

# CVTE | 20200203

作者: [douniwan](#)

原文链接: <https://ld246.com/article/1580832966847>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

## 问题

### 计算机网络方面

- 1.你了解http吗?
- 2.http是无状态的，那么怎么让它变得有状态?
- 3.如果后台服务器有多台，如何保存session?
- 4.http的报文格式?
- 5.http的请求头可以有中文吗？
- 6.编码和乱码的区别?

### 数据库

- 7.你知道InnoDB中的索引吗?
- 8.你知道最左匹配原则吗?

### Java

- 9.说说你对Java中的线程的了解
- 10.Java中的线程和计算机操作系统的线程是一样的吗?
- 11.说说你对Java中的多线程的了解
- 12.说说你对线程池的了解，为什么需要线程池？你知道线程池的哪几个参数?
- 13.你知道volatile吗?
- 14.你知道synchronized吗?
- 15.现在有一张订单表有库存表，需要你写一个下单的业务逻辑来保证库存表和订单表的一致性，你怎么写?
- 16.在Java中你一般怎么使用事物?
- 17.如果不使用 @Transactional，你将怎么实现事物?

### 补充

- 18.简述一下JAVA用JDBC连接数据库的步骤
- 19.preparedStatement与statement的区别

### 答案整理

#### **1.你了解http吗?**

答：了解，http(Hypertext transfer protocol)超文本传输协议，通过浏览器和服务器进行数据交互进行超文本（文本、图片、视频等）传输的规定。也就是说，http协议规定了超文本传输所要遵守的则。他有一个重要的特性就是无状态的，无状态就是指协议对于事务处理没有记忆能力，服务器不知客户端是什么状态。即我们给服务器发送 HTTP 请求之后，服务器根据请求，会给我们发送数据过来

但是，发送完，不会记录任何信息。

## **<span id="2">2.http是无状态的，那么怎么让它变得有状态? </span>**

答：可以采用Cookie或者session来实现http的有状态。

- Cookie 的实现过程

Cookie 会根据从服务器端发送的响应报文内的一个叫做 Set-Cookie 的首部字段信息，通知客户端存Cookie，当下次客户端再往该服务器发送请求时，客户端会自动在请求报文中加入 Cookie 值后发出去。

也就是 **Cookie 是服务器生成的，但是发送给客户端，并且由客户端来保存**。每次请求加上 Cookie 行了。服务器端发现客户端发送过来的 Cookie 后，会去检查究竟是从哪一个客户端发来的连接请求然后对比服务器上的记录，最后得到之前的状态信息。

- Session的实现过程

Session 是另一种记录客户状态的机制，不同的是 Cookie 保存在客户端浏览器中，而 Session 保存在服务器上。

客户端浏览器访问服务器的时候，服务器把客户端信息以某种形式记录在服务器上，这就是 Session 客户端浏览器再次访问时，只需要从该 Session 中查找该客户的状态就可以了。

虽然 Session 保存在服务器，对客户端是透明的，它的正常运行仍然需要客户端浏览器的支持。这是为 Session 需要使用Cookie 作为识别标志。HTTP协议是无状态的，Session 不能依据HTTP连接来判断是否为同一客户，因此服务器向客户端浏览器发送一个名为 JSESSIONID 的 Cookie，它的值为该 Session 的 id（即放在HTTP响应报文头部信息里的Set-Cookie）。Session依据该 Cookie 来识别是否同一用户。

## **<span id="3">3.如果后台服务器有多台，如何保存session? </span>**

答：可以利用cookie共享Session数据，利用Mysql数据库共享Session数据的方式也可以使用内存来享Session数据。

- 利用cookie共享Session数据

当用户请求后产生的session，我们把他的sessionId和值都存在cookie里面。这样，当你访问a服务后，产生了session放在客户端的cookie里面，你在访问被分配到b服务器上。这时候，b服务器先判本身服务器上有没有这个用户的session，如果没有，在去看看客户端的cookie里面有没有这个session，如果有，就获取客户端的这个cookie里面的session。这样就实现了session的同步。

- 利用Mysql数据库共享Session数据的方式

使用一个mysql服务器做共享服务器，把所有的session的数据保存mysql服务器上，所有的web服务都来这台mysql服务器来获取session数据。这里有一个关键的地方，用来存放session的数据表不要其他数据库表放在一起，要独立开来，专门放在一个低端的服务器上面。不然，数据库本身压力就很大了，再加上session是需要频繁的读取的，这使得数据库很容易达到瓶颈，从而导致过高的响应延迟。

- 使用内存来共享Session数据。

这里建议可以选择采用开源的缓存系统来完成session的共享，比如redis等。原理跟mysql一样，不哪个服务器产生的session都放在一个"内存池"里面。要获取session数据的时候都统一到这里获取。

