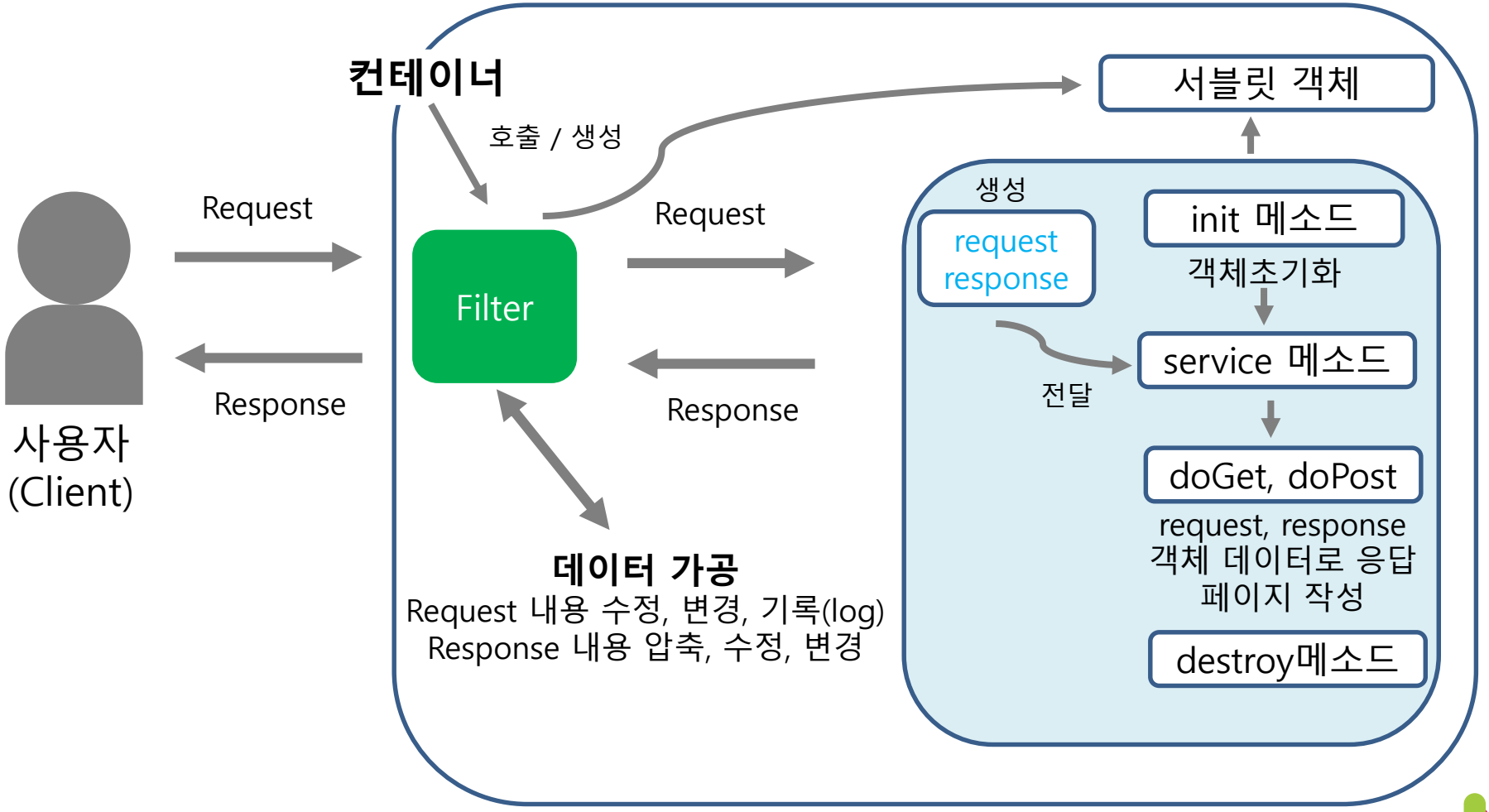
KH정보교육원

서블릿 필터와 래퍼

서블릿 필터

KH정보교육원

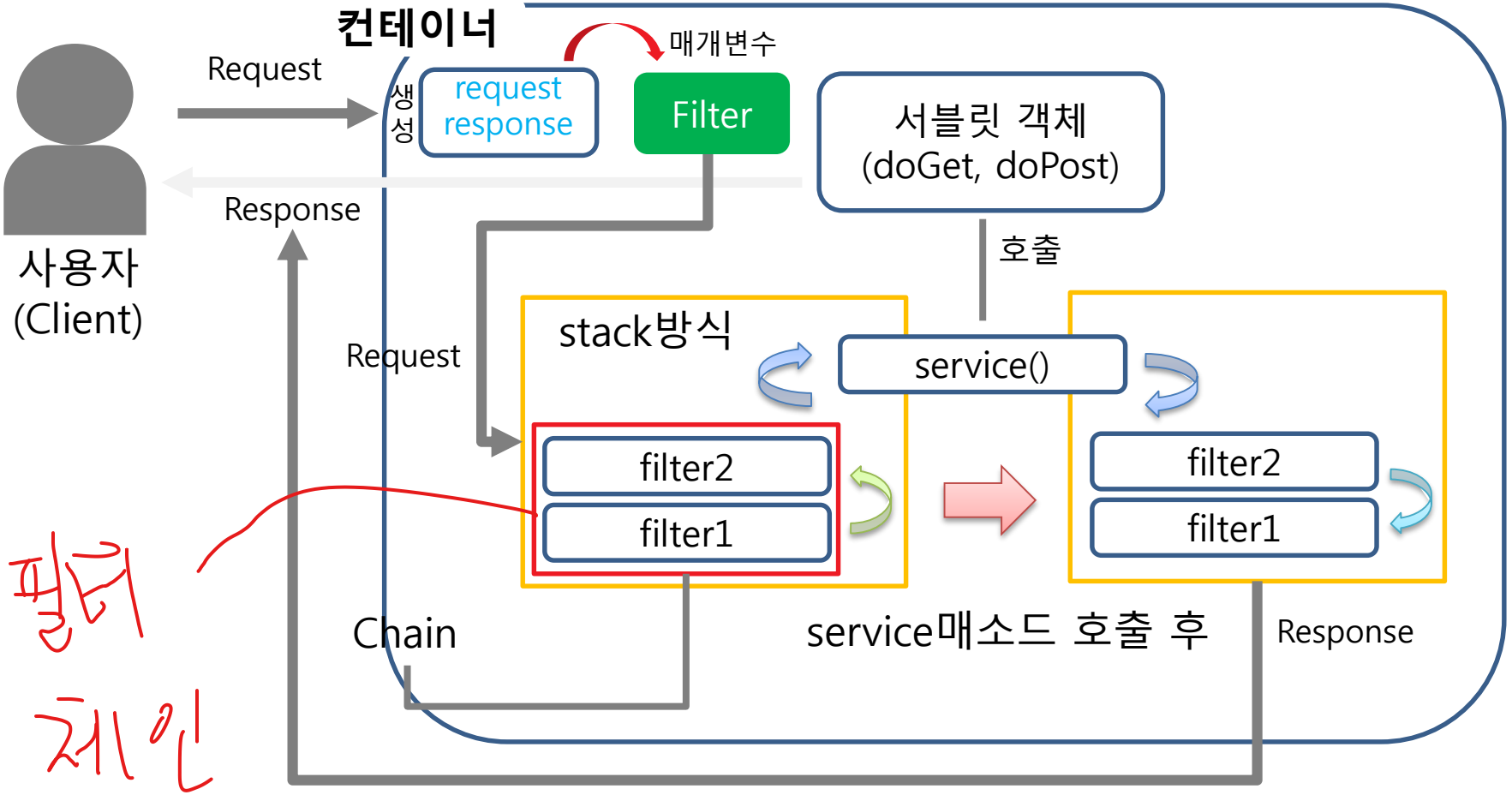
서블릿 필터 동작 구조



서블릿 필터

KH정보교육원

서블릿 필터 내부동작



서블릿 필터

KH정보교육원

서블릿 필터란?

커스텀 필터 제작 가능

javax.servlet.Filter 인터페이스를 상속 받아 구현하는 클래스.
HTTP 요청과 응답 사이에서 전달되는 데이터를 가로채어 서비스에 맞게 변경하고 걸러내는 필터링 작업을 수행할 수 있는 클래스이다.

** Servlet과 비슷한 모습을 보임 (init, destroy, doFilter)

처리내용

Request : 보안관련사항, 요청헤더와 바디 형식지정, 요청에 대한 log기록 유지
Response : 응답 스트림압축, 응답스트림 내용 추가 및 수정, 새로운 응답 작성

** 여러 가지 필터를 연결(chain / 서로 호출)하여 사용가능

서블릿 필터

KH정보교육원

DD설정(web.xml)

Filter등록

```
<filter>
    <filter-name>필터설정이름</filter-name>
    <filter-class>필터를 구현한 클래스</filter-class>
    <init-param> // filter에서 사용할 값 설정
      <param-name>초기값설정이름</param-name>
       <param-value>설정값</param-value>
    </init-param>
 </filter>
```

서블릿 필터

KH정보교육원

DD설정(web.xml)

Filter-Mapping(url패턴과 매핑)

```
<filter-mapping>
    <filter-name>등록된 필터이름</filter-name>
    <url-pattern>요청한 페이지 형식</url-pattern>
</filter-mapping>
```

