

第九天

Request

代表HTTP请求的对象

继承结构

ServletRequest --- request请求的最高级别的接口

HttpServletRequest 在原有的接口之上实现了和HTTP协议相关的方法。这个接口更善于HTTP相关开

HttpServletRequest的API操作：

功能一：获取浏览器相关的信息

getContextPath -- 获得当前web应用虚拟目录名称 -- 在写路径时不要将web应用的虚拟路径的名称写死, 应该在需要写web应用的名称的地方通过getContextPath方法动态获取

getRequestURL方法 -- 返回客户端发出请求完整URL

getRequestURI方法 -- 返回请求行中的资源名部分

getQueryString方法 -- 返回请求行中的参数部分

getRemoteAddr方法 -- 返回发出请求的客户机的IP地址

getRemoteAddr方法 -- 返回发出请求的客户机的IP地址

getMethod -- 得到客户机请求方式

功能二：获取请求头信息

getHeader(name)方法 --- String

getHeaders(String name)方法 --- Enumeration

getHeaderNames方法 --- Enumeration

getIntHeader(name)方法 --- int

getDateHeader(name)方法 --- long(日期对应毫秒)

功能三：获取请求参数

getParameter(String name) --- String 通过name获得值

getParameterValues(String name) --- String[] 通过name获得多值 checkbox

getParameterMap() --- Map key :name value: 多值 将查询的参数保存在一个Map中

getParameterNames() --- Enumeration 获得所有name

乱码的解决

请求参数乱码

```
//通知服务器使用utf-8接收数据
request.setCharacterEncoding("utf-8");
```

get提交乱码处理

```
String username = request.getParameter("username");//? ? ? ? ?
//重新编码：使用字符集iso8859-1字符集
byte[] bytes = username.getBytes("iso8859-1");
//再次解码：使用utf-8字符集
username = new String(bytes,"utf-8");
System.out.println("username:"+username);
```

request功能四---请求转发

一次请求一次响应

代码实现

```
//1.创建一个调度器
RequestDispatcher rd = request.getRequestDispatcher("/servlet/RequestDemo5");
//2.利用调度器实现请求转发
rd.forward(request, response);
```

功能五---作为域对象

域对象：如果一个对象有一个可以被看见的范围，在这个范围内利用对象身上的map实现数据的共享，这样一个对象就称之为域对象。

setAttribute(String name,Object value) ---- 设置域属性。

getAttribute(String name); --- 根据域属性获取域属性的值

removeAttribute(String name) --- 删除域属性

getAttributeNames() ---- 获取全部域属性的名称

RequestDemo7

```
package cn.tedu.request;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
//作为域对象使用
public class RequestDemo7 extends HttpServlet {
    throws ServletException, IOException {
        //在同一个请求转发的范围内，共享数据
        //先向域中添加数据
        request.setAttribute("name","曹洋");
        request.setAttribute("age",18);
```

```

request.setAttribute("addr", "安徽");
request.setAttribute("height", 180);
//再完成请求转发
request.getRequestDispatcher("/index.jsp").forward(request, response);
public void doGet(HttpServletRequest request, HttpServletResponse response)
{
    // request.getRequestDispatcher("/servlet/RequestDemo8").forward(request,
    response);
    throws ServletException, IOException {
        doGet(request, response);
        public void doPost(HttpServletRequest request, HttpServletResponse response)
        {

```

index.jsp

```

<%@ page language="java" import="java.util.*" pageEncoding="UTF-8"%>
<!DOCTYPE HTML >
<html>
<head>

</head>

<body>
姓名: <%=request.getAttribute("name") %><br/>
芳龄: <%=request.getAttribute("age") %><br/>
地址: <%=request.getAttribute("addr") %><br/>
身高: <%=request.getAttribute("height") %><br/>
</body>
</html>

```

请求包含

```

package cn.tedu.request;
import java.io.IOException;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
//请求包含 和RequestDemo10一组
public class RequestDemo9 extends HttpServlet {
    throws ServletException, IOException {
        //1. 创建一个调度器
        RequestDispatcher rd = request.getRequestDispatcher("/servlet/RequestDemo10");
        response.getWriter().write("RequestDemo9.....");
        //2. 利用调度器实现请求包含
        rd.include(request, response);
        public void doGet(HttpServletRequest request, HttpServletResponse response)
        {
            throws ServletException, IOException {
                doGet(request, response);
                public void doPost(HttpServletRequest request, HttpServletResponse response)
                {

```

```
}
```

Response

代表HTTP响应的对象。

ServletResponse --- response的最高级接口

HttpServletResponse --- 包含和HTTP协议相关的API，善于HTTP相关开发

HttpServletResponse的API

设置状态码的方法

void setStatus(int sc)

void setStatus(int sc, String sm)

设置响应头的方法

void setHeader(String name, String value)

void setDateHeader(String name, long date)

void setIntHeader(String name, int value)

void addHeader(String name, String value)

void addDateHeader(String name, long date)

void addIntHeader(String name, int value)

设置响应内容的方法

ServletOutputStream getOutputStream() PrintWriter getWriter()

response功能一---向浏览器发送数据

response字节流和字符流

字节流

response.getOutputStream().write("abc".getBytes());

字符流

response.getWriter().write("中国");

通知浏览器接收数据:

response.setHeader("Content-Type","text/html;charset=utf-8");

简写:

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

通知服务器发送数据:

```
response.setCharacterEncoding("utf-8");
```

response功能二---请求重定向

第一种:

```
response.setStatus(302);
response.setHeader("location", "http://www.tmooc.cn");
response.setHeader("location", "/Day07-servlet/servlet/RequestDemo6");
```

第二种:

```
response.sendRedirect("http://www.baidu.com")
```

response功能三--定时刷新

```
//乱码处理
response.setContentType("text/html;charset=utf-8");
//向缓冲区添加内容
response.getWriter().write("<h1 align='center'><font color='red'>恭喜注册成功, 3秒
之后跳转到首页! </font>
</h1>");
//定时刷新实现原理: 设置refresh响应头
response.setHeader("refresh", "3;url=http://www.tmooc.cn");
```

response功能四---操作浏览器缓存

```
package cn.tedu.response;

import java.io.IOException;
import java.util.Date;

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

public class ResponseDemo6 extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        //控制浏览器不要缓存资源
        //response.setDateHeader("Expires", -1); //1970-1-1
        //response.setHeader("Cache-control", "no-cache");

        //控制浏览器缓存一个资源
        response.setDateHeader("Expires",
            System.currentTimeMillis()+1000*60*60*24); //1970-1-1
    }
}
```

```
        response.setHeader("Cache-control", "max-age=5");
        response.getWriter().write("hello servlet..." + new
Date().toLocaleString());
    }

    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        doGet(request, response);
    }
}
```