**每日作业卷**

**javaweb第37天request**

# 关卡1

## 训练案例 打印客户端IP和浏览器版本信息

### 训练考核知识点

1\_利用request获取客户端IP信息

2\_利用request获取浏览器和操作系统平台信息

### 训练描述

当我们访问网络上的某些网速测试平台时,用浏览器访问到其网站时,对方就可以检测到我们

采用的浏览器版本,所用的操作系统平台.如下图:



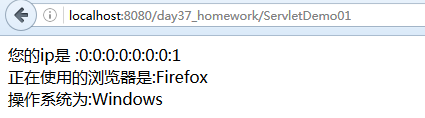
服务端如何识别客户端的IP和操作系统等平台信息呢?

request中有对应的API可以获取到客户端的ip,

向服务端发起请求是,有消息头 user-agent可以将客户端信息发送到服务端.

在本次关卡中,当我们对服务端ServletDemo01发起请求时,服务端向客户端响应客户端的IP

和操作系统等信息



提示:

1\_在本案例中,我们利用工具类UserAgent来出现客户端的信息

public class UserAgent {

private String browserType;//浏览器类型

private String browserVersion;//浏览器版本

private String platformType;//平台类型

private String platformSeries;//平台系列

private String platformVersion;//平台版本

//getter.. and setter...

}

2\_为了方便判断准确各种不同的浏览器版本和系统所处平台信息. 我们抽取了 UesrAgentUtil工具类. 这个工具类大家只需看懂即可.

3\_在工具类UserAgentUtils中用到了StringUtils工具类,这个工具类的使用方式参见

StringUtils的帮助文档

### 操作步骤答案描述

1\_在项目day37\_homework下新建ServletDemo01

2\_向浏览器响应客户端的IP信息

response.setContentType("text/html;charset=utf-8");

response.getWriter().print("您的ip是 :"+request.getRemoteAddr()+"<br/>");

3\_为了描述客户端的浏览器以及平台等信息,我们设计一个类UserAgent,代表客户端

的平台信息以及浏览器信息,内容如下:

public class UserAgent {

private String browserType;//浏览器类型

private String browserVersion;//浏览器版本

private String platformType;//平台类型

private String platformSeries;//平台系列

private String platformVersion;//平台版本

//getter.. and setter...

}

4\_利用在客户端向服务端发起请求的过程中,有一个User-Agent的消息头对应的内容中会

存放本次客户端请求所采用的浏览器以及操作系统平台信息,我们可以利用工具

类UserAgentUtil获取用户的浏览器以及系统的平台信息.

代码如下:

UserAgent userAgent = UserAgentUtil.getUserAgent(request.getHeader("User-Agent"));

response.getWriter().print("正在使用的浏览器是:"+userAgent.getBrowserType()+"<br/>");

response.getWriter().print("操作系统为:"+userAgent.getPlatformType()+"<br/>");

5\_UserAgentUtil的工具类的详细代参考UserAgent.java文件(这个文件只需大家看懂即可)

在UserAgentUtils的工具类中,我们用到了StringUtils工具类,StringUtils工具类是

Apache-commons系列的的工具类,响应的使用参看文档参见有关说明文档.

**最终答案**

## 训练案例1 打印本次请求中所有的HTTP请求头信息

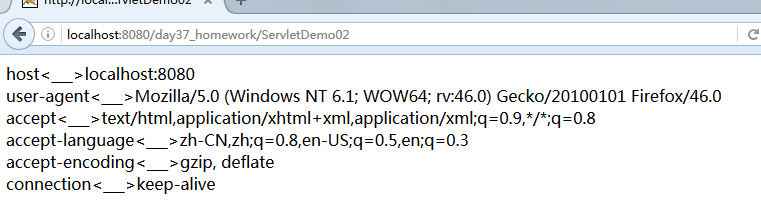
### 训练考核知识点

1\_request获取请求消息头

2\_request获取消息头对应的内容

### 训练描述

打印本次请求中所有的HTTP请求头信息,效果如下:



### 操作步骤描述

1\_在项目day38\_homework下建立ServletDemo02.

2\_设置本次响应内容类型

response.setContentType("text/html;charset=utf-8");

3\_获取到本次请求所有的消息头

Enumeration<String> em = request.getHeaderNames();

4\_通过遍历本次请求的所有消息头获取到对应所有消息头对应的内容

while(em.hasMoreElements()){

String name = em.nextElement();

String value = request.getHeader(name);

response.getWriter().print(name+"<\_\_\_>"+value+"<br/>");

}

**5\_最终答案**

# 关卡2 录入一名学生信息

### 训练考核知识点

1\_利用request接收表单数据

### 训练描述

1\_页面部分实现一个效果如下的表单



2\_在ServletDemo02中接收表单数据

3\_将表单数据封装在stu对象上

4\_调用业务层将stu携带的数据保存到DB中

5\_向客户端响应录入信息成功,响应可以跳转到查看全部学生信息的链接

效果如下:



