**web服务器**

* **定义：实现提供资源或应答的提供者都可以谓之为服务器！**
* **从不同形式划分，服务器有以下几种：**
  1. **Apache tomcat和Microsoft IIS，它们是一种应用服务器**
  2. **购买的服务器**
  3. 嵌入式服务器

#### web服务器应该做些什么

1. **接受建立连接请求**
2. **接受请求**
3. **处理请求**
4. **访问报文中指定的资源**
5. **构建响应**
6. **发送响应**
7. **记录事务处理过程**

**第一步————接受客户端连接**

* 客户端收到一条连接之后，那么它将会把新连接添加到现存web服务器连接列表中，用于监视当前连接上的数据传输情况。期间服务器还应该做到通过一定的设备机制阻止未认证或已知恶意黑名客户端的连接，相关设别技术有：客户端主机名设别、通过ident设别客户端用户等！

第二步————接收请求报文

1. 解析请求行，得知方法、url、协议版本，以及crlf符
2. 解析得到以crlf结尾的首部
3. 得到以crlf结尾，标志首部结束的空行（如果有的话）
4. 解析得到主体，（如果有的话）

* web服务可能还会把请求报文用一种自己能快速处理的内部数据结构来存储请求报文！

**不同的服务器配置：**

1. **单线程web服务器：只能处理一个请求，待当前请求处理完成之后才能处理下一个请求**

**优点：简单已于实现，适用于低负荷服务器。**

**缺点：不能及时处理其他请求，容易引发延迟过长而导致性能问题。**

1. **多线程及多进程web服务器：能同时处理多个请求**

**优点：响应及时**

**缺点：构建复杂，容易快速引起内存消耗过大而死机,最好应该对能同时处理的连接数量进行限制！**

1. 复用i/o的web服务器：复用i/o
2. 复用i/o和多线程的web服务器：2和3的结合

#### 第四步————对资源的映射及访问

* 找到客户端请求资源在服务器的上的目录路径！相关概念有：docroot(文档根目录)、不允许访问根目录的上一级目录。
* 虚拟托管的docroot:在一个服务器上挂了几个web站点，那么这样当请求的资源路径相同时，服务器应该从请求报文首部的host、uri字段找出真正的资源目录，这些目录都是可以配置的！

#### 第五步————构建响应

* 构建响应报文：

**1、正确设置响应主体的长度（content-length）；**

**2、设置报文的mime类型（content-type）**

3、控制重定向