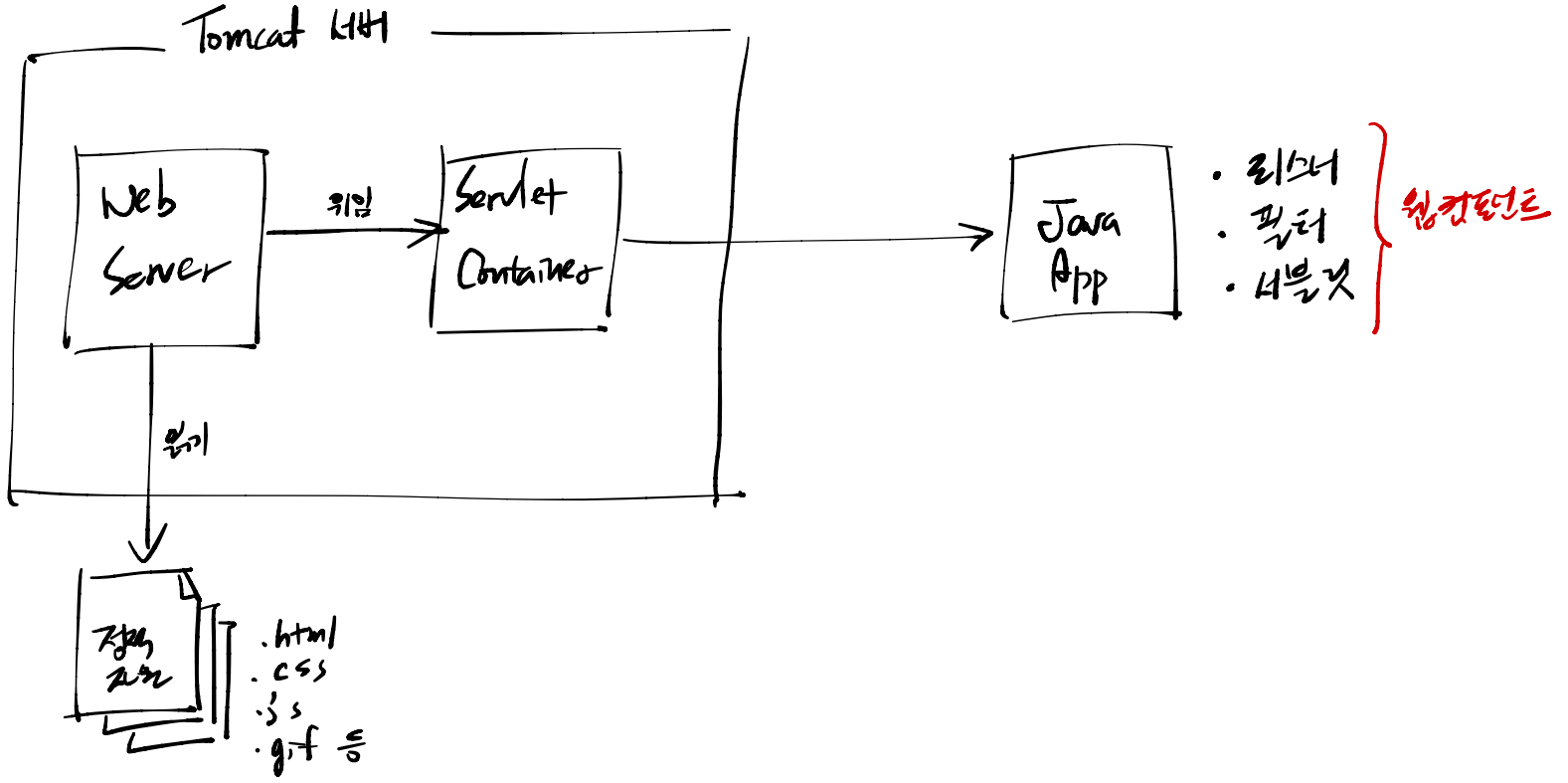
* 웹 컴포넌트(부품)

↳ 한 개 이상의 객체로 구성되어 특정 작업을 수행하는 것. = 특별한 작업을 수행하는 객체

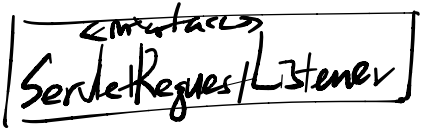


* 웹 컴포넌트 — 리스너 = 옵저버(Observer) GoF

★ 내가 특정 상태에 놓일 때 실행되는 객체 ⇒ "리스너"

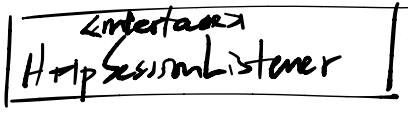

← 서블릿 컨테이너가 시작되거나 종료될 때 호출됨

contextInitialized() contextDestroyed()




← 클라이언트의 요청이 들어오거나 응답하거나 끝났을 때 호출됨

requestInitialized() requestDestroyed()

