socket

socket 是通信的基石。支持 TCP/IP 等协议的基本操作单元。

应用层通过传输层进行数据通信时，TCP 会遇到同时为多个应用程序进程提供并发服务的问题。多个 TCP 连接或多个应用程序进程可能需要通过同一个 TCP 协议端口传输数据。为了区别不同的应用程序进程和连接，许多计算机操作系统为应用程序与 TCP / IP 协议交互提供了套接字 (Socket) 接口。应用层可以和传输层通过 Socket 接口，区分来自不同应用程序进程或网络连接的通信，实现数据传输的并发服务。

Session 与 Cookie 的对比

HTTP 是一种不保存状态，即无状态（stateless）协议。也就是说 HTTP 协议自身不对请求和响应之间的通信状态进行保存。那么我们保存用户状态呢？Session 机制的存在就是为了解决这个问题，Session 的主要作用就是通过服务端记录用户的状态。

实现机制：Session 的实现常常依赖于 Cookie 机制，通过 Cookie 机制回传 SessionID；
大小限制：Cookie 有大小限制并且浏览器对每个站点也有 cookie 的个数限制，Session 没有大小限制，理论上只与服务器的内存大小有关；
安全性：Cookie 存在安全隐患，通过拦截或本地文件找得到 cookie 后可以进行攻击，而 Session 由于保存在服务器端，相对更加安全；
服务器资源消耗：Session 是保存在服务器端上会存在一段时间才会消失，如果 session 过多会增加服务器的压力。
Application（ServletContext）：与一个 Web 应用程序相对应，为应用程序提供了一个全局的状态，所有客户都可以使用该状态。
交换机和路由器分别的实现原理是什么？分别在哪个层次上面实现的？
交换机用于局域网，利用主机的 MAC 地址进行数据传输，而不需要关心 IP 数据包中的 IP 地址，它工作于数据链路层。路由器识别网络是通过 IP 数据包中 IP 地址的网络号进行的，所以为了保证数据包路由的正确性，每个网络都必须有一个唯一的网络号。路由器通过 IP 数据包的 IP 地址进行路由的（将数据包递交给哪个下一跳路由器）。路由器工作于网络层。由于设备现在的发展，现在很多设备既具有交换又具有路由功能，两者的界限越来越模糊。

从输入网址到获得页面的过程

浏览器查询 DNS，获取域名对应的 IP 地址：具体过程包括浏览器搜索自身的 DNS 缓存、搜索操作系统的 DNS 缓存、读取本地的 Host 文件和向本地 DNS 服务器进行查询等。对于向本地 DNS 服务器进行查询，如果要查询的域名包含在本地配置区域资源中，则返回解析结果给客户机，完成域名解析 (此解析具有权威性)；如果要查询的域名不由本地 DNS 服务器区域解析，但该服务器已缓存了此网址映射关系，则调用这个 IP 地址映射，完成域名解析（此解析不具有权威性）。如果本地域名服务器并未缓存该网址映射关系，那么将根据其设置发起递归查询或者迭代查询；

浏览器获得域名对应的 IP 地址以后，浏览器向服务器请求建立链接，发起三次握手；

TCP/IP 链接建立起来后，浏览器向服务器发送 HTTP 请求；

服务器接收到这个请求，并根据路径参数映射到特定的请求处理器进行处理，并将处理结果及相应的视图返回给浏览器；

浏览器解析并渲染视图，若遇到对 js 文件、css 文件及图片等静态资源的引用，则重复上述步骤并向服务器请求这些资源；

浏览器根据其请求到的资源、数据渲染页面，最终向用户呈现一个完整的页面。

总结：ip——tcp——http——指定 url——资源——渲染页面展示页面

URI 和 URL 的区别是什么？

URI (Uniform Resource Identifier) 是统一资源标志符，可以唯一标识一个资源。
URL (Uniform Resource Location) 是统一资源定位符，可以提供该资源的路径。它是一种具体的 URI，即 URL 可以用来标识一个资源，而且还指明了如何 locate 这个资源。
URI 的作用像身份证号一样，URL 的作用更像家庭住址一样。URL 是一种具体的 URI，它不仅唯一标识资源，而且还提供了定位该资源的信息。

状态码

	类别	原因短语
1XX	Informational（信息性状态码）	接收的请求正在处理
2XX	Success（成功状态码）	请求正常处理完毕
3XX	Redirection（重定向状态码）	需要进行附加操作以完成请求
4XX	Client Error（客户端错误状态码）	服务器无法处理请求
5XX	Server Error（服务器错误状态码）	服务器处理请求出错