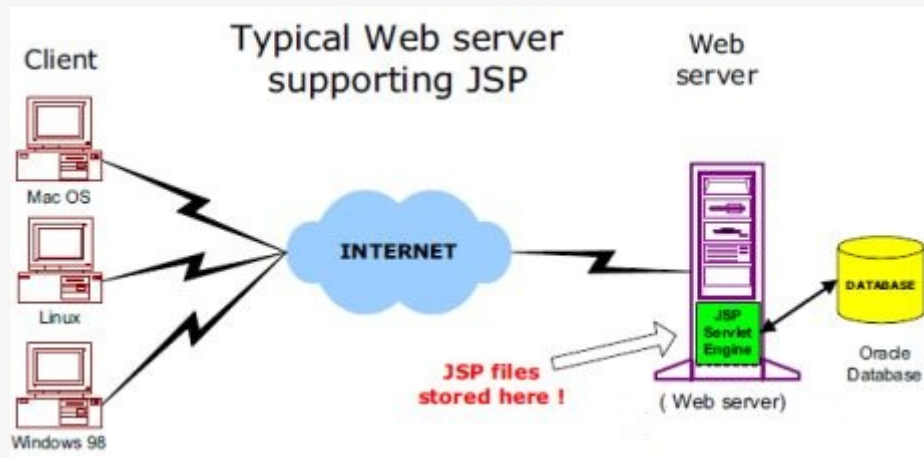


JSP 结构

网络服务器需要一个 JSP 引擎，也就是一个容器来处理 JSP 页面。容器负责截获对 JSP 页面的请求。本教程使用内嵌 JSP 容器的 Apache 来支持 JSP 开发。

JSP 容器与 Web 服务器协同合作，为 JSP 的正常运行提供必要的运行环境和其他服务，并且能够正确识别专属于 JSP 网页的特殊元素。

下图显示了 JSP 容器和 JSP 文件在 Web 应用中所处的位置。

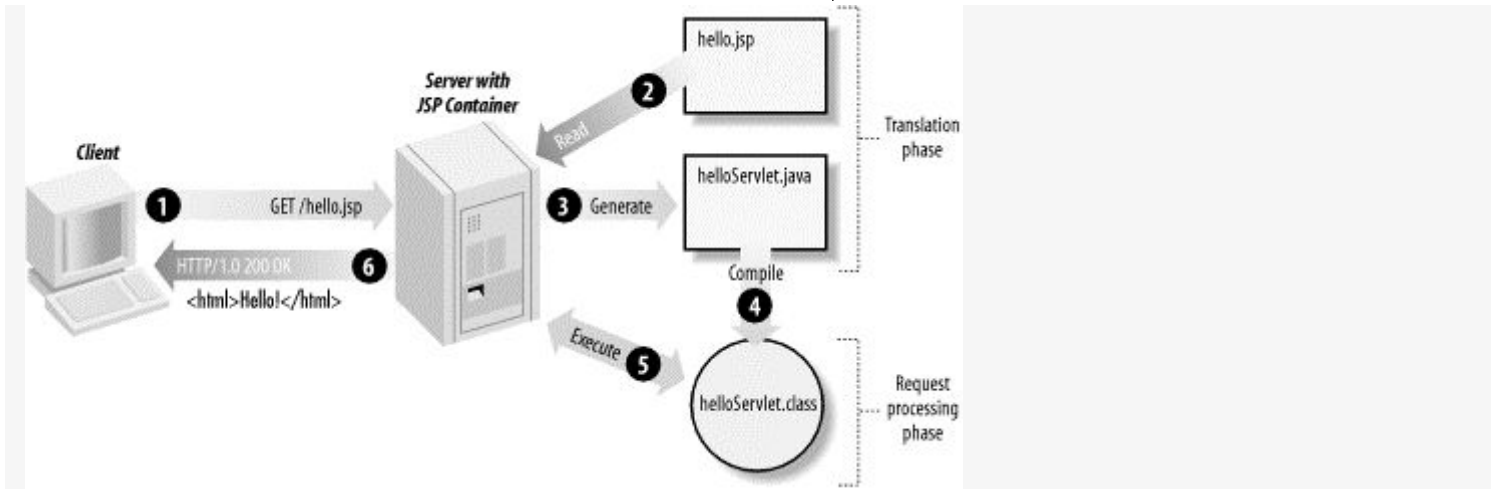


JSP 处理

以下步骤表明了 Web 服务器是如何使用 JSP 来创建网页的：

- 就像其他普通的网页一样，您的浏览器发送一个 HTTP 请求给服务器。
- Web 服务器识别出这是一个对 JSP 网页的请求，并且将该请求传递给 JSP 引擎。通过使用 URL 或者 .jsp 文件来完成。
- JSP 引擎从磁盘中载入 JSP 文件，然后将它们转化为 Servlet。这种转化只是简单地将所有模板文本改用 `println()` 语句，并且将所有的 JSP 元素转化成 Java 代码。
- JSP 引擎将 Servlet 编译成可执行类，并且将原始请求传递给 Servlet 引擎。
- Web 服务器的某组件将会调用 Servlet 引擎，然后载入并执行 Servlet 类。在执行过程中，Servlet 产生 HTML 格式的输出并将其内嵌于 HTTP response 中上交给 Web 服务器。
- Web 服务器以静态 HTML 网页的形式将 HTTP response 返回到您的浏览器中。
- 最终，Web 浏览器处理 HTTP response 中动态产生的 HTML 网页，就好像在处理静态网页一样。

以上提及到的步骤可以用下图来表示：



一般情况下，JSP 引擎会检查 JSP 文件对应的 Servlet 是否已经存在，并且检查 JSP 文件的修改日期是否早于 Servlet。如果 JSP 文件的修改日期早于对应的 Servlet，那么容器就可以确定 JSP 文件没有被修改过并且 Servlet 有效。这使得整个流程与其他脚本语言（比如 PHP）相比要高效快捷一些。

总的来说，JSP 网页就是用另一种方式来编写 Servlet 而不用成为 Java 编程高手。除了解释阶段外，JSP 网页几乎可以被当成一个普通的 Servlet 来对待。

← JSP 开发环境搭建

JSP 生命周期 →

点我分享笔记