

Tomcat常见面试题

Tomcat的缺省端口是多少，怎么修改

Tomcat 端口 多少， 怎么修

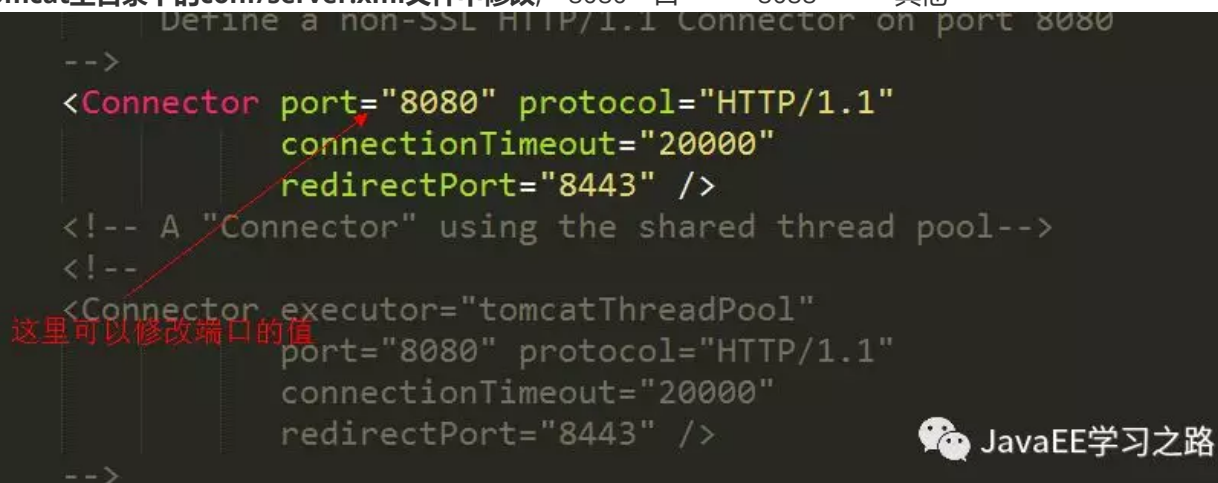
1. 到Tomcat 下 conf 文件夹
2. 入conf 文件夹 到server.xml 文件
3. 开server.xml 文件
4. 在server.xml 文件 到下列信

```
<Service name="Catalina">

    <Connector port="8080" protocol="HTTP/1.1"
        connectionTimeout="20000"
        redirectPort="8443" />
```

1. port="8080" 改为 port="8888", 并且保存
2. 启动Tomcat, 并且在IE 浏览器 地址 栏 输入<http://127.0.0.1:8888/>

到tomcat主目录下的conf/server.xml文件中修改, 8080 改为 8888 或其他



```
-->
    Define a non-SSL HTTP/1.1 Connector on port 8080
    <Connector port="8080" protocol="HTTP/1.1"
        connectionTimeout="20000"
        redirectPort="8443" />
    <!-- A "Connector" using the shared thread pool-->
    <!--
    <Connector executor="tomcatThreadPool"
        port="8080" protocol="HTTP/1.1"
        connectionTimeout="20000"
        redirectPort="8443" />
-->
```

这里可以修改端口的值

JavaEE学习之路

Tomcat 有哪几种Connector 运行模式(优化)?

tomcat 有哪几 种 Connector 运行模式 (优化)?

1. bio(blocking I/O)
2. nio(non-blocking I/O)
3. apr(Apache Portable Runtime/Apache可移植运行库)

关键 :

- bio: 传统的Java I/O操作, 同步且阻塞IO。
- nio: JDK1.4开始支持, 同步阻塞或同步非阻塞IO

- aio(nio.2): **JDK7开始支持，异步非阻塞IO**
- apr: Tomcat将以JNI 调用 Apache HTTP 服务器 动态 库 处 件 取 传 作, 从 大大地 **提高Tomcat对静态文件的处理性能**

