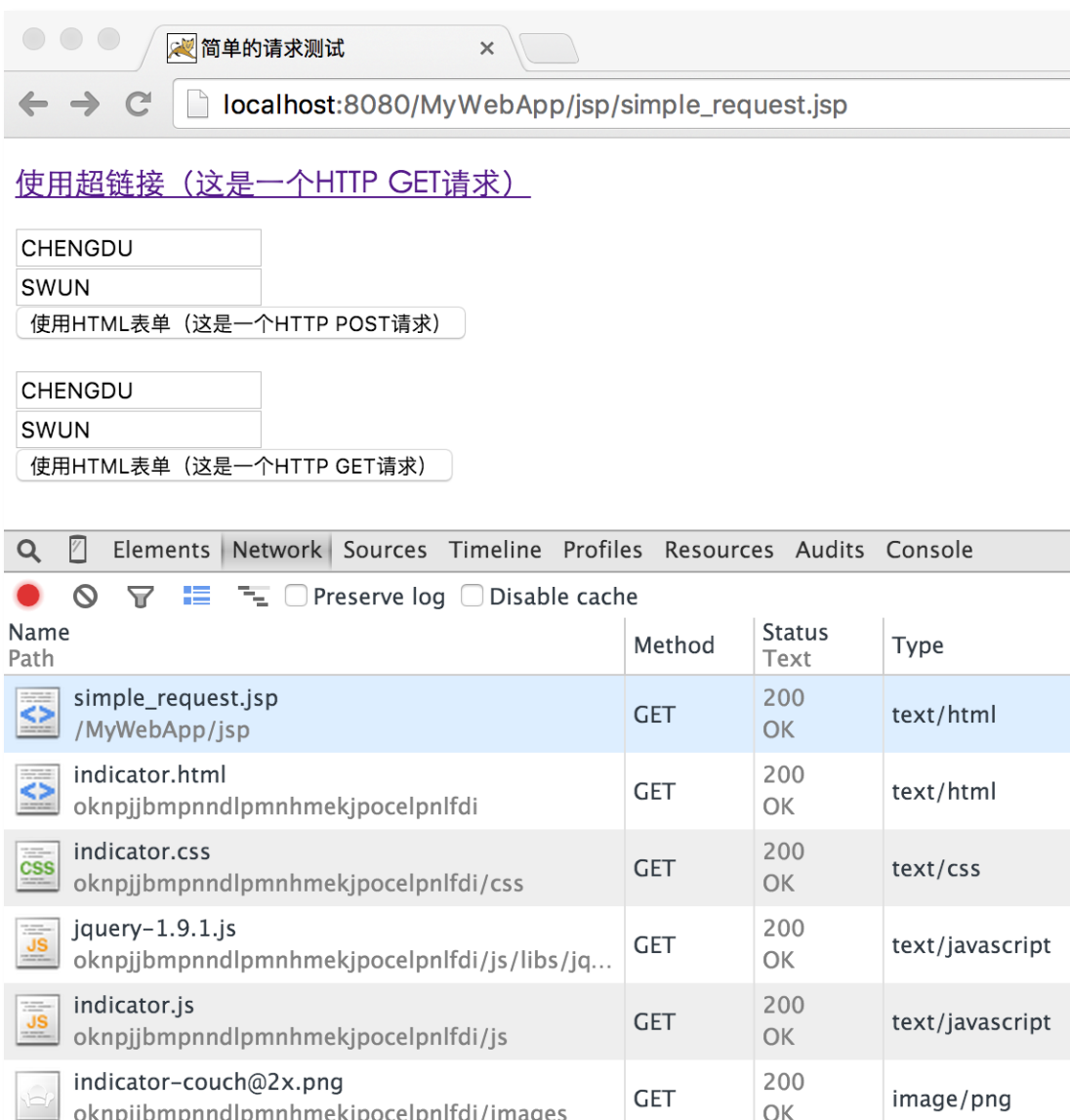
HttpServlet 可用于处理 HTTP 请求，下面将通过一个示例介绍如何使用 HttpServlet，本示例的具体步骤如下。新建一个 Dynamic Web Project，在 WebContent 文件夹下新建一个文件夹 jsp。

