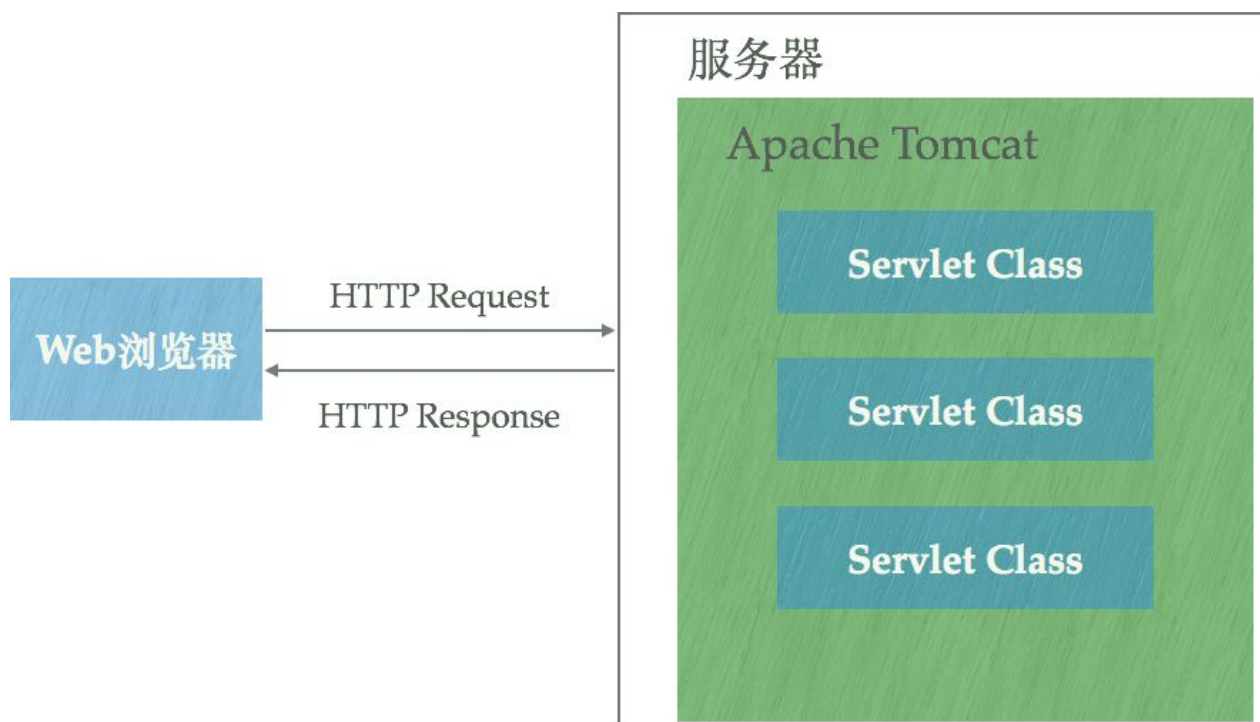


03-Tomcat和Servlet原理

Tomcat



Tomcat运行在JVM之上，它和HTTP服务器一样，绑定IP地址并监听TCP端口，同时还包含以下职责：

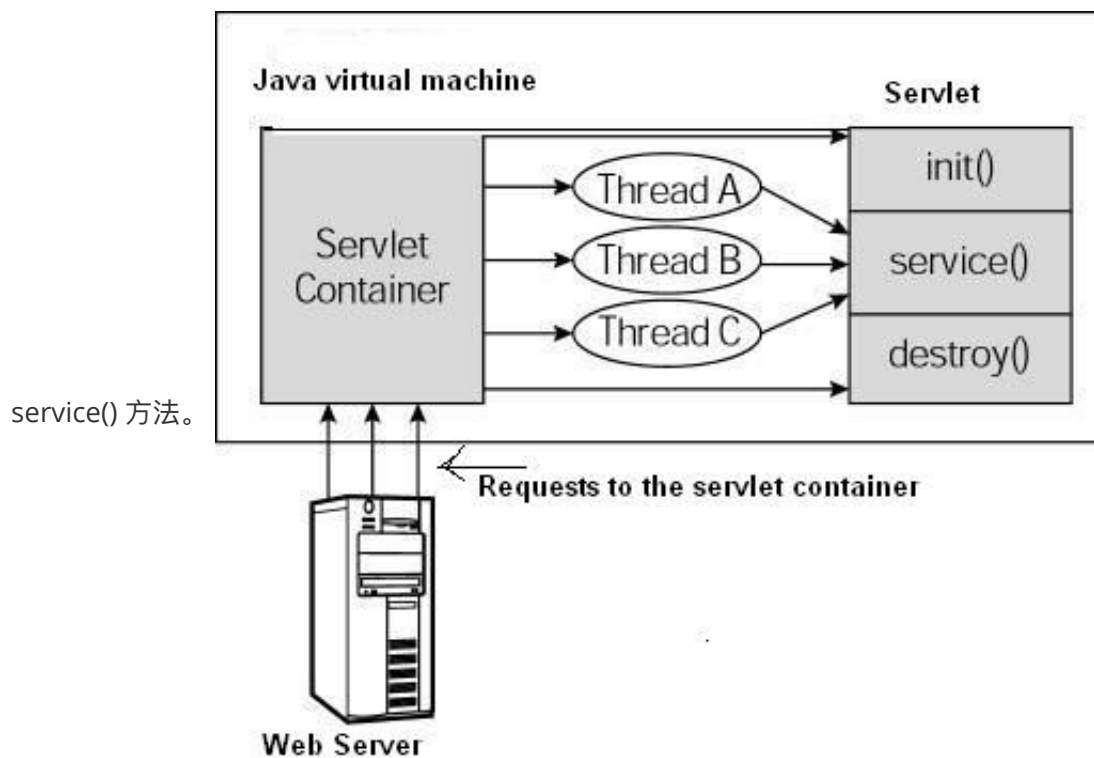
- 管理Servlet程序的生命周期
- 将URL映射到指定的Servlet进行处理
- 与Servlet程序合作处理HTTP请求——根据HTTP请求生成HttpServletResponse对象并传递给Servlet进行处理，将Servlet中的HttpServletResponse对象生成的内容返回给浏览器

Servlet

生命周期

一个典型的 Servlet 生命周期方案。

- 第一个到达服务器的 HTTP 请求被委派到 Servlet 容器。
- Servlet 容器在调用 service() 方法之前加载 Servlet。
- 然后 Servlet 容器处理由多个线程产生的多个请求，每个线程执行一个单一的 Servlet 实例的



service() 方法

`service()` 方法是执行实际任务的主要方法。Servlet 容器（即 Web 服务器）调用 `service()` 方法来处理来自客户端（浏览器）的请求，并把格式化的响应写回给客户端。

每次服务器接收到一个 Servlet 请求时，服务器会产生一个新的线程并调用服务。`service()` 方法检查 HTTP 请求类型（GET、POST、PUT、DELETE 等），并在适当的时候调用 `doGet`、`doPost`、`doPut`、`doDelete` 等方法。

下面是该方法的特征：

```
public void service(ServletRequest request,
                    ServletResponse response)
    throws ServletException, IOException{
}
```

`service()` 方法由容器调用，`service` 方法在适当的时候调用 `doGet`、`doPost`、`doPut`、`doDelete` 等方法。所以，您不用对 `service()` 方法做任何动作，您只需要根据来自客户端的请求类型来重写 `doGet()` 或 `doPost()` 即可。

`doGet()` 和 `doPost()` 方法是每次服务请求中最常用的方法。下面是这两种方法的特征。

doGet() 方法

GET 请求来自于一个 URL 的正常请求，或者来自于一个未指定 METHOD 的 HTML 表单，它由 `doGet()` 方法处理。

```
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
    throws ServletException, IOException {
    // Servlet 代码
}
```

doPost() 方法

POST 请求来自于一个特别指定了 METHOD 为 POST 的 HTML 表单，它由 doPost() 方法处理。

```
public void doPost(HttpServletRequest request,
                  HttpServletResponse response)
    throws ServletException, IOException {
    // Servlet 代码
}
```

helloworld servlet

```
// 导入必需的 java 库
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

// 扩展 HttpServlet 类
public class HelloWorld extends HttpServlet {

    private String message;

    public void init() throws ServletException
    {
        // 执行必需的初始化
        message = "Hello World";
    }

    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws ServletException, IOException
    {
        // 设置响应内容类型
        response.setContentType("text/html");

        // 实际的逻辑是在这里
        PrintWriter out = response.getWriter();
        out.println("<h1>" + message + "</h1>");
    }

    public void destroy()
```

```
{  
    // 什么也不做  
}  
}
```

