

[02] Servlet 获取请求和页面跳转

1、Tomcat 和 Servlet 的关系

之前提到过，Servlet 是运行在 Web 容器里的，Tomcat 作为容器的一种，在这里自然也要大概说说两者之间的大致关系。

首先，如上所述，Tomcat 是 Web 应用服务器，是一个 Servlet/JSP 容器。它负责处理客户请求，把请求传送给 Servlet，并将 Servlet 的响应传回给客户。而 Servlet 是运行在支持 Java 语言的服务器上的组件。

从 HTTP 协议中的请求和响应就可以得知，浏览器发出的请求是一个请求文本，而浏览器接收到的也应该是一个响应文本。而在 Web 服务器中，容器将其分别封装成了对象，以便于我们进行操作：

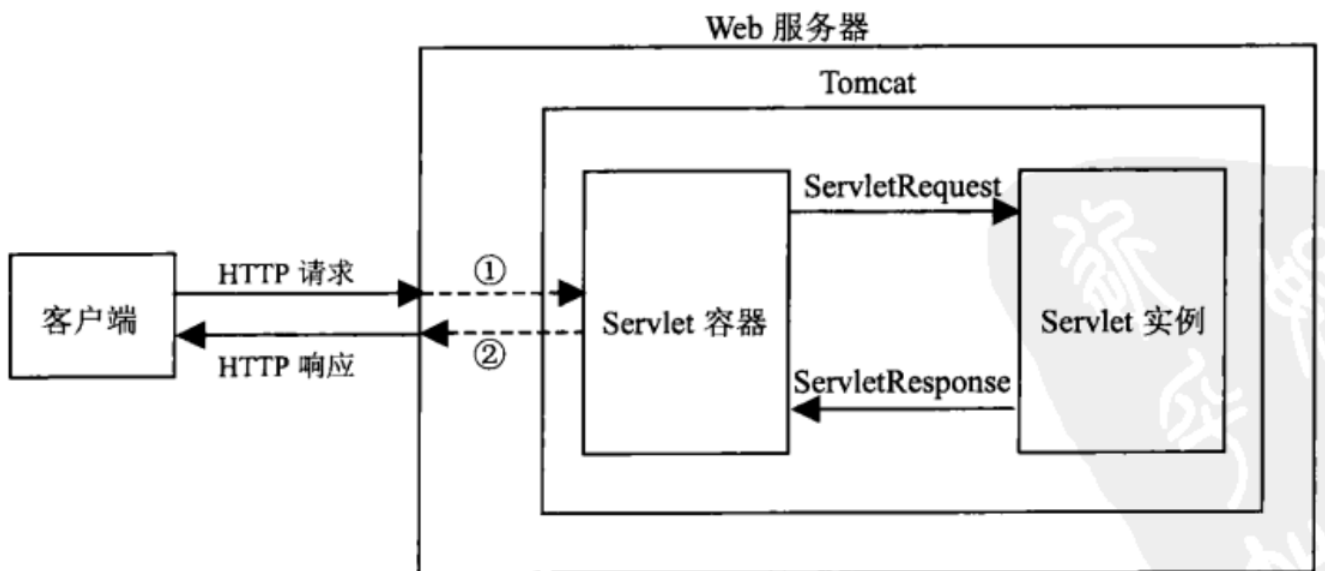


图 1-1 Tomcat 服务器响应客户请求过程

- Tomcat 将 HTTP 请求文本接收并解析，然后封装成 `HttpServletRequest` 类型的 `request` 对象，所有的 HTTP 头数据读取可以通过 `request` 对象调用对应的方法查询到；
- Tomcat 同时将响应的信息封装为 `HttpServletResponse` 类型的 `response` 对象，通过设置 `response` 属性就可以控制要输出到浏览器的内容，然后将 `response` 交给 Tomcat，Tomcat 就会将其变成响应文本的格式发送给浏览器

2、请求对象的由来

Servlet 类中我们往往要覆盖两个方法，doGet 和 doPost，而它们都有两个参数，分别是 HttpServletRequest 和 HttpServletResponse，分别为请求和响应。

实际上，这两个类型都是接口，具体的实现类实际上是由容器实现的，其实也就是我们上面已经说明了的，是由 Tomcat 服务器将请求和响应进行了分别的封装，然后拿给我们使用。

我们可以使用 System.out.println 方法分别打印 doGet 和 doPost 方法中的形参 request 和 response，你可以在控制台看到：

- org.apache.catalina.connector.RequestFacade
- org.apache.catalina.connector.ResposneFacade

我们可以看下 API，实际上也确实是这样说明的，以请求对象为例：

Extends the ServletRequest interface to provide request information for HTTP servlets.

The servlet container creates an HttpServletRequest object and passes it as an argument to the servlet's service methods (doGet, doPost, etc).