(1) 构建请求 JSP 页面 simple\_request.jsp:

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html" charset="UTF-8">
  <title>简单的请求测试</title>
</head>
<body>
  <p>
    <a href="<%=request.getContextPath() %>/http-request?param1=CHENGDU&param2=SWUN">
      使用超链接（这是一个HTTP GET请求）
    </a>
  </p>
  <p>
    <form action="<%=request.getContextPath() %>/http-request" method="post">
      <input type="text" name="param1" value="CHENGDU"><br>
      <input type="text" name="param2" value="SWUN"><br>
      <input type="submit" value="使用HTML表单（这是一个HTTP POST请求）"><br>
    </form>
  </p>
  <p>
    <form action="<%=request.getContextPath() %>/http-request" method="get">
      <input type="text" name="param1" value="CHENGDU"><br>
      <input type="text" name="param2" value="SWUN"><br>
      <input type="submit" value="使用HTML表单（这是一个HTTP GET请求）"><br>
    </form>
  </p>
</body>
</html>
```

使用本句代码获取Web应用的根路径。若直接在JSP中使用/，则直接获取到服务器的根路径

在浏览器访问该页面效果如下，若使用的 chrome 浏览器，可以打开“开发者工具”视图，帮助理解，如何打开，请自行搜索。是在打不开也不妨碍。



(2) 创建 Servlet 类 RequestServlet，类 RequestServlet 继承了 HttpServlet，并重写了 HttpServlet 的 doGet() 方法和 doPost() 方法，doGet() 方法用于处理 HTTP GET 方式的请求，doPost() 方法用于处理 HTTP POST 方式的请求类，RequestServlet 的代码如下：

```

package cx.course.jweb.servlet;

import java.io.IOException;

public class RequestServlet extends HttpServlet{
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {

        /*在Servlet中跳转到指定页面的方法如下，
        * 页面名可通过手动方式指定，如下面的/jsp/http_get.jsp*/
        req.getRequestDispatcher("/jsp/http_get.jsp").forward(req, resp);
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {

        /*在Servlet中跳转到指定页面的方法如下，
        * 页面名也可通过web.xml文件中的servlet参数指定【推荐】*/
        String postPage = this.getInitParameter("post-page");
        /*也可将某些内容打印至Console控制台（而不会打印到JSP页面），
        * 以供某些情况下的测试之用*/
        System.out.println(postPage);
        req.getRequestDispatcher(postPage).forward(req, resp);
    }
}

```

(3) 构建响应 JSP 页面，共两个，一个名为 http\_get.jsp，另一个名为 http\_post.jsp，分别用于响应 doGet() 和 doPost() 方法

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>doGet方法处理之后跳转的页面</title>
</head>
<body>
doGet方法处理之后跳转的页面
</body>
</html>

