为什么要把HTTP协议设计成无状态的？

把HTTP设计成无状态的可以使得每个HTTP请求是独立的。即使同一个keep-alive的连接里的请求，都不知道前一个请求的状态，这简化了HTTP协议的实现，不需要状态机，而且每一次请求都不会占用过多连接，超过keep-alive，就断开连接。

什么是Keep-Alive？

最初，HTTP协议设计出来是无连接的，即一个客户端发起一个请求，建立连接，服务的处理请求响应，客户端接收响应，马上断开连接。计算机科学家这样设计的原因是服务器在互联网中，面对成千上万太客户端发访问。每个客户端与服务器之间交换数据具有突发性和瞬时性。网页浏览具有联想性、发散性导致两次传输数据的关联性很低。如果设置成有连接的，会导致大部分通道处于空闲状态。

后来，随着网页变得越来越复杂，里面嵌入了很多图片，每次访问图片都需要进行TCP连接就显得很低效，所以Keep-Alive被提出来。Keep-Alive并不能解决HTTP协议是无状态的这一事实，即使处于同一个Keep-Alive里的连接相互之间都是无关的。

什么是servlet？

let：是little，后缀，表示“小”的意思

serv：serve 服务，服务器

servlet就是小型服务器

一个servlet可以处理多个请求

浏览器请求页面的流程？

1. 建立TCP连接（因为HTTP协议是建立在TCP协议之上的）
2. 客户端发送HTTP请求
3. 服务端响应HTTP请求，客户端接收HTTP响应

HTTP请求格式？

GET / HTTP/1.1

Host:发送请求的主机ip地址

User-Agent:表示客户端本身

Accept：表示客户端能接收的资源类型。例如text/\*，表述可以接收文本媒体，具体的文件后缀名是\*，表示任意，任意后缀的文本类型文件。

大类型/小类型

|  |  |
| --- | --- |
| 大类型 | 小类型 |
| text | html, plain, css, … |
| image | jpeg, png, gif, … |
| … | … |

HTTP响应格式？

HTTP/1.1 200 OK

Content-Type: text/html

Content-Length: 21932

Content-Encoding: gzip

Cache-Control: max-age=300

<html>…网页数据…



如何编写一个HTTP服务器？

HTTP协议是基于TCP协议的，要想实现一个HTTP服务器，需要编写基于多线程的TCP服务，TCP的实现细节，HTTP服务器不需要理解。在一个TCP中获取请求，如果请求是以GET / HTTP/1.1这种特点类型的格式开始的，那么就是http请求。对HTTP请求进行解析，发送响应即可。

在JavaEE中，处理TCP连接，解析HTTP协议就不给程序员做了，因为这些工作都是重复的且很繁琐，servlet就是干这个的，serv指server，即服务器，这里是指HTTP服务器，一个servlet就是一个小型的HTTP服务。一个servlet就可以实现处理TCP请求，解析HTTP协议，响应…功能

聊一聊servlet？

servlet = serv + let = server + little；

这里的server指HTTP 服务器

所以

servlet = 小型HTTP服务器。

所以，一个servlet类总是继承于HTTPServlet，然后重写doGet和doPost方法。

小型服务器不能像Java程序一样直接运行，而是放到一个web服务器，由web服务器加载“小型服务器”来运行

jar和war的区别？

jar是Java Archive文件，Java归档文件。包含java类文件和这些类相关的元数据和资源。用于分发java类文件，多个类可以打包成一个jar到不同平台运行

war是Web Application Archive web应用归档文件，将多个servelet和相关的资源打成一个war包。因为servlet运行不是直接通过main方法启动运行的，需要有一个支持servle API的web服务，有这个web服务器去加载我们的“小型服务器”。

Servlet和SpringMVC中的概念

|  |  |
| --- | --- |
| Servlet | servlet, jsp(可以编译成servlet), filter, listener |
| SpringMVC | controller,handler, DispatcherServlet, HandlerIntercepter |