建议用这个方法。

#### **<span id="4">4.http的报文格式? </span>**

答： HTTP请求报文由**请求行**（request line）、**请求头部**（header）、**空行**和**请求数据**4个部分组成。

#### **<span id="5">5.http的请求头可以有中文吗? </span>**

答：http的请求头中可以有中文，但是必须对中文进行编码，这样在服务端才可以对中文进行译码，不会使得传输的中文变成乱码。

#### **<span id="6">6.编码和乱码的区别? </span>**

答：编码就是一个编号\*\*（数字）到字符的一种映射关系，就仅仅是一种一对一的映射而已，可以理解一个很大的对应表格；乱码的原因就是文本字符编码过程与字节流解码过程使用了不同的编码格式，个往往归咎于解码格式选择错误，也就是说在解码的过程中出现了问题。如 果我的字符是用utf-8编，你用GBK解码那肯定出问题。

注：这里还可以区分一下[编码和编码格式的区别](#)

#### **<span id="7">7.你知道InnoDB中的索引吗? </span>**

答：知道，InnoDB引擎中是索引的数据结构采用B+树，然后索引分为两种，一个是主键索引一个是通索引。

- 主键索引

主键索引的b+树中，叶子节点里面存储的是该主键所在行的所有信息。

- 普通索引

普通索引的b+树中，叶子节点里面存储的是该索引所在行的主键信息。

也就是说，当我们在利用主键索引查询数据的时候，我们通过b+树找到相应的主键索引后，就可以接将信息查取出来；而利用普通索引查询的时候，通过该普通索引在b+树中找到它存储的相应主键然后再根据该主键去查询获取到数据。而普通索引根据查询到的主键再去查询相应的信息的这个过程为**回表**

#### **<span id="8">8.你知道最左匹配原则吗? </span>**

答：知道，最左匹配原则就是指在聚集索引中，如果你的sql语句中用到了聚集索引中的最左边的索引，那么这条sql语句就可以利用这个聚集索引去进行定位。例如某表有索引(a,b,c)现在你有如下语：

```
select * from t where a=1 and b=1; #这样可以利用到定义的索引 (a,b,c)
```

```
select * from t where a=1; #这样也可以利用到定义的索引 (a,b,c)
```

```
select * from t where b=1 and c=1; #这样不可以利用到定义的索引 (a,b,c)
```

```
select * from t where a=1 and c=1; #这样不可以利用到定义的索引 (a,b,c)
```

也就是说通过最左匹配原则你可以定义一个聚集索引，但是使得多中查询条件都可以用到该索引。

## **<span id="9">9.说说你对Java中的线程的了解</span>**

答：世间万物都可以同时完成很多工作。例如，人体可以同时进行呼吸、血液循环、思考问题等活动。用户既可以使用计算机听歌，也可以编写文档和发送邮件，而这些活动的完成可以同时进行。这种同时执行多个操作的“思想”在Java中被称为并发，而将并发完成的每一件事称为线程。

## **<span id="10">10.Java中的线程和计算机操作系统的线程是一样的吗？</span>**

答：不一样。

从实际意义上来讲，操作系统中的线程除去new和terminated状态，一个线程真实存在的状态，只有：

- ready：表示线程已经被创建，正在等待系统调度分配CPU使用权。
- running：表示线程获得了CPU使用权，正在进行运算
- waiting：表示线程等待（或者说挂起），让出CPU资源给其他线程使用

为什么除去new和terminated状态？是因为这两种状态实际上并不存在于线程运行中，所以也没什么实际讨论的意义

对于Java中的线程状态：

无论是Timed Waiting，Waiting还是Blocked，对应的都是操作系统线程的**waiting（等待）**状态。而Runnable状态，则对应了操作系统中的ready和running状态。

注：[关于操作系统线程和Java线程](#)

## **<span id="11">11.说说你对Java中的多线程的了解</span>**

答：在Java中，并发机制非常重要，但并不是所有程序语言都支持线程。在以往的程序中，多以一任务完成以后再进行下一个任务的模式进行，这样下一个任务的开始必须等待前一个任务的结束。Java语言提供了并发机制，允许开发人员在程序中执行多个线程，每个线程完成一个功能，并与其他线程并发执行。这种机制被称为多线程

## **<span id="12">12.说说你对线程池的了解，为什么需要线程池？你知道线程池的哪几个参数？</span>**

- 线程池

线程池是一种多线程处理形式，处理过程中将任务添加到队列，然后在创建线程后自动启动这些任务。

一个线程池包括以下四个基本组成部分：

- 1、线程池管理器（ThreadPool）：用于创建并管理线程池，包括 创建线程池，销毁线程池，添加新任务；
- 2、工作线程（PoolWorker）：线程池中线程，在没有任务时处于等待状态，可以循环的执行任务；
- 3、任务接口（Task）：每个任务必须实现的接口，以供工作线程调度任务的执行，它主要规定了任务的入口，任务执行完后的收尾工作，任务的执行状态等；
- 4、任务队列（taskQueue）：用于存放没有处理的任务。提供一种缓冲机制。

