

课程回顾

1 HTTP协议

```
1 概念: HTTP用在浏览器和服务器之间的一个约定。
2 作用: 约束浏览器和服务器交互的数据格式
3 浏览器遵循请求协议:
4 post请求协议:
5 一个请求首行      post /web项目名称/资源名称  HTTP1.1
6 多个请求头        遵循key: value  user-agent  accept accet-language
7 一个空白行
8 一个请求体        form提交给后台的数据: name属性值: 用户输入或选择值
9
10 get请求协议:
11 一个请求首行      get /web项目名称/资源名称  HTTP1.1
12 多个请求头        遵循key: value  user-agent  accept accet-language
13 一个空白行
14 GET没有请求体
15
16
17 服务器遵循响应协议:
18 响应协议
19 一个响应首行      HTTP1.1  200  ok   响应码 (200 404-not found 307--重定向 500--
-服务器错误)
20 多个响应头        遵循key: value  host  time content-type content-length
21 一个空白行
22 响应体            html
```

```
3 浏览器遵循请求协议:
4 post请求协议:
5 一个请求首行      post /web项目名称/资源名称  HTTP1.1
6 多个请求头        遵循key: value  user-agent  accept accet-language
7 一个空白行
8 一个请求体        form提交给后台的数据: name属性值: 用户输入或选择值
9
10 get请求协议:
11 一个请求首行      get /web项目名称/资源名称  HTTP1.1
12 多个请求头        遵循key: value  user-agent  accept accet-language
13 一个空白行
14 GET没有请求体
```

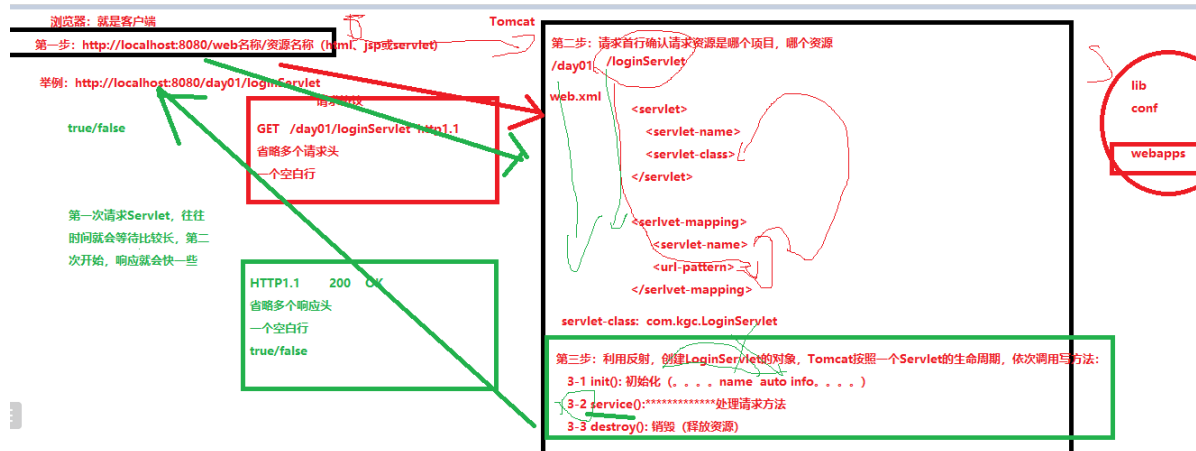


```
16
17 服务器遵循响应协议:
18 响应协议
19 一个响应首行      HTTP1.1  200  ok   响应码 (200 404-not found 307--重定向 500---服务器错误)
20 多个响应头        遵循key: value  host  time content-type content-length
21 一个空白行
22 响应体            html
```



2 Servlet生命周期

- 1 1. Tomcat负责创建Servlet对象, 调用init(), 初始化Servlet
- 2 2. Tomcat调用service()执行处理请求代码, 响应浏览器
- 3
- 4
- 5
- 6 3. Tomcat关闭之前, 调用destroy()销毁所有servlet, 释放资源



