

第六讲 服务端编程

盛羽

中南大学计算机学院

shengyu@csu.edu.cn

1. 概述

2. Servlet

3. JSP



■ 服务端编程

- 在需要的时候动态展示不同的信息，这些信息通常会从服务器上的数据库中取出，然后发送给客户端，并通过一些代码（比如 HTML 和 Javascript）展示在客户端

- Web 浏览器通过超文本传输协议（HTTP）来和 Web 服务器进行通信

■ HTTP 请求

- 一个标识所请求资源的 URL

- 一个定义所需操作的方法 (比如获取，删除或者发布资源)

- 还可以包括编码在 URL 参数中的附加信息

■ HTTP 响应

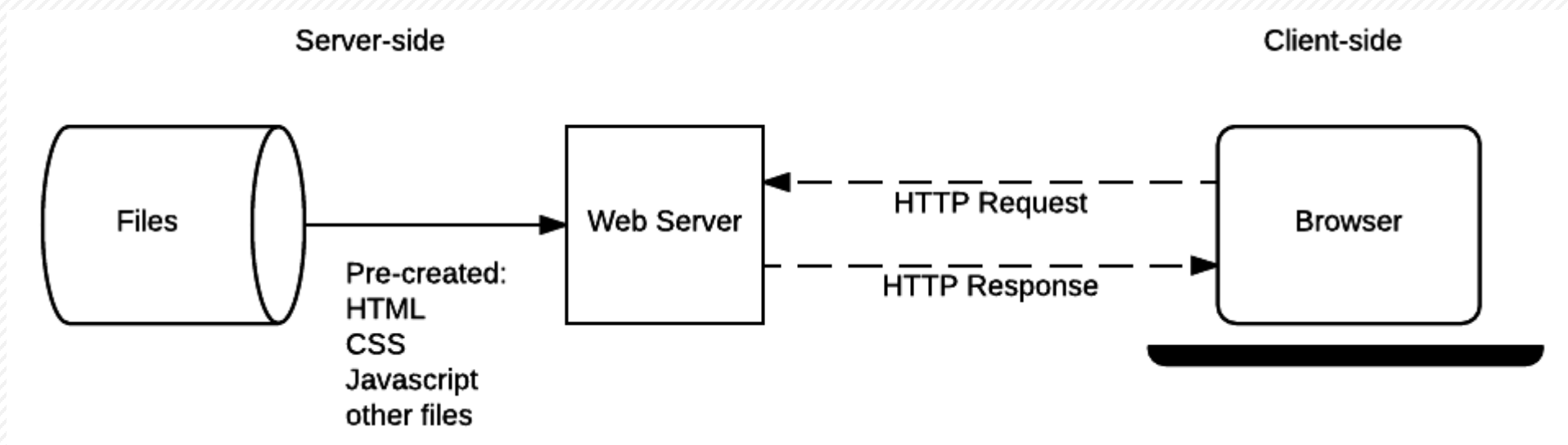
- Web 服务器等待客户端的请求信息，在它们到达的时候处理它们，并且回复 Web 浏览器一个 HTTP 响应信息。