## ● 为什么需要线程池

在Java中，如果每当一个请求到达就创建一个新线程，开销是相当大的。在实际使用中，每个请求创建新线程的服务器在创建和销毁线程上花费的时间和消耗的系统资源，甚至可能要比花在处理实际的用请求的时间和资源要多得多。

除了创建和销毁线程的开销之外，活动的线程也需要消耗系统资源。如果在一个JVM里创建太多的线程，可能会导致系统由于过度消耗内存或“切换过度”而导致系统资源不足。为了防止资源不足，服务应用程序需要一些办法来限制任何给定时刻处理的请求数目，尽可能减少创建和销毁线程的次数，特是一些资源耗费比较大的线程的创建和销毁，尽量利用已有对象来进行服务，这就是“池化资源”技术产生的原因。线程池主要用来解决线程生命周期开销问题和资源不足问题。通过对多个任务重用线程，线程创建的开销就被分摊到了多个任务上了，而且由于在请求到达时线程已经存在，所以消除了线程创建所带来的延迟。这样，就可以立即为请求服务，使应用程序响应更快。另外，通过适当地调整线程中的线程数目可以防止出现资源不足的情况。

## <span id="13">13.你知道volatile吗? </span>

答：

在多线程环境下，线程可以将线程间共享的变量保存在本地内存（如寄存器）中，而不是从内存中读取，这就可能会引发不一致的问题，另一个进程可能在此线程运行期间改变了变量的值，而此线程并没看到变化。

而volatile修饰的成员变量在每次被线程访问时，都强迫从共享内存中重读该成员变量的值。而且，成员变量发生变化时，强迫线程将变化值回写到共享内存。这样在任何时刻，两个不同的线程总是看某个成员变量的同一个值。

Java语言规范中指出：为了获得最佳速度，允许线程保存共享成员变量的私有拷贝，而且只当线程进入或者离开同步代码块时才与共享成员变量的原始值对比。

也就是说，JVM会对线程变量的访问进行优化，这样当多个线程同时与某个对象交互，就必须要注意到要让线程及时的得到共享成员变量的变化。

而volatile关键字就是提示VM：对于这个成员变量不能保存它的私有拷贝，而应直接与共享成员变量交互。

使用建议：

- 1.在两个或者更多的线程访问的成员变量上使用volatile
- 2.当要访问的变量已在synchronized代码块中，或者为常量时，不必使用
- 3.volatile屏蔽掉了VM中必要的代码优化，所以在效率上比较低，因此一定在必要时才使用此关键字

## <span id="14">14.你知道synchronized吗? </span>

答：

synchronized是Java中的关键字，是一种同步锁，它一次只允许一个线程进入特定代码段，从而避免线程同时修改同一数据。它修饰的对象有以下几种：

- 修饰一个代码块，被修饰的代码块称为同步语句块，其作用的范围是大括号{}括起来的代码，作用对象是调用这个代码块的对象
- 修饰一个方法，被修饰的方法称为同步方法，其作用的范围是整个方法，作用的对象是调用这个方法



的对象

- 修改一个静态的方法，其作用的范围是整个静态方法，作用的对象是这个类的所有对象
- 修改一个类，其作用的范围是synchronized后面括号括起来的部分，作用主对象是这个类的所有对象

**<span id="15">15.现在有一张订单表有库存表，需要你写一个下单的业务逻辑来保证库存表和订单表的一致性，你会怎么写？</span>**

答：由于需要保证两件事情要么都成功要么都不成功，所以需要用到事务，这里可以把这两个操作放在同一个事务中。

**<span id="16">16.在Java中你一般怎么使用事物？</span>**

答：在Java中我一般通过@Transactional来使用事务。。

这里后面详细补充！！！！！！

**<span id="17">17.如果不使用 @Transactional，你将怎么实现事物？</span>**

这里后面详细补充！！！！！！

**<span id="18">18.简述一下JAVA用JDBC连接数据库的步骤</span>**

答：

1. 加载JDBC驱动程序,利用 `Class.forName`
2. 获取到对应数据库连接对象,利用 `DriverManager`类中`getConnection`方法
3. 通过连接对象创建\*Statement, `Statement`,`PreparedStatement`,`CallableStatement`
4. 执行SQL语句
5. 处理结果
6. 关闭资源

**<span id="19">19.preparedStatement与statement的区别</span>**

答：

Statement执行不带参数的简单SQL语句,并返回它所生成结果的对象,每次执行SQL语句时,数据库都编译该sql语句

PreparedStatement用来执行带参数的预编译的SQL语句

PreparedStatement的优点:

- 效率高.

使用PreparedStatement执行SQL命令时,命令会被数据库编译和解析,并放到命令缓冲区.以后每当执行同一个PreparedStatement对象时,预编译的命令就可以重复使用

- 代码可读性和可维护性好

- 安全性好.

使用PreparedStatement可以防止SQL注入.