

JSP 프로그래밍

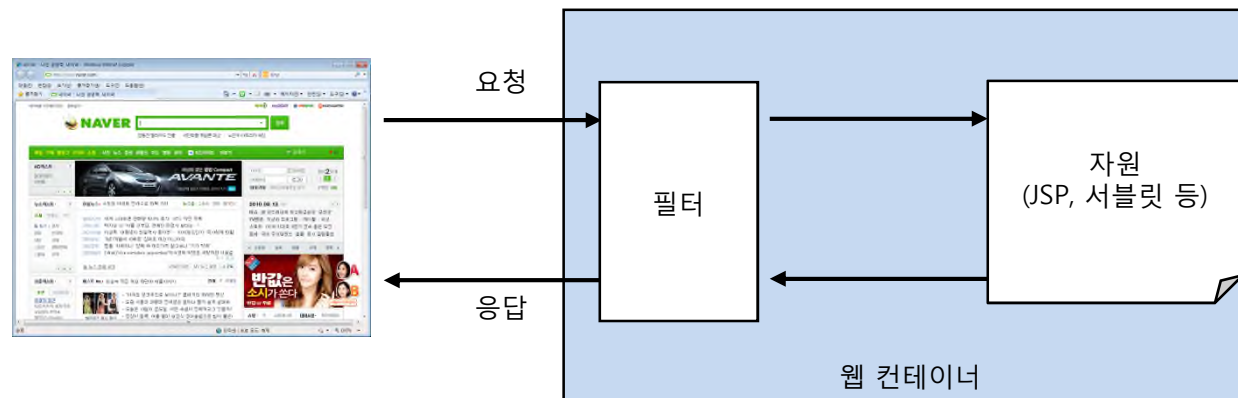
19 필터(Filter)

목차

- ❖ 필터란 무엇인가
- ❖ 필터의 구현
- ❖ 필터의 응용

필터란 무엇인가?

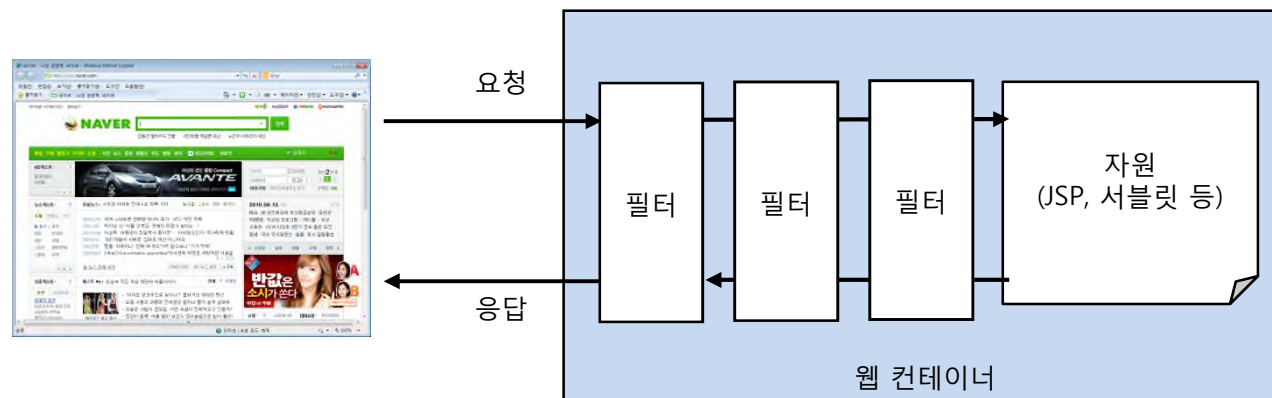
- ❖ 필터는 "HTTP 요청과 응답을 변경할 수 있는 재사용 가능한 코드"이다.
- ❖ 필터는 객체의 형태로 존재하며 클라이언트로부터 오는 요청(request)과 최종 자원(서블릿/JSP/기타 문서) 사이에 위치하여 클라이언트의 요청 정보를 알맞게 변경할 수 있으며, 또한 필터는 최종 자원과 클라이언트로 가는 응답(response) 사이에 위치하여 최종 자원의 요청 결과를 알맞게 변경할 수 있다.



[그림] 필터의 기본구조

필터란 무엇인가?

- ❖ 필터는 정보를 변경하는 역할뿐만 아니라 흐름을 변경하는 역할도 할 수 있다. 즉, 필터는 클라이언트의 요청을 필터 체인의 다음 단계(결과적으로는 클라이언트가 요청한 자원)에 보내는 것이 아니라 다른 자원의 결과를 클라이언트에 전송할 수도 있다.
- ❖ 필터의 이러한 기능은 사용자 인증이나 권한 체크와 같은 기능을 구현할 때 사용할 수 있다.



[그림] 필터 체인의 구성

필터의 구현

- ❖ 필터를 구현하는 데 있어 핵심은 다음의 3개의 타입이다.
- ❖ `javax.servlet.Filter` 인터페이스 : 클라이언트와 최종 자원 사이에 위치하는 필터를 나타내는 객체가 구현해야 하는 인터페이스.
- ❖ `javax.servlet.ServletRequestWrapper` 클래스 : 필터가 요청을 변경한 결과를 저장하는 래퍼이다.
- ❖ `javax.servlet.ServletResponseWrapper` 클래스 : 필터가 응답을 변경하기 위해 사용하는 래퍼이다.

필터의 구현

❖ Filter 인터페이스

- `public void init(FilterConfig config) throws ServletException`
 - : 필터를 초기화할 때 호출된다.
- `public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain) throws IOException, ServletException`
 - : 필터 기능을 수행한다. chain을 이용해서 체인의 다음 필터로 처리를 전달할 수 있다.
- `public void destroy()`
 - : 필터가 웹 컨테이너에서 삭제될 때 호출된다.

필터의 구현

```
public void init(FilterConfig config) throws ServletException {  
    // 필터 초기화 작업  
}  
  
public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)  
    throws IOException, ServletException {  
  
    // 1. 전처리(Pre-Process)지역, request 파라미터를 이용하여 요청의 필터 작업 수행  
    ...  
    // 2. 체인의 다음 필터 처리  
    chain.doFilter(request, response);  
  
    // 3. 후처리 (Post-process) 지역, response를 이용하여 응답의 필터링 작업 수행  
    ...  
}  
  
public void destroy() {  
    // 주로 필터가 사용한 자원을 반납  
}
```

필터의 구현

❖ FilterConfig가 제공하는 메서드

메서드	리턴 타입	설 명
getFilterName()	String	설정 파일에서 <filter-name>에 지정한 필터의 이름을 리턴한다.
getInitParameter(String name)	String	설정 파일의 <init-param>에서 지정한 초기화 파라미터의 값을 읽어온다. 존재하지 않는 경우 null을 리턴한다.
getInitParameterNames()	Enumeration<String>	초기화 파라미터의 이름 목록을 구한다.
getServletContext()	ServletContext	서블릿 컨텍스트 객체를 구한다.

필터 설정하기 : web.xml 이용

```
<filter>
  <filter-name>encodingFilter</filter-name>
  <filter-class>kr.or.nextit.web.filter.CharacterEncodingFilter</filter-class>
  <init-param>
    <param-name>encoding</param-name>
    <param-value>utf-8</param-value>
  </init-param>
</filter>

<filter-mapping>
  <filter-name>encodingFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>

<!-- 특정 서블릿에 대한 요청에 대해서 필터링. -->
<filter-mapping>
  <filter-name>AuthCheckFilter</filter-name>
  <servlet-name>FileDownload</servlet-name>
</filter-mapping>
```