

Spring Framework

-세션 , redirect(redirectAttributes)

1. Connectionless Protocol
2. session(스프링 세션)
3. redirect(리다이렉트)
4. Interceptor(인터셉터)

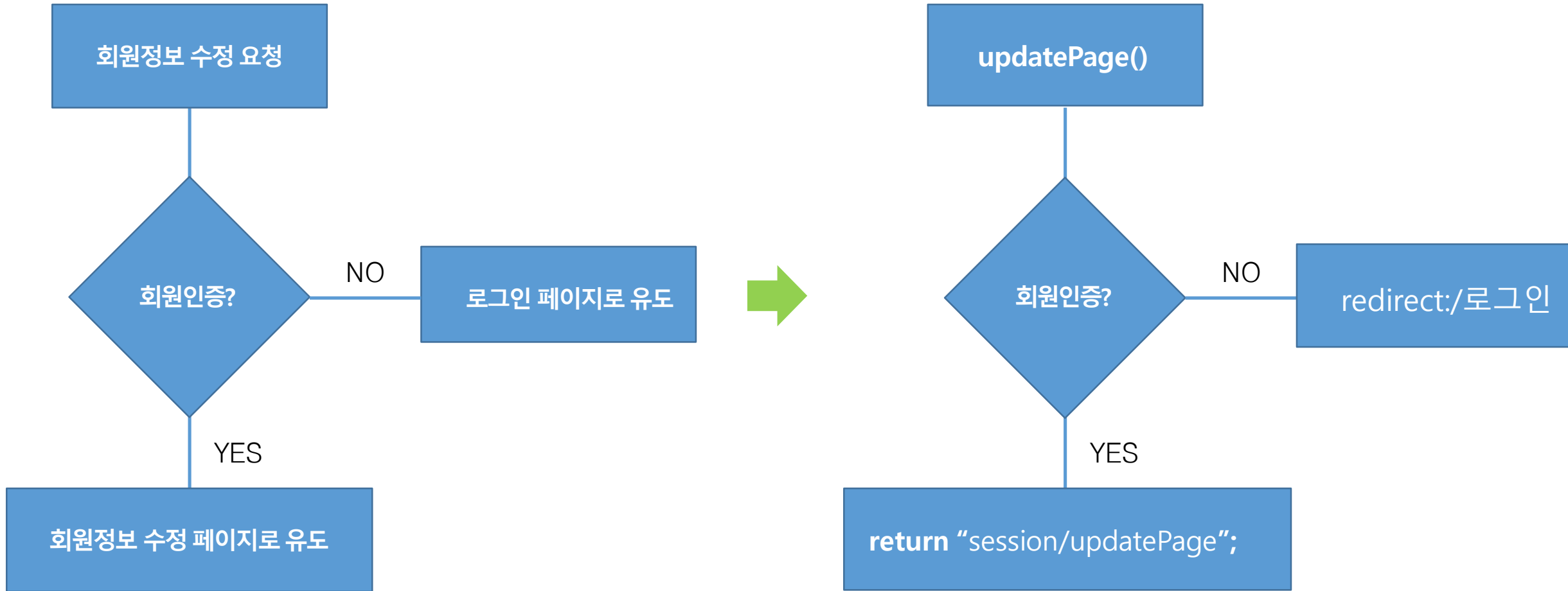
1. Connectionless Protocol

Connectionless Protocol

- 웹 서비스는 HTTP 프로토콜을 기반으로 하는데, 요청에 대한 응답 후 관계를 끊는다.
- 즉, 인증되지 않은 사용자는 모든 페이지에서 인증 과정을 거쳐야 한다
- 이런 불편함 때문에 지속적인 인증 수단으로 **세션**과 쿠키를 사용 한다.



2. 스프링 세션



2. 스프링 세션

스프링MVC의 세션을 사용하는 방법

1. HttpServletRequest 이용
2. HttpSession 이용

HttpServletRequest

파라미터로 HttpServletRequest를 받은 후 getSession()으로 세션 얻음.

HttpSession

파라미터로 HttpSession을 받아 세션 사용.

