Review tomcat

1.XML的作用以及它的解释和它的声明

2.XML的主要约束

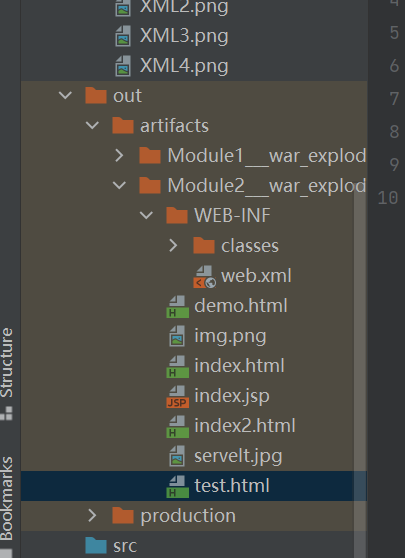
3.名称空间的作用

4.XML的属性名以及它的意思

5.属性文件格式以及它内部的语法

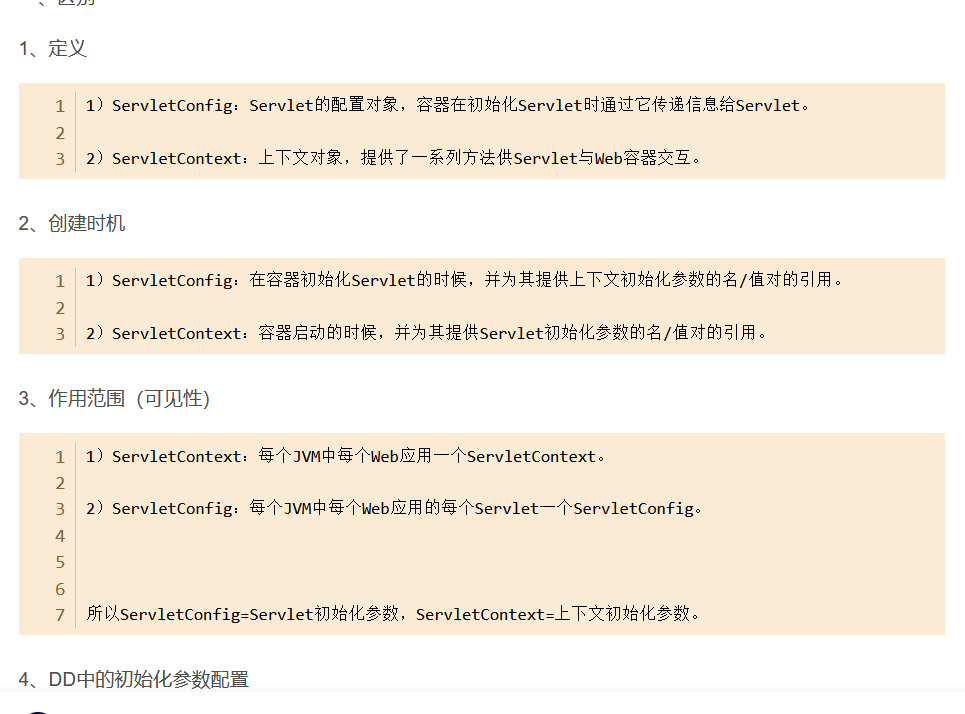
6.XML具体是个什么东西

1. Tomcat是个什么东西（从两方面来讲）以及他们的结构图
2. Servlet的含义
3. Tomcat占用了几个端口号，分别是哪几个
4. Tomcat本身是一个什么？
5. 当我们运行的时候，其实运行的是部署目录，若我们部署目录中没看见我们新增的文件，则重新部署



1. 导入别人的module (操作)，点击xxx.iml文件
2. Module重新部署（操作）
3. Edit里面的on update action和on frame deactivation的含义以及选择
4. Application text的意思
5. Context root的意思

