**面试**

# 1 Http

## 1.1http协议介绍

http是一个基于请求与响应模式的、无状态、应用层的协议，基于TCP的连接方式。特点：无连接，限制每次连接只处理一个请求，节省传输时间；无状态，无状态是指协议对于事务处理没有记忆能力，即如果后续处理需要前面的信息，必须重传。

* *我们在传输数据时，可以只使用(传输层)TCP/IP协议，但是那样的话，如果没有应用层，便无法识别数据内容。如果想要使传输的数据有意义，则必须使用到应用层协议。应用层协议有很多，比如HTTP、FTP、TELNET等，也可以自己定义应用层协议。WEB使用HTTP协议作应用层协议，以封装HTTP文本信息，然后使用TCP/IP做传输层协议将它发到网络上*
* *socket是对TCP/IP协议的封装，Socket本身并不是协议，而是一个调用接口(API)；TCP/IP只是一个协议栈，就像操作系统的运行机制一样，必须要具体实现，同时还要提供对外的操作接口。*

*这个就像操作系统会提供标准的编程接口，比如win32编程接口一样，TCP/IP也要提供可供程序员做网络开发所用的接口，这就是Socket编程接口。*

* *传输层的TCP是基于网络层的IP协议的，而应用层的HTTP协议又是基于传输层的TCP协议的，而Socket本身不算是协议，它只是提供了一个针对TCP或者UDP编程的接口。*
* *TCP的三次握手：****第一次握手****：客户端发送syn包(syn=j)到服务器，并进入SYN\_SEND状态，等待服务器确认;****第二次握手****：服务器收到syn包，必须确认客户的SYN(ack=j+1)，同时自己也发送一个SYN包(syn=k)，即SYN+ACK包，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包，向服务器发送确认包ACK(ack=k+1)，此包发送完毕，客户端和服务器进入ESTABLISHED状态，完成三次握手。握手过程中传送的包里不包含数据，三次握手完毕后，客户端与服务器才正式开始传送数据。*

## 1.2 http请求

* http请求由三部分组成：请求行、消息报头、请求正文
* 请求方法：
  + get：请求获取request-URL所标识的资源
  + post：在request-URL所标识的资源后附加新的数据
  + head：获取资源响应消息报头，方法常用于测试超链接的有效性，是否可以访问，以及最近是否更新。
  + put：请求服务器存储一个资源，并用request-URL作为其标识
  + delete：请求服务器删除request-URL所标识的资源
  + trace：请求服务器回送收到的请求信息，主要用于测试或诊断
  + options：请求查询服务器的性能，或者查询与资源相关的选项和需求，返回服务器支持的请求方法
* header内容：
  + Accept:image/gif/text/html
  + Content-length:22
  + Connection:keep-alive
  + Cache-Control:no-cache

## 1.3 http响应

* 响应状态码：
  + 1xx : 指示信息，表示请求已接收，继续处理
  + 2xx： 成功，表示请求已被成功接收、理解、接受
  + 3xx： 重定向，要完成请求必须进行进一步的操作；301-moved permanently；302-
  + 4xx：客户端错误，请求语法错误或请求无法实现；400-bad request；401-Unauthorized；403-Forbidden；404-Not Found；
  + 5xx：服务器错误，服务器未能实现合法的请求；500-Internal Server Error；502-bad gateway；504-gateway timeout

## 1.4其他（网关代理通道）

* **代理(Proxy)**：一个中间程序，它可以充当一个服务器，也可以充当一个客户机，为其它客户机建立请求。请求是通过可能的翻译在内部或经过传递到其它的 服务器中。一个代理在发送请求信息之前，必须解释并且如果可能重写它。代理经常作为通过防火墙的客户机端的门户，代理还可以作为一个帮助应用来通过协议处 理没有被用户代理完成的请求。
* **网关(Gateway)**：一个作为其它服务器中间媒介的服务器。与代理不同的是，网关接受请求就好象对被请求的资源来说它就是源服务器；发出请求的客户机并没有意识到它在同网关打交道。网关经常作为通过防火墙的服务器端的门户，网关还可以作为一个协议翻译器以便存取那些存储在非HTTP系统中的资源。
* **通道(Tunnel)**：是作为两个连接中继的中介程序。一旦激活，通道便被认为不属于HTTP通讯，尽管通道可能是被一个HTTP请求初始化的。当被中继 的连接两端关闭时，通道便消失。当一个门户(Portal)必须存在或中介(Intermediary)不能解释中继的通讯时通道被经常使用。

