



北京圣思园科技有限公司
<http://www.shengsiyuan.com>

主讲人：张龙

使用Servlet过滤器(重要)

- 教学目标
 - 了解Servlet过滤器的概念
 - 了解Servlet过滤器的工作过程
 - 掌握创建Servlet过滤器的方法
 - 掌握发布Servlet过滤器的方法

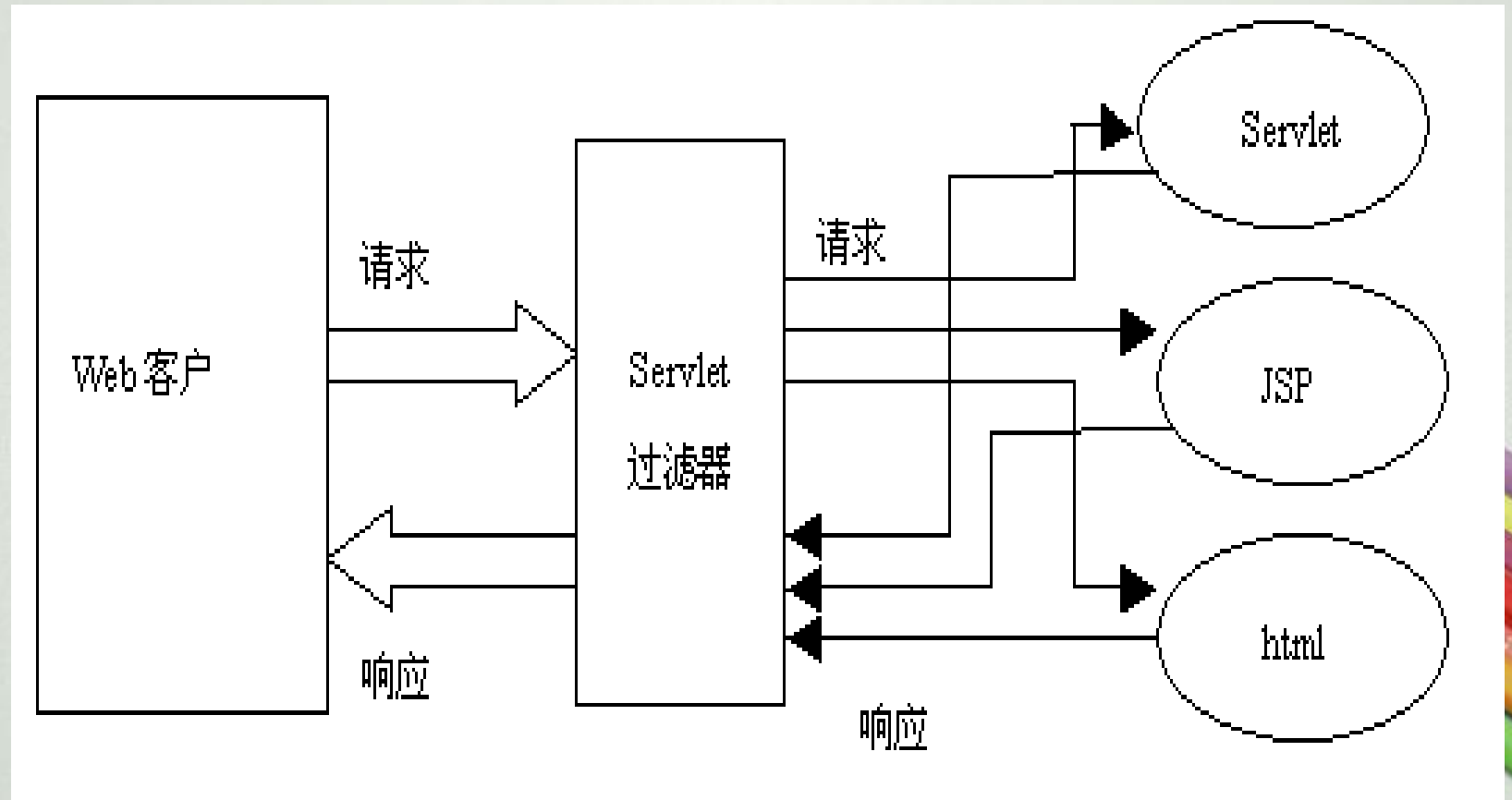


Servlet过滤器的概念

- Servlet过滤器是在Java Servlet规范2.3中定义的，它能够对Servlet容器的请求和响应对象进行检查和修改。
- Servlet过滤器本身并不生成请求和响应对象，它只提供过滤作用。
- Servlet过滤器能够在Servlet被调用之前检查Request对象，修改Request Header和Request内容；
- 在Servlet被调用之后检查Response对象，修改Response Header和Response内容。Servlet过滤器负责过滤的Web组件可以是Servlet、JSP或HTML文件。