答：Context root用于设置项目访问路径

1. HTTP协议全名以及它的作用
2. 请求报文，请求行，请求消息头，请求体，他们四个的关系
3. 请求消息头的6个重要的属性以及他们的解释
4. 请求体的解释
5. HTTP定义的八种请求方式，常用的两种以及他们的作用和特征
6. 响应报文包括哪三个？
7. 响应体在导航栏，响应消息头的四个重要属性的作用
8. 响应状态码 200，302，403，404，50x
9. 请求数据和响应数据又称为请求报文和响应报文
10. Servlet（类和接口）的作用（底层实现）
11. 在程序中想用servlet类，必须从tomcat的lib中获取servlet.api给程序附上依赖
12. 在浏览器端打印出hello world
13. Web.xml中的配置以及配置的具体分析
14. Web工程中的资源分为两大类，具体指出两大类中的资源
15. 访问静态资源地址格式?
16. 访问动态资源地址格式？
17. 虚拟路径是指的什么？
18. Web应用名称指的是什么
19. Servlet生命周期？
20. Servlet创建对象的时机？（创建对象后自动初始化）
21. 我们可以修改servlet创建对象的时间点，怎么修改？
22. Servlet生命周期各个方法的总结
23. ServletConfig里面的方法
24. 在web.xml中配置初始化参数
25. Enumeration遍历方法
26. 在web.xml中配置Web应用的初始化参数
27. ServletContext里面的方法（7个）
28. 使用idea创建servlet（操作）
29. Idea生成的servlet自动继承了HttpServlet
30. **实现Servlet接口**而是**继承HttpServlet类**
31. Servlet的继承关系（熟悉）
32. 工程目录和部署目录的区别和基准
33. 工程目录和部署目录的对应关系
34. 获取web应用名（context root/application text）方法
35. 重定向的特点
36. 转发和重定向的对比
37. 转发和重定向的应用场景
38. WEB-INF的解释
39. 请求参数的概念，以及例子
40. 服务器将请求参数的封装
41. 获取请求参数的方法
42. 在post方式下，我们对请求报文进行编码，方式以及注意点？（get目前不需要）
43. 响应报文进行编码，两种方式以及注意点
44. Servletconfig和servletContext的区别？
45. 
46. 获取应用名称application context的方法？
47. 我们可以用过request.setAttribute(“”,””)，将数据存储到web容器内，后面通过getAttribute(“”)获取即可，具有共享性质
48. Thymeleaf是一个什么技术？
49. Thymeleaf的优势
50. 物理视图的解释以及它的组成
51. lib一般都放在WEB-INFO下，
52. 在上下文参数(web.xml)中配置视图前缀和视图后缀
53. Html中thymeleaf声明位置
54. th:text=””的作用，修改内部文本节点，以及没有它又怎样？
55. Thymeleaf指定属性值方式
56. 加上thymeleaf的属性都可以通过thymeleaf设定新值，修改属性节点
57. 解析URL地址基本语法,”@{}“作用是什么？
58. 注意：上下文路径=contextPath=context root=web应用名
59. 直接执行表达式${key}具体看文档
60. 我们通常用哪种方式设置key，域对象
61. Object getAttribute(String name) 获取指定的属性
62. Enumeration getAttributeNames(); 获取所有名字属性组成的 Enumeration对象。
63. removeAttribute(String name) 移除指定的属性
64. void setAttribute(String name,Object o); 设置属性

有这些方法的对象,request(操作请求域，一次请求响应范围),session(操作绘画域，一次会话范围),servletcontext(操作应用域，整个应用程序范围)对象都有这些方法，也称之为域对象``

**第六节访问域对象，重点看**

1. 获取请求参数（一个值，或多个值）
2. getContextPath()是哪个对象，
3. Thymeleaf的四个基本内置对象

#request:获取HttpServletRequest对象

#response:获取HttpServletResponse 对象

#session：获取HttpSession 对象

#servletContext:获取ServletContext对象

1. 内置判空方法，参数一般
2. OGNL语法，在Thymeleaf环境下，${表达式}，里面的对象可以是 8个

访问属性域的起点

* 1. 请求域属性名
  2. session
  3. application

1. param
2. 内置对象
   1. #request
   2. #session
   3. #lists
   4. #strings

### 25. 属性访问语法

* 访问对象属性：使用getXxx()、setXxx()方法定义的属性
  + 对象.属性名
* 访问List集合或数组
  + 集合或数组[下标]
* 访问Map集合
  + Map集合.key
  + Map集合['key']

