

게시판 구현하기 (접근제어)

게시판 구현하기 (코드로 접근 제어하기)

게시판 구현하기 - 코드로 접근제어 하기

- 실습 상황

로그인 한 사용자만 볼 수 있는 페이지인데, URL에 입력하면 바로 접속 되는 경우
→ 로그인 한 사용자만 볼 수 있도록 설정

게시판 구현하기 - 코드로 접근제어 하기

• 설명

1. kr.co.greenart.common.templat에 Login 클래스 생성
2. loginCheck 메서드를 생성
3. 세션에 담긴 memberId가 null 인지 확인하고 true/false 반환
 - null : 로그인하지 않은 유저
 - not null : 로그인한 유저

```
package kr.co.greenart.common.template;

import javax.servlet.http.HttpSession;

public class Login {
    public static boolean loginCheck(HttpSession session) {
        Integer memberId = (Integer) session.getAttribute("memberIdx");

        if(memberIdx == null) {
            return false;
        } else {
            return true;
        }
    }
}
```

게시판 구현하기 - 코드로 접근제어 하기

• 설명

1. loginCheck 객체를 생성
2. if문에 loginCheck 메서드를 실행
→ if문으로 하단의 return 부분까지 모두 묶어주어야 함
3. 반환된 값이 true일 경우 if문 실행

```
@GetMapping("list.do")
public String boardList(@RequestParam(value="cpage", defaultValue="1") int currentPage,
                        HttpSession session,
                        Model model) {
```

```
    Login loginCheck = new Login();
    if(loginCheck.LoginCheck(session)) {
```

```
        // 전체 게시물 수 구하기
        int listCount = boardService.selectListCount();
        int pageLimit = 10;
        int boardLimit = 10;
        // ...
    }
```

게시판 구현하기 - 코드로 접근제어 하기

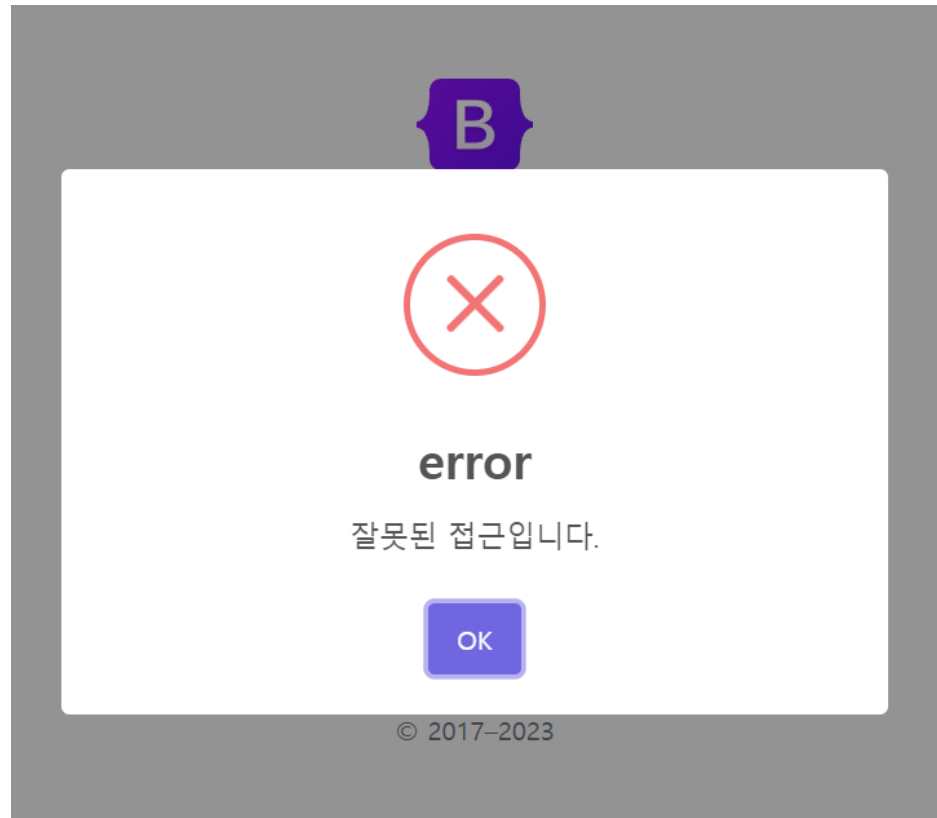
• 설명

1. 로그인하지 않은 사용자일 경우 "잘못된 접근입니다." 메시지 출력 후 login 페이지로 이동

```
session.removeAttribute("msg");
session.removeAttribute("status");
return "board/boardList"; // /WEB-INF/views/boardList.jsp 파일로 포워딩
} else {
    model.addAttribute("msg", "잘못된 접근입니다.");
    model.addAttribute("status", "error");
    return "member/login";
}
```