Servlet过滤器的过滤过程



Filter接口

- 所有的 Servlet 过滤器类都必须实现 `javax.servlet.Filter` 接口。这个接口含有 3 个过滤器类必须实现的方法：
 - `init()`
 - `doFilter()`
 - `destroy()`



Filter接口

- **init(FilterConfig):** 这是Servlet过滤器的初始化方法，Servlet容器创建Servlet过滤器实例后将调用这个方法。在这个方法中可以读取web.xml 文件中Servlet过滤器的初始化参数
- **doFilter(ServletRequest, ServletResponse, FilterChain):**
这个方法完成实际的过滤操作。当客户请求访问与过滤器关联的URL时，Servlet容器将先调用过滤器的doFilter方法。FilterChain参数用于访问后续过滤器
- **destroy():** Servlet容器在销毁过滤器实例前调用该方法，在这个方法中可以释放Servlet过滤器占用的资源



过滤器的例子

- 下面创建一个**NoteFilter**过滤器，它可以拒绝列在黑名单上的客户访问留言簿，
- 参考**NoteFilter.java**



NoteFilter过滤器

- 当 NoteFilter 初始化时，它调用 `config.getInitParameter("blacklist")` 方法，从 `web.xml` 文件中读取初始化参数 `blacklist`，这个参数表示被禁止访问留言簿的客户黑名单。

```
public void init(FilterConfig config) throws ServletException {  
    this.config = config;  
    blackList=config.getInitParameter("blacklist");  
}
```



NoteFilter过滤器

- 在NoteFilter的doFilter()方法中首先从request对象中读取客户姓名，然后将客户姓名转换为中文字符编码。如果客户姓名中包含黑名单里的字符串，那么将直接向客户端返回一个拒绝网页。由于在这种情况下没有调用chain.doFilter()方法，因此客户请求不会到达所访问的Web组件。



NoteFilter过滤器

//获得username请求参数，进行字符编码转换

```
String username =
```

```
    ((HttpServletRequest) request).getParameter("username");
```

```
if (username!=null )
```

```
    username=
```

```
        new  
        String(username.getBytes("ISO-8859-  
1"),"GB2312");
```

```
String(username.getBytes("ISO-8859-
```



NoteFilter过滤器

//进行身份验证

```
if (username!=null && username.indexOf(blackList) != -1 ) {  
    response.setContentType("text/html;charset=GB2312");  
    PrintWriter out = response.getWriter();  
    out.println("<html><head></head><body>");  
    out.println("<h1>对不起,"+username + ",你没有权限留言  
                </h1>");  
    out.println("</body></html>");  
    out.flush();  
    return;  
}
```