```

http\_get.jsp

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>doPost方法处理之后跳转的页面</title>
</head>
<body>
doPost方法处理之后跳转的页面
</body>
</html>

```

http\_post.jsp

(4) 在 web.xml 文件中配置类 RequestServlet，配置的关键代码如下：

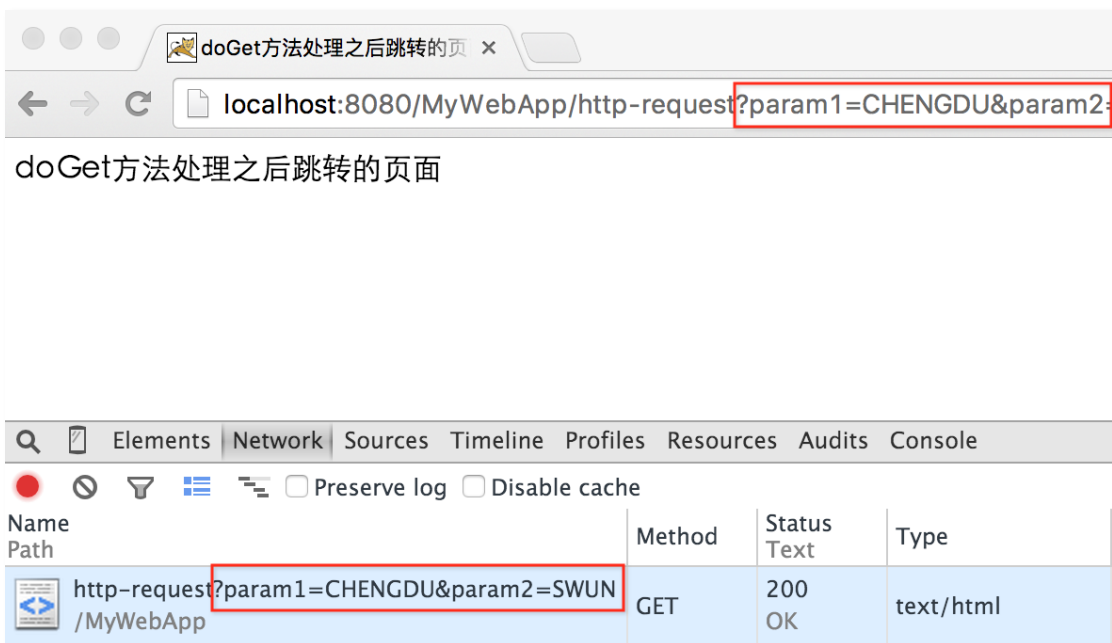
```

