

2025년 상반기 K-디지털 트레이닝

# 필터

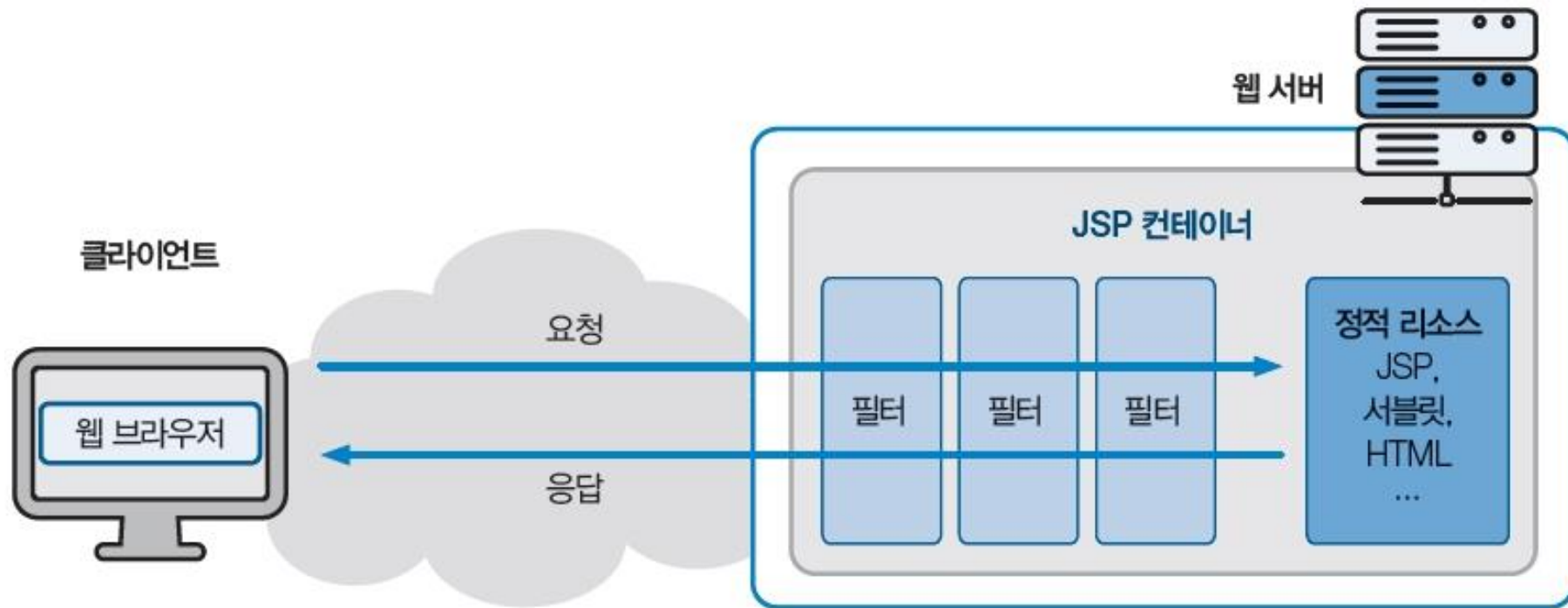
---

[KB] IT's Your Life

# 1 필터의 개요

## ✓ 필터(filter)

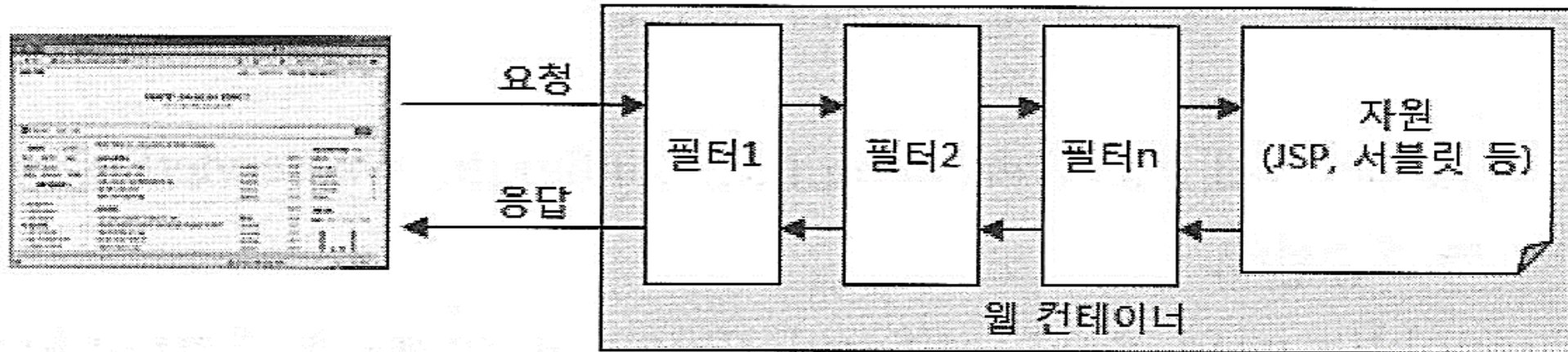
- 클라이언트와 서버 사이에서 request와 response 객체를 먼저 받아 사전/사후 작업 등 공통적으로 필요한 부분을 처리하는 것



