安卓面试突破专题课程

**Android 基础与底层机制**

1. 数据库的操作类型有哪些，如何导入外部数据库？（Ricky）

2. 是否使用过本地广播，和全局广播有什么差别？（Ricky）

3. 是否使用过 IntentService，作用是什么， AIDL 解决了什么问题？ (小米) （Ricky）

4. Activity、 Window、 View 三者的差别， fragment 的特点？（360）（Ricky）

5. 描述一次网络请求的流程（新浪）（Jason）

一次完整的HTTP请求所经历的7个步骤

HTTP通信机制是在一次完整的HTTP通信过程中，Web浏览器与Web服务器之间将完成下列7个步骤：

1. 建立TCP连接：

在HTTP工作开始之前，Web浏览器首先要通过网络与Web服务器建立连接，该连接是通过TCP来完成的，该协议与IP协议共同构建Internet，即著名的TCP/IP协议族，因此Internet又被称作是TCP/IP网络。HTTP是比TCP更高层次的应用层协议，根据规则，只有低层协议建立之后才能进行更高层协议的连接，因此，首先要建立TCP连接，一般TCP连接的端口号是80。

2. Web浏览器向Web服务器发送请求命令：

一旦建立了TCP连接，Web浏览器就会向Web服务器发送请求命令。例如：GET/sample/hello.jsp HTTP/1.1。

3. Web浏览器发送请求头信息 ：

浏览器发送其请求命令之后，还要以头信息的形式向Web服务器发送一些别的信息，之后浏览器发送了一空白行来通知服务器，它已经结束了该头信息的发送。

4. Web服务器应答 ：

客户机向服务器发出请求后，服务器会客户机回送应答， HTTP/1.1 200 OK ，应答的第一部分是协议的版本号和应答状态码。

5. Web服务器发送应答头信息：

正如客户端会随同请求发送关于自身的信息一样，服务器也会随同应答向用户发送关于它自己的数据及被请求的文档。

6. Web服务器向浏览器发送数据：

Web服务器向浏览器发送头信息后，它会发送一个空白行来表示头信息的发送到此为结束，接着，它就以Content-Type应答头信息所描述的格式发送用户所请求的实际数据。

7. Web服务器关闭TCP连接 ：

一般情况下，一旦Web服务器向浏览器发送了请求数据，它就要关闭TCP连接，然后如果浏览器或者服务器在其头信息加入了这行代码：Connection:keep-alive；TCP连接在发送后将仍然保持打开状态，于是，浏览器可以继续通过相同的连接发送请求。保持连接节省了为每个请求建立新连接所需的时间，还节约了网络带宽。

6. Handler、 Thread 和 HandlerThread 的差别（小米）（Jason）

Handler会关联一个单独的线程和消息队列，Handler默认关联主线程，如果要在其他线程执行，可以使用HandlerThread。

HandlerThread继承于Thread，所以它本质就是个Thread。与普通Thread的差别就在于，主要的作用是建立了一个线程，并且创立了消息队列，有来自己的looper,可以让我们在自己的线程中分发和处理消息。

7. 低版本 SDK 实现高版本 api（小米）（Ricky）

8. launch mode 应用场景（百度、小米、乐视）（Ricky）

9. touch 事件传递流程（小米）（Ricky）

10. view 绘制流程（百度）（Ricky）

11. 什么情况导致内存泄漏（美团）（Ricky）

12. ANR 定位和修正（Ricky）

13. 什么情况导致 oom（乐视、美团）（Ricky）

14. Android Service 与 Activity 之间通信的几种方式（Ricky）

15. Android 各个版本 API 的区别（Ricky）

16. 如何保证一个后台服务不被杀死,比较省电的方式是什么？（百度）（Ricky）

17. Requestlayout， onlayout， onDraw， DrawChild 区别与联系（猎豹）（Ricky）

18. invalidate()和 postInvalidate() 的区别及使用（百度）（Ricky）

19. Android 动画框架实现原理（Ricky）

20. Android 为每个应用程序分配的内存大小是多少？（美团）（Ricky）

21. Android View 刷新机制（百度、美团）（Ricky）

22. LinearLayout 对比 RelativeLayout（百度）（Ricky）

23. 优化自定义 view（百度、乐视、小米）（Ricky）

24. ContentProvider（乐视）（Ricky）

25. fragment 生命周期（Ricky）

26. volley 解析（美团、乐视）（Ricky）

27. Android Glide 源码解析（Ricky）

28. Android 属性动画特性（乐视、小米）（Ricky）

29. Handler 机制及底层实现（Danny）

30. Binder 机制及底层实现（Danny）

**Java 基础**

1. 接口的意义（百度）（Jason）

1、重要性：在Java语言中， abstract class 和interface 是支持抽象类定义的两种机制。正是由于这两种机制的存在，才赋予了Java强大的 面向对象能力。