서블릿과 필터 맵핑(필터 적용 서블릿지정)

```
<filter-mapping>
    <filter-name>등록된 필터이름</filter-name>
    <servlet-name>적용할 서블릿명</servlet-name>
</filter-mapping>
```

** 매핑하는방법이 두가지 / url-pattern이 우선적용

↳ 어노테이션 메핑도 ㅇㅇ

서블릿 필터

KH정보교육원

클래스 설정(java코드)

```
public class 클래스명 implements Filter
{
    @Override
    public void init(FilterConfig config) throws ServletException
    {          Filter호출시 작업 설정          }
    @Override
    public void doFilter(ServletRequest req, ServletResponse resp,
                         FilterChain chain) throws ServletException,
                                                   IOException
    {          필터링 작업할 내용          }
    @Override
    public void destroy()
    {  삭제시 작업 설정                    }
}
```

서블릿 필터

KH정보교육원

@(annotation)적용 설정

@WebFilter를 활용하여 annotation으로 필터를 설정할 수 있다.

** 필터를 구현한 클래스에 선언하면 된다.

설정방법

구 분
@WebFilter("요청url")
@WebFilter(value="요청url")
@WebFilter(urlPatterns="패턴")
@WebFilter(servletNames="서블릿이름")
@WebFilter(servletnames={"서블릿1", "서블릿2"})
@WebFilter(urlPatterns="url패턴", initParams=@webInitParam(name="이름", value="값"))

서블릿 필터

KH정보교육원

Filter Interface

init(FilterConfig config);

웹 컨테이너가 필터를 호출할 경우 해당 메소드가 호출되어
필터 객체를 생성하며 초기화한다.
** 매개변수 FilterConfig는 web.xml에 있는 <filter>정보를 가지고 있음.

doFilter(ServletRequest req, ServletResponse res, FilterChain chain)

필터가 수행될 때 구동하는 메소드로, 요청 객체와 응답 객체를 사용해
일련의 작업을 수행한 뒤, chain을 통해 가공된 값을 목적지로 전송한다.

destroy();

역할이 끝난 필터는 웹 컨테이너에 의해 해당 메소드를 호출하고
소멸된다.

서블릿 필터

KH정보교육원

예제

CharsetEncodingFilter Class

```
package test.common.filter;

import java.io.IOException;

public class CharsetEncodingFilter implements Filter {

    @Override
    public void init(FilterConfig config) throws ServletException {}

    @Override
    public void doFilter(ServletRequest req, ServletResponse res, FilterChain chain)
            throws IOException, ServletException {

        req.setCharacterEncoding("UTF-8");
        res.setContentType("text/html; charset=UTF-8");

        chain.doFilter(req, res);

    }

    @Override
    public void destroy() {}
}
```

예제 파일 : test.common.filter.CharsetEncodingFilter.java

서블릿 필터

KH정보교육원

예제

작성한 Filter를 web.xml 에 등록한다.

```
<filter>
 <filter-name>EncodingFilter</filter-name>
 <filter-class>test.common.filter.CharsetEncodingFilter</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>
```

예제 파일 : web/WEB-INF/web.xml

서블릿 필터 참고

KH정보교육원

FilterChain (interface)

필터는 chain처럼 서로 연결되어 있는데 연결 되어있는 필터를 순차별로 doFilter() 매소드를 이용하여 실행시키는 인터페이스, 마지막 필터가 실행된 후에는 service() 매소드를 실행시켜 서블릿의 매소드(doGet(), doPost())를 실행

doFilter(ServletRequest req, ServletResponse)

chain으로 연결되어있는 다음 필터를 실행하는 매소드

서블릿 필터 참고

KH정보교육원

dispatcher처리

2.4버전부터 dispatcher의 요청도 필터를 적용할 수 있음.

Filter-Mapping(url패턴과 매핑)

```
<filter-mapping>
    <filter-name>등록된 필터이름</filter-name>
    <url-pattern>요청한 페이지 형식</url-pattern>
<dispatcher>
   REQUEST || INCLUDE || FORWARD || ERROR
</dispatcher>
 </filter-mapping>
```

서블릿 래퍼

KH정보교육원

서블릿 래퍼란?

관련 클래스(ServletRequest, ServletResponse, HttpServletRequest, HttpServletResponse)를 내부에 보관하며, 내부에 있는 클래스가 구현하고 있는 매소드로 위임(구현이 되어있다는 의미)하여 모든 매소드가 구현된 상태이기 때문에 필요한 매소드만 구현하면 됨.