# 1 필터의 개요

## ✓ Filter

- 다수의 Filter를 체인으로 연결 가능



- ex04의 form method를 post로 변경하면?
  - 모든 서블릿에서 utf-8인코딩 설정을 처리해야 함.
  - 필터에서 미리 인코딩을 설정해 줌

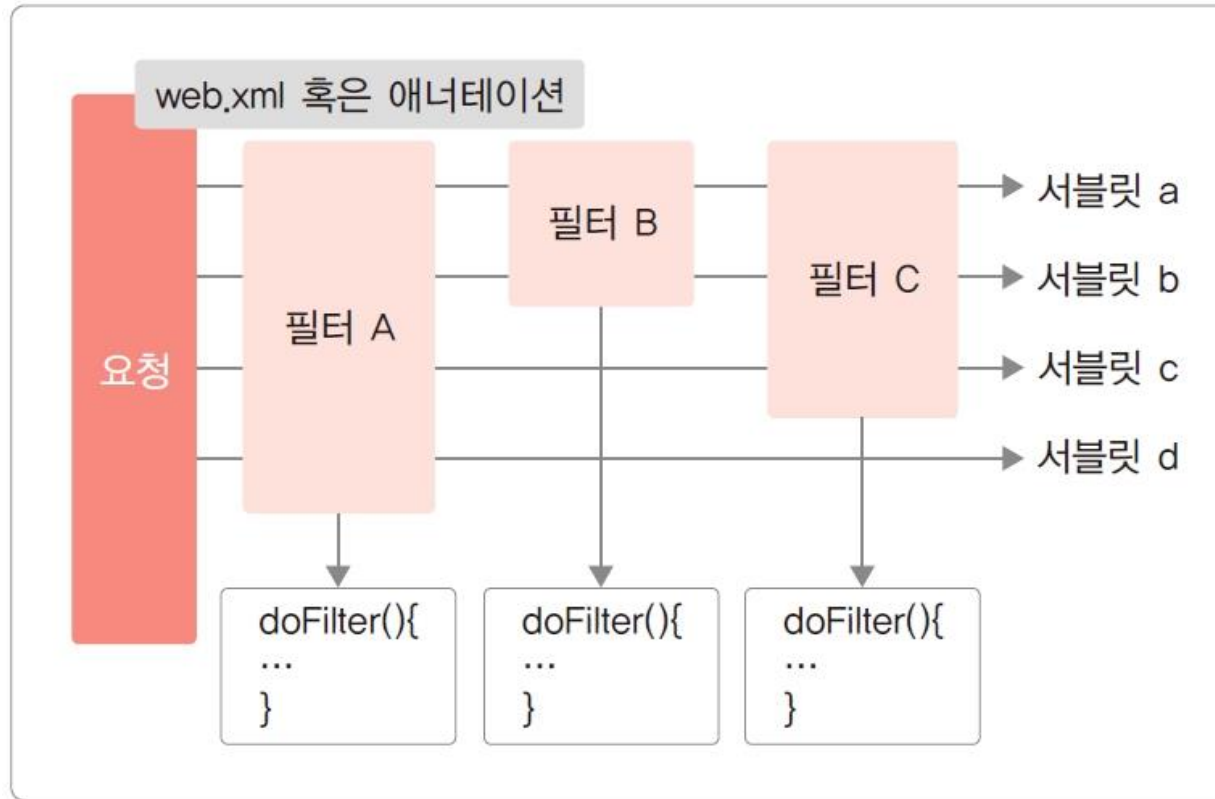
# 1 필터의 개요

## ✓ 필터의 활용

필터	기능
Request 필터	<ul style="list-style-type: none"><li>· 인증(사용자 인증)</li><li>· 요청 정보를 로그 파일로 작성</li><li>· 암호화 인코딩 작업</li></ul>
Response 필터	<ul style="list-style-type: none"><li>· 응답 결과 데이터 압축</li><li>· 응답 결과에 내용 추가/수정</li><li>· 총 서비스 시간 측정</li></ul>

# 1 필터의 개요

## ✓ 필터의 동작 구조



## 2 Filter 인터페이스

### ✓ Filter 인터페이스

- 필터 기능을 구현하는 데 핵심적인 역할을 함
- 클라이언트와 서버의 리소스 사이에 위치한 필터의 기능을 제공하기 위해 자바 클래스로 구현

```
import jakarta.servlet.*;
public class 클래스 이름 implements Filter {
    ...(생략)...
}
```

### ○ Filter 인터페이스 메소드

메소드	설명
init()	필터 인스턴스의 초기화 메소드이다.
doFilter()	필터 기능을 작성하는 메소드이다.
destroy()	인스턴스의 종료 전에 호출되는 메소드이다.