2、简单、规范性：如果一个项目比较庞大，那么就需要一个能理清所有业务的架构师来定义一些主要的接口，这些接口不仅告诉开发人员你需要实现那些业务，而且也将命名规范限制住了（防止一些开发人员随便命名导致别的程序员无法看明白）。

3、维护、拓展性：比如你要做一个画板程序，其中里面有一个面板类，主要负责绘画功能，然后你就这样定义了这个类。可是在不久将来，你突然发现这个类满足不了你了，然后你又要重新设计这个类，更糟糕是你可能要放弃这个类，那么其他地方可能有引用他，这样修改起来很麻烦。

如果你一开始定义一个接口，把绘制功能放在接口里，然后定义类时实现这个接口，然后你只要用这个接口去引用实现它的类就行了，以后要换的话只不过是引用另一个类而已，这样就达到维护、拓展的方便性。

4、安全、严密性：接口是实现软件松耦合的重要手段，它描叙了系统对外的所有服务，而不涉及任何具体的实现细节。这样就比较安全、严密一些（一般软件服务商考虑的比较多）。

2. 抽象类的意义（乐视）（Jason）

抽象类往往用来表征对问题领域进行分析、设计中得出的抽象概念，是对一系列看上去不同，但是本质上相同的具体概念的抽象。具体分析如下：

1.因为抽象类不能实例化对象，所以必须要有子类来实现它之后才能使用。这样就可以把一些具有相同属性和方法的组件进行抽象，这样更有利于代码和程序的维护。

2.当又有一个具有相似的组件产生时，只需要实现该抽象类就可以获得该抽象类的那些属性和方法。

1. 内部类的作用(百度，乐视) （Jason）

定义：放在一个类的内部的类我们就叫内部类。

作用：

1.内部类可以很好的实现隐藏，一般的非内部类，是不允许有 private 与protected权限的，但内部类可以

2.内部类拥有外围类的所有元素的访问权限

3.可是实现多重继承

4.可以避免修改接口而实现同一个类中两种同名方法的调用。

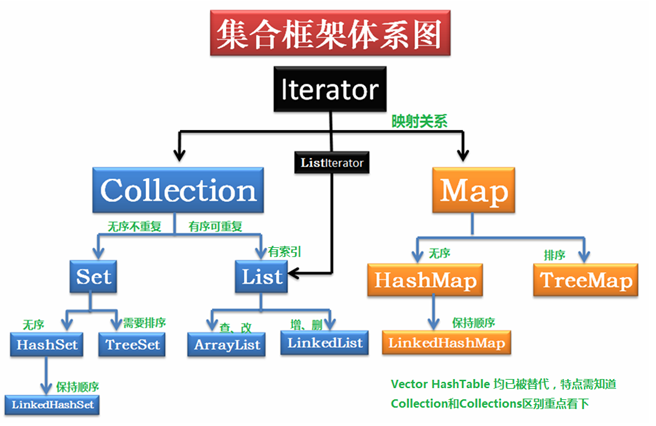
4. 父类的静态方法能否被子类重写，为什么？（猎豹）（Jason）

父类的静态方法是不能被子类重写的，其实重写只能适用于实例方法，不能用于静态方法，对于上面这种静态方法而言，我们应该称之为隐藏。

Java静态方法形式上可以重写，但从本质上来说不是Java的重写。因为静态方法只与类相关，不与具体实现相关。声明的是什么类，则引用相应类的静态方法(本来静态无需声明，可以直接引用)。并且static方法不是后期绑定的，它在编译期就绑定了。换句话说，这个方法不会进行多态的判断，只与声明的类有关。

