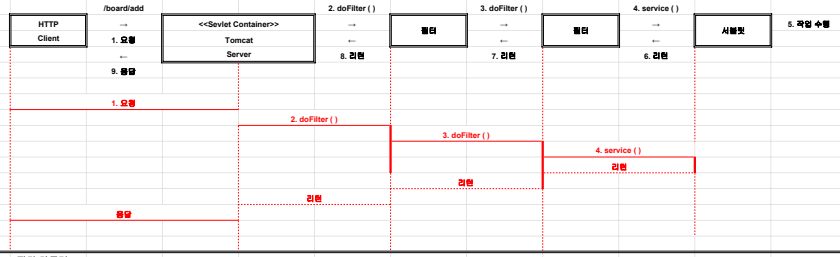
*067 필터 구동 원리

필터 구동원리



*필터 만들기

- init ( ) ← 필터 객체 생성 후 즉시 호출된다 = 필터가 작업하는데 필요한 자원을 준비시키는 코드를 둔다.
- doFiter ( ) ← 요청이 들어올 때 호출된다.
- destroy ( ) ← 웹 애플리케이션이 멈출때 호출된다. = init ( ) 에서 준비한 자원을 해제시키는 코드를 둔다.

필터 배치

1. 에노테이션 자기가 만든 필터 → 소스 갖고 있음 → 에노테이션 붙일 수 있음.

★@WebFilter(url)

2. DD file (web.xml)

<filter>~</filter> 남이 만든 필터 클래스 → 소스 없음 → 에노테이션 못 붙임

<filter-mapping>~</filter-mapping>

"필터 배치(deplyment)"

이런 필터가 있는데, 이런 요청이 들어올때 실행하라고 웹 애플리케이션에 설정하는 것!

↓

서블릿 컨테이너는 배치된 컴포넌트만 관리한다.

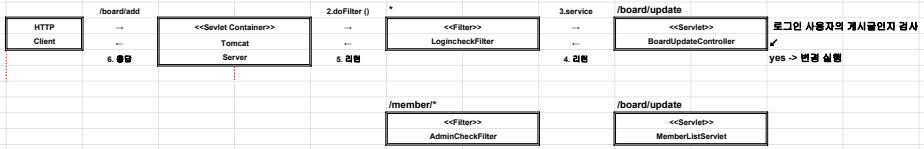
필터

서블릿

리스터

등

*067 필터로 로그인 검사하기



관리자

필터는 "실행" 순서를 제어할 수 없음.

*HTTP GET 요청

이메일	user2@test.com
암호	1111
이메일 저장	v
로그인	

Request Line →

파라미터

path

파라미터 명 파라미터 값

GET/app/auth/login?email=user1@test.com&password=1111&saveEamil=on HTTP/1.1 ┘

Query String(서버에 보내는 데이터)

→ 요청

Request URI

general-header <- 요청, 응답에 모두 붙일 수 있는 헤더

cache-control : no - cashe

connection : keep-alive

Accept : text/html~

Accept - encoding : gzip, deflate,br

host:localhost:8888 요청을 받는 서버 주소

user-agent : mozilla/5.0~ 요청을 보내는 브라우저 정보

request-header

요청과 관련된 정보

┘ > Get 요청은 message-body가 없다.
> entity-header가 없다.

*장점 ← >QueryString형태로 데이터를 URL에 포함해서 보낸다. → *단점 : 텍스트만 가능

- URL에 데이터를 포함키실 수 있다.
└ 특정 콘텐트 조회하는 페이지를 가르킬 때 적절하다.
https://www.youtube.com/watch?v=AcWIE9qazLI
└ 바로 해당 콘텐트로 이동(QueryString 형태)

- 즉, binary 보내기 힘듬
- 주소창에 노출(보안취약)
- 웹 브라우저는 URL을 캐시에 저장(보안에 매우 취약)

---

*HTTP POST 요청

URL에 데이터가 포함되지 않는다.

이메일	user2@test.com		POST /app/auth/login	HTTP/ 1.1
암호	1111	→	cache-control : ____┘	General-header
이메일 저장	v	요청	connection : ____┘	
로그인			Accept : ____┘	request-header
			Accept - encoding : ____┘	
			Host : localhost : 8888 ____┘	
			User-Agent : Moziller /5.0 ____┘	

*장점
> message-body 영역에 실어서 보냄
> 보내는 데이터의 크기에 제약 없음
> binary 데이터를 보낼 수 있음
보안에 유리 > URL에 데이터가 포함되지 않기 때문에 적절 노출되는 것을 막을 수 있다.
> 웹 브라우저가 캐시하지 않는다.

content-length : 49	POST 요청시에 추가된다.
content-type : application/x-www-from-urlencoded ____┘	entity-header
-	message-body로 보내는 데이터에 대한 정보
-	
____┘	
email = user1@test.com&password=1111&saveEmail=on	message-body

↓
대용량의 데이터 전달에 적합
- 게시글 등록/ 변경
- 첨부파일 등록 /변경
- 로그인 데이터 전달

---

*HttpServlet 클래스와 GET/POST 요청

---

*POST 방식 요청에서 한글 깨지는 원리

POST / ____ HTTP/1.1 ┘
HOST : localhost : 8888 ┘
-
-
- '가' '라' 간'
┘ ↓ URL 인코딩 ↓ URL 인코딩 ↓ URL 인코딩 URL 디코딩
title = ABC%EA%BO%80%EA%BO%81 - - - -
→
↓ 바이트 배열
t i t l e = A B C - - - -
url 인코딩 한 바이트 74 69 74 6C 65 3D 41 42 43 - - - - 문자가 16진수 바이트 코드로 변환
↓ ok! ↓ 서버에 전송 → URL 디코딩
url 디코딩 한 바이트 74 67 74 6C 65 3d 41 42 43 EA BO 80
↓ ↓ getParameter("title") <= Unicode2(UTF-16)코드로 변환
0041 0042 0043
UTF-8 한글코드를 ASCII 코드와 같은 방식으로 변환하면 문제가 발생
↓
UTF- 16으로 제대로 못바꾸고 ASCII 코드와 같은 방식으로 변환해서 문제가 발생 "한글 깨짐"

*해결책
URL 디코딩한 바이트를
UTF-16 코드로 변환하기 전에
원래 바이트가 어떤 CharSet
으로 인코딩 된 것인지 알려줘!

how?

requesr.setCharacterEncoding("UTF-8")
getParameter () 를 호출하기 전에 먼저 실행
단 한번이라도 호출한 후에 실행하면 무효하다.

자바 문자(2byte)
char = 2byte