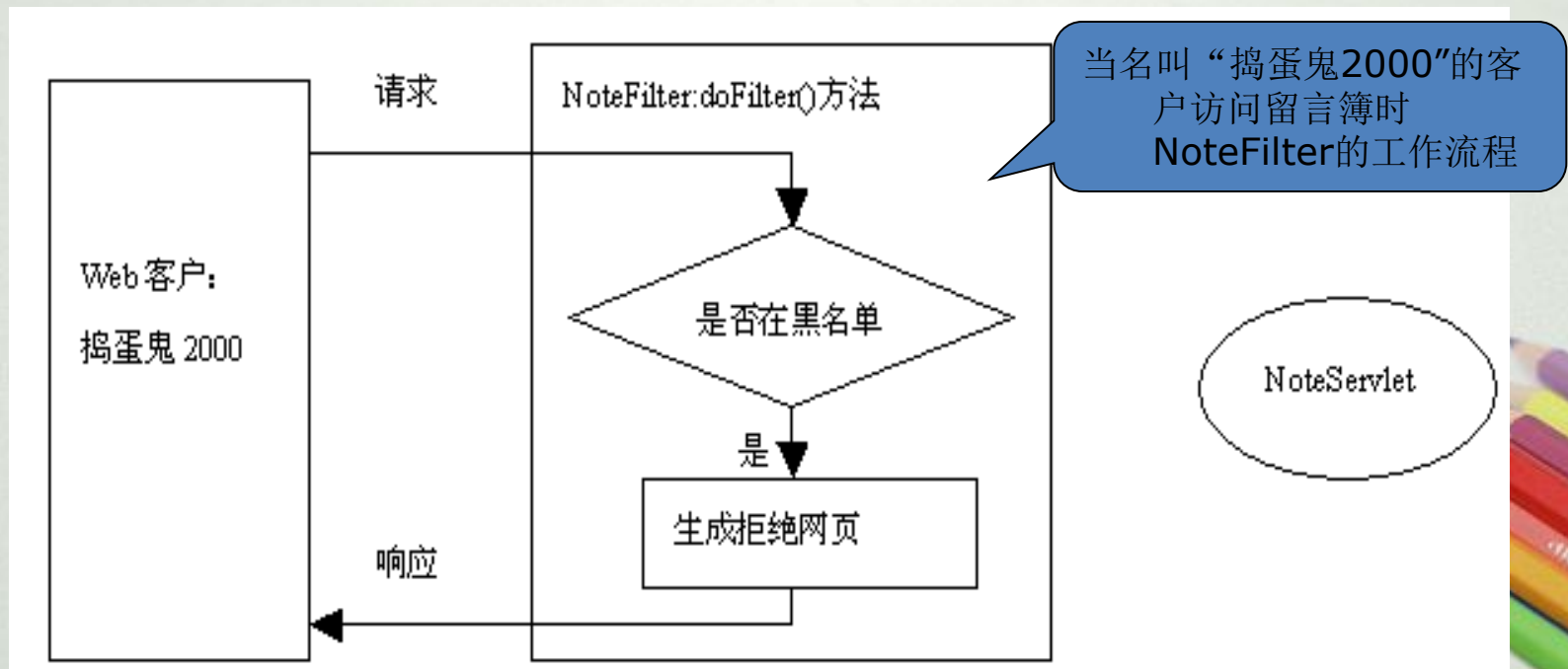


NoteFilter过滤器拒绝HTTP请求

- 假定姓名中包含“捣蛋鬼”字符串的客户将被禁止访问留言簿，并且留言簿由NoteServlet类来实现，当名叫“捣蛋鬼2000”的客户访问留言簿时，将被NoteFilter过滤器拒绝访问。



NoteFilter过滤器拒绝HTTP请求




NoteFilter过滤器转发HTTP请求

- 如果客户名不在黑名单里，NoteFilter的doFilter方法就会调用chain.doFilter()方法，这个方法用于调用过滤器链中后续过滤器的doFilter()方法。
- 假如没有后续过滤器，那么就把客户请求传给相应的Web组件。
- 在本例中，在调用chain.doFilter()方法前后记录了调用的时间，从而计算出Web组件响应客户请求所花的时间。

