

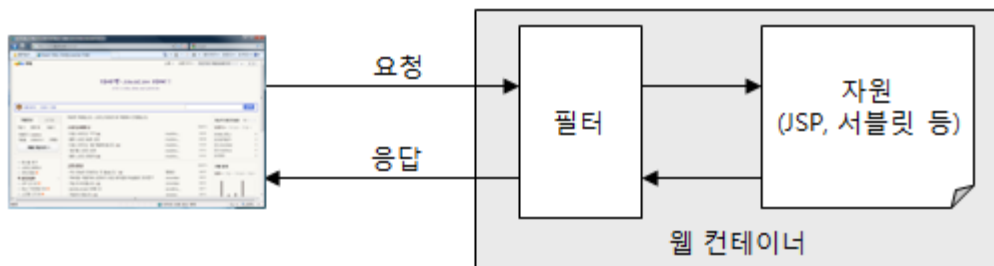
필터

# TOC

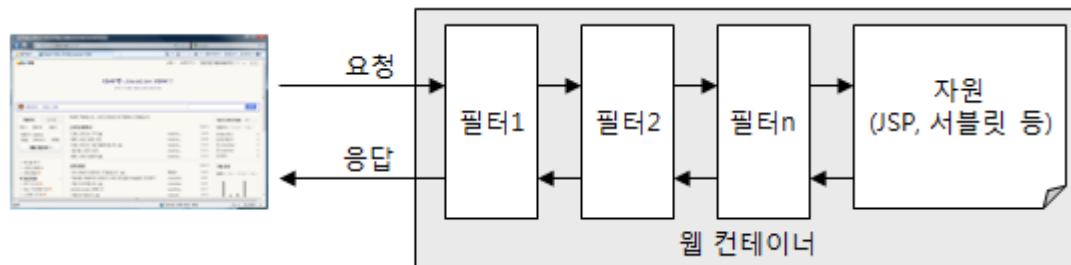
- 필터란
- 필터의 구현
- 필터 응용

# 필터

- HTTP 요청과 응답을 변경할 수 있는 재사용 가능한 코드
- 필터의 기본 구조



- 요청의 내용을 변경하거나 응답의 내용을 변경 가능
- 1개 이상의 필터 연동 가능



# 필터 구현

- Filter 인터페이스 사용
- Filter 인터페이스의 메서드
  - `public void init(FilterConfig filterConfig) throws ServletException`  
필터를 초기화할 때 호출된다.
  - `public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain) throws java.io.IOException, ServletException`  
체인을 따라 다음에 존재하는 필터로 이동한다. 체인의 가장 마지막에는 클라이언트가 요청한 최종 자원이 위치한다.
  - `public void destroy()`  
필터가 웹 컨테이너에서 삭제될 때 호출된다.

# 필터 구현

- doFilter() 메서드에서 필터 기능 구현

```
public void doFilter(ServletRequest request,
                    ServletResponse response
                    FilterChain chain)
    throws IOException, ServletException {
    // 1. request 파라미터를 이용하여 요청의 필터 작업 수행
    ...
    // 2. 체인의 다음 필터 처리
    chain.doFilter(request, response);

    // 3. response를 이용하여 응답의 필터링 작업 수행
    ...
}
```

# 필터 설정

- web.xml에 URL 별 매핑 설정 추가

```
<web-app ...>

  <filter>
    <filter-name>FilterName</filter-name>
    <filter-class>javacan.filter.FileClass</filter-class>
    <init-param>
      <param-name>paramName</param-name>
      <param-value>value</param-value>
    </init-param>
  </filter>

  <filter-mapping>
    <filter-name>FilterName</filter-name>
    <url-pattern>*.jsp</url-pattern>
  </filter-mapping>

  ...
</web-app>
```

# 필터 설정

- <dispatcher> 를 통한 필터 적용 시점 설정

```
<filter-mapping>  
  <filter-name>AuthCheckFilter</filter-name>  
  <servlet-name>FileDownload</servlet-name>  
  <dispatcher>INCLUDE</dispatcher>  
</filter-mapping>
```

- REQUEST - 클라이언트의 요청인 경우에 필터를 사용 (기본값)
- FORWARD - forward()를 통해서 제어를 이동하는 경우에 필터를 사용
- INCLUDE - include()를 통해서 포함하는 경우에 필터를 사용

# 필터 설정

- @WebFilter 어노테이션 이용

```
import javax.servlet.annotation.WebFilter;

@WebFilter(filterName="xsltFilter", urlPatterns={ "/xml/*", "/xsl/*" })
public class XSLTFilter implements Filter{
    ... 필터 구현
}
```

- urlPatterns : 필터를 적용할 URL 패턴 목록을 지정.
- servletNames : 필터를 적용할 서블릿 이름 목록을 지정.
- filterName : 필터의 이름을 지정.
- initParams : 초기화 파라미터 목록을 지정.
- dispatcherTypes : 필터를 적용할 범위를 지정. 기본값은  
DispatchType.REQUEST



# 요청 및 응답 래퍼

- 요청이나 응답을 변경할 때 사용
  - 요청 래퍼: `HttpServletRequestWrapper`
  - 응답 래퍼: `HttpServletResponseWrapper`
- `doFilter()`에서 `chain.doFilter()`를 호출할 때, 래퍼 객체를 전달해 줌으로써 다음 작업 가능
  - 요청 변경: 파라미터 변경, 헤더 변경, 전송 데이터 변경 등
  - 응답 변경: 응답 데이터 변경, 압축 등

## 필터의 응용

- 데이터 변환(다운로드 파일의 압축/데이터 암호화/이미지 변환 등)
- XSL/T를 이용한 XML 문서 변경
- 사용자 인증
- 캐싱 필터
- 자원 접근에 대한 로깅