

Filter와 interceptor 차이점과 용도

Spring은 공통적으로 여러 작업을 처리함으로써 중복된 코드를 제거할 수 있도록 많은 기능을 지원하고 있습니다. 그 중에서 필터(Filter) 와 인터셉터(interceptor) 의 차이에 대해 알아보려고 합니다.

필터(Filter)

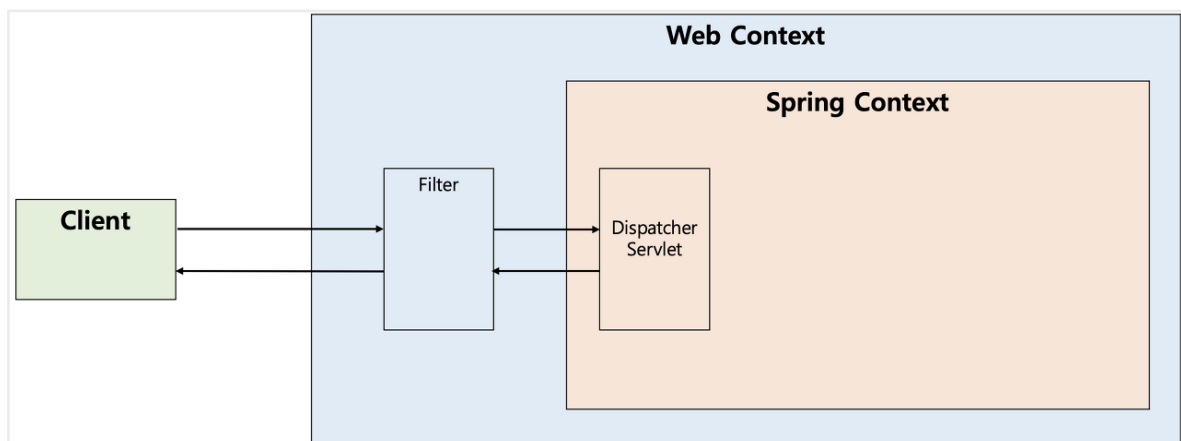
필터란 J2EE 표준 스펙 기능으로 디스패처 서블릿(Dispatcher Servlet)에 요청이 전달되기 전/후에 url 패턴에 맞는 모든 요청에 대해 부가작업을 처리할 수 있는 기능을 제공합니다.

? 여기서 J2EE란? 자바 기술로 기업환경의 애플리케이션을 만드는데 필요한 스펙 집합입니다.

? 여기서 디스패처 서블릿(Dispatcher Servlet)이란? HTTP 프로토콜로 들어오는 모든 요청을 가장 먼저 받아서 적합한 컨트롤러에게 위임해주는 프론트 컨트롤러입니다.

장점

- web.xml에 서블릿을 URL매핑을 위해 모두 등록해야 했지만, dispatcher Servlet이 해당 어플리케이션으로 들어오는 모든 요청을 핸들링 해준다
- 비즈니스로직에 집중 가능



Filter는 디스패처 서블릿보다 먼저 요청에 대한 기능을 제공합니다.

그렇다면 필터(Filter)는 왜 사용할까요?

예시를 들어보자면

- 공통된 보안 및 인증/인가 관련 작업
- 모든 요청에 대한 로깅 또는 감사

- 이미지/데이터 압축 및 문자열 인코딩
- Spring과 분리되어야 하는 기능 등이 있습니다.

필터에서는 기본적으로 스프링과 무관하게 전역적으로 처리해야 하는 작업들을 처리 가능합니다.

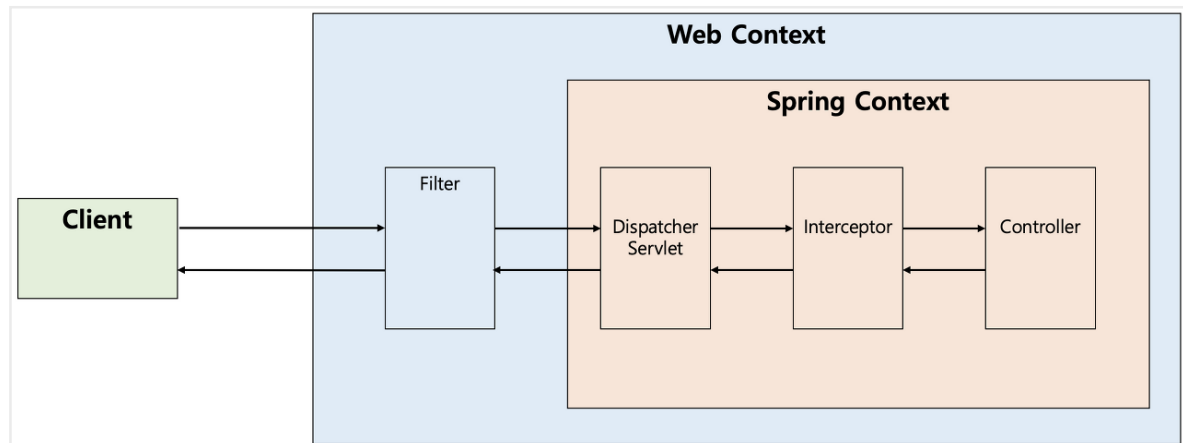
대표적으로는 보안 공통작업이 있습니다. 필터는 인터셉터보다는 먼저 동작하므로 전역적으로 즉, 제일 먼저 처리해야하는 보안검사를 하여 올바른 요청인지 아닌지 인증/인가 관련 작업을 할수 있습니다. 이렇게 하게 되면 스프링에 접근하기 이전에 차단이 가능하므로 안정성이 올라간다는 장점이 있습니다.