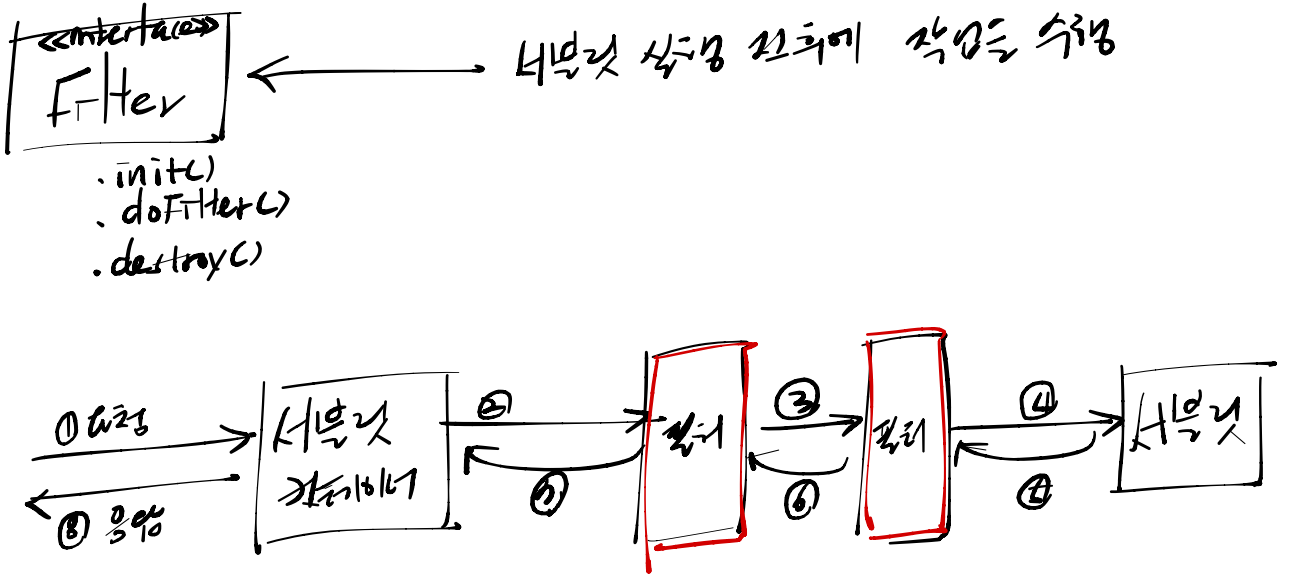



← 클라이언트 세션이 생성되었을 때 또는 종료되었을 때 호출됨

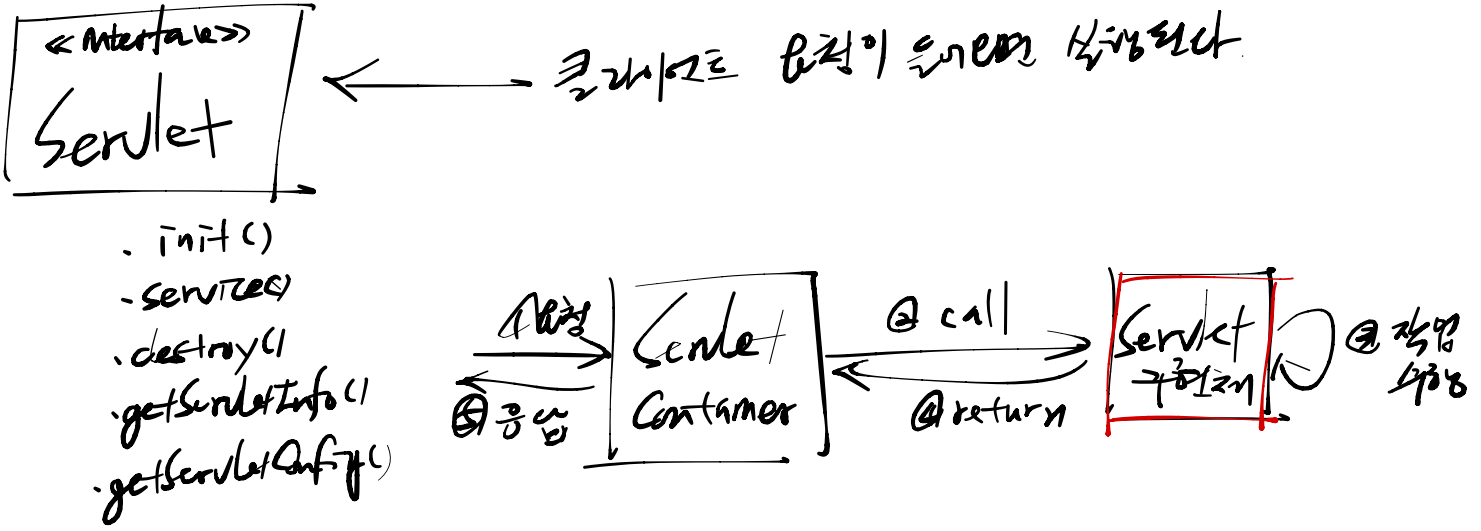
sessionCreated() sessionDestroyed()

⋮

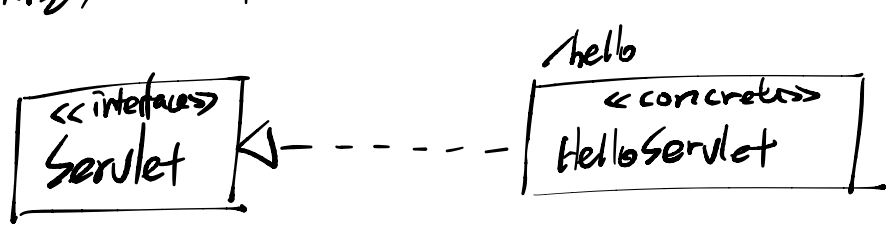
* 웹 애플리케이션 - 필터 = Chain of Responsibility
GoF의 디자인 패턴



* 웹 컴포넌트 — 서블릿 = Command
GoF의 디자인 패턴

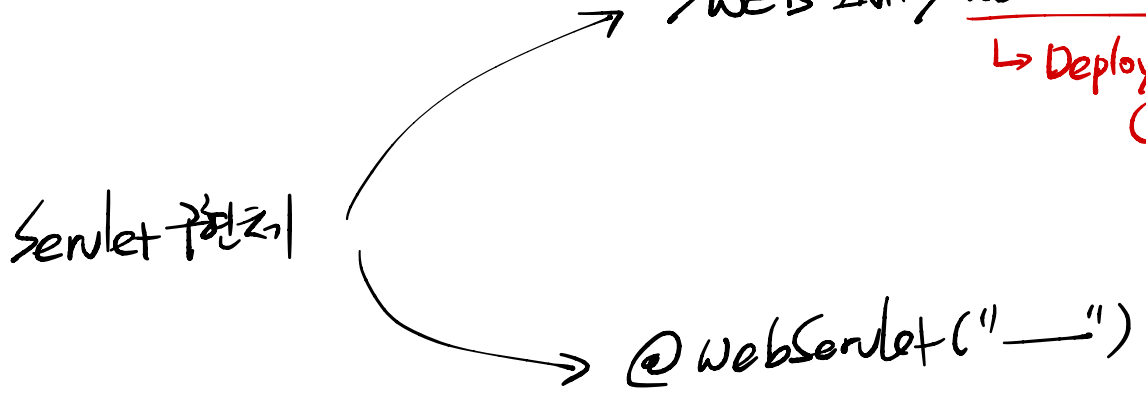


* 서블릿 만들기

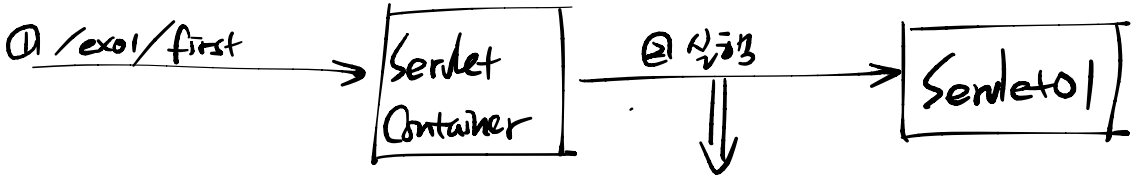



.init()
.service()
. destroy()
. getServletInfo()
. getServletConfig()

* 서블릿 배치 하기


/WEB-INF/web.xml
↳ Deployment Descriptor 파일
(DD File)
Servlet 관리
@WebServlet("—")

