

C/S 架构:

socket、serversocket、awt/swing 做一个客户端软件。

建好 socket 连接后,通过 IO 流来交换数据。数据格式由各个开发者自己确定。name:ddd
uname:ddd。

B/S 架构:

浏览器也是一种客户端软件。

HTTP 协议定义了浏览器和服务器通讯的基本规则之一。

浏览器和服务器的交互:

1. 交互模式是: 请求-响应。
2. 客户端先发出请求,服务器根据请求内容发送响应内容。客户端根据响应内容进行显示。

客户端发出请求的格式应该是什么样的?

1. 首先服务器的地址、端口、资源地址。 请求行
2. 请求头
3. 请求实体

注: 客户端向服务器传参, 如果将参数放到请求行, 即为 get 方式! 如果将参数放到请求实体中, 即为 post 方式!

服务器受到请求内容后发送响应内容

1. 响应内容怎么定义:
 - a) 状态行(200,404,500)
 - b) 响应头
 - c) 响应实体

客户端收到响应内容之后, 该如何显示这些内容?

1. 首先 HTML 诞生了。 哈哈哈哈哈 五星红旗
2. 希望 HTML 元素可以动起来, 这样就诞生了 javascript(livescript)。
3. 我们希望有个技术可以精确的、更加漂亮的控制页面元素的显示。于是 CSS 诞生了。

服务器根据请求内容发送响应内容

1. 如何收到请求内容?
 - a) 建立监听。ServerSocket, s.accept()。
 - b) 启动一个线程(轻量级的进程), 为请求的客户提供服务。
 - c) 为了封装客户端提供的请求信息, 于是我定义一个对象: HttpServletRequest 对象。
 - d) 为了封装回送给客户端的响应内容, 我们定义一个对象: HttpServletResponse 对象。
2. 根据请求内容调用相应的程序(Servlet)
 - a) 我们通过 web.xml 中定义的 url 和 servlet 的关系, 找到相应的 servlet 程序。
 - b) 于是我们定义了 HttpServlet, 并且增加 service(HttpServletRequest req, HttpServletResponse response) 方法
 - c) 将 response 对象中的信息, 转译成基于 http 协议的响应内容。再将这些响应内容通过 socket 传送给客户端。

为了统一处理上述流程，于是我们编写了一个程序，叫做 tomcat 服务器。 我们也把服务器叫做 web 容器，servlet 容器。

下面要解决的问题就是怎么样使我们的 Servlet 程序，更加规范，更加强大？

1. 浏览器发送给服务器的请求内容。 `HttpServletRequest` 对象
2. 服务器回送给浏览器的响应内容。 `HttpServletResponse` 对象
3. servlet 类读取 web.xml 中自己的配置信息。 `ServletConfig`
4. 需要读取 web.xml 中的公共信息<context-param>。 `ServletContext`
5. 将所有 servlet 放入一个容器中统一管理。 `ServletContext`
6. 保存以前操作的状态。 客户端保存 Cookie 服务器保存 Session

这时，我们发现 servlet 擅于做业务逻辑处理，但是不善于做页面展示的代码处理。于是，我们对 servlet 进行了改进处理，jsp 诞生了！ jsp 也是 servlet，也是 java 类。你访问一个 jsp 实际上是执行了对应的 java 类。 我们需要一个统一的转译程序，可以将 jsp 文件内容转译成 java 类代码，于是产生了 jsp 引擎(一般服务器软件自带了这样的功能)。

ajax—>异步模式，局部刷新效果。

通过增加 XMLHttpRequest 对象，使用 XHR 对象发送请求，获得数据后，也是首先传给 XHR 对象，再通过 javascript 代码操作 dom 模型，实现局部刷新效果。