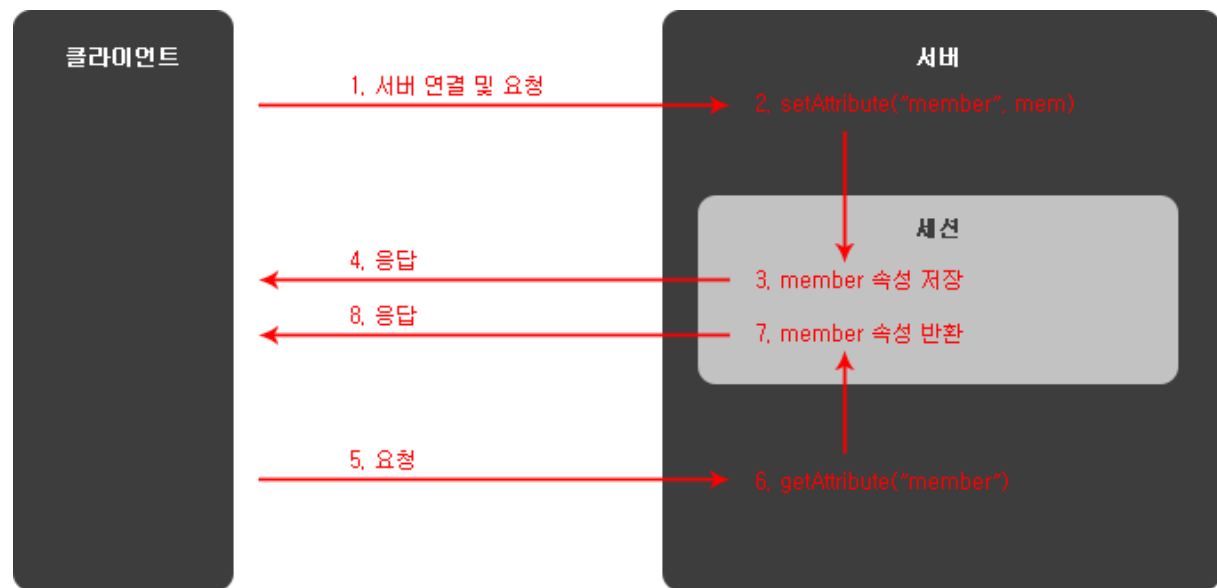
```
public String updatePage(HttpServletRequest request) { ... }
```

```
public String MyPage(HttpSession session) { ... }
```

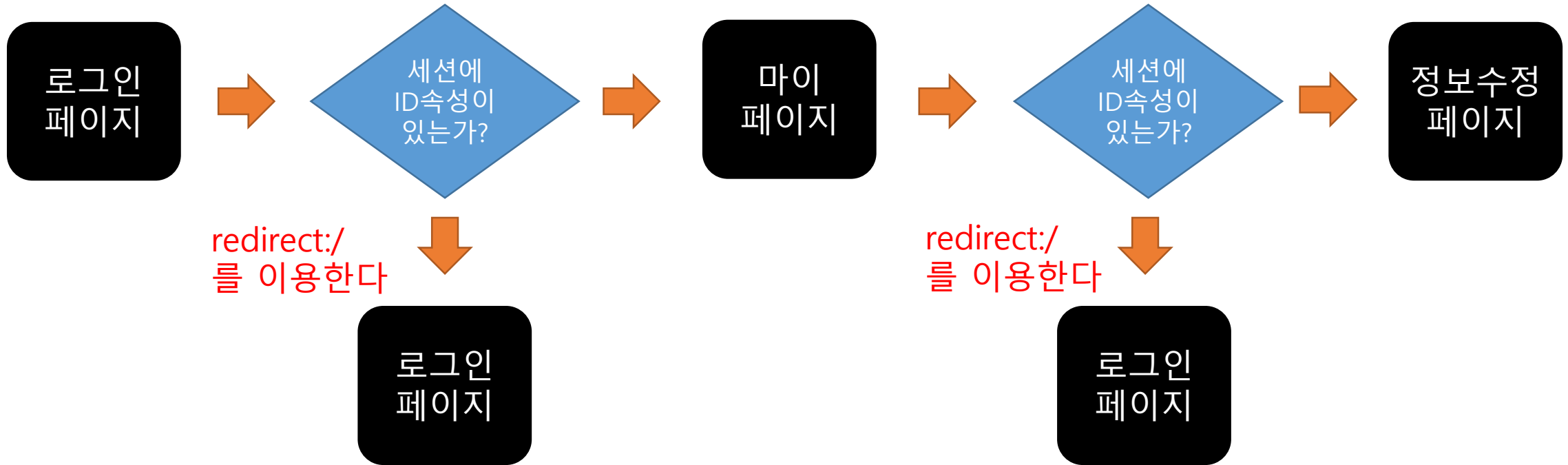
2. 스프링 세션

세션의 주요 기능

세션 메소드	기능
<code>getId()</code>	세션 ID를 반환한다.
<code>setAttribute()</code>	세션 객체에 속성을 저장한다.
<code>getAttribute()</code>	세션 객체에 저장된 속성을 반환한다.
<code>removeAttribute()</code>	세션 객체에 저장된 속성을 제거한다.
<code>setMaxInactiveInterval()</code>	세션 객체의 유지시간을 설정한다.
<code>getMaxInactiveInterval()</code>	세션 객체의 유지시간을 반환한다.
<code>invalidate()</code>	세션 객체의 모든 정보를 삭제한다.