5. 举 1-2 个排序算法，并使用 java 代码实现（美团）（Danny）

6. 列举 java 的集合和继承关系（百度、美团）（Jason）



7. java 虚拟机的特性（百度、乐视）（Jason）

1.运行时数据区域：程序计数器、java虚拟机栈、本地方法栈、java堆、方法区

2.主流对象访问方式：使用句柄和直接指针；

3垃圾收集：

4.jdk1.2后分为强引用、软引用、弱引用、虚引用；

5.类加载器：启动类加载器、扩展类加载器、应用程序类加载器

6.虚拟机字节码执行引擎（解释执行、编译执行）

7.解释器和编译器

8.即时编译器编译优化技术

8. 哪些情况下的对象会被垃圾回收机制处理掉（乐视、美团、小米）（Jason）

（1）超出对象的引用作用域时，这个对象就变成垃圾。

（2）没有超出对象的引用作用域，给这个引用赋值为空时，这个引用的对象就变成垃圾。

（3）创建匿名对象时，匿名对象用完以后即成垃圾。

9. 进程和线程的区别（猎豹）（Jason）

进程和线程都是一个时间段的描述，是CPU工作时间段的描述，不过是颗粒大小不同。

1) 简而言之,一个程序至少有一个进程,一个进程至少有一个线程.

2) 线程的划分尺度小于进程，使得多线程程序的并发性高。

3) 另外，进程在执行过程中拥有独立的内存单元，而多个线程共享内存，从而极大地提高了程序的运行效率。

4) 线程在执行过程中与进程还是有区别的。每个独立的进程有一个程序运行的入口、顺序执行序列和程序的出口。但是线程不能够独立执行，必须依存在应用程序中，由应用程序提供多个线程执行控制。

5) 从逻辑角度来看，多线程的意义在于一个应用程序中，有多个执行部分可以同时执行。但操作系统并没有将多个线程看做多个独立的应用，来实现进程的调度和管理以及资源分配。这就是进程和线程的重要区别。

11. Java 中==和 equals 的区别， equals 和 hashCode 的区别（乐视）（Jason）

==操作符用来比较两个基本类型变量时，比较的是值；用来比较引用类型变量时，比较的是引用的地址

equals方法是基类Object的方法，用于比较两个对象的内容是否相同，Object类中equals方法的默认实现使用的==操作符比较，实际开发过程中，我们会重写equals方法，重新定义比较规则

当涉及到像HashMap等与哈希表结构相关的一些类时，会使用到hashCode方法

默认的hashCode实现一般是内存地址对应的数字，所以不同的对象，hashCode（）的返回值是不一样的

equals(object)相同时，hashCode（）的返回值也要尽量相同，当equals(object)不相同时，hashCode（）的返回没有特别的要求，但是也是尽量不相同以获取好的性能

12. ArrayList 和 HashMap 的实现原理（美团，百度）（Danny）

13. java 中 int char long 各占多少字节数（Jason）

char 2个字节，int 4个字节，long 8个字节

14. java int 与 integer 的区别（Jason）

int 是基本数据类型，integer是包装数据类型（引用类型），JDK1.5之后，自动装包/拆包大大方便了基本类型数据和它们包装类地使用。

15. string stringbuffer stringbuilder 区别（小米、乐视、百度）（Jason）

共同点：String StringBuffer StringBuilder 都是字符串相关函数，StringBuffer StringBuilder都是AbstractStringBuilder的子类

不同点：

String内容不可变，StringBuffer StringBuilder内容可变

String与StringBuffer线程安全，StringBuilder非线程安全

如果程序不是多线程的，那么使用StringBuilder效率高于StringBuffer

16. Java 多态（乐视）（Jason）

静态多态：方法重载

动态多态：方法重写

多态发生的条件：

1）要有继承

2）要有重写

3）父类引用指向子类对象

多态的好处：

1. 可替换性

2）可扩充性

3）接口性

4）简化性

17. 什么导致线程阻塞（58、美团）（Jason）

阻塞状态的线程的特点是：该线程放弃CPU的使用，暂停运行，只有等到导致阻塞的原因消除之后才恢复运行。或者是被其他的线程中断，该线程也会退出阻塞状态，同时抛出InterruptedException

1）线程执行了Thread.sleep(int millsecond);方法，当前线程放弃CPU，睡眠一段时间，然后再恢复执行

2）线程执行一段同步代码，但是尚且无法获得相关的同步锁，只能进入阻塞状态，等到获取了同步锁，才能回复执行。

3）线程执行了一个对象的wait()方法，直接进入阻塞状态，等待其他线程执行notify()或者notifyAll()方法。

4）线程执行某些IO操作，因为等待相关的资源而进入了阻塞状态。比如说监听system.in，但是尚且没有收到键盘的输入，则进入阻塞状态。

18. 抽象类接口区别（360）（Jason）

1）抽象类要被子类继承，接口要被类实现

2）接口只能做方法申明，抽象类中可以做方法申明，也可以做方法实现

3）接口里定义的变量只能是公共的静态的常量，抽象类中的变量是普通变量。

4）抽象类里的抽象方法必须全部被子类所实现，如果子类不能全部实现父类抽象方法，那么该子类只能是抽象类。同样，一个实现接口的时候，如不能全部实现接口方法，那么该类也只能为抽象类。