下 **配置Tomcat运行模式改成是NIO模式，并配置连接池相关参数来进行优化:**

```
<!--
<Connector port="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443" />

-->
<!-- protocol 启 nio , (tomcat8 使 nio)(apr 利 io) -->
<!-- minProcessors 小 -->
<!-- maxProcessors 大 -->
<!-- acceptCount允 大 , 应大于 于maxProcessors-->
<!-- enableLookups 如 为true,request.getRemoteHost会 DNS , 反向 ip对应域名 主 名-->
<Connector port="8080" protocol="org.apache.coyote.http11.Http11NioProtocol"
    connectionTimeout="20000"
    redirectPort="8443

    maxThreads="500"
    minSpareThreads="100"
    maxSpareThreads="200"
    acceptCount="200"
    enableLookups="false"
/>
```

Tomcat有几种部署方式

1. Web 在webapps下, Tomcat会 动将其 对
2. 在server.xml 件上 `<Context>` , 关 属 即可
3. Catalina : 入到conf\Catalina\localhost 件下, 创建 个xml 件, 件 名字就 名字 写XML

部署方式第二点:

- 在其他 下创建 个web , 并创建WEB-INF 和 个html 件



- 到Tomcat 下/conf/server.xml 件



- 在server.xml中
下 加如下代 **path**表示的是访问时输入的web项目名，**docBase**表示的是站点目录的绝对路径

```
<Context path="/web1" docBase="D:\web1"/>
```

- 好 web

部署方式第三点:

- 入到conf\Catalina\localhost 件下，创建 个xml 件，该文件的名字就是站点的名字。
- xml 件 代 如下，**docBase**是你web站点的绝对路径

```
<?xml version="1.0" encoding="UTF-8"?>
<Context
  docBase="D:\web1"
  reloadable="true">
</Context>
```

- web 下 html

Servlet面试题

Servlet生命周期

Servlet 命周 ?

- 第一次访问Servlet， 们发 **init()**和**service()**都被调用了
- 第二次访问Servlet， **service()**被调用了
- 三 Servlet， 还是**service()**被调用了
- 们关闭Tomcat服务器 候， **destroy()**被调用了!

Servlet 生命周期 可分为5个

1. **加载Servlet** Tomcat 加载 Servlet 的时候, Tomcat会负责创建Servlet的实例
 2. **初始化** Servlet 实例化后, Tomcat会调用init()方法初始化这个对象
 3. **处理服务** 浏览器访问Servlet 时候, Servlet 会调用service()方法处理请求
 4. **销毁** Tomcat关闭 到Servlet 从Tomcat删除 时候会 调用 destroy() , 让该实例释放掉所占的资源 一个Servlet如 不 使用 , 也会 Tomcat 调用
 5. **卸载** Servlet 调用完destroy() 后, 垃圾回收 如 有需要再次使用这个Servlet, 会重新调用init()方法进行初始化操作
- **单线程** : 只要访问Servlet, service()就会被调用。init()只有第一次访问Servlet的时候才会被调用。destroy()只有在Tomcat关闭的时候才会被调用。

get方式和post方式有何区别

get 和post 有何区别

带上:

- GET : 在URL地址后 带 参 数 制 , 其 容 量 常不 1K
- POST : 可以在 实体内容中向 服务器发 送 , 传 输 制 度

参 数 位 上:

- GET : 参 数 在URL地址后 , 以?
- POST : 参 数 在HTTP 包中

上:

- GET 取 数据
- POST 交 数据
- - 先 因为GET 带 参 数 小, 带 去 大 数据
 - POST 交 参 数 后台 加容 量 (使 POST 交 中 数据 , 后台也 加容 量 决)
 - GET 比 POST 快
 - 原因:

GET 和 POST

Servlet相关 API

doGet与doPost 两个参 数 什么 对 象

1. HttpServletRequest: 封 了与 客 户 端 信 息 对 象
2. HttpServletResponse: 封 了与响应 客 户 端 信 息 对 象

取 参 数 元 值 几 种 , 分别 下

1. request.getParameter() 回客 户 端 参 数 值
2. request.getParameterNames() 回 客 户 端 属 名 举 例
3. request.getParameterValues() 回包含参 数 值

request.getAttribute()和request.getParameter()区别

上:

- request.getAttribute(), 一般用于获取request域对象的数据(在 之前 使 setAttribute 到 request对 上)
- request.getParameter(), 一般用于获取客户端提交的参数

存储 上:

- request.getAttribute()可以 取Objcet对
- request.getParameter()只 取字 串(也 为什么它 于 取客 交 参)

forward和redirect的区别

forward和redirect 区别

- 实际发生位置不同, 地址栏不同
- - 重定向是由浏览器进行跳转的, 定向 候, 浏览器的地址会发生变化的 介 : 实 定向 原 response 和Location头 合 实 这是由浏览器进行的页面跳转实 定向会发出两个http请求, **request域对 , 因为它不 同 个request对
 - 转发是由服务器进行跳转的, 友会发 , 在 发 候, 浏览器的地址栏是没有发生变化的, 在 Servlet111 候, 即使 到了Servlet222 , 器 地址 Servlet111 也就 浏览器是不知道该跳转的动作, 转发是对浏览器透明的 上 发 序图 们也可以发 , 实现转发只是一次的http请求, 发中request和response对 同 个 也 了, 为什么 可以使 request作为域对象进行Servlet之间的通讯。
 - 发 发 在 务器
 - 定向 发 在 器
- 用法不同:
- - 重定向时"/"代表的是webapps目录
 - 转发时"/"代表的是本应用程序的根目录【zhongfucheng】
 - 多人 不 发和 定向 候, 资源地址究竟怎么写 候 应 名写上, 候不 应 名写上 容 人 住 个原则: 给服务器用的直接从资源名开始写, 给浏览器用的 要把应用名写上
 - request.getRequestDispatcher("/ 名 URI").forward(request,response)
 - response.sendRedirect("/web应 / 名 URI");
- 能够去往的URL的范围不一样:
- - 转发是服务器跳转只能去往当前web应用的资源
 - 重定向是服务器跳转, 可以去往任何的资源
- 传递数据的类型不同
- - 转发的request对象可以传递各种类型的数据, 包括对象
 - 重定向只能传递字符串
- 跳转的时间不同
- - 转发时: 执行到跳转语句时就会立刻跳转
 - 重定向: 整个页面执行完之后才执行跳转

么 发(forward)和 定向(redirect)使 哪 个?

- 上 了 发和 定向 区别也可以 容 出 。转发是带着转发前的请求的参数的。重定向是 新的请求

典型 应 场：

1. 发: Servlet 处 业务 , 后 forward 到 jsp 处 , 器 URL 不变
2. 定向: 交 单, 处 功后 redirect 到另 个 jsp, 单 复 交, 器 URL 变了

tomcat容器是如何创建servlet类实例? 用到了什么原理?

tomcat容器 如何创建servlet 实例? 到了什么原

1. 容器启动 , 会 取在webapps 下 web应 中 web.xml 件, 后对 xml文件进行解析, 并 读取servlet注册信息 后, 将 个应 中 册 servlet 加 , 并 反射的方式实例化 (候也 在 实例化)

