

## 4、请求参数

笔记本： spring

创建时间： 2022/4/4 17:24

作者： 雷丰阳

---

入参处理：

```
package com.atguigu.controller;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletInputStream;
import javax.servlet.ServletOutputStream;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.CookieValue;
import org.springframework.web.bind.annotation.RequestHeader;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;

import com.atguigu.book.Book;

@Controller
public class HelloController {

    /**
     * request.getParameter("").....
     *
     * @return
     */
    @RequestMapping("/hello")
    public String handle01() {
        System.out.println("handle01...");
        return "success";
    }

    /**
     * SpringMVC如何获取请求带来的各种信息 默认方式获取请求参数： 直接给方法入参上
     * 写一个和请求参数名相同的变量。这个变量就来接收请求参数的值；
     * 带：有值，没带：null；
     *
     * @RequestParam：获取请求参数的；参数默认是必须带的；
     * @RequestParam("user")String username username =
     * request.getParameter("user")
     *
     *
     * @RequestParam("user")
     * @PathVariable("user")
     *
     * /book/ 【{user}pathvariable】?
     * 【user=admin(requestparam)
     *
     *
     *
     * value:指定要获取的参数的key required:这个参数是否
     必须的
     *
     * defaultValue:默认值。没带默认是null；
     *
     *
     * @RequestHeader：获取请求头中某个key的值； request.getHeader("User-
     Agent");
```

```

        * @RequestHeader("User-Agent")String userAgent userAgent =
        * request.getHeader("User-Agent")
        * 如果请求头中没有这个值就会报错;
value() required()
        *
        * defaultValue()
        *
        * @CookieValue: 获取某个cookie的值; 以前的操作获取某个cookie; Cookie[]
cookies =
        * request.getCookies(); for(Cookie
c:cookies){
        * if(c.getName().equals("JSESSIONID")){
String
        * cv = c.getValue(); } }
        * value()
        * required()
        * defaultValue()
        */
@RequestMapping("/handle01")
public String handle02(
        @RequestParam(value = "user", required = false, defaultValue =
"你没带") String username,
        @RequestHeader(value = "AHAHA", required = false, defaultValue =
"她也没带") String userAgent,
        @CookieValue(value="JSESSIONID",required=false)String jid) {
    System.out.println("这个变量的值: " + username);
    System.out.println("请求头中浏览器的信息: " + userAgent);
    System.out.println("cookie中的jid的值"+jid);
    return "success";
}

/**
 * 如果我们的请求参数是一个POJO;
 * SpringMVC会自动的为这个POJO进行赋值?
 * 1)、将POJO中的每一个属性,从request参数中尝试获取出来,并封装即可;
 * 2)、还可以级联封装;属性的属性
 * 3)、请求参数的参数名和对象中的属性名一一对应就行
 *
 *
 * 提交的数据可能有乱码:
 * 请求乱码:
 * GET请求: 改server.xml: 在8080端口处URIEncoding="UTF-8"
 * POST请求:
 * 在第一次获取请求参数之前设置
 * request.setCharacterEncoding("UTF-8");
 * 自己写一个filter; SpringMVC有这个filter
 *
 * 响应乱码:
 * response.setContentType("text/html;charset=utf-8")
 * @param book
 * @return
 */
@RequestMapping("/book")
public String addBook(Book book){
    System.out.println("我要保存的图书: "+book);
    return "success";
}

/**
 * SpringMVC可以直接在参数上写原生API;
 *
 * HttpServletRequest
 * HttpServletResponse
 * HttpSession
 *
 *
 * java.security.Principal
 * Locale: 国际化有关的区域信息对象
 * InputStream:
 * ServletInputStream inputStream = request.getInputStream();
 * OutputStream:
 * ServletOutputStream outputStream =
response.getOutputStream();
 * Reader:
 * BufferedReader reader = request.getReader();

```

```
    * Writer:
    *      PrintWriter writer = response.getWriter();
    *
    * @throws IOException
    *
    *
    */
    @RequestMapping("/handle03")
    public String handle03(HttpSession session,
        HttpServletRequest request, HttpServletResponse response) throws
        IOException {
        request.setAttribute("reqParam", "我是请求域中的");
        session.setAttribute("sessionParam", "额我是Session域中的");

        return "success";
    }
}
```