5）抽象方法只能申明，不能实现，接口是设计的结果 ，抽象类是重构的结果

6）接口可继承接口，并可多继承接口，但类只能单根继承。

19. 容器类之间的区别（乐视、美团）（Jason）

1）Vector和ArrayList

vector是线程同步的，所以它也是线程安全的，而arraylist是线程异步的，是不安全的。如果不考虑到线程的安全因素，一般用arraylist效率比较高。

如果集合中的元素的数目大于目前集合数组的长度时，vector增长率为目前数组长度的100%,而arraylist增长率为目前数组长度的50%.如过在集合中使用数据量比较大的数据，用vector有一定的优势。

ArrayList 和Vector是采用数组方式存储数据，此数组元素数大于实际存储的数据以便增加和插入元素，都允许直接序号索引元素，但是插入数据要设计到数组元素移动等内存操作，所以索引数据快插入数据慢，Vector由于使synchronized方法（线程安全）所以性能上比ArrayList要差

2）arraylist和linkedlist

1.ArrayList是实现了基于动态数组的数据结构，LinkedList基于链表的数据结构。

2.对于随机访问get和set，ArrayList觉得优于LinkedList，因为LinkedList要移动指针。

3.对于新增和删除操作add和remove，LinkedList比较占优势，因为ArrayList要移动数据。这一点要看实际情况的。若只对单条数据插入或删除，ArrayList的速度反而优于LinkedList。但若是批量随机的插入删除数据，LinkedList的速度大大优于ArrayList. 因为ArrayList每插入一条数据，要移动插入点及之后的所有数据。

3）HashMap与TreeMap

1、HashMap通过hashcode对其内容进行快速查找，而TreeMap中所有的元素都保持着某种固定的顺序，如果你需要得到一个有序的结果你就应该使用TreeMap（HashMap中元素的排列顺序是不固定的）。集合框架”提供两种常规的Map实现：HashMap和TreeMap (TreeMap实现SortedMap接口)。

2、在Map 中插入、删除和定位元素，HashMap 是最好的选择。但如果您要按自然顺序或自定义顺序遍历键，那么TreeMap会更好。使用HashMap要求添加的键类明确定义了hashCode()和 equals()的实现。

4)、hashtable与hashmap

1.历史原因:Hashtable是基于陈旧的Dictionary类的，HashMap是Java 1.2引进的Map接口的一个实现

2.同步性:Hashtable是线程安全的，也就是说是同步的，而HashMap是线程序不安全的，不是同步的

3.值：只有HashMap可以让你将空值作为一个表的条目的key或value

5)、Set与List

Set中的数据对象没有顺序且不可以重复。

List中的数据对象有顺序且可以重复。

20. Java 中 HashMap 和 HashTable 的区别（乐视、小米）（Danny）

21. ArrayMap VS HashMap（Danny）

**数据结构与算法**

1. 堆和栈在内存中的区别是什么(数据结构方面以及实际实现方面) （Danny）

2. 最快的排序算法是哪个？给阿里 2 万多名员工按年龄排序应该选择哪个算法？堆和树

的区别；写出快排代码；链表逆序代码（阿里）（Danny）

3. 求 1000 以内的水仙花数以及 40 亿以内的水仙花数（百度）（Danny）

4. 子串包含问题(KMP 算法)写代码实现（Danny）

5. 万亿级别的两个 URL 文件 A 和 B，如何求出 A 和 B 的差集 C,(Bit 映射->hash 分组->多文（Danny）

件读写效率->磁盘寻址以及应用层面对寻址的优化) （Danny）

6. 蚁群算法与蒙特卡洛算法（Danny）

7. 写出你所知道的排序算法及时空复杂度，稳定性（小米）（Danny）

**其他**

1. 死锁的四个必要条件（Jason）

1）互斥条件，即某个资源在一段时间内只能由一个线程占有，不能同时被两个或两个以上的线程占有

2）不可抢占条件，线程所获得的资源在未使用完毕之前，资源申请者不能强行地从资源占有者手中夺取资源，而只能由该资源的占有者线程自行释放

3）占有且申请条件，线程至少已经占有一个资源，但又申请新的资源；由于该资源已被另外线程占有，此时该线程阻塞；但是，它在等待新资源之时，仍继续占用已占有的资源。

4）循环等待条件，存在一个线程等待序列{P1，P2，...，Pn}，其中P1等待P2所占有的某一资源，P2等待P3所占有的某一源，......，而Pn等待P1所占有的的某一资源，形成一个线程循环等待环

