Servlet 生命周期 Servlet 表单数据

Servlet 实例

Servlet 是服务 HTTP 请求并实现 **javax.servlet.Servlet** 接口的 Java 类。Web 应用程序开发人员通常编写 Servlet 来扩展 javax.servlet.http.HttpServlet，并实现 Servlet 接口的抽象类专门用来处理 HTTP 请求。

Hello World 示例代码

下面是 Servlet 输出 Hello World 的示例源代码：

```
// 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

让我们把上面的代码写在 HelloWorld.java 文件中，把这个文件放在 C:\ServletDevel（在 Windows 上）或 /usr/ServletDevel（在 UNIX 上）中，您还需要把这些目录添加到 CLASSPATH 中。

假设您的环境已经正确地设置，进入 **ServletDevel** 目录，并编译 HelloWorld.java，如下所示：

```
$ javac HelloWorld.java
```

如果 Servlet 依赖于任何其他库，您必须在 CLASSPATH 中包含那些 JAR 文件。在这里，我只包含了 servlet-api.jar JAR 文件，因为我没有在 Hello World 程序中使用任何其他库。

该命令行使用 Sun Microsystems Java 软件开发工具包（JDK）内置的 javac 编译器。为使该命令正常工作，您必须 PATH 环境变量中使用的 Java SDK 的位置。

如果一切顺利，上面编译会在同一目录下生成 HelloWorld.class 文件。下一节将讲解已编译的 Servlet 如何部署在生产中。

Servlet 部署

默认情况下，Servlet 应用程序位于路径 <Tomcat-installation-directory>/webapps/ROOT 下，且类文件放在 <Tomcat-installation-directory>/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 <Tomcat-installation-directory>/webapps/ROOT/WEB-INF/classes 中，并在位于 <Tomcat-installation-directory>/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>
```

上面的条目要被创建在 web.xml 文件中的 <web-app>...</web-app> 标签内。在该文件中可能已经有各种可用的条目，但不要在意。

到这里，您基本上已经完成了，现在让我们使用 <Tomcat-installation-directory>\bin\startup.bat（在 Windows 上）或 <Tomcat-installation-directory>/bin/startup.sh（在 Linux/Solaris 等上）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 生命周期 Servlet 表单数据

2 篇笔记 写笔记

destory 方法被调用后，servlet 被销毁，但是并没有立即被回收，再次请求时，并没有重新初始化。
代码示例：

```
private String message;

@Override
public void init() throws ServletException {
    message = "Hello World , Nect To Meet You: " + System.currentTimeMillis();
    System.out.println("servlet初始化……");
    super.init();
}

@Override
public void doGet(HttpServletRequest req, HttpServletResponse response) throws ServletExc
eption, IOException {
    response.setContentType("text/html");
    PrintWriter writer = response.getWriter();
    writer.write("<h1>" + message + "</h1>");
    destroy();
}

@Override
public void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletExcept
ion, IOException {
    // TODO Auto-generated method stub
    super.doPost(req, resp);
}

@Override
```

```
public void destroy() {
    System.out.println("servlet销毁! ");
    super.destroy();
}
```

控制台打印：

```
servlet初始化……
servlet销毁!
2017-7-6 19:48:52 org.apache.catalina.core.StandardContext reload
信息: Reloading Context with name [/myServlet] has started
servlet销毁!
2017-7-6 19:48:52 org.apache.catalina.core.StandardContext reload
信息: Reloading Context with name [/myServlet] is completed
servlet初始化……
servlet销毁!
servlet销毁!
servlet销毁!
servlet销毁!
servlet销毁!
servlet销毁!
servlet销毁!
```

wk_我主沉浮 2年前 (2017-07-06)

servlet 浏览器访问路径配置有个小问题：

1、java 类里的注解 —— @WebServlet("/HelloServlet") 对应浏览器路径：

```
http://localhost:8080/TomcatTest/HelloServlet
```

2、配置文件（web.xml）里对应的浏览器访问路径：

```
http://localhost:8080/TomcatTest/TomcatTest/HelloServlet
```

这两种配一个就好了，不然路径重名的话反而会让tomcat启动不了。

例如这样就启动不了：

修改 web.xml：

```
<url-pattern>/HelloServlet</url-pattern>
```

修改后，web.xml 和 java 类的注解，对应路径都是：

```
http://localhost:8080/TomcatTest/HelloServlet
```

导致

```
命名的 servlet[HelloServlet]和 [com.runoob.test.HelloServlet] 都被映射到 URL 模式 [/ HelloServlet] 这是不允许的。
```

解决办法：

将**注解去掉**或者**保留注解进入web.xml将映射删除**既可以。

董大dj 2年前 (2017-09-15)