2. 在servlet 册 加上

1

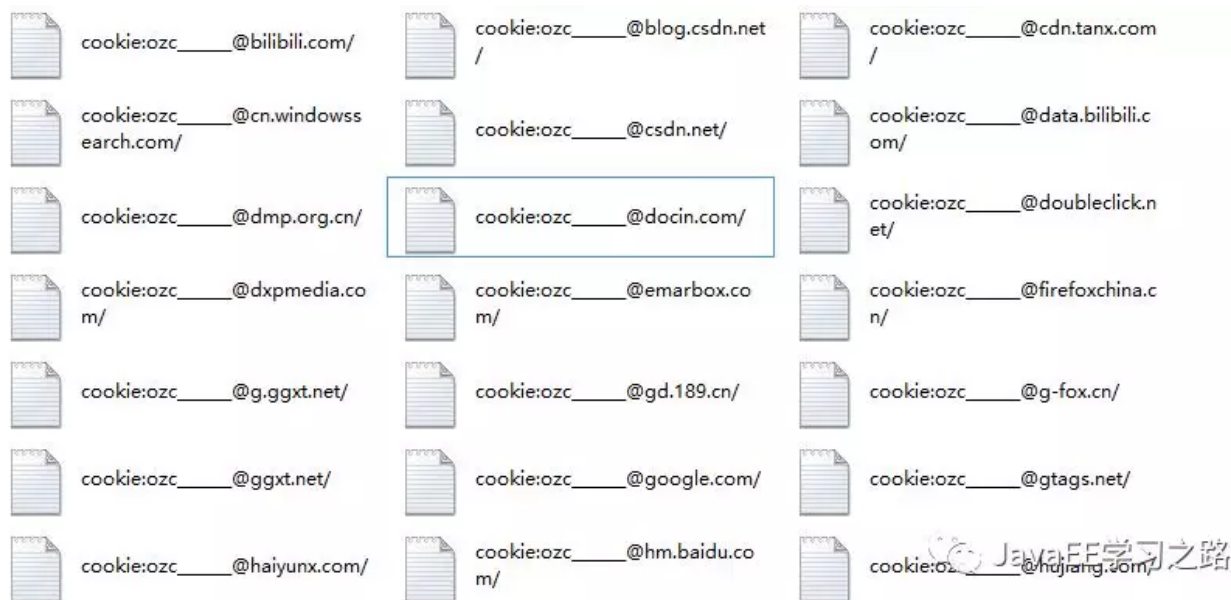
如 为 , 则在 开始就实例化, 如 不写 为 , 则 实例化

什么是cookie? Session和cookie有什么区别?

什么 cookie?

Cookie W3C 出, netscape 区发展 制

- 之 交互是通过HTTP协议传输数据的, Http协 无状态的协议 协 什么 ? 一旦数据提交完后, 浏览器和服务器的连接就会关闭, 再次交互的时候需要重新建立新的连接
- 务器 信 , 于 乎, W3C就 出了: 给每一个用户都发一个通行证, 无论谁访问的时候都需要携带通行证, 这样服务器就可以从通行证上确认用户的信息 就 Cookie



Session和cookie 什么区别?

- 从存储方式上比较
 - Cookie只 存储字 串, 如 存储 ASCII字 串 对其
 - Session可以存储任何 型 , 可以 Session 个容器
- 从隐私安全上比较

- Cookie存储在浏览器中，对客户端是可见的 信 容 出去 如 使 Cookie， 好将Cookie加密
 - Session存储在服务器上，对客户端是透明的 不存在 信
- 从有效期上比较
- Cookie保存在 中，只 maxAge属 为 大 ，即使关 器，Cookie 存在
 - Session的保存在服务器中，设置maxInactiveInterval属性值来确定Session的有效期。并且Session依赖于名为JSESSIONID的Cookie，该Cookie默认的maxAge属性为-1。如果关闭了浏览器，该Session虽然没有从服务器中消亡，但也就失效了。
- 从对服务器的负担比较
- Session 保存在 务器 ， 个 会产 个Session，如 并发 常多， 不 使 Session ， Session会 大 内存
 - Cookie 保存在客 不占 务器 像baidu Sina 大型 ， 使 Cookie 会
- 从浏览器的支持上比较
- 如 器 了Cookie， 么Cookie 了！
 - 如 器 了Cookie， Session可以 URL地址 写 会
- 从跨域名上比较
- Cookie可以 domain属 实 域名
 - Session只在 前 域名内 ， 不可跨域名

Servlet安全性问题

于Servlet 单例 ， 多个 Servlet 候， 服务器会为每个用户创建一个线程 多个 并发 Servlet共享 候就会出 安全

原则：

- 如 个变量需要多个用户共享，则应 在 变 候， 加同步机制synchronized (对象){}
- 如 个变 不需要共享，则直接在 doGet() 或者 doPost()定义. 不会存在 安全