解决死锁的办法：加锁顺序，死锁检测

2. 常见编码方式； utf-8 编码中的中文占几个字节；数字几个字节（Jason）

一个utf8数字占1个字节，一个utf8英文字母占1个字节，少数是汉字每个占用3个字节，多数占用4个字节。

1. 实现一个 Json 解析器(可以通过正则提高速度) （Jason）

String json = "{name:\"jason\",father:\"jason\",age:18}";

//name:"jason"

//age:18

//\"\\w+\" 字符串属性

Pattern p = Pattern.compile("\\w+:(\"\\w+\"|\\d\*)");

Matcher m = p.matcher(json);

while(m.find()){

String text = m.group();

int dotPos= text.indexOf(":");

String key = text.substring(0, dotPos);

String value = text.substring(dotPos+1, text.length());

//替换字符串的开始结束的双引号

value = value.replaceAll("^\\\"|\\\"$", "");

System.out.println(key);

System.out.println(value);

}

4. Android App 的设计架构： MVC,MVP,MVVM 与架构经验谈（搜狐）（Jason）

5. 写出观察者模式的代码（Jason）

触发联动

6. TCP 的 3 次握手和四次挥手； TCP 与 UDP 的区别（Jason）

建立TCP需要三次握手才能建立，而断开连接则需要四次握手。



三次握手：

TCP是面向连接的，无论哪一方向另一方发送数据之前，都必须先在双方之间建立一条连接。在TCP/IP协议中，TCP协议提供可靠的连接服务，连接是通过三次握手进行初始化的。三次握手的目的是同步连接双方的序列号和确认号并交换 TCP窗口大小信息。

第一次握手：建立连接。客户端发送连接请求报文段，将SYN位置为1，Sequence Number为x；然后，客户端进入SYN\_SEND状态，等待服务器的确认；

第二次握手：服务器收到SYN报文段。服务器收到客户端的SYN报文段，需要对这个SYN报文段进行确认，设置Acknowledgment Number为x+1(Sequence Number+1)；同时，自己自己还要发送SYN请求信息，将SYN位置为1，Sequence Number为y；服务器端将上述所有信息放到一个报文段（即SYN+ACK报文段）中，一并发送给客户端，此时服务器进入SYN\_RECV状态；

第三次握手：客户端收到服务器的SYN+ACK报文段。然后将Acknowledgment Number设置为y+1，向服务器发送ACK报文段，这个报文段发送完毕以后，客户端和服务器端都进入ESTABLISHED状态，完成TCP三次握手。

四次分手：

当客户端和服务器通过三次握手建立了TCP连接以后，当数据传送完毕，肯定是要断开TCP连接的啊。那对于TCP的断开连接，这里就有了神秘的“四次分手”。

第一次分手：主机1（可以使客户端，也可以是服务器端），设置Sequence Number和Acknowledgment Number，向主机2发送一个FIN报文段；此时，主机1进入FIN\_WAIT\_1状态；这表示主机1没有数据要发送给主机2了；

第二次分手：主机2收到了主机1发送的FIN报文段，向主机1回一个ACK报文段，Acknowledgment Number为Sequence Number加1；主机1进入FIN\_WAIT\_2状态；主机2告诉主机1，我“同意”你的关闭请求；

第三次分手：主机2向主机1发送FIN报文段，请求关闭连接，同时主机2进入LAST\_ACK状态；

第四次分手：主机1收到主机2发送的FIN报文段，向主机2发送ACK报文段，然后主机1进入TIME\_WAIT状态；主机2收到主机1的ACK报文段以后，就关闭连接；此时，主机1等待2MSL后依然没有收到回复，则证明Server端已正常关闭，那好，主机1也可以关闭连接了。