## 1.5 缓存

### 直接本地缓存

直接使用浏览器端的Cache，不会发出真正的request，直到Cache过期；最省流量和服务器资源，但可能存在数据不一致，适合实时性不高的场景。

Header信息：Cache-Control: public, max-age=31536000

* public : 用户浏览器和中间proxy都会cache
* private : 只有用户端会cache
* max-age=xxx : 设置cache的最大存活时间，单位s
* no-store: 不要缓存
* no-cache: 同上，但是浏览器的支持可能不一致，最好两个一起用
* Expires: Mon, 19 Nov 2012 08:40:01 GMT   
  指定cache的*绝对*过期时间，和Cache-Control一起使用时以后者为准

### 304校验缓存

将浏览器端的Cache信息发送到服务器校验，如果服务器认为依然有效，则返回304，浏览器使用Cache；会发出一次请求且需要服务器校验，没有第一种方式那么有效，但可以保证数据实时性。

校验的两种方式：

* 基于内容ETag，ETag是Entity Tag的缩写，它由服务器端指定，通常是一个资源的内容的hash。

服务器返回给浏览器：Etag: 427fe7b6442f2096dff4f92339305444

浏览器请求传回给服务器：If-None-Match: 427fe7b6442f2096dff4f92339305444

服务器收到请求将两个值对比，若不一致，返回新数据，否则使用缓存

* 基于最后修改时间：Last-Modified

Request将本地缓存的最后修改时间通过 If-Modified-Since 发送到服务器

## 1.6 cookie与session区别

<https://www.zhihu.com/question/19786827>

sessionID是服务器发送给客户端的用于标识同一用户，比如购物车、结算类似需要验证是哪一个用户对服务器的请求，sessionID在服务器可存入内存、数据库、文件等等，客户端则会存入cookie中，在每次请求中都会把sessionID发送到服务器，若客户端禁用了cookie功能，则会通过一种重写URL的技术来识别用户（在每次请求的URL后追加sid=xxx的参数）。Cookie可以记录用户的信息（如登录账号密码，在下次访问同一网页时自动填入账号密码）

* session在服务器，cookie在客户端
* session默认被存在服务器的一个文件里（不是内存）
* session的运行依赖sessionID 而sessionID是存在cookie中的，也就是说，如果浏览器禁用了cookie 同时session也会失效（但可通过在URL中传递sessionID的方式实现）
* session可放在文件、数据库、或内存中
* 用户验证这种场合一般会用session

# linux命令

将当前目录下的文件a变成隐藏文件：

mv a .a

# 提高产品质量

## 产品需求设计的合理性

失败的产品设计上线后会经常性的变动需求，导致整个团队频繁的修改和测试，尤其在上线前还在修改产品逻辑的场景。需要PM前期做很充分的需求调研。

## 技术架构设计，逻辑实现要合理

技术框架设计不合理，导致在线上用户量达到量级时，爆发很严重的问题。

## 开发编码规范

开发对变量定义不规范、对输入输出参数的不规范、对异常处理的不规范、对日志处理的不规范，导致出现类似空指针、数组越界、崩溃这样低级的bug而且很难找到引起bug的原因。

## 需求评审

应起到实际的评审目的，提前将评审内容发给团队相关人，提前查阅、记录问题，带着问题去评审，效率更高、效果也会更好。

## 测试流程的规范

根据公司自己的实际情况制定，参考流程：测试需求分析、case编写、case评审、接口测试、冒烟测试、正式4轮测试、beta测试、monkey测试、提交testin兼容众测、线上环境回归测试、发布版本后安装卸载升级流程测试。

## 开发流程规范

开发代码的管理同步，代码分支开发、合并主干，提交代码review，SQL上线进行必要的review

## 上线流程规范

管理提交代码，确保分支没问题的代码才能合并到主干，再做主干整体回归测试。

## 优化功能测试的范围界定

考虑新功能对其他模块的影响，对老版本的兼容，每一次上线一定要对高频功能、核心功能做最充分的回归测试。

## 静态代码扫描

Checkstyle、FindBugs、PMD、Jtest、php depend、PHPMD等，仅通过分析或检查源程序的语法、结构、过程、接口等来检查程序的正确性，找出代码隐藏的错误和缺陷，如参数不匹配，有歧义的嵌套语句，错误的递归，非法计算，可能出现的空指针引用、错误的变量定义等等。使用静态代码分析工具自动化执行代码检查和分析，能够极大地提高软件可靠性并节省软件开发和测试成本。

## 接口测试、单元测试

一般情况下后端完成接口开发，就可以提前提测，尽早的暴露隐含的问题，让开发及时解决，方便后面的功能测试顺利进行，可缩短产品的研发周期，提高效率。

## UI走查

最好在冒烟测试时就走查一遍，test环境测试完毕走查一遍，确保不因后期修改bug带来UI问题。

## Bug预防

对bug进行分析，找出真正原因，及对应的预防避免措施，提醒测试、开发人员注意。

## 运维监控

## 复盘总结

每次项目结束进行及时的复盘总结，针对项目中出现的问题，及时做出调整，避免犯同样的错误。

## 沟通机制

产品与开发与测试之间的信息同步