3、HttpServletRequest 接口

如下 API，可以看到，因为请求中大部分数据是直接封装进去的，所以基本上都是 get 方法：

拿 getParameter(String name) 这个方法为例，该方法是获取随着请求一起传递过来的额外信息，常见的如我们的账户密码登陆的表单提交，其中的 name 对应的是表单中 input 的 name 属性值，这其实在提交后的过程中也是容器帮我们封装好的。

如果是诸如 checkbox 多选框表单的提交，那么获取值的时候则使用 getParameterValues(String name)，返回的是 String 类型的数组。

Method Summary	
String	getAuthType() Returns the name of the authentication scheme used to protect the servlet.
String	getContextPath() Returns the portion of the request URI that indicates the context of the request.
Cookie[]	getCookies() Returns an array containing all of the Cookie objects the client sent with this request.
long	getDateHeader(String name) Returns the value of the specified request header as a long value that represents a Date object.
String	getHeader(String name) Returns the value of the specified request header as a String.
Enumeration	getHeaderNames() Returns an enumeration of all the header names this request contains.
Enumeration	getHeaders(String name) Returns all the values of the specified request header as an Enumeration of String objects.
int	getIntHeader(String name) Returns the value of the specified request header as an int.
String	getMethod() Returns the name of the HTTP method with which this request was made, for example, GET, POST, or PUT.
String	getPathInfo() Returns any extra path information associated with the URL the client sent when it made this request.
String	getPathTranslated() Returns any extra path information after the servlet name but before the query string, and translates it to a real path.
String	getQueryString() Returns the query string that is contained in the request URL after the path.
String	getRemoteUser() Returns the login of the user making this request, if the user has been authenticated, or null if the user has not been authenticated.
String	getRequestedSessionId() Returns the session ID specified by the client.
String	getRequestURI() Returns the part of this request's URL from the protocol name up to the query string in the first line of the HTTP request.
StringBuffer	getRequestURL() Reconstructs the URL the client used to make the request.
String	getServletPath() Returns the part of this request's URL that calls the servlet.
HttpSession	getSession() Returns the current session associated with this request, or if the request does not have a session, creates one.

其他的 get 方法这里不再详细进行展开，可以参看 API。

4、响应重定向和请求转发

我们知道，请求到达 Servlet 之后，往往还需要进一步的跳转到其他页面，因为之前已经说过，Servlet 现在的作用还是用来做控制，而不再进行页面资源输出了。

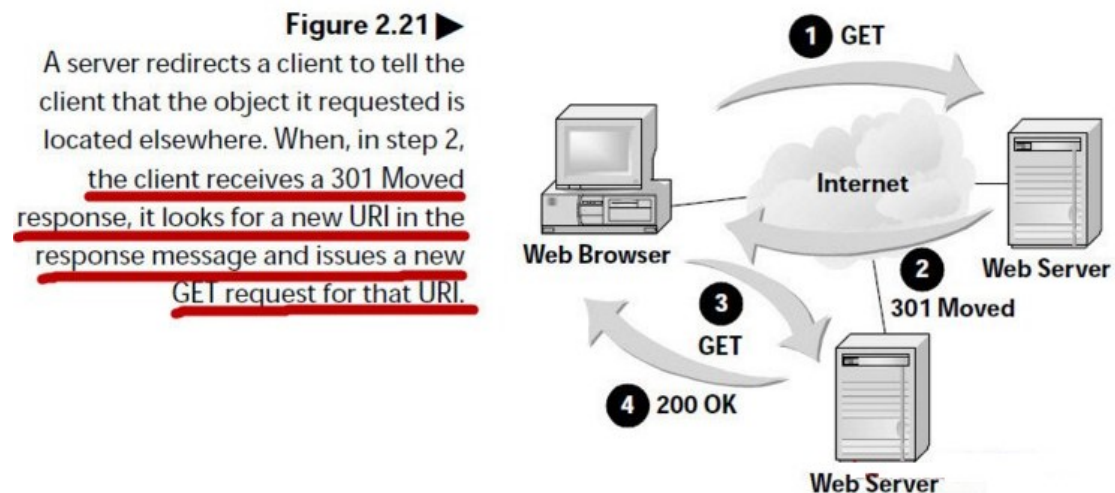
所以在 Servlet 中有两种跳转方式：

- 响应重定向
- 请求转发

4.1 响应重定向

标准意义上的“重定向”指的是 HTTP 重定向，它是 HTTP 协议规定的一种机制。

这种机制是这样工作的：当 client 向 server 发送一个请求，要求获取一个资源时，在 server 接收到这个请求后发现请求的这个资源实际存放在另一个位置，于是 server 在返回的 response 中写入那个请求资源的正确的 URL，并设置 response 的状态码为 301（表示这是一个要求浏览器重定向的 response），当 client 接受到这个 response 后就会根据新的 URL 重新发起请求。重定向有一个典型的特征，即当一个请求被重定向以后，最终浏览器上显示的 URL 往往不再是开始时请求的那个 URL 了（**浏览器地址会变化**）。