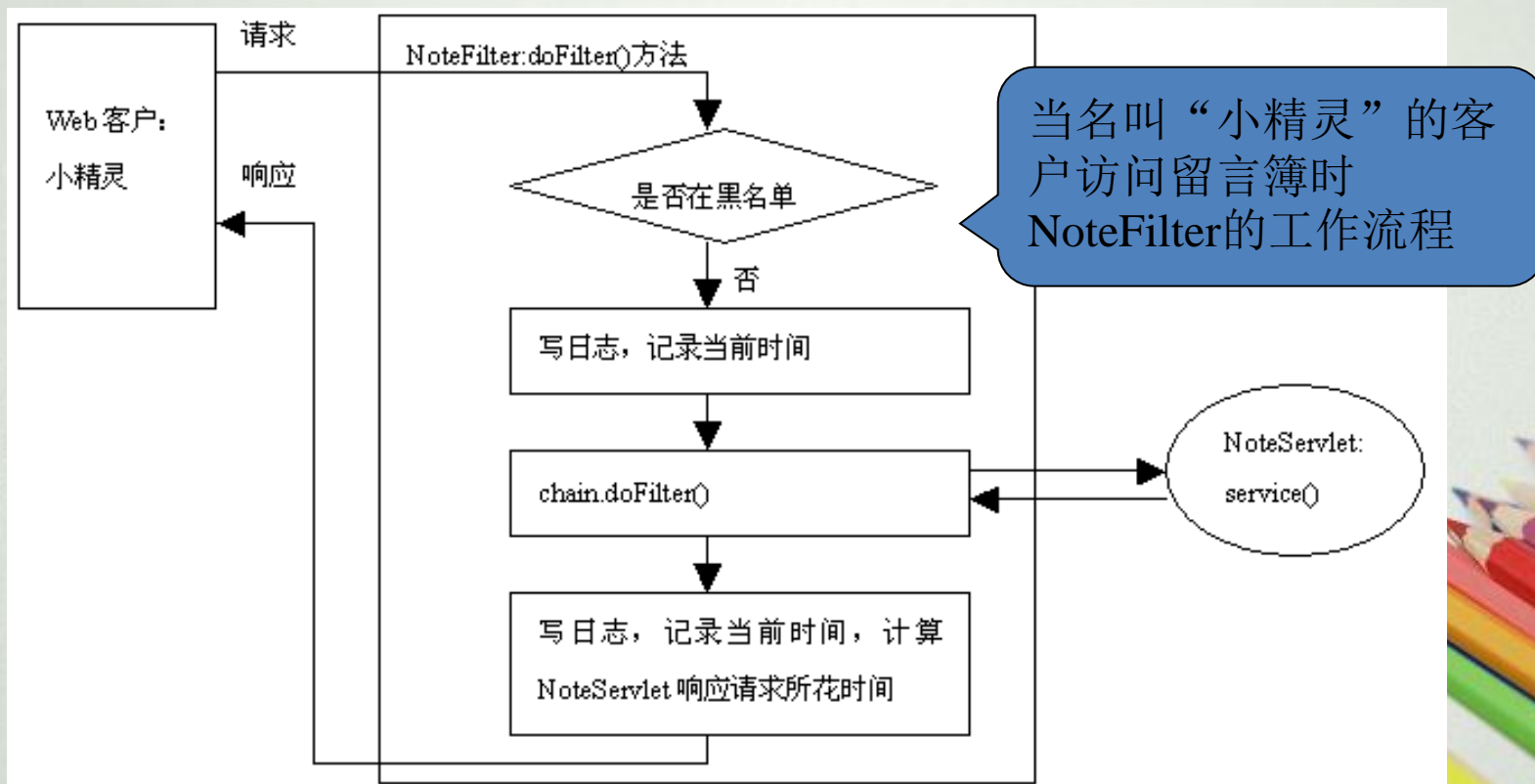


NoteFilter过滤器转发HTTP请求

```
long before = System.currentTimeMillis();  
chain.doFilter(request, response);  
long after = System.currentTimeMillis();  
String name = "";  
if (request instanceof HttpServletRequest) {  
    name  
    =  
    ((HttpServletRequest)request).getRequestURI(  
    );  
}  
Long time = after - before;  
System.out.println("所用时间: " + time + "ms");
```



NoteFilter过滤器转发HTTP请求



发布Servlet过滤器

- 发布 Servlet 过滤器时，必须在 web.xml 文件中加入 `<filter>` 元素和 `<filter-mapping>` 元素。`<filter>` 元素用来定义一个过滤器：

`<filter>`

`<filter-name>NoteFilter</filter-name>`

`<filter-class>NoteFilter</filter-class>`

`<init-param>`

`<param-name>blacklist</param-name>`

`<param-value>捣蛋鬼</param-value>`

`</init-param>`

`</filter>`



发布Servlet过滤器

- `<filter-mapping>` 元素用于将过滤器和URL关联:

```
<filter-mapping>
```

```
    <filter-name>NoteFilter</filter-name>
```

```
    <url-pattern>/note</url-pattern>
```

```
</filter-mapping>
```