* 서블릿 실행하기



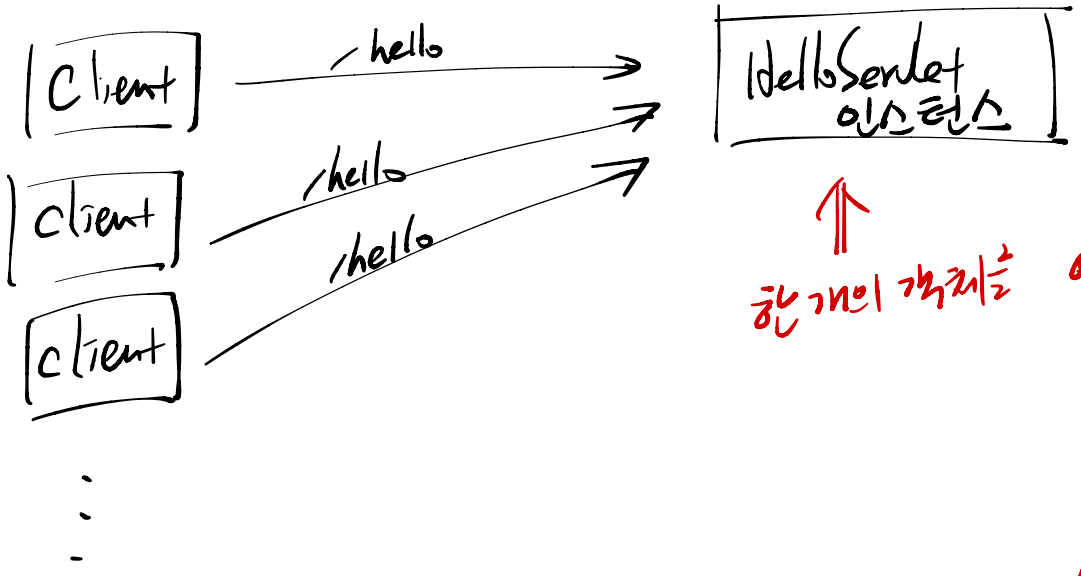

1) 객체가 없으면,
   객체생성 → init() 호출 ← 서블릿이 작업할 때 사용한 객체를 초기화하기 위해

2) service() 호출

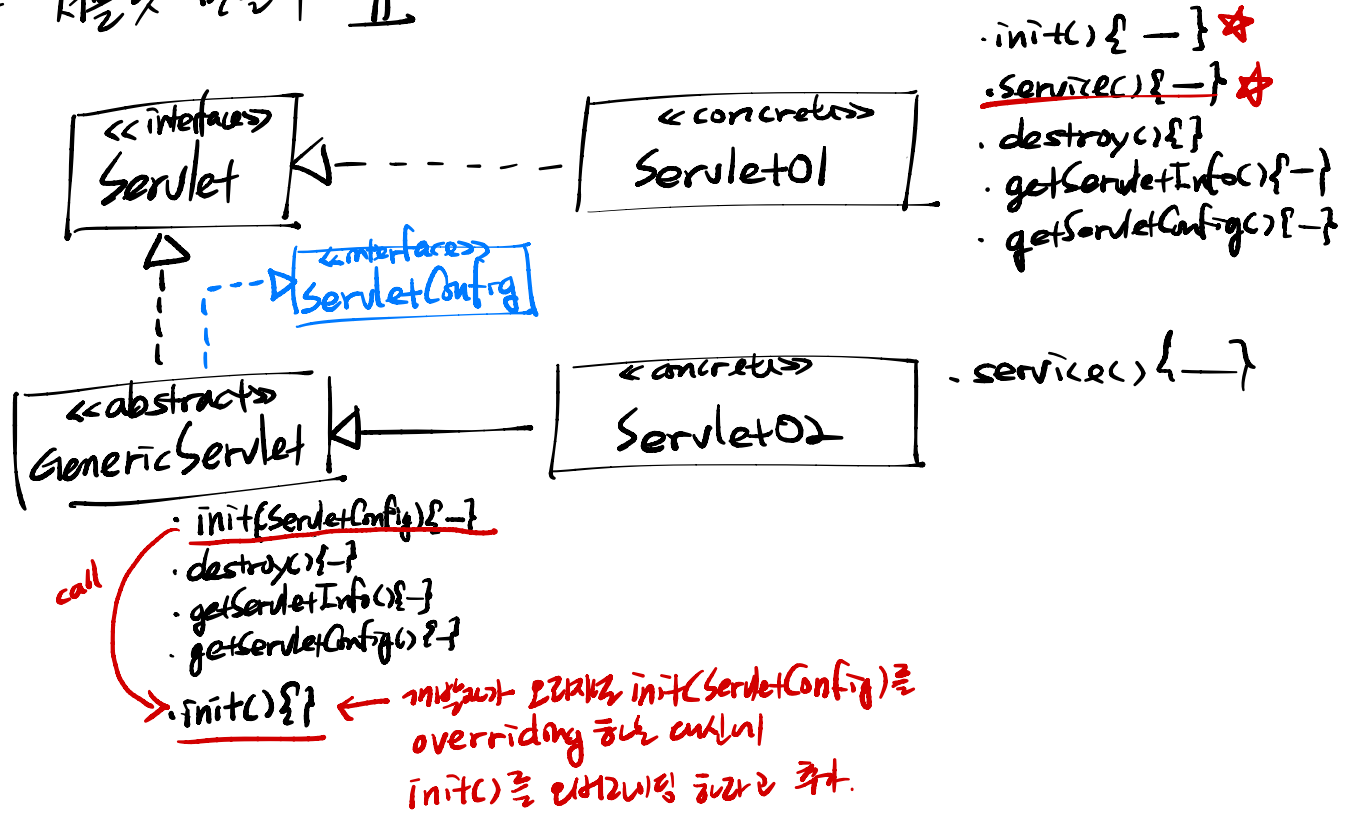
3) 웹애플리케이션종료,
   destroy() 호출 ← 서버종료전에 init() 에서 초기화했던 자원을 해제하기 위해