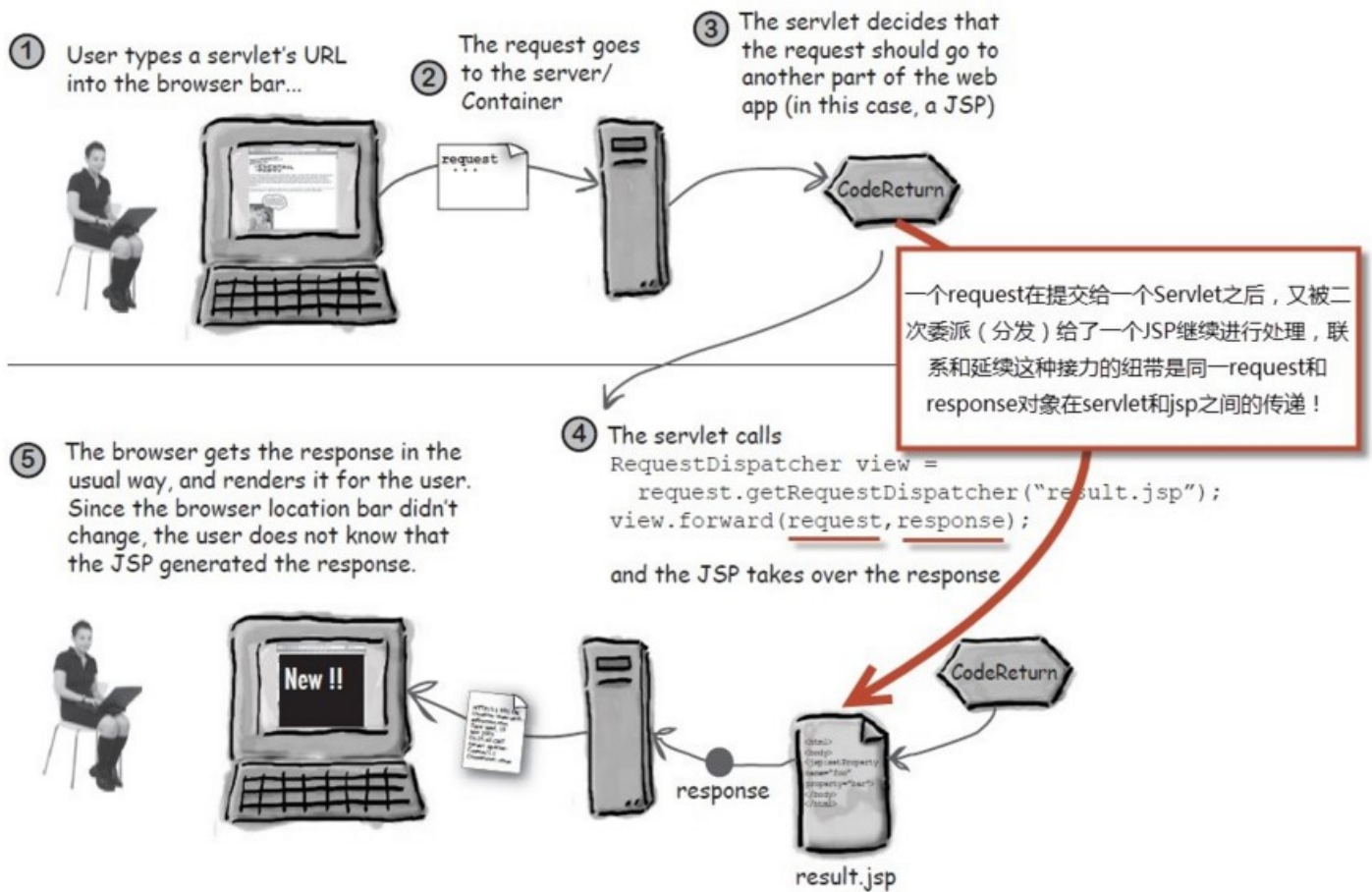
`HttpServletResponse` 中定义了响应重定向的方法 `sendRedirect(String path)`，该方法向目标资源重新发送请求，生成新的响应，并且：

- 重定向使用 GET 方式提交
- 重定向生成的是新的请求，所以当前请求中的数据不会随着新的请求而传递

4.2 请求转发

请求转发也称之为服务器端重定向，它指的是服务器在处理 request 的过程中将 request 先后委托多个 Servlet 或 JSP 接替进行处理的过程。

最为常见的是先由一个 Servlet 处理（比如实现业务逻辑的计算），然后 forward 给一个 JSP 进行视图的渲染，或者 forward 给另一个 Servlet 进一步处理。使用服务器端重定向，**浏览器地址不会发生变化**。因为浏览器只执行了对某个页面 A 的请求，而这个页面的服务器内部的跳转（或许还经过了页面 B 和页面 C 的处理）情况浏览器是不知道的，它只知道自己获得的响应是 A 页面发回来的，甚至不知道服务器中有多个 Servlet 处理。



在 Servlet 中，要先使用 `HttpServletRequest` 请求中的 `getRequestDispatcher(String path)` 获取请求转发器对象，然后调用该对象的 `forward(request, response)` 方法进行跳转。

往往以连写的方式，如下示例：

```
request.getRequestDispatcher(String path).forward(request, response);
```