3 Servlet使用步骤 *****

特点: Servlet由Tomcat服务器根据用户的请求, 来自主完成对象的创建

- 1 1. 创建一个类xxxServlet, 实现Servlet接口
- 2 重写接口所有的方法:
- 3 init()
- 4 service()
- 5 destroy()
- 6 2. WEB-INF里面web.xml配置实现类
- 7 <servlet>
- 8 <servlet-name>随便写</servlet-name>
- 9 <servlet-class></servlet-class>
- 10 </servlet>
- 11 <servlet-mapping>
- 12 <servlet-name>servlet配置servlet-name一样</servlet-name>
- 13 <!-- url-pattern浏览器请求servlet时使用的url地址 -->
- 14 <url-pattern>/路径名, 随便写, 建议与Servlet类名一样的</url-pattern>
- 15 </servlet-mapping>
- 16
- 17 3. form的action设置为配置Servlet的url-pattern
- 18 3-1 打开浏览器, 浏览器地址栏: `http://服务器ip地址:端口/web项目名/url-pattern配置`

4 idea自带模板实现Servlet创建

- 1 1. 创建一个类, 继承javax提供一个HttpServlet
- 2 doGet(): 处理get请求
- 3 doPost(): 处理Post请求
- 4
- 5
- 6 HttpServlet特点: 已经实现了Servlet接口, 程序里面因为只关注请求的处理, 不再关注init
- 7 destroy()

课程目标

1 请求协议对象HttpServletRequest

2 响应协议对象HttpServletResponse

3 总结javaWeb书写路径的套路

课程实施

1 HttpServletRequest

1-1 HttpServletRequest概念

1 | HttpServletRequest: 一个请求协议对象

1-2 HttpServletRequest作用

1 | HttpServletRequest封装哪些方法?
2 | 1. 获取请求首行信息
3 | getHeader()
4 | 2. 获取请求头
5 |
6 | 3. 获取请求体
7 |
8 |

1-2-1 如何通过request对象获取表单提交的数据

.42\webapps\manager

username=lucy getParameter(username)

pwd: [123456] getParameterMap()

username: [lucy]

hobby: [篮球2, 篮球3, 篮球4] form提交的数据量特别多情

pwd:123456

username:lucy getParameterNames()

hobby: 篮球2

[篮球2, 篮球3, 篮球4] getParameterValues(hobby)

| | | |
|-------------|---|---|
| | getLocalPort() | Returns the host name of the Internet Protocol (IP) interface on which the request was received. |
| int | getLocalPort() | Returns the Internet Protocol (IP) port number of the interface on which the request was received. |
| String | getParameter(String name) | 获取表单一个name对应的值 Returns the value of a request parameter as a String, or null if the parameter does not exist. |
| Map | getParameterMap() | 获取一个表单所有的提交数据: key=value Returns a java.util.Map of the parameters of this request. |
| Enumeration | getParameterNames() | 获取所有的表单项的name属性值集合 Returns an Enumeration of String objects containing the names of the parameters contained in this request. |
| String[] | getParameterValues(String name) | 获取一个name对应的所有的value: 复选框 Returns an array of String objects containing all of the values the given request parameter has, or null if the parameter does not exist. |
| String | getProtocol() | Returns the name and version of the protocol the request uses in the form <i>protocol/majorVersion.minorVersion</i> , for example, HTTP/1.1. |

1-3 解决request获取中文乱码问题?

1-3-1 POST方式

- 1 POST提交数据，走post请求协议，提交的数据都在请求体，
- 2 因此解决中文乱码，设置请求体的编码格式就可以，GET不适用

```
void setCharacterEncoding(String env) // 设置请求体的编码格式  
    Overrides the name of the character encoding used in the body of this request.
```

1-3-2 GET方式

- 1 Tomcat默认的编码格式配置：URIEncoding除了TOMCAT8.0,其他的TomcatURIEncoding的编码格式ISO8859-1
- 2 TOMCAT8.0默认配置URIEncoding都是UTF-8
- 3
- 4 URL:地址
- 5 URI: 是url的父
- 6 //这个代码针对get乱码使用，如果get没有乱码，就不要写这行代码
- 7 String username=new String(username.getBytes("iso8859-1"),"utf-8");