init()
service()
destroy()
} Servlet Lifecycle과 관련된 메서드

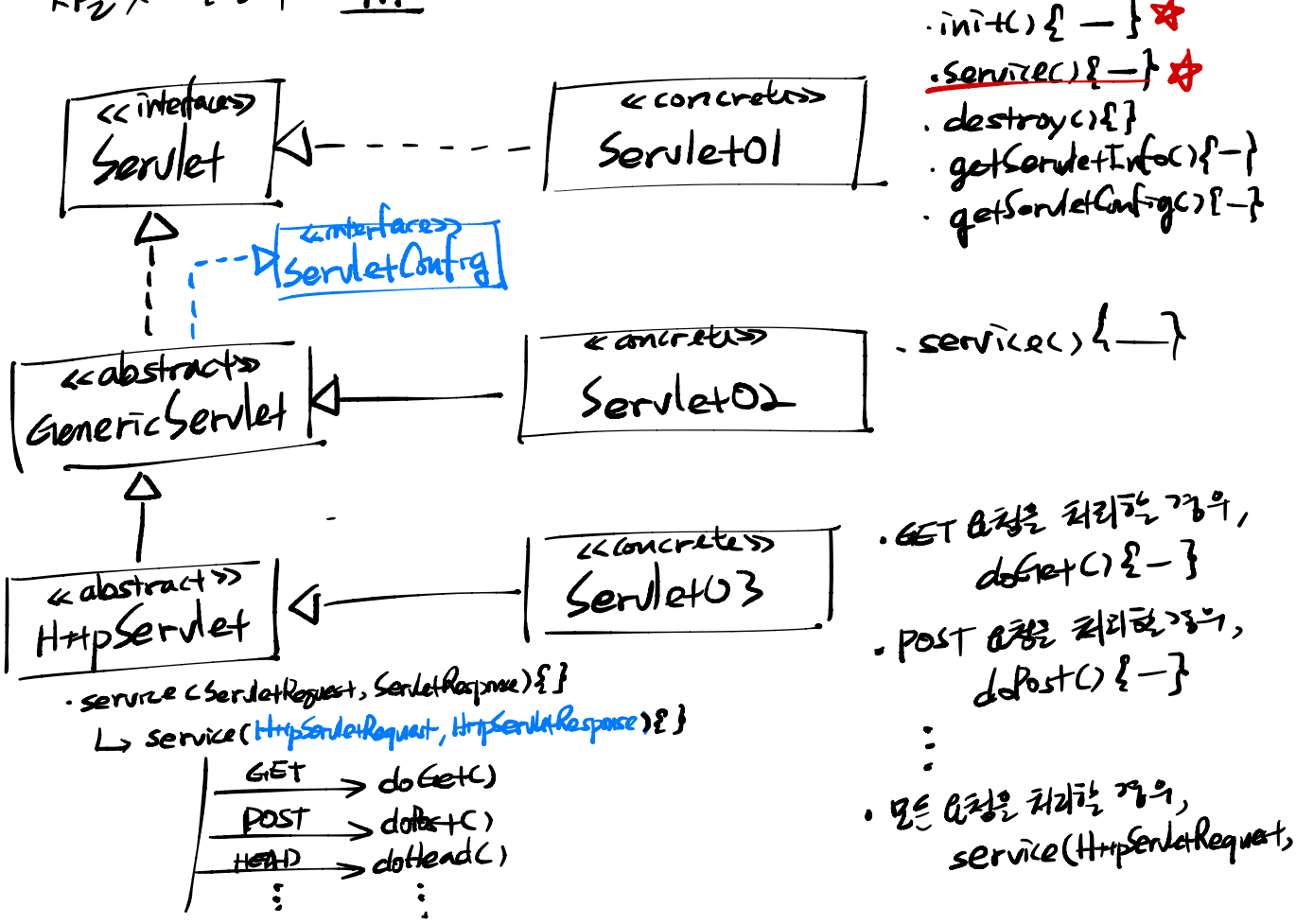
* 클라이언트 요청과 서블릿



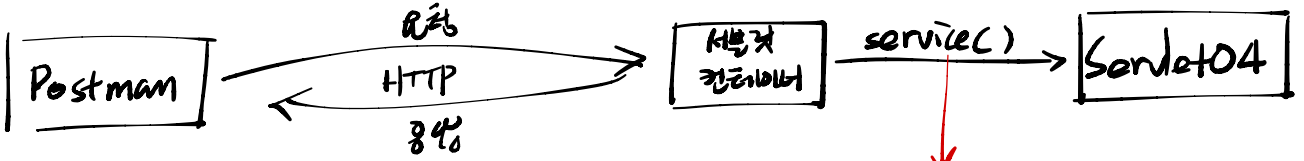
* 서블릿 만들기 II


<<interface>>
Servlet
<<concrete>>
Servlet01
.init(){ — } ☆
.service(){ — } ☆
.destroy(){}
.getServletInfo(){ — }
.getServletConfig(){ — }
<<interface>>
ServletConfig
<<abstract>>
GenericServlet
<<concrete>>
Servlet02
.service(){ — }
.init(ServletConfig){ — }
.destroy(){ — }
.getServletInfo(){ — }
.getServletConfig(){ — }
call
.init(){}
개발자가 오버라이딩 init(ServletConfig)를
overriding 하는 대신에
init()를 오버라이딩 하라고 함.

# * 서블릿 만들기 III


<<interface>>
Servlet
<<concrete>>
Servlet01
.init(){ — } ☆
.service(){ — } ☆
.destroy(){}
.getServletInfo(){ — }
.getServletConfig(){ — }
<<interface>>
ServletConfig
<<abstract>>
GenericServlet
<<concrete>>
Servlet02
.service(){ — }
<<abstract>>