1. 分支和迭代

th:if th:unless(if内部取反)意思以及作用 display

th:switch th:case用法

th:each用法

26．其他模板文件：4个语法的使用

1. 补充 ：

HttpServlet的service方法底层实现

1. 动态资源是什么？静态资源又是什么？
2. 一个servlet接口实现类意味着只能创建一个实例？
3. Httpservlet的doxxx方法默认都是返回“xxx”？
4. 当void destroy()调用时，所用的servlet都会被？
5. Load-on-startup的意思以及解释？
6. Http是无状态的，解释？
7. 我们怎么解决http会话无状态的难题？
8. 会话跟踪技术原理？
9. 创建会话的三个方法
10. 方法：获取sessionId,判断session是不是新的，使会话强制失效，设置获着获取闲置时间（默认是多少？），设置作用域
11. 四种获取servletcontext的方法？
12. Cookie的本质？
13. Cookie的作用？
14. 没创建cookie就没有cookie
15. 创建一个cookie
16. 可以创建多个cookie
17. 把cookie对象保存到浏览器端
18. 获取cookie
19. Cookie的应用
20. Cookie分为
21. 设置cookie过期时间
22. Kaptcha验证码的使用
    1. 添加jar包
    2. 在web.xml文件中注册kaptchaservlet,并设置验证码图片的属性
    3. 在html页面上编写img标签，然后设置src等于kaptchaservlet对应的url-pattern
23. 生成验证码时，会将验证码信息保存到session中，将文本框输入的值和session的值判断

通过查看源码，找到验证码存入Session域时使用的属性名是：

KAPTCHA\_SESSION\_KEY

1. Filter开发步骤
   1. 新建实现了filter接口的实现类
   2. 然后重写其中的三个方法init(FilterConfig),doFilter(ServletRequest, ServletResponse,FilterChain),destroy()

然后配置filter，在web.xml中用<filter>和<filter-mapping>,与配置servlet差不多。注意：filter-mapping中，url-pattern应该设置为你要拦截的资源，所以filter实现类和servlet实现类中的url-pattern的值应相等，才能精准拦截，我们通过请求地址模式来设置要拦截的资源 （url-pattren）

1. dofilter()方法中的，filterChain.doFilter(servletRequest,servletResponse)，filter放行代码，响应回来后执行放行代码下面的代码
2. 过滤器链，了解一下？
3. Filter生命周期？
4. 过滤器匹配规则：精确匹配，模糊匹配（以…开头，以…结尾，所有，3种）
5. 过滤器链中的每一个过滤器执行的顺序，都是由web.xml中的filter-mapping配置顺序来决定的，
6. 观察者模式？
7. Servlet监听器？
8. 监听器（监听器主要针对域对象，都是接口）的分类？
9. Servletcontext什么时候创建的？
10. Servletrequest什么时候创建的？
11. 监听器列表？？？
12. 监听器用法：
    1. 创建监听器类
    2. 注册监听器（每一个listener标签对应着一个监听器，若有多个监听器，则配置多个listener标签即可）
13. 服务器端渲染和ajax渲染
14. Ajax全名
15. 同步和异步的区别
16. Ajax的应用
17. Axios是什么，以及他的作用？
18. Ajax以什么方式传递数据？
19. Axios操作ajax的步骤
    1. 添加并引入axios的js文件
    2. 客户端向服务器端异步发送普通参数值
20. Vue+axios代码解析
21. Axios基本格式
22. Axios的基本格式的示例
23. 客户端向服务器异步发送普通参数值（操作）
24. Axios收到的响应对象结构的属性名及其他的作用
25. 请求失败的属性名及其他的作用、
26. 发送请求体JSON，以及他的效果查看
27. GSON的作用
28. JSON的在axios的存储位置在“”,而不是param(存储请求参数的地方)
29. GSON的servlet代码
30. 创建GSON以及GSON两个重要的API
31. 服务器端返回的JSON数据在data中
32. Ajax 的应用
33. Ajax开发模式与传统模式的比较？
34. Ajax的底层实现
35. Ajax浏览器兼容问题？
36. 大部分浏览器创建XMLHttpRequest对象的方式（多层try-catch嵌套）
37. 兼容不同浏览器的创建XMLHttpRequest对象的方式
38. XMLHttpRequest的主要方法