<servlet>
  <servlet-name>requestServlet</servlet-name>
  <servlet-class>cx.course.jweb.servlet.RequestServlet</servlet-class>
  <init-param>
    <param-name>post-page</param-name>
    <param-value>/jsp/http_post.jsp</param-value>
  </init-param>
</servlet>
<servlet-mapping>
  <servlet-name>requestServlet</servlet-name>
  <url-pattern>/another-url</url-pattern>
  <url-pattern>/http-request</url-pattern>
</servlet-mapping>

```







可以为Servlet添加初始化参数，并在Servlet类的代码中获取参数。这有利于配置和源代码的分离。其好处在于，不必修改源代码，即可改变系统的行为。

在第一步中的 simple\_request.jsp 中，点击其中的链接、表单按钮等，将会被 RequestServlet 处理，如下面几个图所示。特别注意：对于 HTTP GET 请求，页面参数会被置于 URL 的尾部；对于 HTTP POST 请求，页面参数不会出现在 URL 内，请注意区别，并根据实际需要选用不同的请求方式。



Q Elements Network Sources Timeline Profiles Resources Audits Console

● ○ 🔍 📄 ⚙️ ☐ Preserve log ☐ Disable cache

| Name<br>Path  | Headers   | Preview | Response | Cookies | Timing |
|---|---|---------|----------|---------|--------|
|  http-request?param1...<br>/MyWebApp             | <b>General</b><br>Remote Address: [::1]:8080<br>Request URL: http://localhost:8080/MyWebApp/http-request?<br>Request Method: GET<br>Status Code: 200 OK   |         |          |         |        |
|  indicator.html<br>oknpjjbmpnndlpmnh...          | <b>Response Headers</b> <a href="#">view source</a><br>Content-Length: 332<br>Content-Type: text/html; charset=UTF-8<br>Date: Thu, 19 Nov 2015 12:56:04 GMT<br>Server: Apache-Coyote/1.1  |         |          |         |        |
|  indicator.css<br>oknpjjbmpnndlpmnh...           | <b>Request Headers</b> <a href="#">view source</a><br>Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8<br>Accept-Encoding: gzip, deflate, sdch<br>Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.6,en;q=0.4<br>Connection: keep-alive<br>Cookie: JSESSIONID=3A0BB4D3BF6B0E8DF62E133F50631BEB<br>Host: localhost:8080<br>Referer: http://localhost:8080/MyWebApp/jsp/simple_request.do<br>User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_0; rv:53.0) Gecko/20100101 Firefox/53.0 |         |          |         |        |
|  jquery-1.9.1.js<br>oknpjjbmpnndlpmnh...         | <b>Query String Parameters</b> <a href="#">view source</a> <a href="#">view URL encoded</a><br>param1: CHENGDU<br>param2: SWUN  |         |          |         |        |