HttpServlet
<<concrete>>
Servlet03
.service(ServletRequest, ServletResponse){}
↳ service(HttpServletRequest, HttpServletResponse){}
GET → doGet()
POST → doPost()
HEAD → doHead()
.GET 요청을 처리할 경우, doGet(){ — }
.POST 요청을 처리할 경우, doPost(){ — }
.모든 요청을 처리할 경우, service(HttpServletRequest, HttpServletResponse)

* Backend App 테스트




```
GET /ex01/s4 HTTP/1.1
User-Agent: PostmanRuntime/7.41.2
Accept: */*
Cache-Control: no-cache
Postman-Token: e5384013-7560-4dd8-af7d-
c9b5bbad22a7
Host: localhost:8888
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

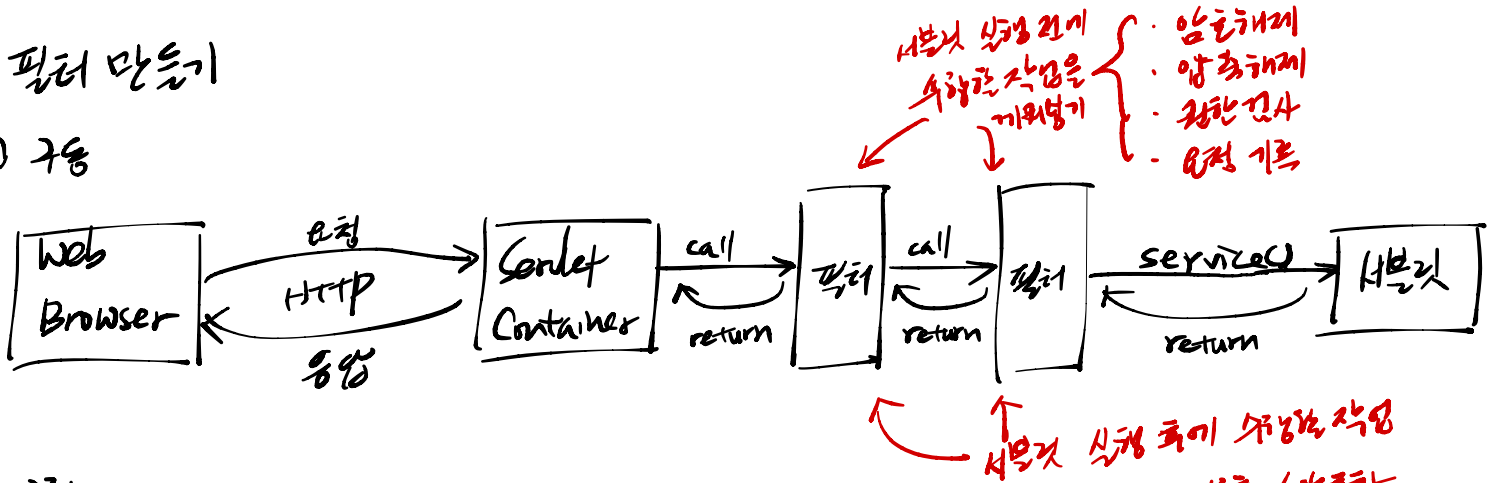
service(HttpServletRequest, HttpServletResponse)

getMethod()

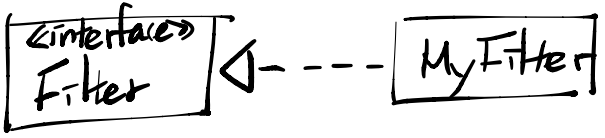
"GET"

# * 필터 만들기

① 구동



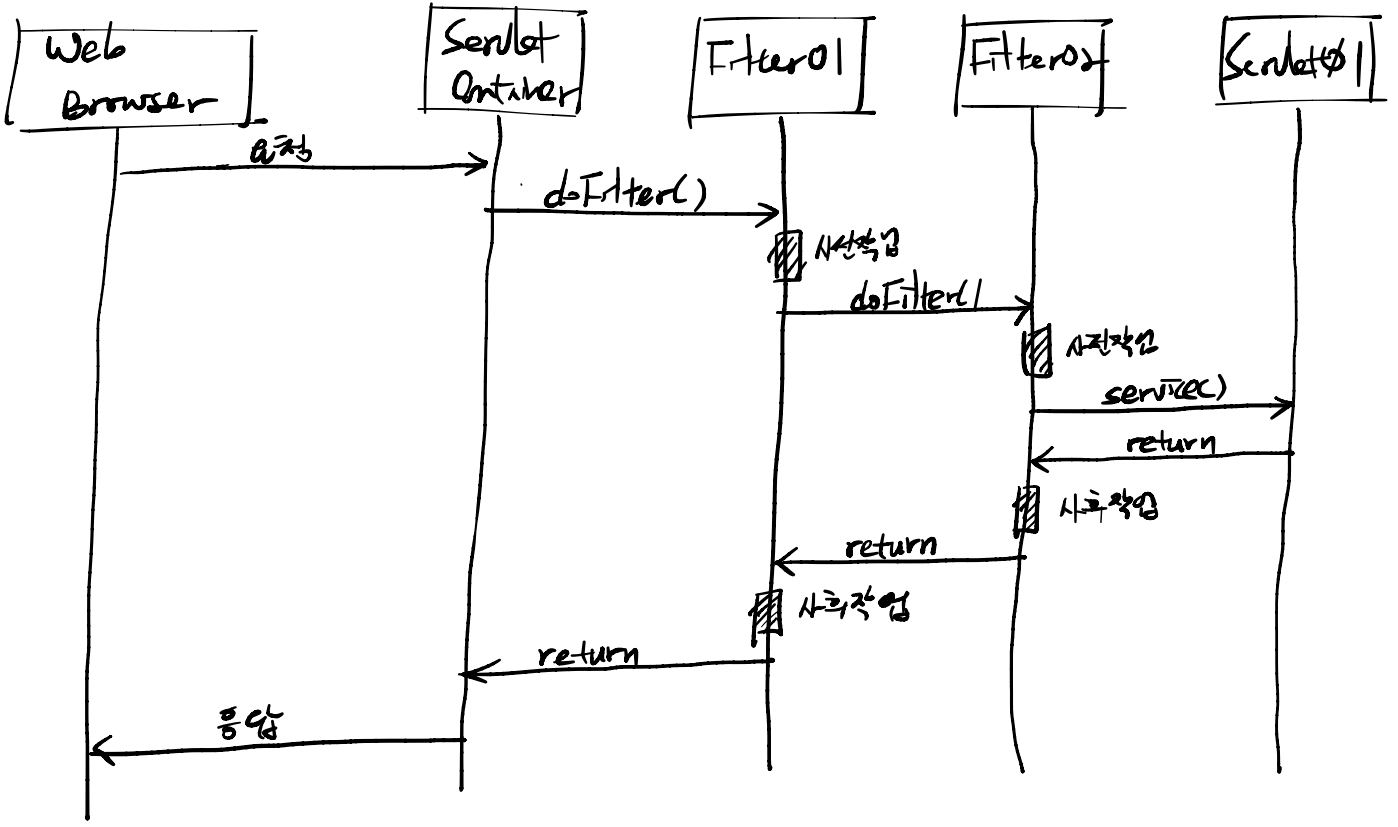

② 구현



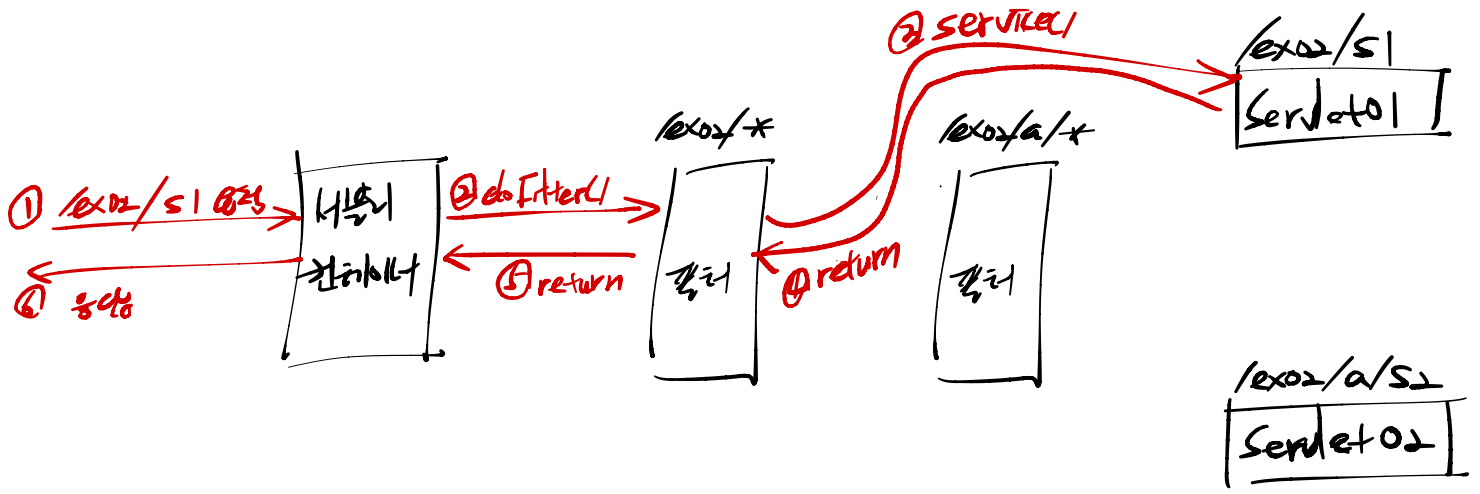

- init()
- doFilter()
- destroy()