## 2 Filter 인터페이스

### ✓ init() 메소드

- JSP 컨테이너가 필터를 초기화할 때 호출되는 메소드
- JSP 컨테이너 내에서 초기화 작업을 수행할 필터 인스턴스를 생성한 후 한번만 호출됨
- 메소드는 JSP 컨테이너에 의해 호출되어 필터의 서비스가 시작되고 있음을 나타냄

```
public void init(FilterConfig filterConfig) throws ServletException
```

### ○ FilterConfig 인터페이스 메소드

메소드	반환 유형	설명
getFilterName()	String	web.xml 파일에 설정된 필터 이름을 반환한다.
getInitParameter(String name)	String	web.xml 파일에 설정된 매개변수에 대한 매개변수 값을 반환한다. 초기화 매개변수가 존재하지 않으면 null을 반환한다.
getInitParameterNames()	Enumeration<String>	web.xml 파일에 설정된 모든 매개변수 이름을 포함하는 Enumeration 객체 타입을 반환한다. 초기화 매개변수가 존재하지 않으면 비어 있는 Enumeration을 반환한다.
getServletContext()	ServletContext	ServletContext 객체를 반환한다.

## 2 Filter 인터페이스

### ✓ doFilter() 메소드

- JSP 컨테이너가 필터를 리소스에 적용할 때마다 호출되는 메소드
- init( ) 메소드 후에 호출되며, 필터가 어떤 기능을 수행할 필요가 있을 때마다 호출

```
public void doFilter(ServletRequest request,  
                    ServletResponse response,  
                    FilterChain chain)  
    throws java.io.IOException, ServletException
```

- ① 첫 번째 매개변수 ServletRequest 객체 : 체인을 따라 전달하는 요청
- ② 두 번째 매개변수 ServletResponse 객체 : 체인을 따라 전달할 응답
- ③ 세 번째 매개변수 FilterChain 객체 : 체인에서 다음 필터를 호출하는 데 사용됨

→ 만약 호출 필터가 체인의 마지막 필터이면 체인의 끝에서 리소스를 호출함



## 2 Filter 인터페이스

### ✓ destroy() 메소드

- 필터 인스턴스를 종료하기 전에 호출하는 메소드
- JSP 컨테이너가 필터 인스턴스를 삭제하기 전에 청소 작업을 수행하는 데 사용되며, 이는 필터로 열린 리소스를 모두 닫을 수 있는 방법
- destroy( ) 메소드는 필터의 수명 동안 한 번만 호출됨

```
public void destroy()
```

## 2 Filter 인터페이스

### ✓ 필터 등록

#### ○ web.xml에 등록하는 경우

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app ...>
  :
  <filter>
    <filter-name>myFilter</filter-name>
    <filter-class>org.scoula.filter.MyFilter</filter-class>
  </filter>

  <filter-mapping>
    <filter-name>myFilter</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
  :
</web-app>
```

#### ○ @WebFilter을 이용해 등록하는 경우

```
@WebFilter (urlPatterns={ "/" })
public class CharacterEncodeFilter implements Filter {
  ...
}
```

## 2 Filter 인터페이스

### ✓ CharacterEncodingFilter

- 모든 요청에 대한 요청 문자셋 설정을 필터에서 하도록 함
  - Servlet에서 개별적으로 설정할 필요 없어짐
  - 일반적으로 제일 먼저 해야하는 작업이 됨

## 2 Filter 인터페이스

### org.scoula.ex04.filter.CharacterEncodeFilter.java

```
@WebFilter (urlPatterns={ "/" })
public class CharacterEncodeFilter implements Filter {
    @Override
    public void init(FilterConfig fConfig) throws ServletException {
    }

    @Override
    public void destroy() {
    }

    @Override
    public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
        throws IOException, ServletException {
        request.setCharacterEncoding("UTF-8");

        chain.doFilter(request, response);
    }
}
```

## 2 Filter 인터페이스

### ✓ ex04의 form method를 post로 변경

- 모든 서블릿에 doPost() 처리 추가!

@Override

```
protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
    doGet(req, resp);  
}
```