3. redirect

**redirect:/**

다시 컨트롤러에 요청을 보내는 행위
즉. 다시 컨트롤러의 요청으로 이동한다

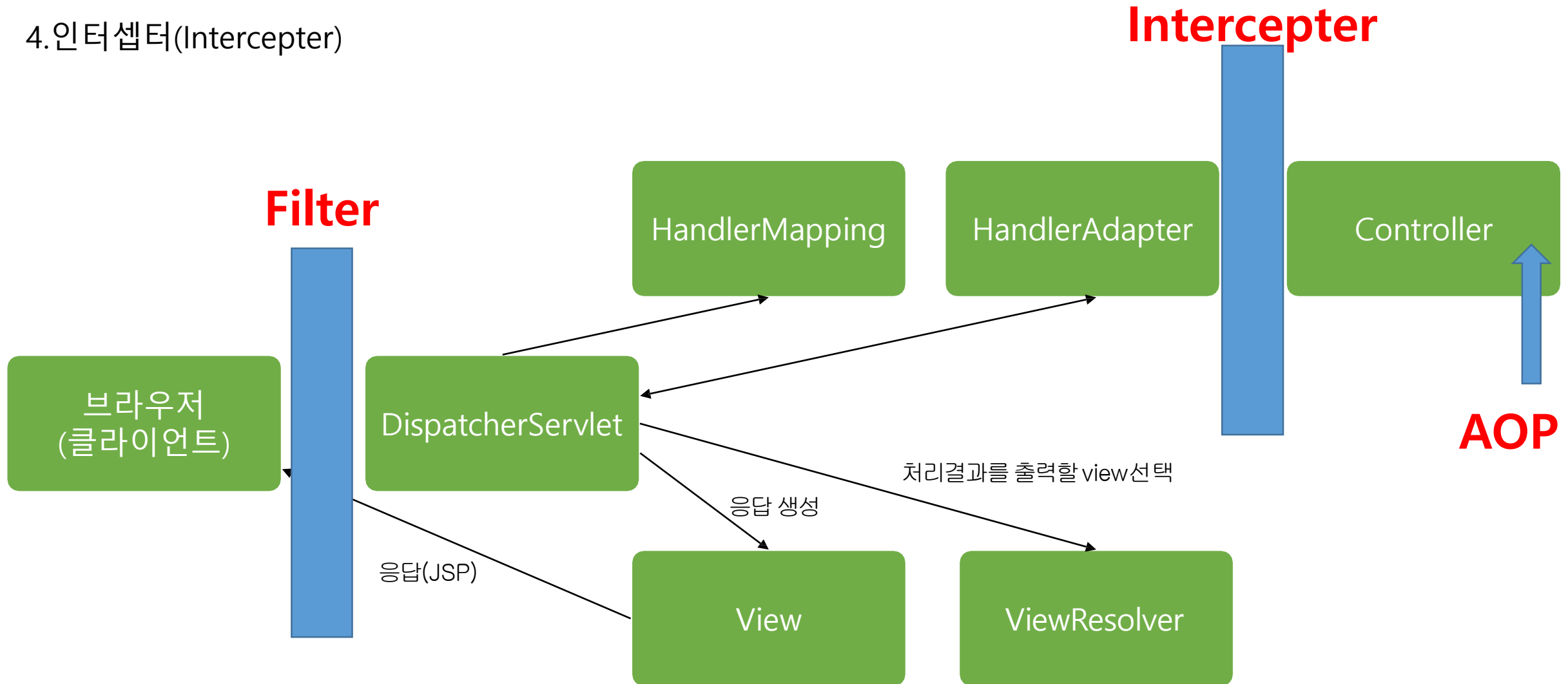
```
return "redirect:/session/myPage";  
return "redirect:/session/mainPage";
```

RedirectAttributes 전달자**1.addFlashAttribute(변수명, 값)**

-리다이렉트 이동시 파라미터 값을 전달하는 방법

-model의 addAttribute(변수명, 값) 과 흡사

4. 인터셉터(Interceptor)