- 这个响应包含一个表明该请求是否成功的状态行（比如“HTTP/1.1 200 OK”代表请求成功）

- 相应一个请求的成功回应包含被请求的资源（比如一个新的 HTML 页面，或者图片等），然后这些会被展示在客户端的 Web 浏览器上

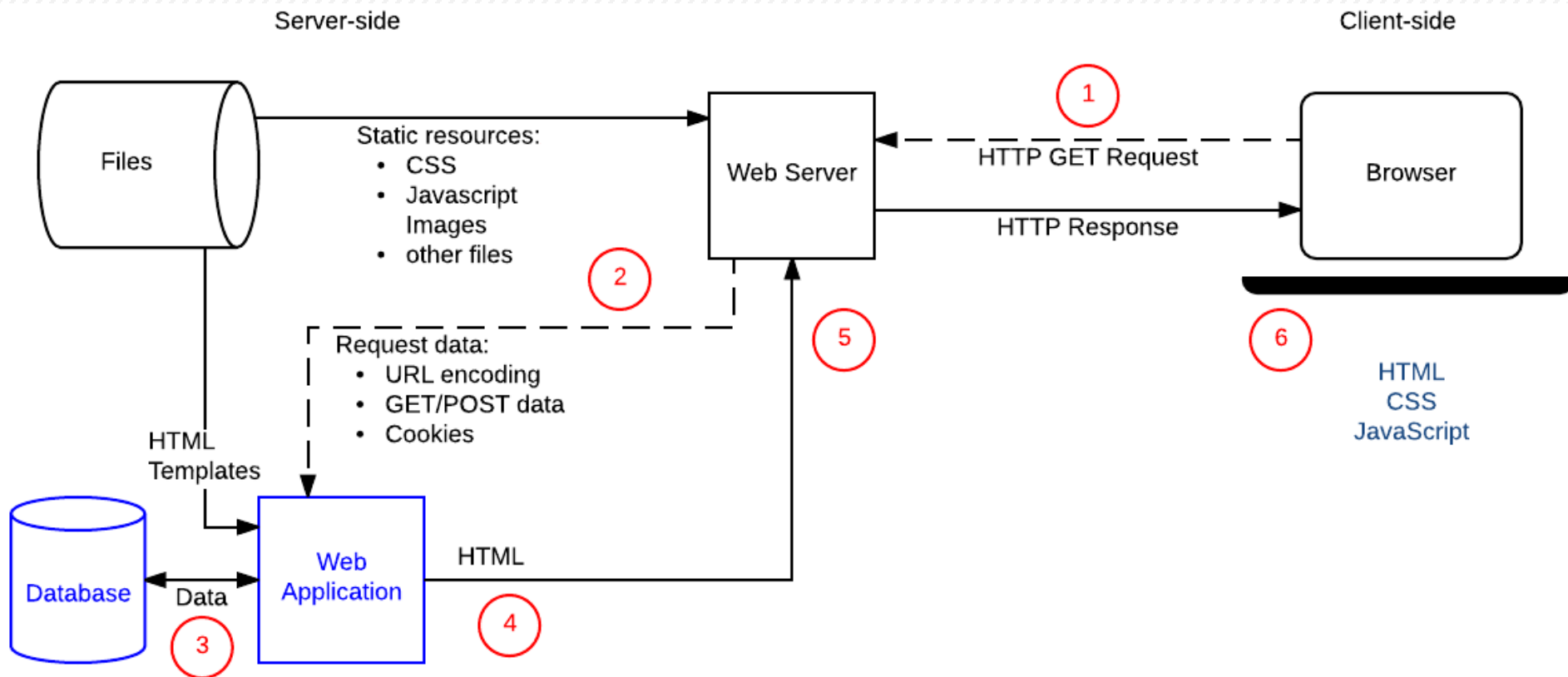
■ 静态网站

- 静态网站是指无论何时当一个特定资源被请求的时候都返回相同的被硬编码的内容
- 浏览器会发送一个指定到这个页面的 URL 的 HTTP“GET”请求。服务器从它的文件系统中检索被请求的文件，然后返回一个 HTTP 回应，该回应包括被请求的文件和一个状态码（通常 200 代表操作成功）



■ 动态网站

- 动态网站是指，一些响应内容只有在被需要的时候才会生发的网站
- 在一个动态网站上，页面通常是通过将数据库的数据植入到 HTML 模板中的占位符中而产生的（这是一种比使用静态网站有效得多的存储大量内容的方式）
- 大多数支持动态网站的代码必须运行在服务器上。编写这些代码就是所谓的“服务器端编程”（有些时候也称“后端脚本编写”）