1-4 课堂案例

```
1 package cn.kgc.servlet;  
2  
3 import javax.servlet.ServletException;  
4 import javax.servlet.http.HttpServlet;  
5 import javax.servlet.http.HttpServletRequest;  
6 import javax.servlet.http.HttpServletResponse;  
7 import java.io.IOException;  
8  
9 /**  
10  * @Author: lc  
11  * @Date: 2022/5/16  
12  * @Description: 建议：一个Servlet中处理一个请求  
13  * LoginServlet处理登录，通常doGet或doPost其中一个方法中写代码即可  
14  * @Version: 1.0  
15  */  
16 public class LoginServlet extends HttpServlet {  
17     //专门处理get请求协议  
18     @Override  
19     protected void doGet(HttpServletRequest request, HttpServletResponse  
20 resp) throws ServletException, IOException {  
21         //设置请求体的编码格式，POST请求提交的数据解决中文乱码  
22         request.setCharacterEncoding("UTF-8");//请求体的编码格式，只要支持中文即可  
23         //登录  
24         //1. 获取数据  
25         //数据从什么地方提交给Tomcat服务器?? 请求协议  
26         //掌握单个key-value获取  
27         String username = request.getParameter("username");//获取username表单  
28         //项用户输入的值  
29         //username获取?????  
30         /*byte[] bytes = username.getBytes("iso8859-1");
```

```

29     username = new String(bytes, "utf-8");*/
30     username=new String(username.getBytes("iso8859-1"),"utf-8");
31     System.out.println("username="+username);
32
33     //一次性获取所有 理解, BeanUtils使用到
34     //Map<form上name的值,name对应的用户输入的或选择的值>
35     Map<String, String[]> map = request.getParameterMap();
36     //如何循环
37     Set<Map.Entry<String, String[]>> entries = map.entrySet();
38     for(Map.Entry<String, String[]> e:entries){
39         System.out.println(e.getKey()+" "+
Arrays.toString(e.getValue()));
40     }
41
42     //方式三: 了解
43     //Enumeration就是迭代器的前身
44     Enumeration<String> names = request.getParameterNames();//form表单所
有的name属性的值
45     while (names.hasMoreElements()){
46         String name = names.nextElement();
47         System.out.println(name+" "+request.getParameter(name));
48     }
49     //方式四: 掌握
50     //getParameterValues:获取复选框的选中
51     String[] hobbies = request.getParameterValues("hobby");//form提交的用
户输入的值
52     System.out.println(Arrays.toString(hobbies));
53     //2.创建依赖的Service对象
54
55     //3.将方法执行结果使用响应协议回送给浏览器显示
56     }
57
58     //专门处理post请求协议
59     @Override
60     protected void doPost(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
61         //登录
62         doGet(req,resp);
63     }
64 }
65

```

2 HttpServletResponse

2-1 HttpServletResponse概念

1 | HttpServletResponse:就是服务器回送给浏览器的响应协议

2-2 HttpServletResponse作用

```
1 HttpServletResponse提供方法:
2 1 设置响应首行
3 setStatus(int 状态码); //状态码在定义常量, 自己找
4 2 设置响应头
5
6 3 设置响应体
7 getOutputStream()
8 getWriter();
9
```

Methods inherited from interface javax.servlet.ServletResponse 服务器向浏览器输出字节流

flushBuffer, getBufferSize, getCharacterEncoding, getContentType, getLocale, getOutputStream, getWriter, isCommitted, reset, resetBuffer,
setBufferSize, setCharacterEncoding, setContentLength, setContentType, setLocale 服务器向浏览器输出字符流

2-3 设置响应体中文乱码问题

Methods inherited from interface javax.servlet.ServletResponse

flushBuffer, getBufferSize, getCharacterEncoding, getContentType, getLocale, getOutputStream, getWriter, isCommitted, reset, resetBuffer,
setBufferSize, setCharacterEncoding, setContentLength, setContentType, setLocale

设置响应体的编码格式 设置浏览器解析响应体的编码格式

```
1 setContentType(), setCharactorEncoding就会按照setContentType()设置的编码设置响应体
```