```
doFilter() {
        사전작업
        다음 필터 또는 서블릿
        사후 작업
}
```

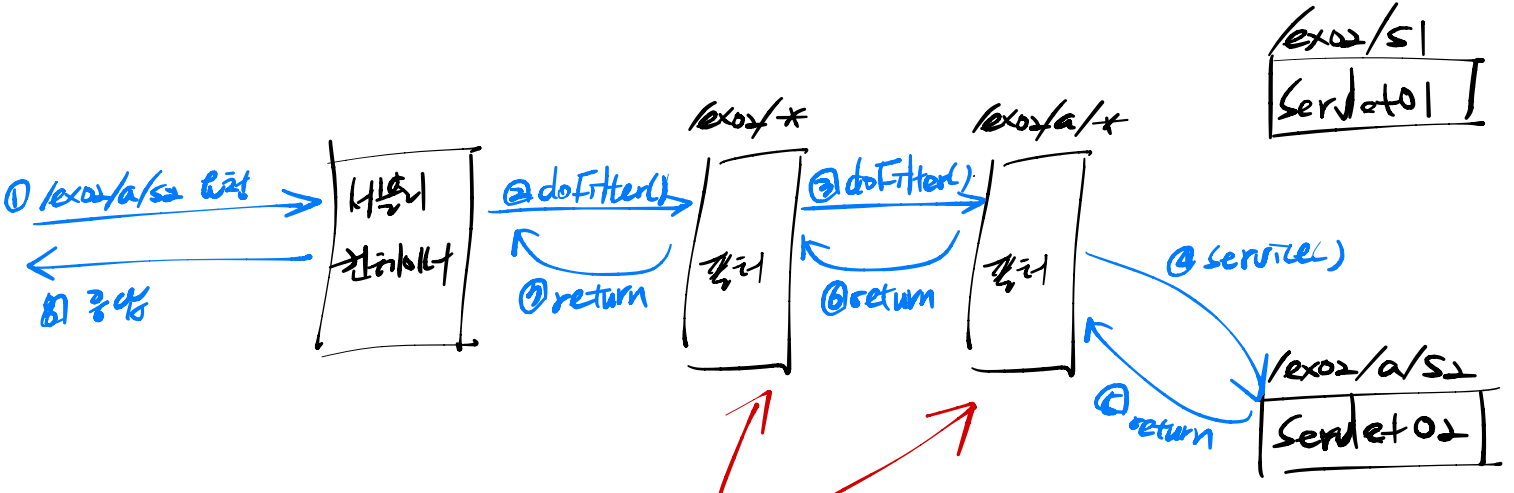
* 필터 실행 흐름


Web Browser
Servlet Container
Filter01
Filter02
Servlet01
요청
doFilter()
사전작업
doFilter()
사전작업
service()
return
사후작업
return
사후작업
return
응답

* 필터 실행 예) /ex02/s1 요청



* 필터 실행 예) /ex02/a/s2 요청



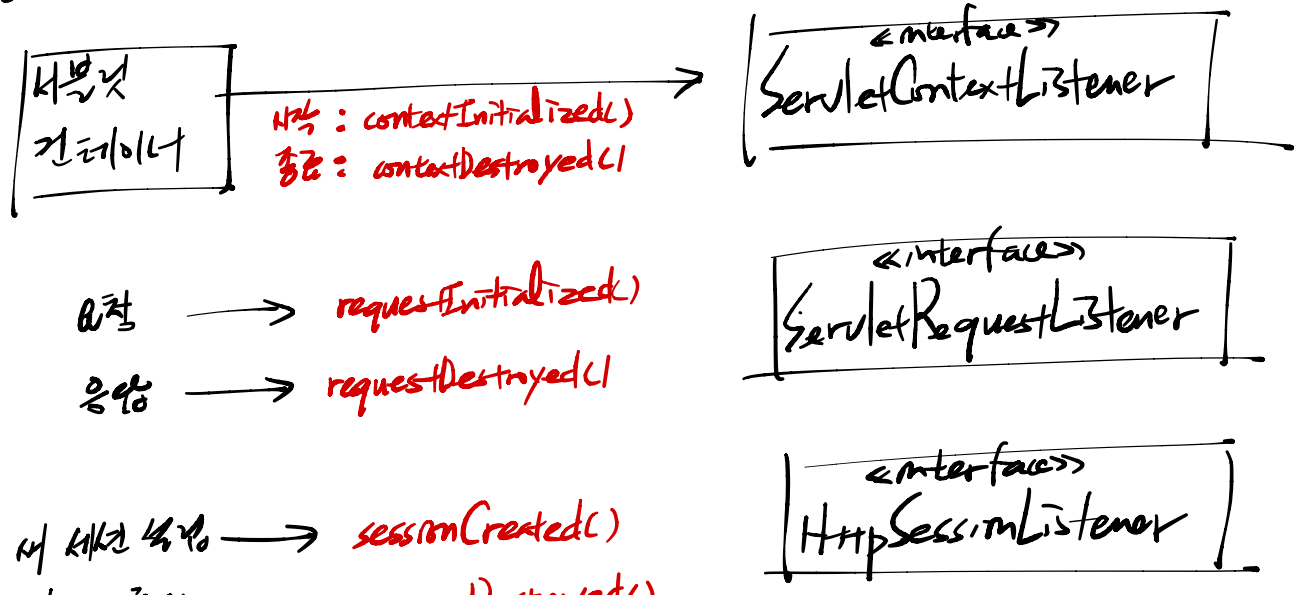

필터 실행 순서는
서블릿 컨테이너의
내부 동작에 따라 결정된다 ⇒ 서블릿 컨테이너 마다 다를 수 있다

⇓

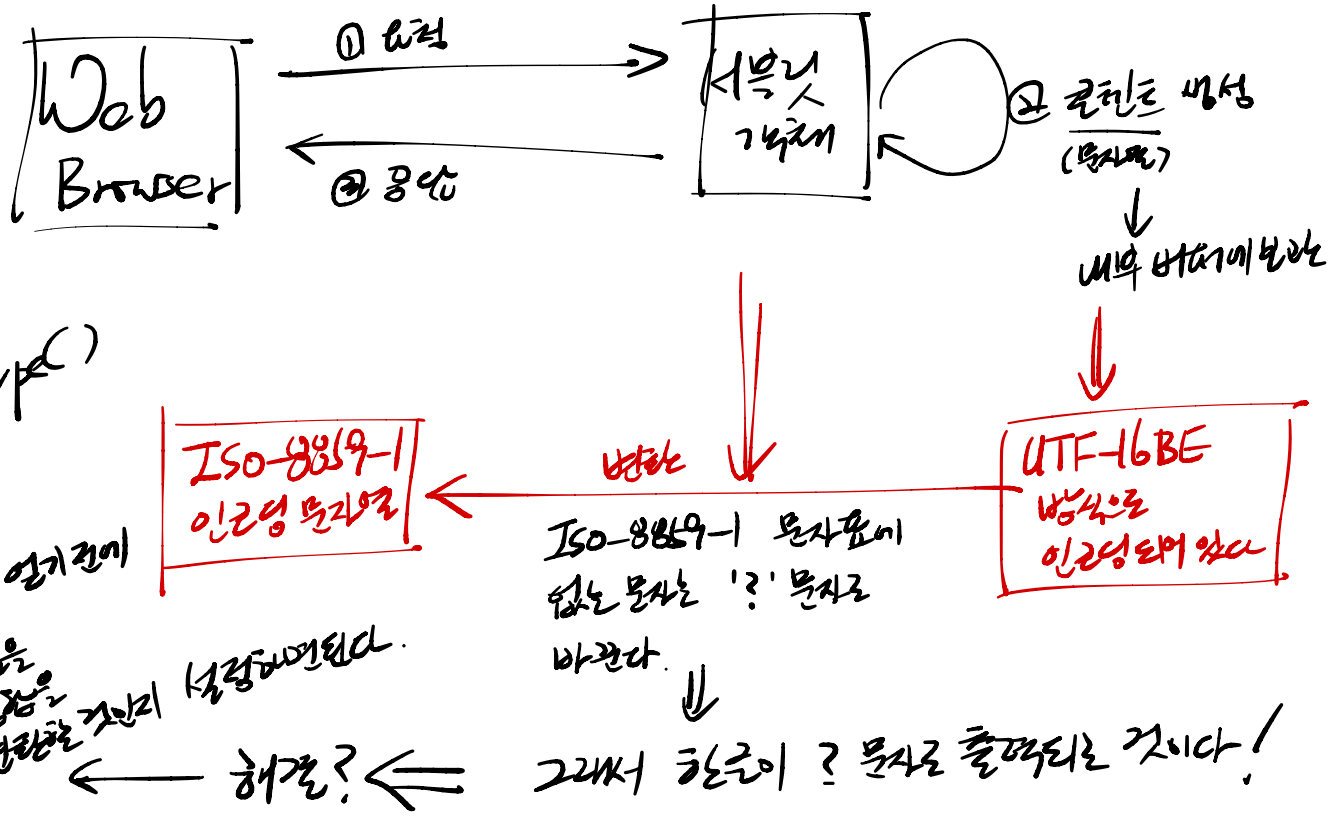
필터 실행 순서를 제어할 수 없다!

# ✳ 리스너 만들기

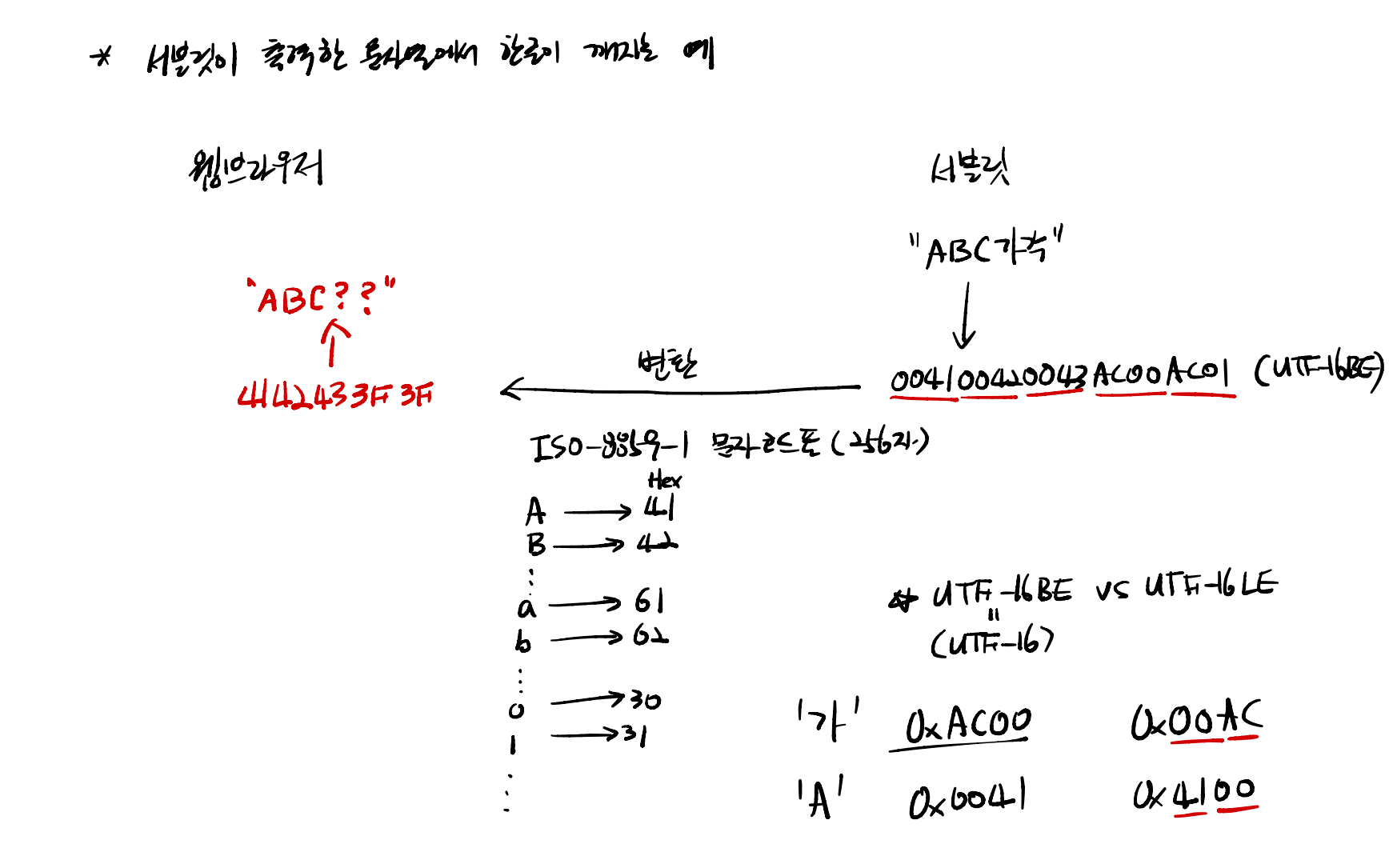
## ① 구동


서블릿 컨테이너
생성 : contextInitialized()
종료 : contextDestroyed()
<<interface>>
ServletContextListener
