

## Java Web知识点复习提纲（二）

---

- 内部跳转和重定向的区别

- 代码实现

内部跳转：`request.getRequestDispatcher("url").forward(request, response);`

重定向跳转：`response.sendRedirect("url");`

- 原理区别

内部跳转也叫请求转发，其过程发生在服务器内部，对于客户端不可见。转发的本质就是服务器内部Java类和方法之间的互调。无论转发多少次，对于客户端来说都只发送了一次请求，所以浏览器地址栏中显示的只有发送请求的路径，可能看不到所访问资源的真实路径。

重定向跳转指的是直接向浏览器返回一封状态码为302的报文，并在响应报文头部中通过Location字段指示重定向路径。客户端会自动向这个路径再次发送请求。重定向会导致访问服务器次数增加。跳转的页面相当于客户端自己发送请求访问到的，所以浏览器地址栏中能够显示出所访问资源的真实URL。

- 范围区别

内部跳转只能跳转到内部资源，例如服务器路径下的某个HTML、JSP或Servlet。

重定向跳转没有限制，可以跳转到外部服务器。

例如，想要跳转到百度首页，只能使用重定向，因为百度资源并不属于我们服务器“内部”。

- Servlet和JSP的异同点

- 相同点

- 都可以接收请求、返回响应，因为JSP的本质就是一个Servlet。

- 不同点

- 表现形式不同。Servlet是一个Java类，而JSP是一个形如HTML格式的文件。
    - 访问方式不同。Servlet可以作为一个Java类直接访问，而JSP需要经过翻译再执行。

- Servlet和JSP的选择问题

- 如果主要需求是处理代码逻辑，则使用Servlet比较合适。
  - 如果主要需求是显示一些资源，则使用JSP比较合适。

- Servlet框架中的监听器可以监听的事件

- 三大作用域对象的声明周期，例如：

- HttpServletRequest的创建和销毁
    - HttpSession的创建和销毁
    - ServletContext的创建和销毁

- 三大作用域对象中数据的活动，例如：

- HttpServletRequest中数据的添加、替换和删除
    - HttpSession中数据的添加、替换和删除
    - ServletContext中数据的添加、替换和删除

- HttpSession的声明周期

- 创建阶段

某个会话中，第一次调用`getSession()`方法。

- 使用阶段

当请求到达时，调用setAttribute()、removeAttribute()、getAttribute()等方法存取数据。

- 销毁阶段

程序中手动调用invalidate()，或超出了所设置的允许的最大不活跃时间，HttpSession对象会被销毁。

- Statement中可以执行SQL命令的三个方法以及描述

三个方法都可以执行任何SQL命令。如果目的仅仅是把SQL发到数据库服务器，使其能够被解析执行，那么调用任意一个方法都可以。

三个方法的区别在于返回值不同。

- execute方法，返回值为boolean值，代表是否有结果集（只有select查询语句有结果集）。
- executeQuery方法，返回值为ResultSet结果集对象，代表一次查询语句的结果集数据。适合执行select查询命令。
- executeUpdate，返回值为int值，表示有多少行数据受到了影响。适合执行DML（insert、update、delete）语句。

- 请求响应模型中浏览器、Servlet和web服务器之间的工作流程

- 首先，浏览器发送请求（请求报文）。
- Web服务器（Tomcat）先接受到请求，对请求报文进行解析，封装为一个ServletRequest对象，再创建一个空的ServletResponse对象。  
然后根据服务器配置的映射规则，将请求对象和响应对象转交给某个Servlet进行处理。
  - 如果该Servlet不存在，则会先创建一个该类实例，并且调用init()方法进行初始化
  - 如果该Servlet存在，则直接调用其service()方法，将request和response对象传入
- 执行我们在Servlet中编写的代码，处理业务。
- Servlet中的方法执行完毕之后，Web服务器会根据代码中对response对象进行的操作，自动编写对应内容的响应报文，返回给浏览器。
- 浏览器接收到响应报文之后，将其中的内容解析执行出来，并显示在浏览器界面中。