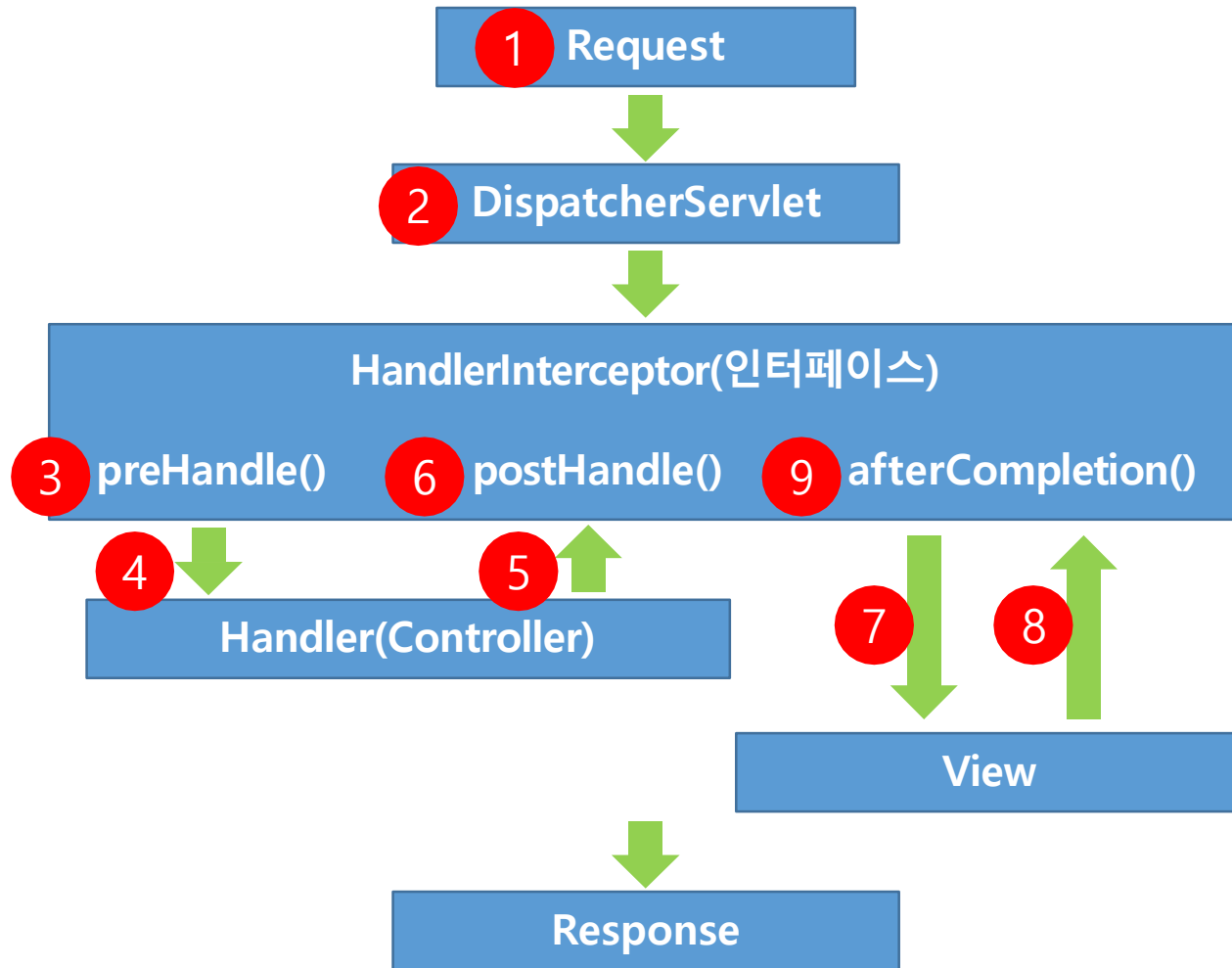


Filter- 디스패처 서블릿을 거치기 전에 처리

Interceptor – 컨트롤러 전 후로 요청을 처리

AOP – 비즈니스 로직에서(메서드) 에서 세밀하게 처리

4. 인터셉터(Interceptor)



3 컨트롤러 실행전 실행됨
false를 리턴하면 다음 내용을 수행하지 않음

6 컨트롤러 실행후 실행됨

9 리졸버뷰에 뷰를 전송한 후 실행됨

-인터셉터의 사용

스프링에서 제공하는

HandlerInterceptorAdapter

클래스를 상속 받아 사용합니다.

4. 인터셉터 설정

servlet-xml 태그 추가

```
<interceptors>
  <interceptor>
    <mapping path="/session/updatePage"/>
    <mapping path="/session/myPage"/>
    <beans:bean class="com.zerock.interceptor.LoginInterceptor"></beans:bean>
  </interceptor>
</interceptors>
```

Mapping태그 요청이 들어오면 bean클래스가 가로챈

