## http状态码

常见的HTTP状态码：

200 ----请求成功

301 ----资源（网页等）被永久转移到其他url

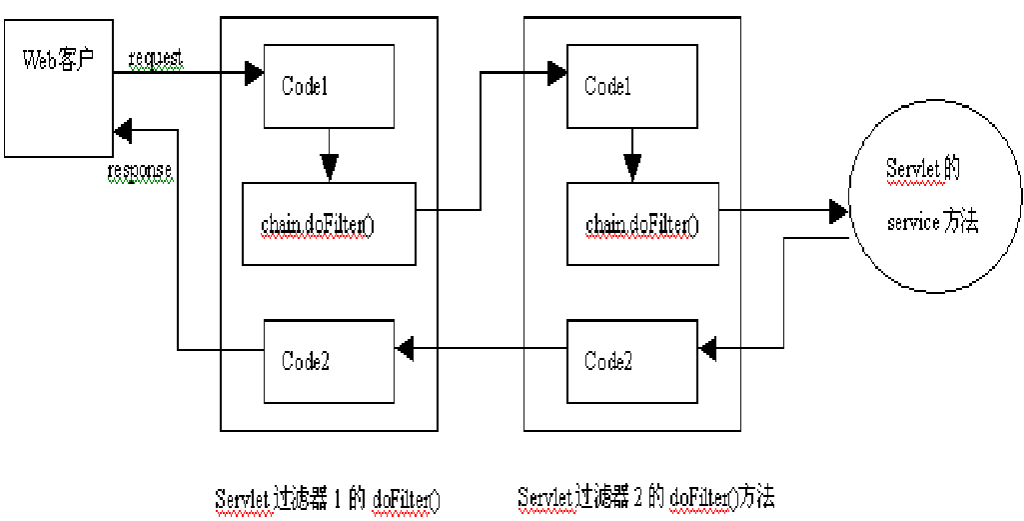
404 ----请求的页面不存在

500 ----内部服务器错误

具体请看：<http://www.runoob.com/http/http-status-codes.html>教程

## Filter学习

首先来看一下执行流程：



过滤器的生命周期，一般经历三个阶段：

初始化：

当容器第一次加载过滤器时，init()方法将被调用，该类在这个方法中包含了一个指向 Filter Config 对象的引用。

过滤：

过滤器的大多数时间都消耗在这里。doFilter方法被容器调用，同时传入分别指向这个请求/响应链中的 Servlet Request、Servlet Response 和 Filter Chain 对象的引用。然后过滤器就有机会处理请求，将处理任务传递给链中的下一个资源(通过调用 Filter Chain 对象引用上的 doFilter方法)，之后在处理控制权返回该过滤器时处理响应。

析构：

容器紧跟在垃圾收集之前调用 destroy()方法，以便能够执行任何必需的清理代码。

其中chain.doFilter(req, resp)方法是把请求转发给过滤器的下一个对象，可以是web.xml里面配置的另外的filter，也可以是servlet。

在这里我们用过滤器配置一个抛出异常到指定错误页面来实践一下。

首先是web.xml的配置：

<filter>

<filter-name>Page Request Timer</filter-name>

<filter-class>cn.filter.UserFilter</filter-class><!-- 指定Filter路径 -->

</filter>

<filter-mapping>

<filter-name>Page Request Timer</filter-name>

<servlet-name>Main Servlet</servlet-name><!-- 指定在哪个servlet下去使用Filter -->

</filter-mapping>

<servlet>

<servlet-name>Main Servlet</servlet-name>

<servlet-class>cn.servlet.FilterAction</servlet-class><!-- 指定Servlet路径 -->

</servlet>

<servlet-mapping>

<servlet-name>Main Servlet</servlet-name>

<url-pattern>/filter</url-pattern><!-- 通过什么请求来访问servlet -->

</servlet-mapping>

然后是Filter类：

package cn.filter;

import java.io.IOException;

import javax.servlet.Filter;

import javax.servlet.FilterChain;

import javax.servlet.FilterConfig;

import javax.servlet.ServletException;

import javax.servlet.ServletRequest;

import javax.servlet.ServletResponse;

import javax.servlet.annotation.WebFilter;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import cn.exception.BsException;

@WebFilter("/UserFilter")

public class UserFilter implements Filter {

private String errorPage;//跳转的错误信息页

public void destroy() {

}

public void doFilter(ServletRequest req, ServletResponse resp,

FilterChain chain) throws IOException, ServletException {

HttpServletRequest request = (HttpServletRequest) req;

HttpServletResponse response = (HttpServletResponse) resp;

try {

chain.doFilter(req, resp);

} catch (RuntimeException e) {

if(e instanceof BsException){//如果是你定义的业务异常

request.setAttribute("BsException", e);//存储业务异常信息类

errorPage = "error.jsp";

request.getRequestDispatcher(errorPage).forward(request, response);//跳转到信息提示页面！！

}

e.printStackTrace();

}

}

public void init(FilterConfig config) throws ServletException {

}

}

其他的只要写一个错误页面error.jsp，一个异常实体类BsException，一个FileDaoImpl方法即可。在浏览器中输入：<http://localhost:8080/Day0918/filter>。

## 一些注意事项

From表单中action中地址=/项目名/servlet名，用submit按钮提交，前提是servlet中有注解@WebServlet("/UserAction")

Servlet中转发重定向地址从WebContent目录下开始

# Day0919

## 总结：今天主要是通过编写一个注册模块和用户列表的增删改查，来熟悉关于jsp+servlet+jdbc的基本操作，通过画图了解有关servlet的生命周期，http的请求与响应，http状态码等相关流程

## Ajax复习

Ajax的定义主要看四个步骤，分别是：

1. New一个新的xmlHttpRequest对象
2. 设置回调函数
3. 设置路径，请求方式，请求方法
4. 调用send传送数据

代码如下：

var xmlHttp = new XMLHttpRequest();

xmlHttp.onreadystatechange = callback;//不要用callback();

xmlHttp.open("post","/Day0919/RegisterAction?method=2&name="+name,true);

xmlHttp.send(null);

function callback(){

if(xmlHttp.status==200&&xmlHttp.readyState==4){

var flag = xmlHttp.responseText;

if(flag=="true"){

message="当前用户名已被使用";

}else{

message="";

}

document.getElementById("showMsg").innerHTML=message;

}

}

## Servlet生命周期，http请求与响应，http状态码