- 数据库连接池的工作机制

数据库连接池在系统初始化时创建一定数量数据库连接对象（Connection），需要时直接从池中取出一个空闲对象，用完后并不直接释放掉对象，而是再放到对象池中，以便下一次对象请求可以直接复用。这样可以消除对象创建和销毁所带来的延迟，从而提高系统的性能。

- JDBC实现批处理的步骤及方法

- 步骤1：将一条SQL命令添加到批处理空间  
addBatch()
- 步骤2：将批处理空间中的SQL真正发送给数据库执行  
executeBatch()

- MVC架构中各个组件的作用

Model：模型，用于封装业务逻辑相关的数据以及对数据的处理方法。

View：视图，用于实现业务的输入和显示。

Controller：控制器，用于视图和模型组件之间的组织和协调。

- Servlet的生命周期以及对应的方法

- 加载类和实例化阶段，调用Servlet类的无参构造器。
- 初始化阶段，调用init方法。
- 处理请求阶段，调用service方法
- 销毁阶段，调用destroy方法
- HTTP请求报文和响应报文的格式

## HTTP请求报文示意图



## HTTP响应报文示意图



- Cookie和Session的关系和区别

相同点：都是数据持久化技术的，可以存储数据。

不同点：Cookie是通过客户端存储数据，Session是通过服务器存储数据。

关系：Session功能是通过Cookie实现的。当服务器端第一次为某个客户端创建Session对象时，会为该Session对象创建一个ID，并通过Cookie返回至浏览器。

- JDBC编程六大步骤

- 注册驱动
  - 获取连接
  - 创建执行SQL命令的Statement对象
  - 编写并执行SQL命令
  - 解析结果集
  - 关闭资源释放连接
- 为什么JDBC使用完毕之后一定要关闭资源？

如果不关闭资源，则会导致与数据库保持的连接数量增加，增加数据库的访问压力。

而且资源类的对象不会被GC回收，如果不手动关闭，可能会出现内存泄漏问题。

- Tomcat根目录下bin、conf、work、webapps、lib五个目录的作用
  - bin：保存Tomcat服务器的可执行程序，例如startup.bat启动脚本
  - conf：保存配置文件
  - work：保存服务器运行过程中生成的文件，例如JSP翻译和编译输出的.java和.class文件
  - webapps：服务器部署项目路径
  - lib：服务器依赖保存路径

- Tomcat启动时如果出现“Address is already in use”的原因及解决办法

报错提示的大致意思是地址已被占用，一般指的是Tomcat启动时要绑定的端口号已经被其他进程占用。

解决方法有两个：

- 将占用该端口号的进程杀死（在CMD中可以使用netstat -ano可以查询端口号被使用情况）。
  - 更换其他端口号（端口号范围是0~65535）。
- JDBC中的批处理技术为什么能够提高效率？

在大批量SQL的执行过程中，程序中拼接一条SQL命令和数据库服务器执行一条SQL命令的速度其实都很快，主要的时间都花费在了程序与数据库的通信过程中。所以批处理技术的思想就是，将SQL积攒起来一次性发送给数据库执行，减少频繁通信产生的时间开销。

- 请求转发和重定向跳转的选择问题

根据每种跳转方式的特点，结合实际需求进行选择。

如果需要对访问的资源路径进行隐藏保护，可以使用内部跳转方法。

如果想要避免表单重复提交问题，或者想要跳转到外部站点，可以使用重定向方法。

- 数据库SQL命令的五大分类及其作用
  - DDL命令：数据库定义语言，用来定义数据库对象结构
  - DCL命令：数据库控制语言，控制用户的权限
  - DML命令：数据库操作语言，用来操作表中的数据（增删改）
  - DQL命令：数据库查询语言，用来查询表中的数据
  - DTL/TCL命令：数据库事务语言，用来控制事务的提交和回滚