1. 概述

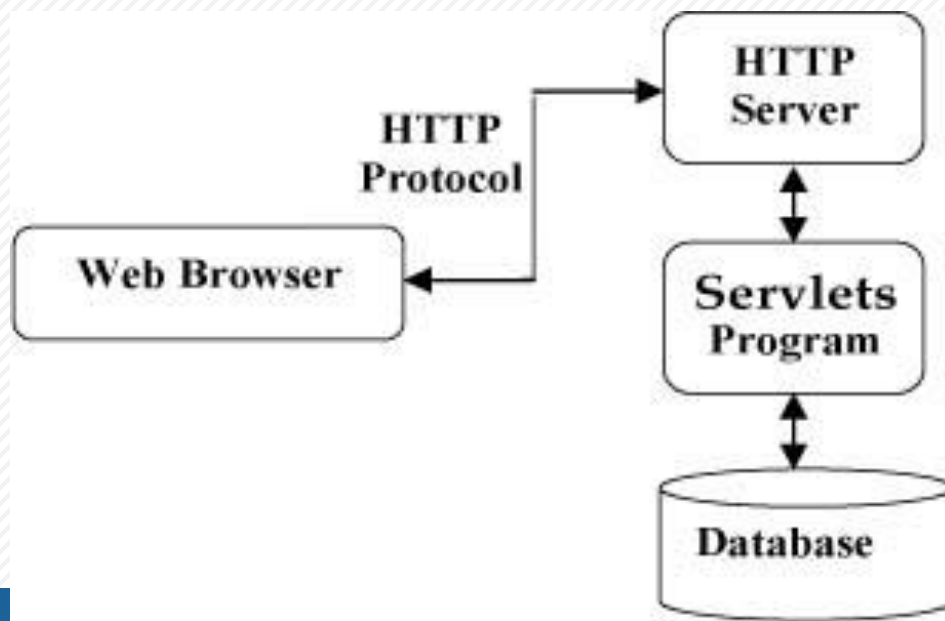
2. Servlet

3. JSP



■ Servlet

- 一种服务器端的Java应用程序
- 可以生成动态的Web页面。
- 担当客户请求（Web浏览器或其他HTTP客户程序）与服务器响应（HTTP服务器上的数据库或应用程序）的中间层。
- Servlet是位于Web 服务器内部的服务器端的Java应用程序，与传统的从命令行启动的Java应用程序不同，Servlet由Web服务器进行加载，该Web服务器必须包含支持Servlet的Java虚拟



■ Servlet的主要功能

■ 交互式地浏览和修改数据，生成动态Web内容。这个过程为：

- 客户端发送请求至服务器端；
- 服务器将请求信息发送至Servlet
- Servlet生成响应内容并将其传给Server。响应内容动态生成，通常取决于客户端的请求
- 服务器将响应返回给客户端

■ Servlet生命周期

- Server创建一个Servlet的实例

- Server调用Servlet的init()方法

- 一个客户端的请求到达Server

 - Server创建一个请求对象

 - Server创建一个响应对象

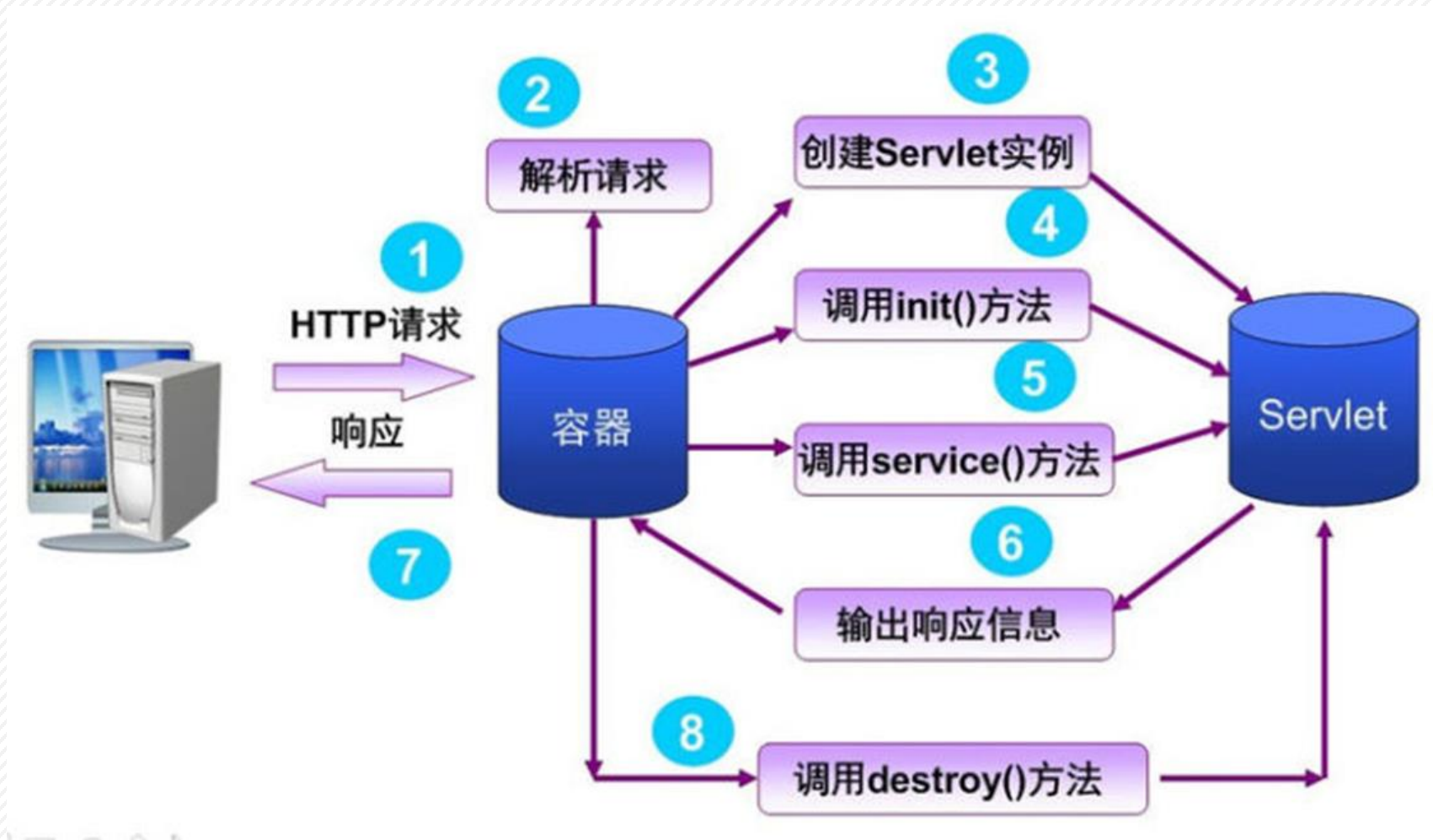
 - Server激活Servlet的service()方法，传递请求和响应对象作为参数