编译 Servlet

\$ javac HelloWorld.java

部署 Servlet

默认情况下，Servlet 应用程序位于路径 /webapps/ROOT 下，且类文件放在 /webapps/ROOT/WEB-INF/classes 中。

如果您有一个完全合格的类名称 com.myorg.MyServlet，那么这个 Servlet 类必须位于 WEB-INF/classes/com/myorg/MyServlet.class 中。

现在，让我们把 HelloWorld.class 复制到 /webapps/ROOT/WEB-INF/classes 中，并在位于 /webapps/ROOT/WEB-INF/ 的 web.xml 文件中创建以下条目：

```
<web-app>  
    <servlet>  
        <servlet-name>HelloWorld</servlet-name>  
        <servlet-class>HelloWorld</servlet-class>  
    </servlet>  
  
    <servlet-mapping>  
        <servlet-name>HelloWorld</servlet-name>  
        <url-pattern>/HelloWorld</url-pattern>  
    </servlet-mapping>  
</web-app>
```

原文链接：<https://www.zhihu.com/question/32212996/answer/87524617>

运行Servlet

启动 tomcat 服务器，最后在浏览器的地址栏中输入 <http://localhost:8080/HelloWorld>。如果一切顺利，您会看到下面的结果：



Servlet Get和Post案例

Get 案例

面是一个简单的 URL，将使用 GET 方法向 HelloForm 程序传递两个值。

<http://localhost:8080/TomcatTest/HelloForm?name=菜鸟教程&url=www.runoob.com>

下面是处理 Web 浏览器输入的 **HelloForm.java** Servlet 程序。我们将使用 **getParameter()** 方法，可以很容易地访问传递的信息：

```
package com.runoob.test;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class HelloForm
 */
@WebServlet("/HelloForm")
public class HelloForm extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public HelloForm() {
        super();
        // TODO Auto-generated constructor stub
    }
}
```

```

/**
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
 */
protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    // 设置响应内容类型
    response.setContentType("text/html;charset=UTF-8");

    PrintWriter out = response.getWriter();
    String title = "使用 GET 方法读取表单数据";
    // 处理中文
    String name =new String(request.getParameter("name").getBytes("ISO-
8859-1"),"UTF-8");
    String docType = "<!DOCTYPE html> \n";
    out.println(docType +
        "<html>\n" +
        "<head><title>" + title + "</title></head>\n" +
        "<body bgcolor=\"#f0f0f0\">\n" +
        "<h1 align=\"center\">" + title + "</h1>\n" +
        "<ul>\n" +
        "    <li><b>站点名</b>: "
        + name + "\n" +
        "    <li><b>网址</b>: "
        + request.getParameter("url") + "\n" +
        "</ul>\n" +
        "</body></html>");
}

// 处理 POST 方法请求的方法
public void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    doGet(request, response);
}
}