1、TCP面向连接（如打电话要先拨号建立连接）;UDP是无连接的，即发送数据之前不需要建立连接

2、TCP提供可靠的服务。也就是说，通过TCP连接传送的数据，无差错，不丢失，不重复，且按序到达;UDP尽最大努力交付，即不保证可靠交付

4、每一条TCP连接只能是点到点的;UDP支持一对一，一对多，多对一和多对多的交互通信

5、TCP首部开销20字节;UDP的首部开销小，只有8个字节

6、TCP的逻辑通信信道是全双工的可靠信道，UDP则是不可靠信道

7. HTTP 协议； HTTP 报文结构；HTTP1.0 与 2.0 的区别；（Jason）

**HTTP协议**，即超文本传输协议(Hypertext transfer protocol)。是一种详细规定了浏览器和万维网(WWW = World Wide Web)服务器之间互相通信的规则，通过因特网传送万维网文档的数据传送协议。

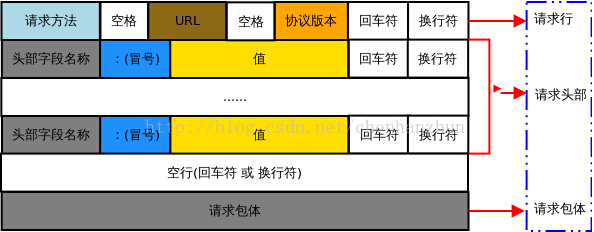
HTTP协议是用于从WWW服务器传输超文本到本地浏览器的传送协议。它可以使浏览器更加高效，使网络传输减少。它不仅保证计算机正确快速地传输超文本文档，还确定传输文档中的哪一部分，以及哪部分内容首先显示(如文本先于图形)等。

HTTP是一个应用层协议，由请求和响应构成，是一个标准的客户端服务器模型。HTTP是一个无状态的协议。

**HTTP 报文结构**

HTTP 请求报文

HTTP 请求报文由请求行、请求头部、空行 和 请求包体 4 个部分组成，如下图所示：



GET /search?hl=zh-CN&source=hp&q=domety&aq=f&oq= HTTP/1.1

Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel, application/vnd.ms-powerpoint,

application/msword, application/x-silverlight, application/x-shockwave-flash, \*/\*

Referer: <a href="http://www.google.cn/">http://www.google.cn/</a>

Accept-Language: zh-cn

Accept-Encoding: gzip, deflate

User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727; TheWorld)

Host: <a href="http://www.google.cn">www.google.cn</a>

Connection: Keep-Alive

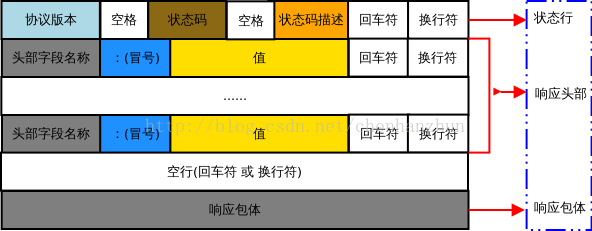
Cookie: PREF=ID=80a06da87be9ae3c:U=f7167333e2c3b714:NW=1:TM=1261551909:LM=1261551917:S=ybYcq2wpfefs4V9g;

NID=31=ojj8d-IygaEtSxLgaJmqSjVhCspkviJrB6omjamNrSm8lZhKy\_yMfO2M4QMRKcH1g0iQv9u-2hfBW7bUFwVh7pGaRUb0RnHcJU37y-