|  indicator.js<br>oknpjjbmpnndlpmnh...            |   |         |          |         |        |
|  indicator-couch@2x.p...<br>oknpjjbmpnndlpmnh... |   |         |          |         |        |


doPost方法处理之后跳转的页面

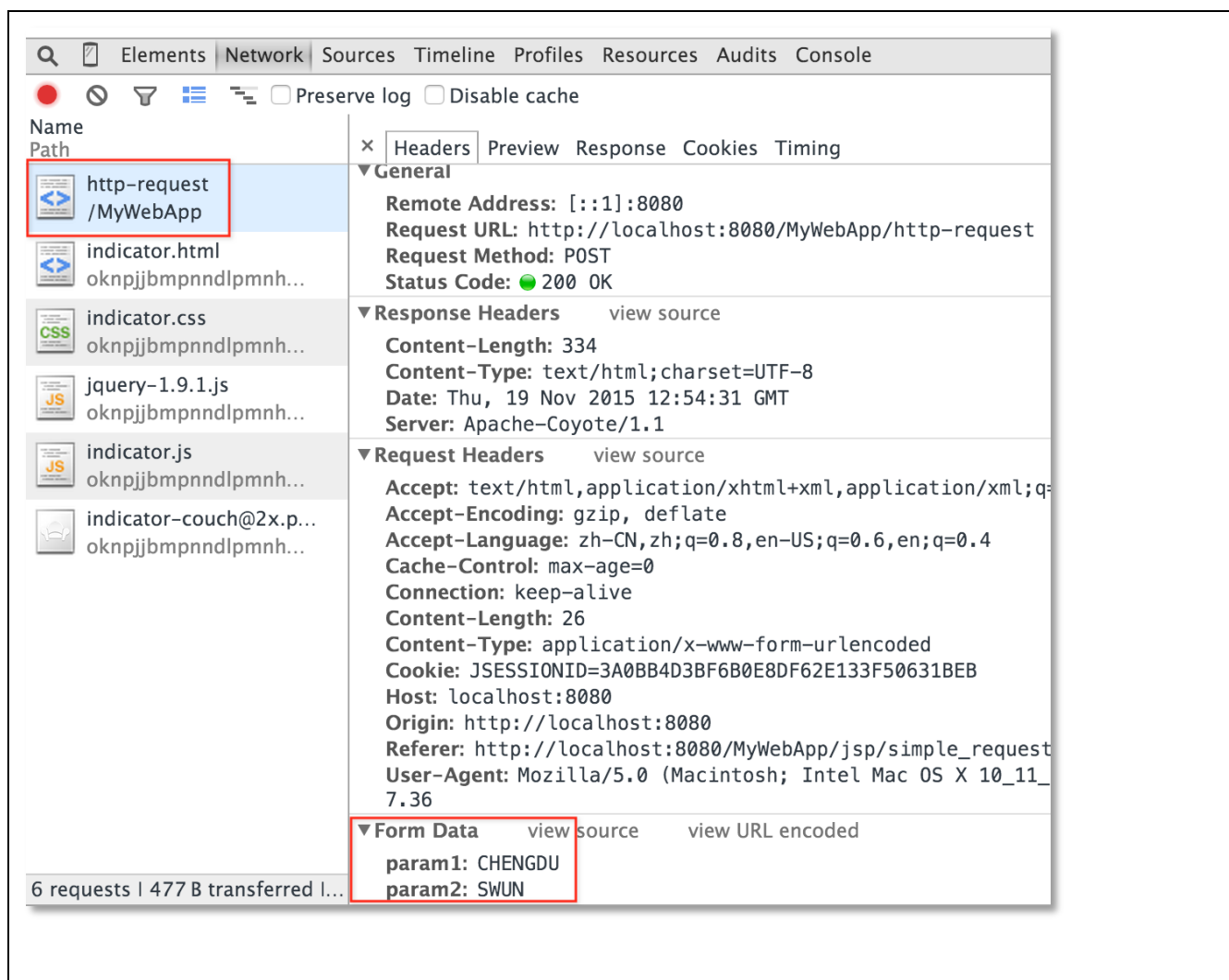
localhost:8080/MyWebApp/http-request

doPost方法处理之后跳转的页面

Q Elements Network Sources Timeline Profiles Resources Audits Console

● ○ 🔍 📄 ⚙️ ☐ Preserve log ☐ Disable cache

| Name<br>Path  | Method | Status<br>Text | Type      |
|---|--------|----------------|-----------|
|  http-request<br>/MyWebApp | POST   | 200<br>OK      | text/html |



## 2.2. 自主编程

新建一个Dynamic Web Project，名为MyStore，在WebContent编写一个页面order.jsp，里面包含两个链接，两个链接指向同一个URL，如/MyStore/buy，但是用于URL传值的查询字符串不同，如/MyStore/buy?product=book1，/MyStore/buy?product=book2。编写一个BuyServlet并配置好web.xml，用于响应/MyStore/buy请求，并在BuyServlet内将book1或book2存在session中。同时order.jsp中包含第三个链接，点击后，利用session显示购物车内容。请完成上述功能。

## 3、实验要求

## 4、实验主要过程与结果

### 4.1. 参照练习

HttpServlet 可用于处理 HTTP 请求，下面将通过一个示例介绍如何使用 HttpServlet，本示例的具体步骤如下。新建一个 Dynamic Web Project，在 WebContent 文件夹下新建一个文件夹 jsp。

(1) 构建请求 JSP 页面 simple\_request.jsp:

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
3 <html>
4 <head>
5   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6   <title>简单的请求测试</title>
7 </head>
8 <body>
9   <p>
10    <a href="<%=request.getContextPath()%>/http-request?param1=CHENGDU&param2=SWUN">
11      使用超链接（这是一个HTTP GET请求）
12    </a>
13  </p>
14  <p>
15    <form action="<%=request.getContextPath()%>/http-request" method="post">
16      <input type="text" name="param1" value="CHENGDU"><br>
17      <input type="text" name="param2" value="SWUN"><br>
18      <input type="submit" value="使用HTML表单（这是一个HTTP POST请求）"><br>
19    </form>
20  </p>
21  <p>
22    <form action="<%=request.getContextPath()%>/http-request" method="get">
23      <input type="text" name="param1" value="CHENGDU"><br>
24      <input type="text" name="param2" value="SWUN"><br>
25      <input type="submit" value="使用HTML表单（这是一个HTTP GET请求）"><br>
26    </form>
27  </p>
28 </body>
29 </html>
30