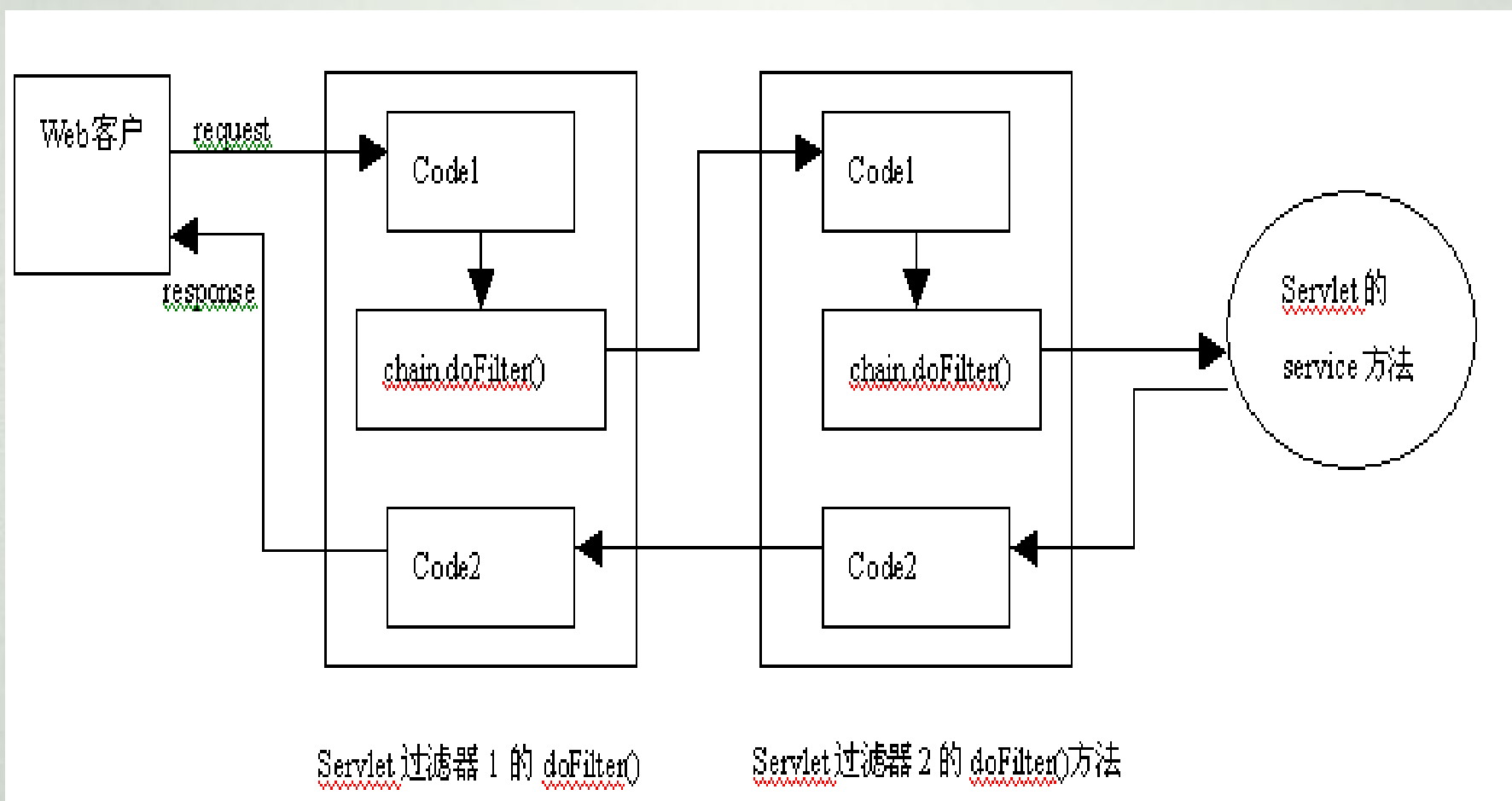
提示:在web.xml文件中,
必须先配置Servlet过滤
器, 再配置Servlet

创建NoteServlet类

- 创建一个Servlet类NoteServlet，它实现一个简单的留言簿。它提供了一个表单，让客户输入姓名和留言，客户提交表单后，再将用户输入的信息显示在客户端的网页上。
- 访问NoteServlet:



串联Servlet过滤器的工作流程



配置串联的Servlet过滤器

```
<filter>
```

```
    <filter-name>Filter1</filter-name>
```

```
    <filter-class>Filter1</filter-class>
```

```
</filter>
```

```
<filter>
```

```
    <filter-name>Filter2</filter-name>
```

```
    <filter-class>Filter2</filter-class>
```

```
</filter>
```



配置串联的Servlet过滤器

```
<filter-mapping>
```

```
    <filter-name>Filter1</filter-name>
```

```
    <url-pattern>/note</url-pattern>
```

```
</filter-mapping>
```

```
<filter-mapping>
```

```
    <filter-name>Filter2</filter-name>
```

```
    <url-pattern>/note</url-pattern>
```

```
</filter-mapping>
```



练习题1

- 问题：在Servlet过滤器中能否访问application范围内的共享数据？



- 答案:可以的, 先调用FilterConfig的getServletContext()方法获得ServletContext, 再调用ServletContext的getAttribute()方法来获得application范围内的共享数据。



练习题2

- 问题：“Servlet过滤器只能对Servlet进行过滤”，这句话是否正确？



- 答案:不正确。Servlet过滤器可以对Servlet、JSP和HTML文件过滤。