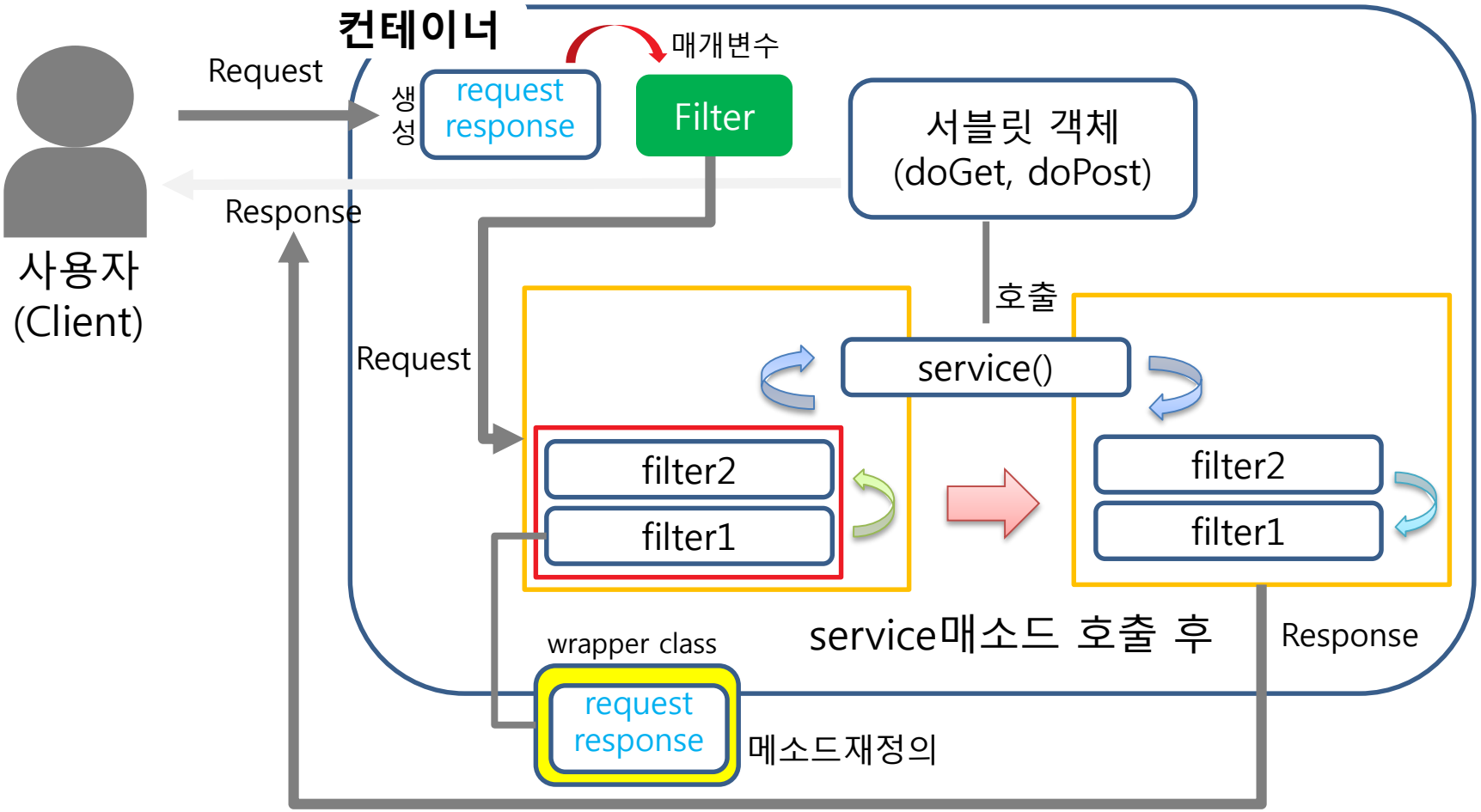
** 즉, java Event처리의 Adapter클래스와 비슷한 기능

서블릿 래퍼 종류

클래스명	내 용
ServletRequestWrapper	ServletRequest객체를 구현한 객체를 가지고 있음
ServletResponseWrapper	ServletResponse객체를 구현한 객체를 가지고 있음
HttpServletRequestWrapper	HttpServletRequest객체를 구현한 객체를 가지고 있음
HttpServletResponseWrapper	HttpServletResponse객체를 구현한 객체를 가지고 있음

KH정보교육원

서블릿 필터 내부동작



서블릿 래퍼생성

KH정보교육원

Wrapper클래스를 상속한 클래스 생성

```
class 클래스명 extends Wrapper클래스명
{
        생성자(HttpServletRequest / response 매개변수명)
        { super(매개변수명); }

        @Override
         // 원하는 매소드 오버라이딩 처리
         ex) public String getParameter(String name)
             {
                  if(super.getParameter(name).equals("kim") return "김";
                  else return "김이 아닙니다";
              }
}
```