```

在浏览器访问该页面效果如下。

简单的请求测试

localhost:8080/myweb/jsp/simple\_r

[使用超链接（这是一个HTTP GET请求）](#)

CHENGDU

SWUN

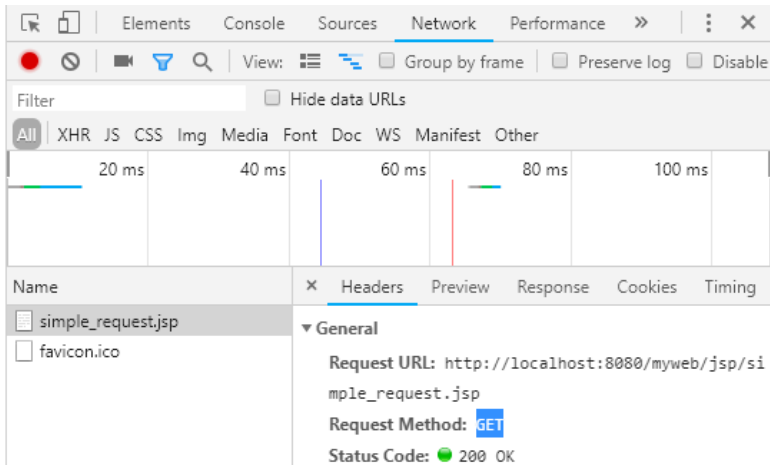
使用HTML表单（这是一个HTTP POST请求）

CHENGDU

SWUN

使用HTML表单（这是一个HTTP GET请求）





(2) 创建 Servlet 类 RequestServlet，类 RequestServlet 继承了 HttpServlet，并重写了 HttpServlet 的 doGet() 方法和 doPost() 方法，doGet() 方法用于处理 HTTP GET 方式的请求，doPost() 方法用于处理 HTTP POST 方式的请求类，RequestServlet 的代码如下：

```

1 package servlets;
2 import java.io.IOException;
3
4
5 public class RequestServlet extends HttpServlet{
6     protected void doGet(HttpServletRequest request, HttpServletResponse response)
7         throws ServletException, IOException{
8         request.getRequestDispatcher("/jsp/http_get.jsp").forward(request, response);
9     }
10    protected void doPost(HttpServletRequest request, HttpServletResponse response)
11        throws ServletException, IOException{
