

Java Web知识点复习提纲

一、Java Web概述

- WWW（万维网）的全称是world wide web
- C/S架构：Client-Server架构，即客户端-服务器架构。
- B/S架构：Browser-Server架构，即浏览器-服务器架构。
- Tomcat目录下bin目录的作用是保存Tomcat下的可执行程序（例如启动和停止脚本）
- 请求报文中，请求行包含的信息依次是：请求方法、URL、HTTP协议版本
- 响应报文中，响应行包含的信息依次是：HTTP协议版本、状态码、状态码描述
- Get请求会将参数拼接在URL后面，Post请求会将参数写入到请求报文体部
- 通过浏览器输入URL访问以及点击 `<a>` 元素发送请求均采用Get方式
- Get方法不适合提交隐私性数据或大量数据，Post方法适合
- HTTP协议中请求方法有：Get、Post、Delete、Put、Trace、Head、Options、Connect
注：浏览器本身只支持Get和Post两种请求方法
- HTML中， `<input>` 元素中决定请求参数名称的属性名是name
- Tomcat默认端口号为8080
- Tomcat目录中可以修改端口号的配置文件名为server.xml
- 一个Web项目中，核心配置文件web.xml放置在WEB-INF下，该目录对客户端不可见
- 常见的HTTP响应状态码及对应的描述
 - 400 BadRequest 不合法的请求格式
 - 404 NotFound 请求的资源未找到
 - 405 MethodNotAllowed 请求的方法不被允许
 - 500 InternalServerError 服务器内部错误
 - 302 Moved Temporarily 重定向

二、Servlet

- javax.servlet.Servlet接口
所有的Servlet类都必须实现javax.servlet.Servlet接口。该接口中有五个方法：
 - init() 初始化时调用，只会调用一次
 - destroy() 销毁时调用，只会调用一次
 - service() 每次请求到来时调用，是Servlet中被调用执行最多的方法
 - getServletConfig() 获取Servlet配置信息（或初始化参数）
 - getServletInfo() 获取Servlet描述信息
- javax.servlet.GenericServlet和javax.servlet.http.HttpServlet都是抽象类
- 创建一个Servlet类的方式有三种：
 - 实现javax.servlet.Servlet接口

- 继承javax.servlet.GenericServlet父类
 - 继承javax.servlet.http.HttpServlet父类
- GenericServlet将Servlet中的四个抽象方法都进行了实现，唯一未实现的方法：service
- GenericServlet中service方法的参数类型是ServletRequest和ServletResponse
- HttpServlet中doGet方法的参数类型是HttpServletRequest和HttpServletResponse
- HttpServletRequest对象和HttpServletResponse都是由Tomcat自动创建的
- Servlet是单例模式，同一个Servlet类中，所有客户端的请求均由同一个对象处理
- HttpServlet中，get请求会由doGet()方法处理，post请求会由doPost()方法处理
- request.getParameter("参数名")用来获取请求报文中的指定参数值，返回值为String类型
- request.getServletContext().getInitParameter("name")可以用来获取web.xml配置文件中 `<context-param>` 中定义的初始化参数
- 跳转方式有两种：内部跳转、重定向跳转
- 实现内部跳转访问某个URL的对象是request
- 如果使用内部跳转，则用户无法在浏览器看到真实的资源路径
- 实现Servlet之间重定向的对象是response
- 重定向跳转会增加访问服务器的次数，例如重定向2次之后，一共访问了服务器3次
- 常用的两种会话跟踪技术是Session和Cookie
- Session是一种通过服务器端保存数据的技术
- HttpSession对象表示一次会话，该对象在每个客户端会话中第一次调用getSession()方法时被创建
- 多个客户端不能共享一个Session对象
- HttpSession对象不能直接使用new关键字创建
- Session的默认有效时长是30分钟，可以通过web.xml中标签进行修改
- Cookie是一种在客户端保存数据的技术
- Cookie被创建在服务器端，通过响应报文返回给浏览器，浏览器会自动保存
- 浏览器在每次提交请求时都会携带所有Cookie，包含在请求报文中
- Cookie的默认的生命周期是会话周期，当关闭浏览器，会话cookie信息会被浏览器删除
- Cookie无法直接使用方法删除，但是可以通过使用同名字段覆盖的方法实现删除功能
- 按照正常流程，请求必然先到达Filter，再到达Servlet（手动转发除外）
- 创建一个Filter的方式是实现javax.servlet.Filter接口
- Filter过滤器中用来处理请求的方法是doFilter
- Listener监听器的作用是监听特定事件的发生，当发生之后调用特定的方法处理
- Tomcat开启时ServletContextListener中的contextInitialized方法将会被执行。
- Tomcat关闭时ServletContextListener中的contextDestroyed方法将会被执行
- HttpServletRequest中获取所有请求参数名和参数值的方法是getParameterMap
- HttpServletResponse中设置响应内容类型的方法是setContentType
- ServletContext表示服务器范围，所有会话中每一次请求都共享同一个ServletContext
- 三大作用域大小关系：ServletContext > HttpSession > HttpServletRequest
- 三大作用域中，存储数据的方法都是setAttribute，删除数据的方法都是removeAttribute
- 路径表示方法中，服务器端"/"斜杠的含义是当前项目的根路径
-