2-4 课堂案例

```
1 package cn.kgc.servlet;
2
3 import javax.servlet.ServletException;
4 import javax.servlet.http.HttpServlet;
5 import javax.servlet.http.HttpServletRequest;
6 import javax.servlet.http.HttpServletResponse;
7 import java.io.IOException;
8 import java.io.PrintWriter;
9
10 /**
11  * @Author: lc
12  * @Date: 2022/5/16
13  * @Description: 建议: 一个Servlet中处理一个请求
14  * LoginServlet处理登录, 通常doGet或doPost其中一个方法中写代码即可
15  * @Version: 1.0
16  */
17 public class LoginServlet extends HttpServlet {
18     //专门处理get请求协议
19     @Override
20     protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
21         //设置请求体的编码格式, POST请求提交的数据解决中文乱码
22         request.setCharacterEncoding("UTF-8"); //请求体的编码格式, 只要支持中文即可
23         //登录
24         //1. 获取数据
25         //数据从什么地方提交给Tomcat服务器? ? 请求协议
26         //掌握单个key-value获取
27         String username = request.getParameter("username"); //获取username表单
项用户输入的值
```

```

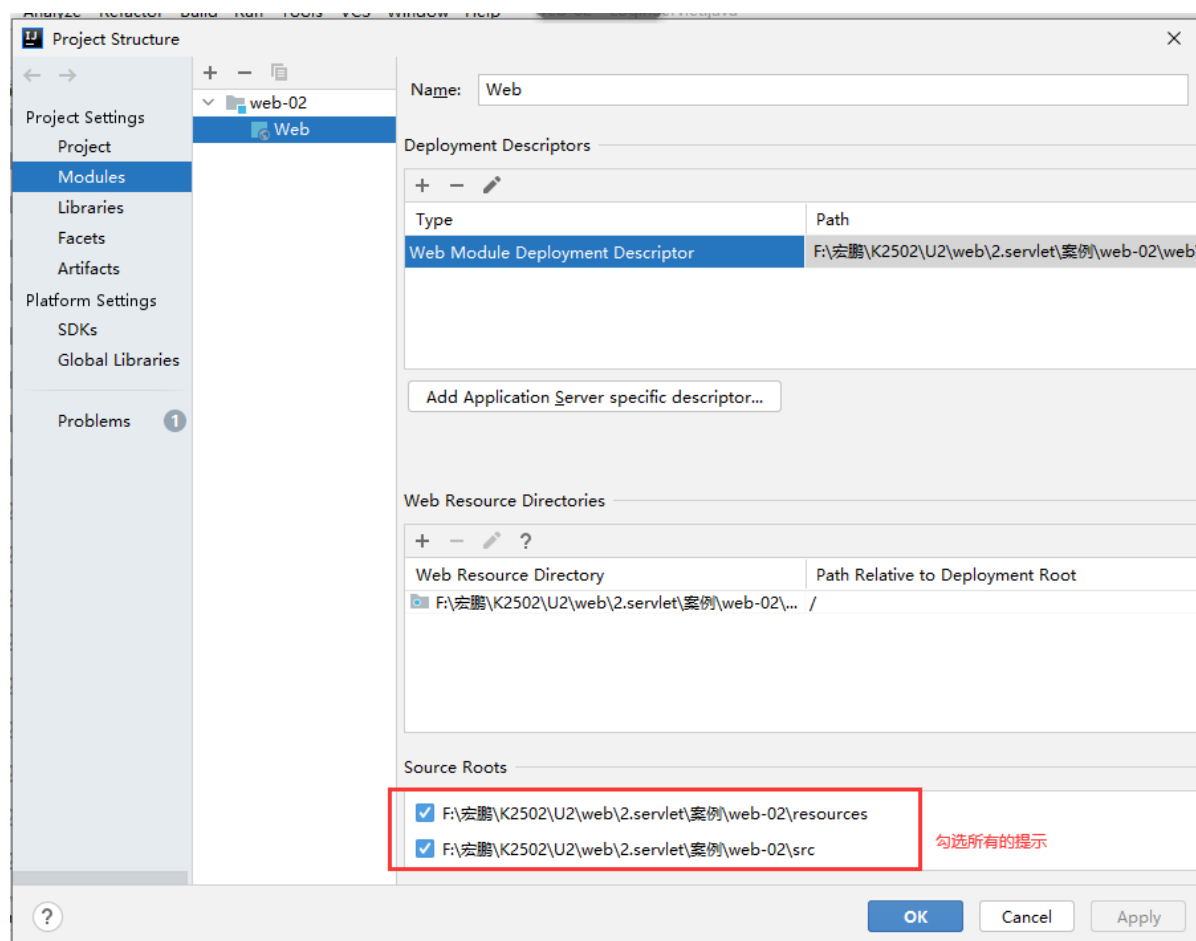
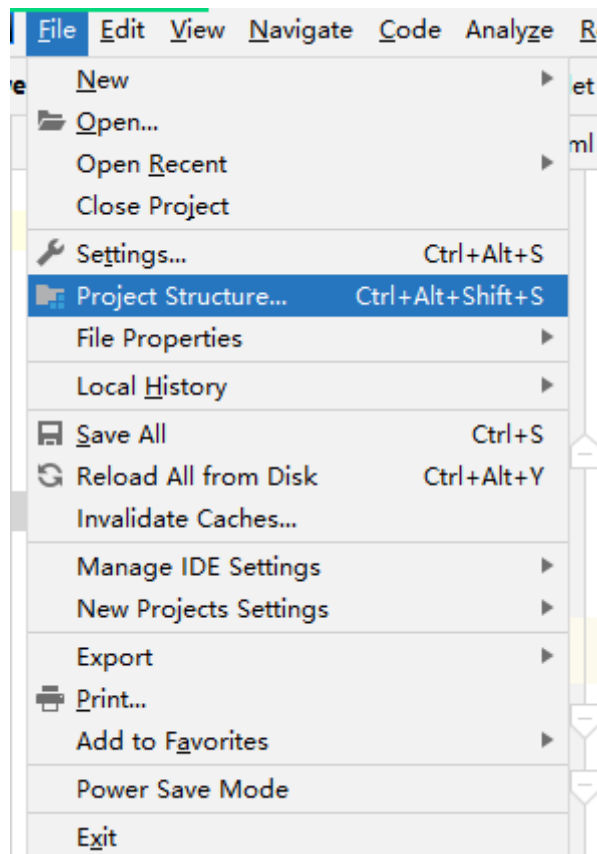
28         //username获取?????
29         /*byte[] bytes = username.getBytes("iso8859-1");
30         username = new String(bytes, "utf-8");*/
31         username=new String(username.getBytes("iso8859-1"),"utf-8");