12        String postPage = this.getInitParameter("post-page");
13        System.out.println(postPage);
14        request.getRequestDispatcher(postPage).forward(request, response);
15    }
16 }

```

(3) 构建响应 JSP 页面，共两个，一个名为 http\_get.jsp，另一个名为 http\_post.jsp，分别用于响应 doGet() 和 doPost() 方法

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html
3 <html>
4 <head>
5     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6     <title>doGet方法处理之后跳转的页面</title>
7 </head>
8 <body>
9     doGet方法处理之后跳转的页面
10 </body>
11 </html>

```

```

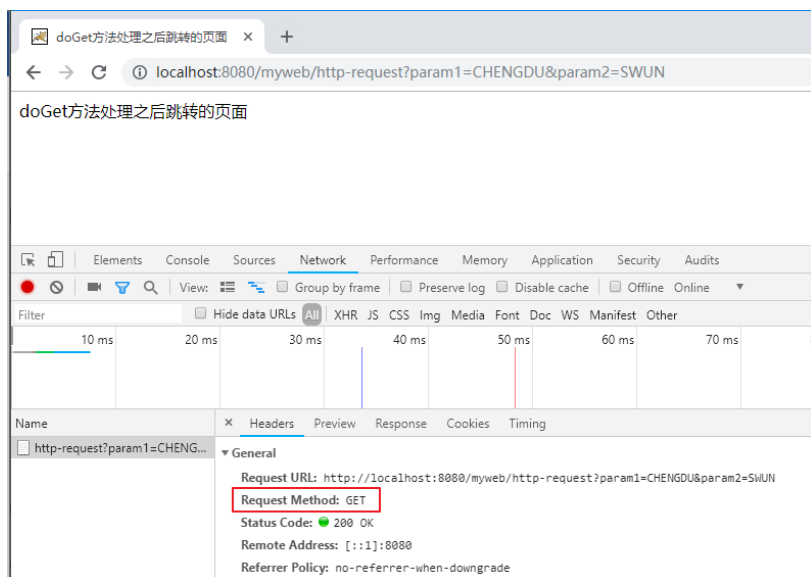
1 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html
3 <html>
4 <head>
5     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6     <title>doPost方法处理之后跳转的页面</title>
7 </head>
8 <body>
9     doPost方法处理之后跳转的页面
10 </body>
11 </html>

```

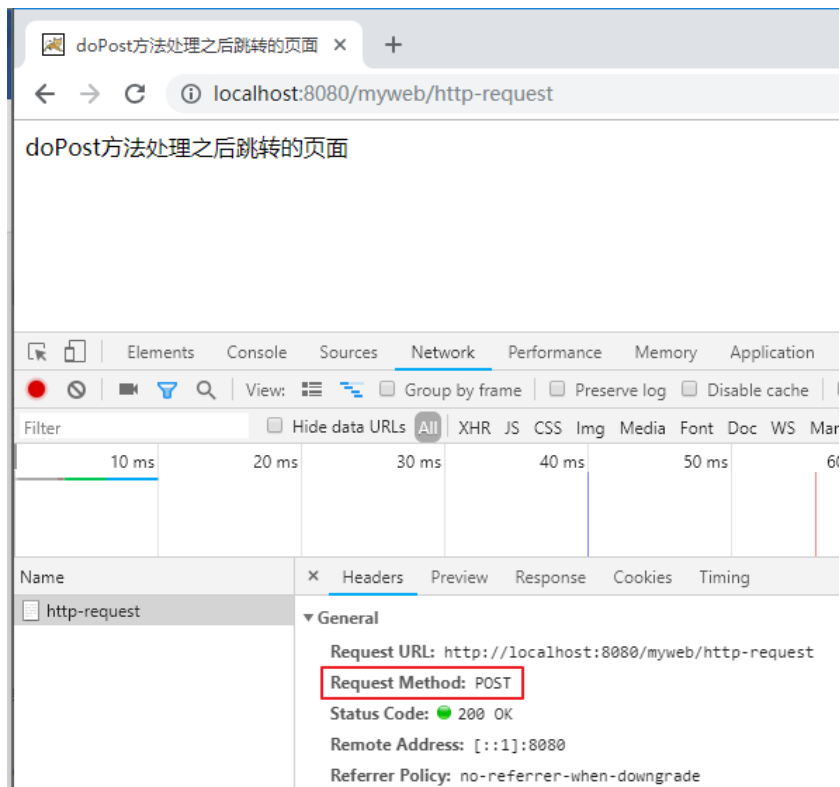
(4) 在 web.xml 文件中配置类 RequestServlet:

