# 第9章

1. 填空
2. Filter
3. HttpSessionListener
4. Content-Type
5. List
6. multipart/form-data
7. 判断
8. 对
9. 对
10. 对
11. 错
12. 错
13. 选择
14. C
15. B
16. D
17. C
18. A
19. 简答
20. Filter的生命周期可分为创建、执行、销毁三个阶段。

（1）创建阶段：Web服务器启动的时候会创建Filter实例对象，并调用init()方法，完成对象的初始化。

（2）执行阶段：当客户端请求目标资源时，服务器会筛选出符合映射条件的Filter，并按照类名的的先后顺序依次执行doFilter() 方法。

（3）销毁阶段：服务器关闭时，Web服务器调用destroy()方法销毁Filter对象。

1. 编程

1. 该功能的实现步骤如下：

（1）创建MyServlet程序类。

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.\*;

@WebServlet(name = "MyServlet",urlPatterns = "/MyServlet")

public class MyServlet extends HttpServlet {

public void doGet(HttpServletRequest request,

HttpServletResponse response)throws ServletException, IOException {

response.getWriter().write("Hello MyServlet ");

}

public void doPost(HttpServletRequest request,

HttpServletResponse response)throws ServletException, IOException {

doGet(request, response);

}

}

（2）创建过滤器MyFilter类。

import javax.servlet.\*;

import javax.servlet.annotation.WebFilter;

import java.io.IOException;

import java.io.PrintWriter;

@WebFilter(filterName = "MyFilter",urlPatterns = "/\*")

public class MyFilter implements Filter {

public void destroy() {

// 过滤器对象在销毁时自动调用，释放资源

}

public void doFilter(ServletRequest request, ServletResponse response,

FilterChain chain) throws IOException, ServletException {

// 用于拦截用户的请求，如果和当前过滤器的拦截路径匹配，该方法会被调用

PrintWriter out=response.getWriter();

out.write("Hello MyFilter");

}

public void init(FilterConfig fConfig) throws ServletException {

// 过滤器对象在初始化时调用，可以配置一些初始化参数

}

}

（3）启动服务器后，访问MyServlet，即可查看拦截效果。

1. 该功能的实现步骤如下：

（1）创建下载页面download.jsp。

<%@ page language="java" contentType="text/html; charset=UTF-8" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>文件下载</title>

</head>

<body>

<a href="http://localhost:8080/chapter09/DownloadServlet?filename=1.png">

文件下载</a>

</body>

</html>

（2）编写DownloadServlet类，该类主要用于设置所要下载的文件以及文件在浏览器中的打开方式，并使用encode(String s, String enc)方法，将URL中的字符串以指定的编码形式输出，防止文件名出现中文乱码问题。

import java.io.\*;

import java.net.URLEncoder;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class DownloadServlet extends HttpServlet {

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=utf-8");

// 获得绝对路径创建文件对象

String path=getServletContext().getRealPath("/download/1.png");

File file=new File(path);

// 通知浏览器以下载的方式打开文件

response.addHeader("Content-Type", "application/octet-stream");

response.addHeader("Content-Disposition","attachment;filename="

+URLEncoder.encode(file.getName(),"utf-8"));

// 通过文件对象获取文件相关的输入流

InputStream in=new FileInputStream(file);

// 获取response对象的输出流

OutputStream out = response.getOutputStream();

byte [] buffer=new byte[1024];

int len;

while((len=in.read(buffer))!=-1){

out.write(buffer,0,len);

}

}

public void doPost(HttpServletRequest req, HttpServletResponse resp)

throws ServletException, IOException {

doGet(req, resp);

}

}

（3）使用浏览器访问该download.jsp，单击页面中的文件下载链接后，即可保存下载该文件。