요청 → requestInitialized()
응답 → requestDestroyed()
<<interface>>
ServletRequestListener
새 세션 생성 → sessionCreated()
세션 무효화 → sessionDestroyed()
<<interface>>
HttpSessionListener

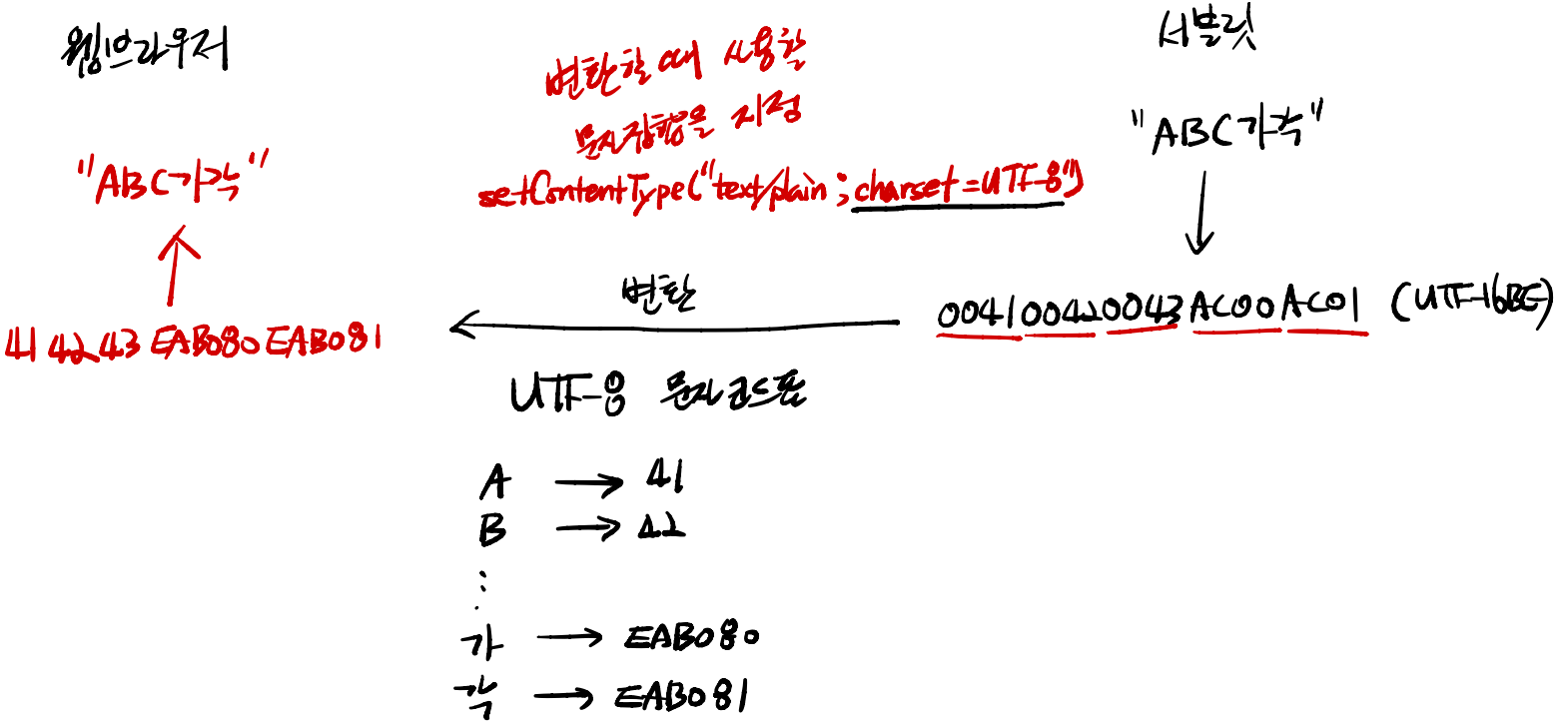
* 문자열 출력시 한글 깨짐


Web Browser
① 요청
② 응답
서블릿 객체
② 클라이언트 응답 (문자열)
내부 버퍼에 보관
UTF-16BE 방식으로 인코딩되어 있다
변환
ISO-8859-1 인코딩 문자열
ISO-8859-1 문자표에 없는 문자는 '?' 문자로 바꾼다.
그래서 한글이 ? 문자로 출력될 것이다!
setContentType()
출력스트림 객체를 얻기 전에 클라이언트 문자표를 어떤 문자집합을 사용할지 변환할 것인지 설정해야한다.
해결?

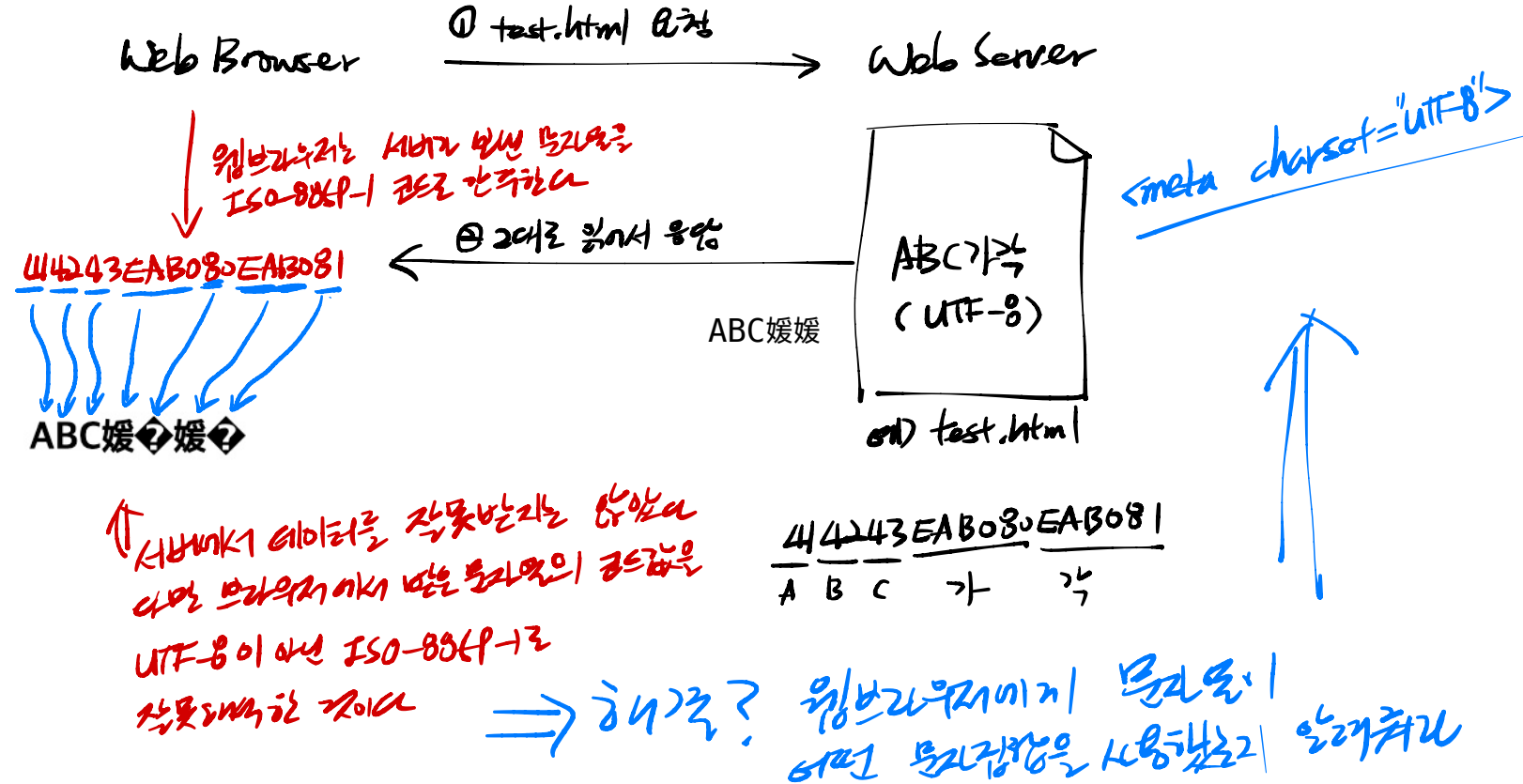
* 서블릿이 출력한 문자열에서 한글이 깨지는 예



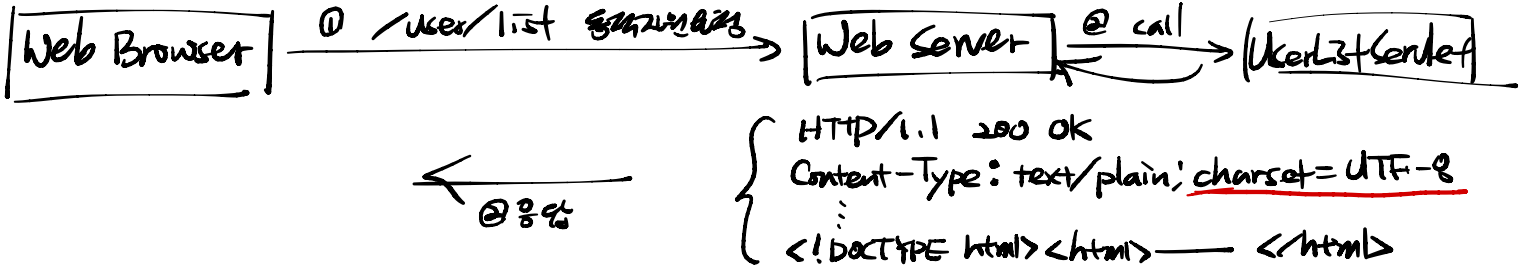
* 서블릿이 출력한 응답메시지에서 한글이 깨지는 이유

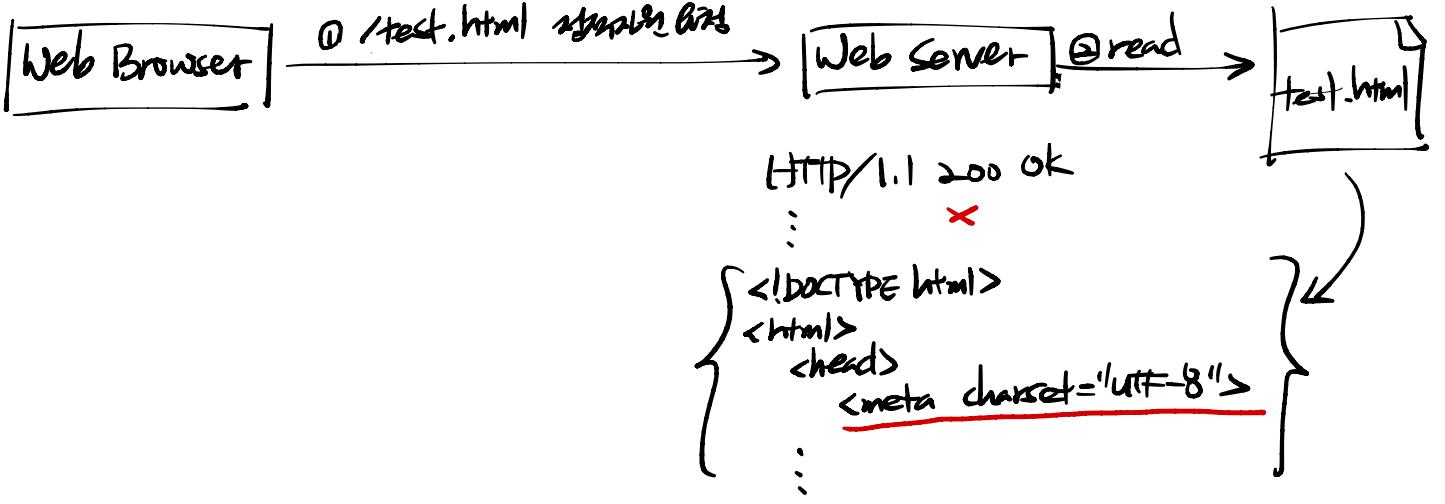


* HTML 문서의 한글 깨짐



* 한글은 깨지나 한글처리


Web Browser
① /user/list 요청메시지전송
Web Server
② call
UserListServlet
② 응답
HTTP/1.1 200 OK
Content-Type: text/plain; charset=UTF-8
<!DOCTYPE html><html>—— </html>


Web Browser
① /test.html 정적자원 요청
Web Server
② read
test.html
HTTP/1.1 200 OK
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">

* setContentType()

Multi-purpose Internet Mail Extension ) 콘텐츠의 타입정보 ⇒ 이메일을 전송하는 콘텐츠가 어떤 타입인지 알려주기 위해 반드시요

↳ 자바는 네트워크로 주고받는 콘텐츠의 타입을 지정하는 용도로 반드시 사용되는 메소드 입니다.

setContentType( "MIME타입; 문자집합" );

↑ 콘텐츠의 인코딩 정보

MIME Type 형식

⇓

type/subtype ; 파라미터=값

예) "text/plain; charset=UTF-8"

⇓

UTF-16BE ——변환——> ~~ISO-8859-1~~ UTF-8