32         System.out.println("username="+username);
33
34         //2.创建依赖的Service对象
35         //假设方法执行完毕，返回boolean  bool=true 登录成功  false-登录失败
36         boolean bool=true;
37         //3.将方法执行结果使用响应协议回送给浏览器显示
38         //resp.setStatus(HttpServletResponse.SC_INTERNAL_SERVER_ERROR);//500
39         //浏览器发送请求，服务器接收到之后，404响应它
40
41         //response.setStatus(HttpServletResponse.SC_INTERNAL_SERVER_ERROR);//SC_NOT_
42         FOUND 404
43
44         //封装响应协议
45         //response.setStatus(200);//ok 一般自己可以不写
46         //response.设置响应头 //没有特殊响应设置需求，响应头默认即可
47         //response.设置响应体// 要不要程序员设置??? 需要!!
48         //设置响应协议编码格式
49         //response.setCharacterEncoding("utf-8");//设置响应体
50         response.setContentType("text/html;charset=utf-8");//设置浏览器
51         //设置文本，使用字符流
52         PrintWriter writer = response.getWriter();
53         writer.print(bool?"登录成功":"登录失败");//不换行
54
55         //writer.println();//换行
56         //设置非文本，使用字节流
57     }
58
59     //专门处理post请求协议
60     @Override
61     protected void doPost(HttpServletRequest req, HttpServletResponse resp)
62     throws ServletException, IOException {
63         //登录
64         doGet(req,resp);
65     }
66 }