```
<servlet>
  <servlet-name>RequestServlet</servlet-name>
  <servlet-class>servlets.RequestServlet</servlet-class>
  <init-param>
    <param-name>post-page</param-name>
    <param-value>/jsp/http_post.jsp</param-value>
  </init-param>
</servlet>
<servlet-mapping>
  <servlet-name>RequestServlet</servlet-name>
  <url-pattern>/servlets/RequestServlet</url-pattern>
</servlet-mapping>
```

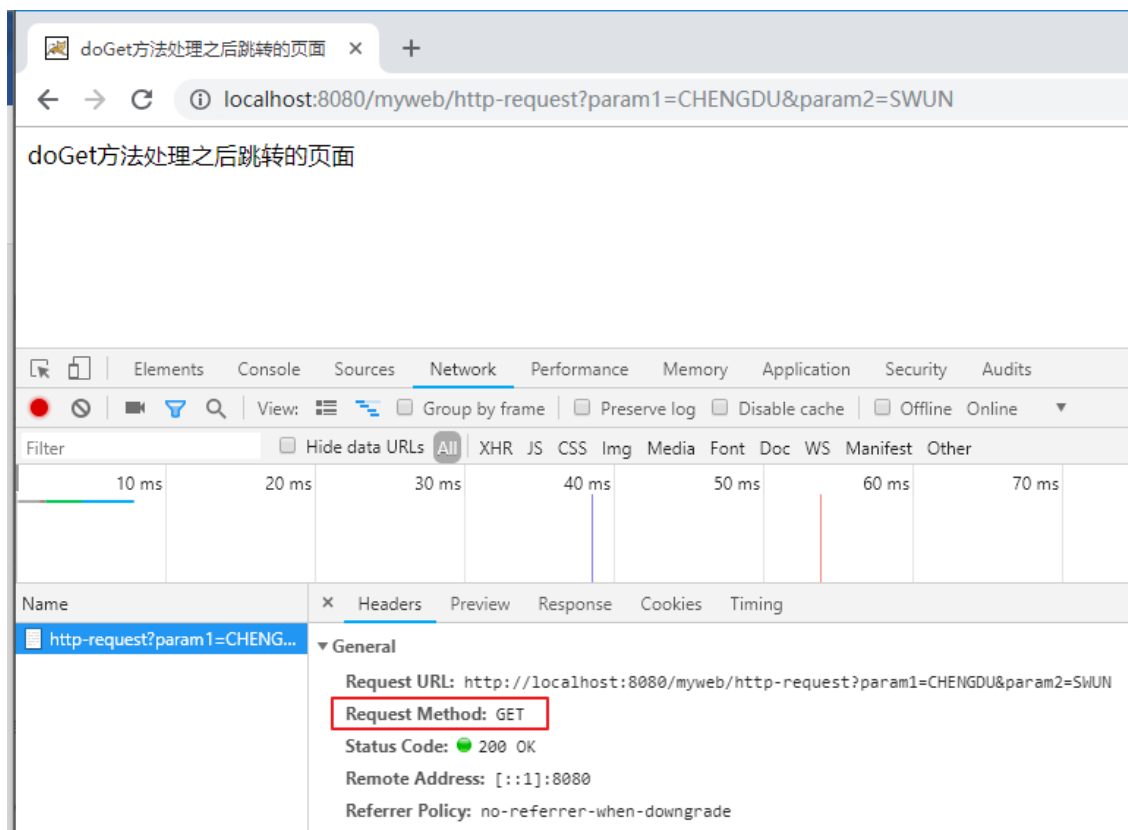
在第一步中的 simple\_request.jsp 中, 点击其中的链接、表单按钮等, 将会被 RequestServlet 处理。  
点击超链接后, 结果如下



点击第一个提交按钮后, 结果如下



点击第二个提交按钮后，结果如下



## 4.2. 自主编程

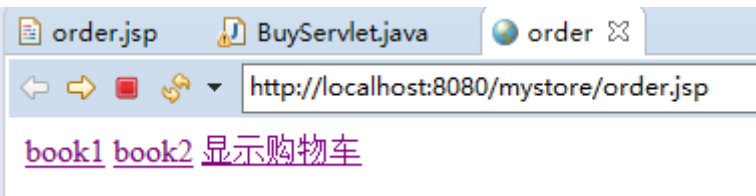
新建一个Dynamic Web Project，名为MyStore，在WebContent编写一个页面order.jsp，里面包含两个链接，两个链接指向同一个URL，如/MyStore/buy，但是用于URL传值的查询字符串不同，如/MyStore/buy?product=book1，/MyStore/buy?product=book2。编写一个BuyServlet并配置好web.xml，用于响应/MyStore/buy请求，并在BuyServlet内将book1或book2存在session中。同时order.jsp中包含第三个链接，点击后，利用session显示购物车内容。请完成上述功能。

### Order 页面

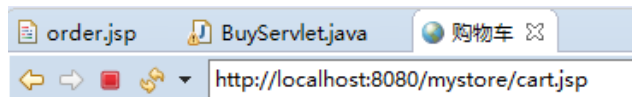
```
order.jsp BuyServlet.java 购物车
1 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding=
2 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w
3 <html>
4 <head>
5     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6     <title>order</title>
7 </head>
8 <body>
9     <a href="buy?product=book1">book1</a>
10    <a href="buy?product=book2">book2</a>
11    <a href="cart.jsp">显示购物车</a>
12 </body>
13 </html>
```

## BuyServlet

```
order.jsp BuyServlet.java 购物车
1 package servlets;
2
3 import java.io.IOException;
4
5 /**
6  * Servlet implementation class BuyServlet
7  */
8 @WebServlet("/BuyServlet")
9 public class BuyServlet extends HttpServlet {
10     private static final long serialVersionUID = 1L;
11
12     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
13         // TODO Auto-generated method stub
14         String product = request.getParameter("product");
15         HttpSession session = request.getSession();
16         ArrayList products = (ArrayList) session.getAttribute("products");
17         if(products == null) products = new ArrayList();
18         if(product != null) {
19             products.add(product);
20             session.setAttribute("products", products);
21         }
22     }
23
24     /**
25      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
26      */
27     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
28         // TODO Auto-generated method stub
29         doGet(request, response);
30     }
31 }
```



点击 book1、book2 后点击显示购物车



## 购物车中的内容是：

book1  
book2

## 5、简答题

注：实验报告的内容及格式可由学院根据学科专业特点确定；全校各专业必须使用学校统一封面。