三、JSP

- JSP的本质是Servlet，所以其运行需要Tomcat环境支持
- Tomcat启动时，并不会直接翻译所有的JSP文件。每个JSP都会在被访问时进行翻译和编译
- JSP被翻译生成的.java文件以及编译生成的.class放置在tomcat目录下的work目录中
- JSP中的注释语法：`<%-- 注释内容 --%>`
- JSP中有九大内置对象：request、response、session、application、page、pageContext、out、exception、config
- JSP中的声明脚本会被翻译为类中属性和方法
- JSP中的代码脚本会被翻译为_jspService()方法中的代码
- JSP中的表达式脚本会被翻译为Java中的引用变量或方法调用
- JSP中四大作用域大小关系：ServletContext > HttpSession > HttpServletRequest > PageContext
- 多个作用域中有同名字段时，EL表达式会默认从最小范围中取
- JSTL标签属于第三方标签库，不可直接使用。必须先安装依赖，并在JSP中使用taglib指令引入
- JSTL标签库中，`<c:forEach>` 用来实现循环，`<c:if>` 用来实现判断
- JavaBean类的要求
 - 属性必须使用private修饰，实现封装
 - 必须提供可访问的无参构造器
 - 每个属性必须提供get/set方法

四、JDBC

- MySQL数据库的默认端口号是3306
 - SQL命令五大分类：DDL、DCL、DML、DQL、DTL(或TCL)
 - JDBC中MySQL数据库驱动类的全限类名是com.mysql.jdbc.Driver
 - DriverManager的作用是管理所有的驱动类，以及获取Connection连接对象
 - JDBC编程六大步骤：
 - 注册驱动
 - 获取连接
 - 创建Statement对象
 - 编写和执行SQL命令
 - 解析结果集
 - 关闭资源
- 注：顺序不可随意颠倒
- 注册驱动三种方式
 - 反射机制
 - 创建驱动类对象
 - 将驱动类全限类名添加至VM启动参数
 - 获取连接两种方法：
 - DriverManager.getConnection(url, user, password)
 - driver.connect(url, info)

注：info是一个Properties集合对象

- 适合执行异构SQL命令的是Statement
- 获取Statement方法需要使用Connection接口提供的createStatement方法
- 适合执行同构SQL命令的是PreparedStatement
- 获取PreparedStatement方法需要使用Connection接口提供的prepareStatement方法
- PreparedStatement是Statement的子接口，特点是需要预编译，不需要对每条SQL单独编译
- JDBC注册驱动的方式中，使用到的反射机制代码是Class.forName()
- JDBC中负责管理事务的对象是Connection
- JDBC中事务的默认提交规则是自动提交
- JDBC中Connection设置事务提交策略的方法是setAutoCommit
- JDBC中可以通过Connection对象的commit()方法手动提交事务
- JDBC中可以通过Connection对象的rollback()方法回滚事务
- execute、executeQuery、executeUpdate都可以执行任意SQL命令，只不过返回值不相同
 - execute返回值为boolean类型
 - executeQuery返回值为ResultSet类型（注意：不是java.util包下的集合组件）
 - executeUpdate返回值为int类型
- ResultSet结果集中，数据库表中的int类型，可以使用ResultSet.getInt()方法获取
- ResultSet结果集中，数据库表中的varchar等类型，可以使用ResultSet.getString()方法获取
- ResultSet结果集中，next方法用于判断是否存在下一条数据
- JDBC中的资源如果不关闭，不影响功能实现（例如SQL的执行和结果集的解析）。但是会造成数据库压力过大甚至崩溃，以及Java内存泄漏风险
- JDBC关闭资源的原则是：先开的后关，后开的先关
- JDBC中将SQL添加到批处理空间的方法是addBatch，执行批处理的方法是executeBatch