```

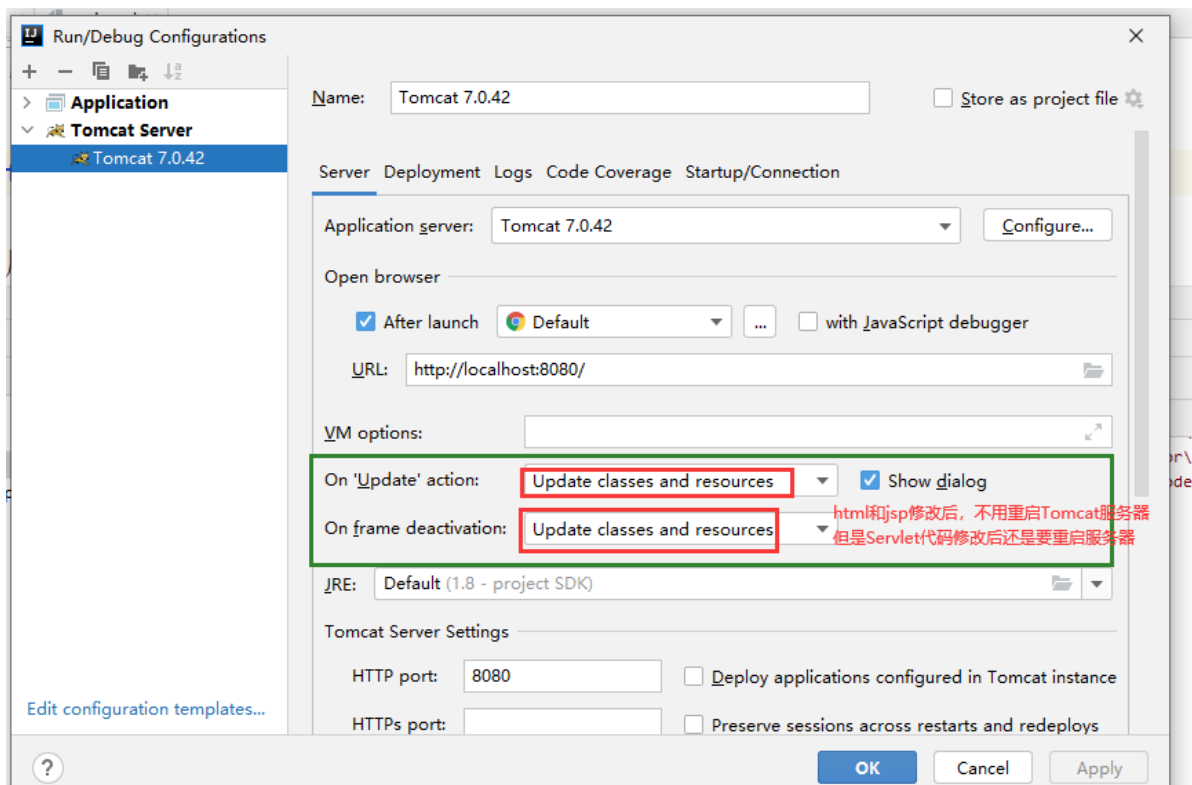
3 总结javaWeb书写路径的套路

- 1 form表单中action:/web项目名/servlet的url-pattern名称
- 2
- 3 浏览器地址栏: http://localhost:8080/web项目名/servlet的url-pattern名称

补充: idea2021中如何出现Servlet的创建模板



补充：如何配置Tomcat在修改html和jsp后不重启



补充：如何区分Tomcat对于GET请求的编码设置

| | |
|-------------|---|
| URIEncoding | This specifies the character encoding used to decode the URI bytes, after %xx decoding the URL. If not specified, UTF-8 will be used unless the org.apache.catalina.STRICT_SERVLET_COMPLIANCE system property is set to true in which case ISO-8859-1 will be used. get请求的默认编码是UTF-8 |
|-------------|---|

课程总结

1. Request获取表单数据的方法

```
getParameter()  
getParameterMap()  
getParameterNames()  
getParameterValues()
```

2. 请求提交中文乱码问题

```
post: request.setCharacterEncoding()  
get: new String(request.getBytes("iso8859-1"), "utf-8")
```

3. response设置状态码和响应体

```
setStatus()  
getWriter().print()
```

4. 响应乱码问题

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

预习安排

request的转发

response的重定向

response实现文件下载