또한 필터는 이미지나 데이터의 압축이나 문자열 인코딩과 같은 웹 애플리케이션에 전반적으로 사용되는 기능을 구현하기에 적합합니다.

인터셉터(Interceptor)

Interceptor는 J2EE의 표준 스펙인 Filter 와 다르게 Spring이 제공하는 기술로써, 디스패처 서블릿이 컨트롤러에 호출되기 전과 후에 요청과 응답을 참조하거나 가공할 수 있는 기능을 제공합니다.

웹컨테이너에서 실행되는 Filter와 달리 Interceptor는 Spring Container에서 실행됩니다.



그렇다면 왜 인터셉터를 사용해야 할까요?

예시를 들어보자면

- 세부적인 보안 및 인증/인가 공통 작업
- API 호출에 대한 로깅 또는 감사
- Controller로 넘겨주는 정보(데이터)의 가공 등이 있습니다.

인터셉터에서는 클라이언트의 요청과 관련되어 전역적으로 처리해야 하는 작업들을 처리할 수 있습니다.

클라이언트가 요청했을때 제일먼저 필터에서 인증/인가를 거치고 나서 세부적으로 인증/인가를 해야할때 사용해야합니다.

예를들어, 다른그룹의 사용자는 접근이 불가능하다는 경우가 있는데, 이러한 작업들은 컨트롤러로 넘어가기 전에 검사해야 하기때문에 인터셉터가 처리하기 좋습니다.

인터셉터(Interceptor)와 AOP의 비교

인터셉터 대신에 컨트롤러들에 적용할 부가기능을 어드바이스로 만들어 AOP(Aspect Onrriented Programming, 관점지향 프로그래밍)를 적용할 수도 있지만 다음과 같은 이유들로 컨트롤러의 호출 과정에 적용되는 부가기능들은 인터셉터를 사용하는 것이 더 좋습니다.

- 1.컨트롤러는 타인과 실행 메소드가 모두 제각각이라 포인트컷(적용할 메소드 선별)의 작성이 어렵다.
- 2.컨트롤러는 파라미터가 리턴값이 일정하지 않다.
- 3.AOP에서는 HttpServletRequest/Response를 객체를 얻기 어렵지만 인터셉터에서는 파라미터로 넘어온다.

필터(Filter)와 인터셉터(Interceptor)의 차이점

대상	필터(Filter)	인터셉터(Interceptor)
관리되는 컨테이너	웹 컨테이너	스프링 컨테이너
Request/Response 객체 조작 가능 여부	O	X
용도	<ul style="list-style-type: none">• 공통된 보안 및 인증/인가 관련 작업• 모든 요청에 대한 로깅 또는 감사• 이미지/데이터 압축 및 문자열 인코딩• Spring과 분리되어야 하는 기능	<ul style="list-style-type: none">• 세부적인 보안 및 인증/인가 공통 작업• API 호출에 대한 로깅 또는 감사• Controller로 넘겨주는 정보(데이터)의 가공

일부에서 필터(Filter)가 스프링 빈으로 등록되지 못하며, 빈을 주입 받을 수도 없다고 설명하는데 이는 잘못된 설명입니다.

예전에는 맞는 설명일지는 몰라도 현재는 스프링 빈으로 등록이 가능하며, 다른 곳에 주입되거나 다른 빈을 주입 받을 수도 있습니다.

여담으로 대표적으로 필터(Filter)를 인증과 인가에 가장 잘 사용되는 도구는 SpringSecurity가 있습니다. SpringSecurity의 특징중 하나는 Spring MVC에 종속적이지 않다는 것인데, 이러한 이유로 필터 기반으로 인증/인가 처리를 하기 때문입니다.

여기에서는 AOP에 대해서 제대로 설명하지 못해서 다른 글에서 제대로 다룰 예정입니다.

참고블로그

<https://mangkyu.tistory.com/173>

<https://mangkyu.tistory.com/18>

<https://mangkyu.tistory.com/221>