FxlRugatx63JLv7CWMD6UB\_O\_r

HTTP 响应报文

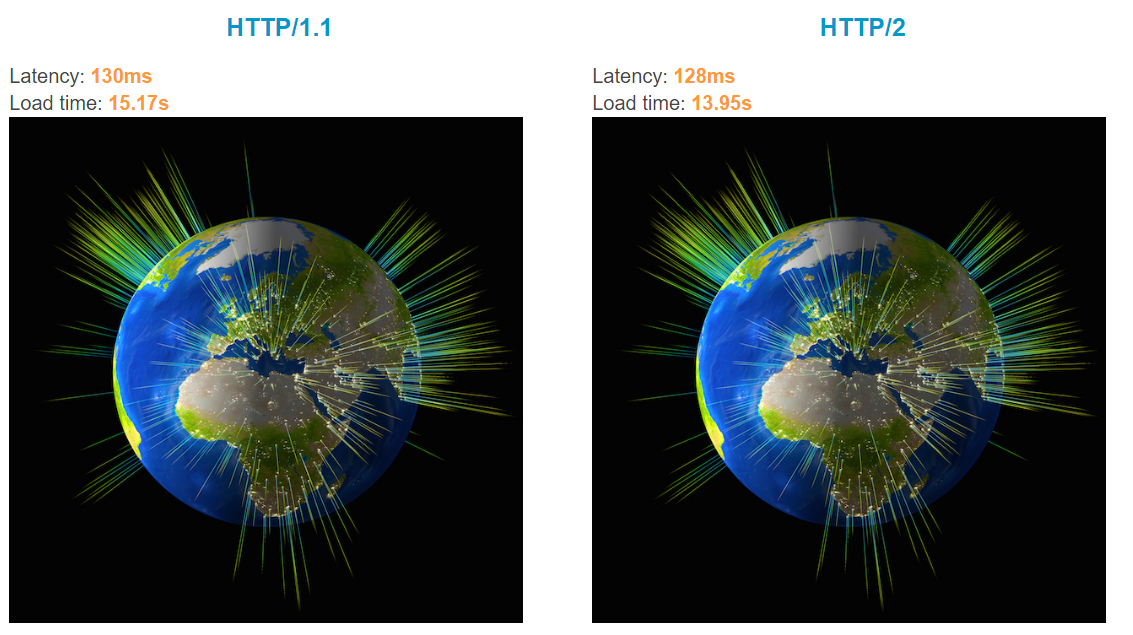
HTTP 响应报文由状态行、响应头部、空行 和 响应包体 4 个部分组成，如下图所示：



**HTTP1.0 与 2.0 的区别**

HTTP 2.0 的出现，相比于 HTTP 1.x ，大幅度的提升了 web 性能。在与 HTTP/1.1 完全语义兼容的基础上，进一步减少了网络延迟。

HTTP/2: the Future of the Internet （https://http2.akamai.com/demo）这是 Akamai 公司建立的一个官方的演示，用以说明 HTTP/2 相比于之前的 HTTP/1.1 在性能上的大幅度提升。 同时请求 379 张图片，从Load time 的对比可以看出 HTTP/2 在速度上的优势。

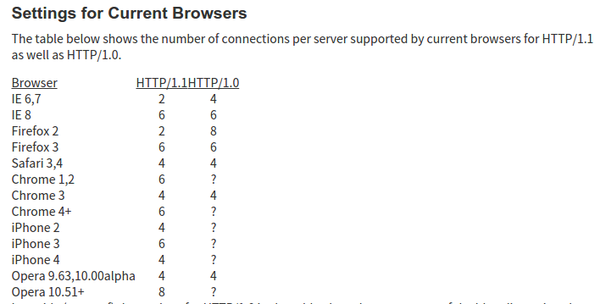


**多路复用 (Multiplexing)**

多路复用允许同时通过单一的 HTTP/2 连接发起多重的请求-响应消息。

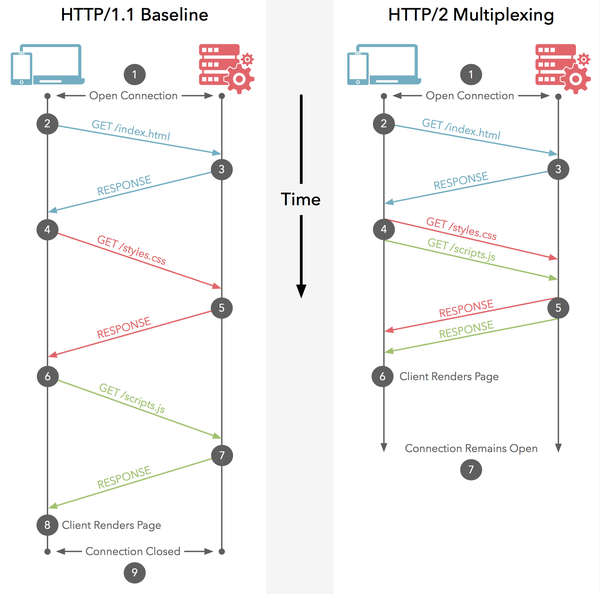
众所周知 ，在 HTTP/1.1 协议中 「浏览器客户端在同一时间，针对同一域名下的请求有一定数量限制。超过限制数目的请求会被阻塞」。

该图总结了不同浏览器对该限制的数目。



这也是为何一些站点会有多个静态资源 CDN 域名的原因之一，拿 Twitter 为例，http://twimg.com，目的就是变相的解决浏览器针对同一域名的请求限制阻塞问题。