```

然后我们在 **web.xml** 文件中创建以下条目：

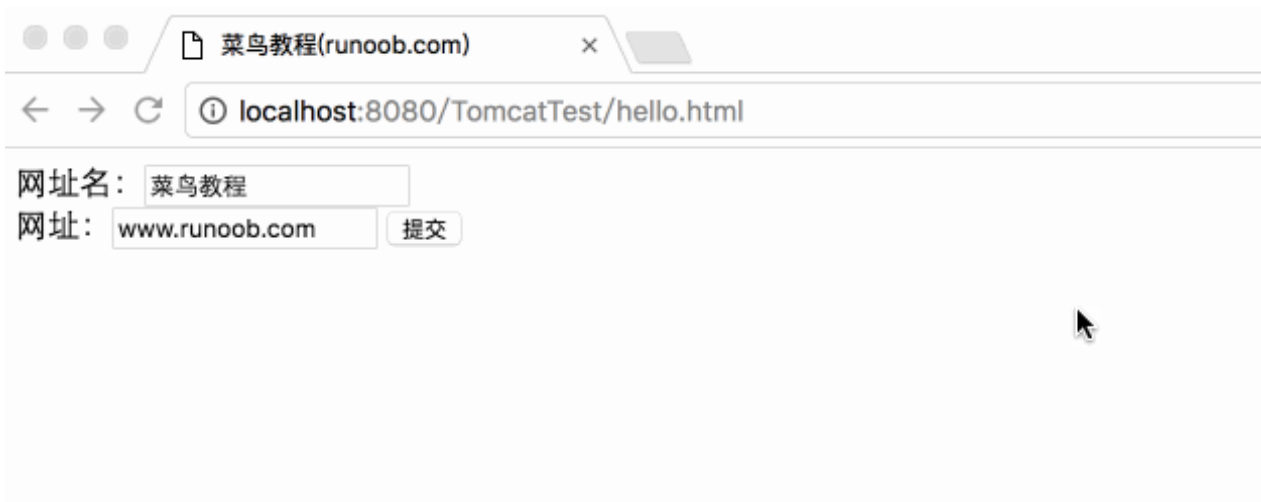
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
  <servlet>
    <servlet-name>HelloForm</servlet-name>
    <servlet-class>com.runoob.test>HelloForm</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>HelloForm</servlet-name>
    <url-pattern>/TomcatTest/HelloForm</url-pattern>
  </servlet-mapping>
</web-app>
```

现在在浏览器的地址栏中输入 <http://localhost:8080/TomcatTest/HelloForm?name=菜鸟教程&url=www.runoob.com>，并在触发上述命令之前确保已经启动 Tomcat 服务器。如果一切顺利，您会得到下面的结果：



下面是一个简单的实例，使用 HTML 表单和提交按钮传递两个值。我们将使用相同的 Servlet HelloForm 来处理输入。

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>菜鸟教程 (runoob.com)</title>
</head>
<body>
<form action="HelloForm" method="GET">
  网址名: <input type="text" name="name">
  <br />
  网址: <input type="text" name="url" />
  <input type="submit" value="提交" />
</form>
</body>
</html>
```



post 案例

让我们对上面的 Servlet 做小小的修改，以便它可以处理 GET 和 POST 方法。下面的 **HelloForm.java** Servlet 程序使用 GET 和 POST 方法处理由 Web 浏览器给出的输入。

注意：如果表单提交的数据中有中文数据则需要转码：

```
String name =new String(request.getParameter("name").getBytes("ISO8859-1"), "UTF-8");
```

```
package com.runoob.test;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class HelloForm
 */
@WebServlet("/HelloForm")
public class HelloForm extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public HelloForm() {
        super();
        // TODO Auto-generated constructor stub
    }
}
```



```

/**
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
 */
protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    // 设置响应内容类型
    response.setContentType("text/html;charset=UTF-8");

    PrintWriter out = response.getWriter();
    String title = "使用 POST 方法读取表单数据";
    // 处理中文
    String name =new String(request.getParameter("name").getBytes("ISO8859-
1"),"UTF-8");
    String docType = "<!DOCTYPE html> \n";
    out.println(docType +
        "<html>\n" +
        "<head><title>" + title + "</title></head>\n" +
        "<body bgcolor=\"#f0f0f0\">\n" +
        "<h1 align=\"center\">" + title + "</h1>\n" +
        "<ul>\n" +
        "    <li><b>站点名</b>: "
        + name + "\n" +
        "    <li><b>网址</b>: "
        + request.getParameter("url") + "\n" +
        "</ul>\n" +
        "</body></html>");
    }

    // 处理 POST 方法请求的方法
    public void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        doGet(request, response);
    }
}

```

现在，编译部署上述的 Servlet，并使用带有 POST 方法的 hello.html 进行测试，如下所示：

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>菜鸟教程(runoob.com)</title>
</head>
<body>
<form action="HelloForm" method="POST">
网址名: <input type="text" name="name">
<br />

```

```
网址: <input type="text" name="url" />  
<input type="submit" value="提交" />  
</form>  
</body>  
</html>
```

下面是上面表单的实际输出，尝试输入网址名和网址，然后点击"提交"按钮，Gif 演示如下：