 - service()方法获得关于请求对象的信息，处理请求，访问其他资源，获得需要的信息

 - service()方法使用响应对象的方法，将响应传回Server，最终到达客户端。service()方法可能激活其它方法以处理请求，如doGet()或doPost()或程序员自己开发的新的方法。

- 一般Servlet只初始化一次(只有一个对象)

2. Servlet



■ 编写流程

- 编写一个类，继承 `javax.sevelet.http.HttpServlet`
- 重写
 - `init`函数
 - 一个 `doGet /doPost/service`函数
 - 一个 `Destroy`函数
- `HttpServletRequest` 和 `HttpServletResponse` 是所有 `doXXX()`方法的默认参数
- 配置文件(`web.xml`) 里进行注册
- 编译
- 访问
- 示例

■ Servlet的内置对象

■ out

- `PrintWriter out=response.getWriter();`

■ request: 参数request

■ response: 参数response

■ session

- `HttpSession session=request.getSession();`

■ application

- `ServletContext application=this.getServletContext();`

1. 概述

2. Servlet

3. JSP



- 在HTML文档中加入Java脚本程序和JSP标记来构成

- 语法结构

 - `<% %>`

 - 脚本元素：规范JSP中所使用的Java代码

 - 指令元素：针对JSP引擎控制转译后的Serverlet的整个结构

 - 动作元素：连接到用到的组件（如Javabenas）

- JSP指令

 - 指示引擎如何处理JSP页面中的内容

 - `<%@ 指令名 属性1=“属性值” 属性2=“属性值” ... %>`

■ Include指令

`<% @ include file="插入文件的URL" %>`

- 读入指定页面的内容。并把这些内容和原来的页面融合到一起
- 这个过程是在翻译阶段:也就是jsp被转化成servlet的阶段进行的

■ 声明

`<%! %>`

□ 表达式

`<% =表达式 %>` 直接输出

□ 代码段/脚本片段

`<% for(int i=1;i<=4;i++){%>`

`<h<%=i %>>你好</h<%=i%>>`

`<%}%>`

■ 注释

- `<%--注释内容--%>`

■ JSP内置对象

■ request对象：获取客户端的输入信息

- `getCookie()`、`getHeaders()`、`getAttribute()`、`getMethod()`、`getParameter()`和`getParameterName()`等

- `<%String name=request.getParameter("userName"); out.println(name); %>`

■ response对象：向客户端返回请求

■ out对象：将结果输出到客户端

■ session：JSP为每一个会话而建立的个人对象，可以存储及提供个别用户独享的永久或半永久信息

- 会话：一个用户和Web服务器之间的一次链接