而 HTTP/2 的多路复用(Multiplexing) 则允许同时通过单一的 HTTP/2 连接发起多重的请求-响应消息。

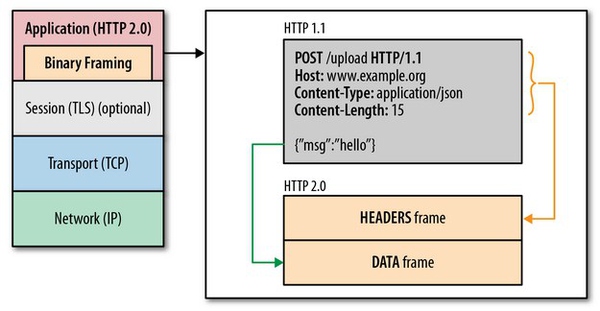


因此 HTTP/2 可以很容易的去实现多流并行而不用依赖建立多个 TCP 连接，HTTP/2 把 HTTP 协议通信的基本单位缩小为一个一个的帧，这些帧对应着逻辑流中的消息。并行地在同一个 TCP 连接上双向交换消息。

**二进制分帧**

在不改动 HTTP/1.x 的语义、方法、状态码、URI 以及首部字段….. 的情况下, HTTP/2 是如何做到「突破 HTTP1.1 的性能限制，改进传输性能，实现低延迟和高吞吐量」的 ?

关键之一就是在 应用层(HTTP/2)和传输层(TCP or UDP)之间增加一个二进制分帧层。



在二进制分帧层中， HTTP/2 会将所有传输的信息分割为更小的消息和帧（frame）,并对它们采用二进制格式的编码 ，其中 HTTP1.x 的首部信息会被封装到 HEADER frame，而相应的 Request Body 则封装到 DATA frame 里面。

HTTP/2 通信都在一个连接上完成，这个连接可以承载任意数量的双向数据流。

在过去， HTTP 性能优化的关键并不在于高带宽，而是低延迟。TCP 连接会随着时间进行自我「调谐」，起初会限制连接的最大速度，如果数据成功传输，会随着时间的推移提高传输的速度。这种调谐则被称为 TCP 慢启动。由于这种原因，让原本就具有突发性和短时性的 HTTP 连接变的十分低效。

HTTP/2 通过让所有数据流共用同一个连接，可以更有效地使用 TCP 连接，让高带宽也能真正的服务于 HTTP 的性能提升。

总结：

单连接多资源的方式，减少服务端的链接压力,内存占用更少,连接吞吐量更大，由于 TCP 连接的减少而使网络拥塞状况得以改善，同时慢启动时间的减少,使拥塞和丢包恢复速度更快

**服务端推送（Server Push）**

服务端推送是一种在客户端请求之前发送数据的机制。在 HTTP/2 中，服务器可以对客户端的一个请求发送多个响应。Server Push 让 HTTP1.x 时代使用内嵌资源的优化手段变得没有意义；如果一个请求是由你的主页发起的，服务器很可能会响应主页内容、logo 以及样式表，因为它知道客户端会用到这些东西。这相当于在一个 HTML 文档内集合了所有的资源，不过与之相比，服务器推送还有一个很大的优势：可以缓存！也让在遵循同源的情况下，不同页面之间可以共享缓存资源成为可能。

8. HTTP 与 HTTPS 的区别以及如何实现安全性（Jason）

1.HTTPS加密传输协议HTTP名文传输协议;

2.HTTPS需要用SSL证书HTTP用;

3.HTTPS比HTTP更加安全搜索引擎更友;

4.HTTPS标准端口443HTTP标准端口80;

5.HTTPS基于传输层HTTP基于应用层;

6.HTTPS浏览器显示绿色安全锁HTTP没显示;

HTTPS协议是标准的HTTP协议架在SSL/TLS协议之上的一种结构，HTTPS实现安全性的几个主要机制：

1. 证书：通过第三方权威证书颁发机构（如VeriSign）验证和担保网站的身份，防止他人伪造网站身份与不知情的用户建立加密连接。

2. 密钥交换：通过公钥（非对称）加密在网站服务器和用户之间协商生成一个共同的会话密钥。

3. 会话加密：通过机制(2)协商的会话密钥，用对称加密算法对会话的内容进行加密。

4. 消息校验：通过消息校验算法来防止加密信息在传输过程中被篡改。