6\_实现service层代码

7\_实现DAO层代码

### 操作步骤描述

1\_在页面demo01.html中实现一个表单

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

<script src="/day37\_homework/jedate/jedate.js"></script>

</head>

<body>

<h1 align="center">录入学生信息</h1>

<form action="/day37\_homework/ServletDemo03" method="post">

<table border="1" style="width:350px;margin:10px auto;border:1px solid #ddd">

<tr>

<td>姓名</td><td><input type="text" name="stuName"/></td>

</tr>

<tr>

<td>性别</td>

<td>

<input type="radio" name="stuSex" value="男" checked="checked"/>男

<input type="radio" name="stuSex" value="女"/>女

</td>

</tr>

<tr>

<td>年龄</td>

<td align="left">

<select name="stuAge" style="width:40px;margin-left:10px">

<option value="18">18</option>

<option value="19">19</option>

<option value="20">20</option>

<option value="21">21</option>

<option value="22">22</option>

<option value="23">23</option>

<option value="24">24</option>

<option value="25" selected="selected">25</option>

<option value="26">26</option>

<option value="27">27</option>

<option value="28">28</option>

<option value="29">29</option>

<option value="30">30</option>

<option value="31">31</option>

</select>

</td>

</tr>

<tr>

<td>生日(yyyy-MM-dd)</td>

<td><input type="text" name="stuBirthday"/></td>

<tr>

<td>电话</td>

<td><input type="text" name="stuPhone"/></td>

</tr>

<tr>

<td>爱好</td>

<td>

<input type="checkbox" name="stuHobby" value="sports"/>运动

<input type="checkbox" name="stuHobby" value="game"/>游戏

<input type="checkbox" name="stuHobby" value="reading"/>阅读

</td>

</tr>

<tr>

<td colspan="2" align="center">

<input type="submit" value="提交"/>

</td>

</tr>

</table>

</form>

</body>

</html>

2\_创建仓库day37\_homework,创建表stu

CREATE DATABASE /\*!32312 IF NOT EXISTS\*/`day37\_homework` /\*!40100 DEFAULT CHARACTER SET utf8 \*/;

USE `day37\_homework`;

/\*Table structure for table `stu` \*/

DROP TABLE IF EXISTS `stu`;

CREATE TABLE `stu` (

`stuId` int(11) NOT NULL AUTO\_INCREMENT,

`stuName` varchar(32) DEFAULT NULL,

`stuAge` int(11) DEFAULT NULL,

`stuSex` varchar(32) DEFAULT NULL,

`stuBirthday` date DEFAULT NULL,

`stuPhone` varchar(32) DEFAULT NULL,

`sports` varchar(10) DEFAULT NULL,

`game` varchar(10) DEFAULT NULL,

`reading` varchar(10) DEFAULT NULL,

PRIMARY KEY (`stuId`)

) ENGINE=InnoDB AUTO\_INCREMENT=10 DEFAULT CHARSET=utf8;

2\_实现ServletDemo02代码:略

response.getWriter().print("录入学生成功,<a href='#'>查看所有学生信息</a>");

3\_实现StuService中代码:略

public void saveStu(Stu stu) throws SQLException{

StuDao stuDao=new StuDao();

stuDao.addStu(stu);

}

4\_完成StuDao代码:略

public void addStu(Stu stu) throws SQLException{

String sql="INSERT INTO stu VALUES(NULL,?,?,?,?,?,?,?,?)";

//stuName stuAge stuBirthday stuPhone sports game reading

Object[] params={stu.getStuName(),stu.getStuAge(),stu.getStuSex(),stu.getStuBirthday(),stu.getStuPhone(),stu.getSports(),stu.getGame(),stu.getReading()};

QueryRunner qr=new QueryRunner(JDBCUtils.getDataSource());

qr.update(sql,params);

}

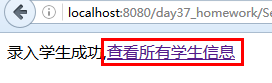
# 关卡3 请求转发实现查询全部学生

### 训练考核知识点

利用request实现请求转发

### 训练描述

在关卡2的案例中我们向客户端输出了一个”查看全部学生”的链接.



1\_点击查看所有学生信息,向服务端的ServletDemo04发起请求,我们要设置这个链接的路径

String str=”<a href='/day37\_homework/ServletDemo04'>录入学生信息</a><br/>”;

response.getWriter().println(str);

2\_ServletDemo04中调用service层查询全部学生信息,并将返回的所有学生放入request

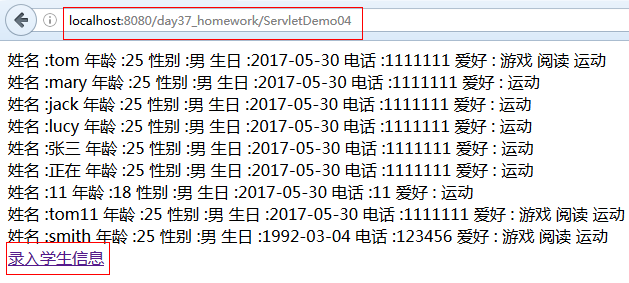
域对象,转发到ServletDemo05

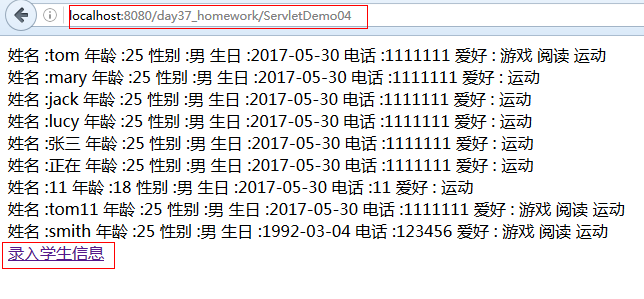
3\_ServletDemo05中获取到request中的所有学生,遍历并输出学生的所有信息到浏览器

4\_在ServletDemo05中向客户端响应一个文本信息为”录入学生信息的”链接,点击这个链接

可以跳转到WebContent下的demo01.html

.





### 操作步骤描述

1\_新建ServletDemo04

try {

StuService StuService = new StuService();

List<Stu> list = StuService.getAllStu();

request.setAttribute("allStu", list);

request.getRequestDispatcher("/ServletDemo05").forward(request, response);

} catch (Exception e) {

e.printStackTrace();

}

2\_建立ServletDemo05:略