게시판 구현하기 - 코드로 접근제어 하기

- 설명



게시판 구현하기 - 코드로 접근제어 하기

- 단점

이런식으로 코드 기반으로 접근제어를 할 경우 모든 페이지마다 로그인하지 않은 사용자의 접근을 막기 위한 코드를 작성해야 함

즉, 불필요한 코드의 중복이 발생하며 변경되는 내용이 있을 경우 유지보수에 악영향을 끼침

게시판 구현하기 (인터셉터 개요)

게시판 구현하기 – 인터셉터

- 인터셉터(Interceptor)

컨트롤러에 들어오는 요청 및 응답을 중간에 가로채는 역할을 하는 객체

게시판 구현하기 – 인터셉터

- **preHandle**

컨트롤러 호출 전에 호출

- **postHandle**

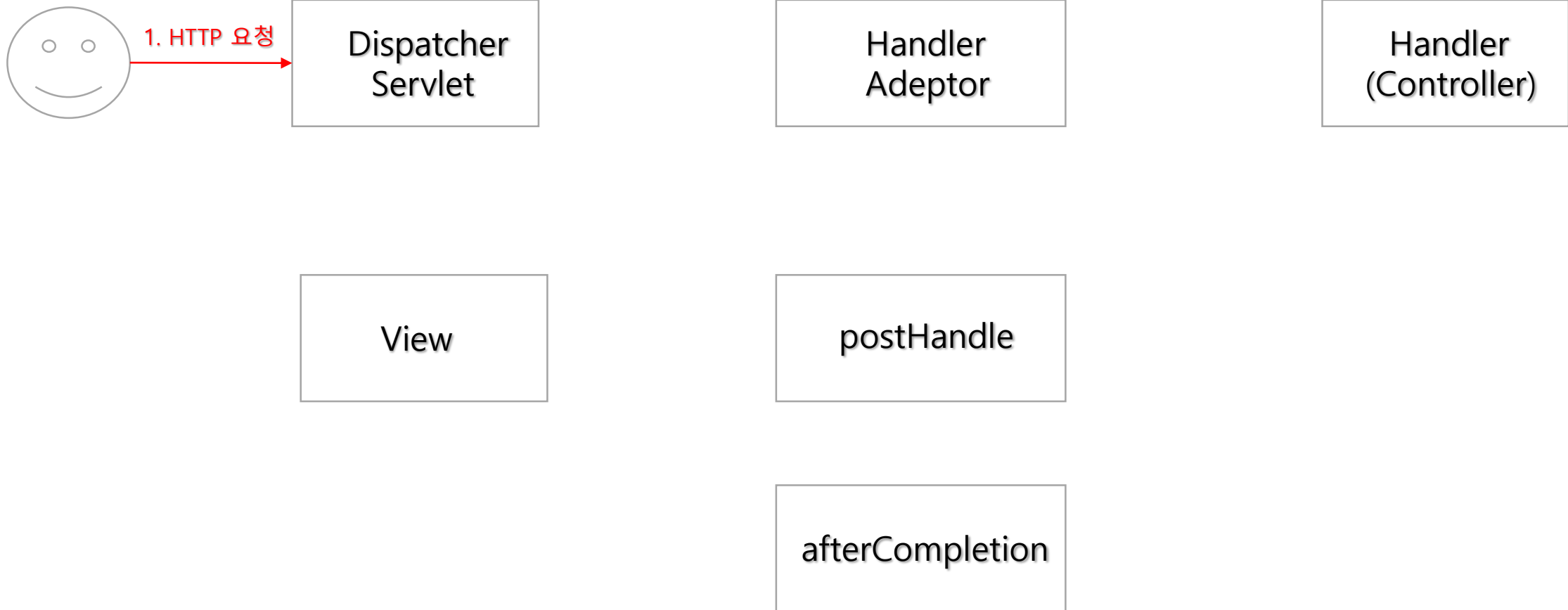
컨트롤러 호출 후 호출

- **afterCompletion**

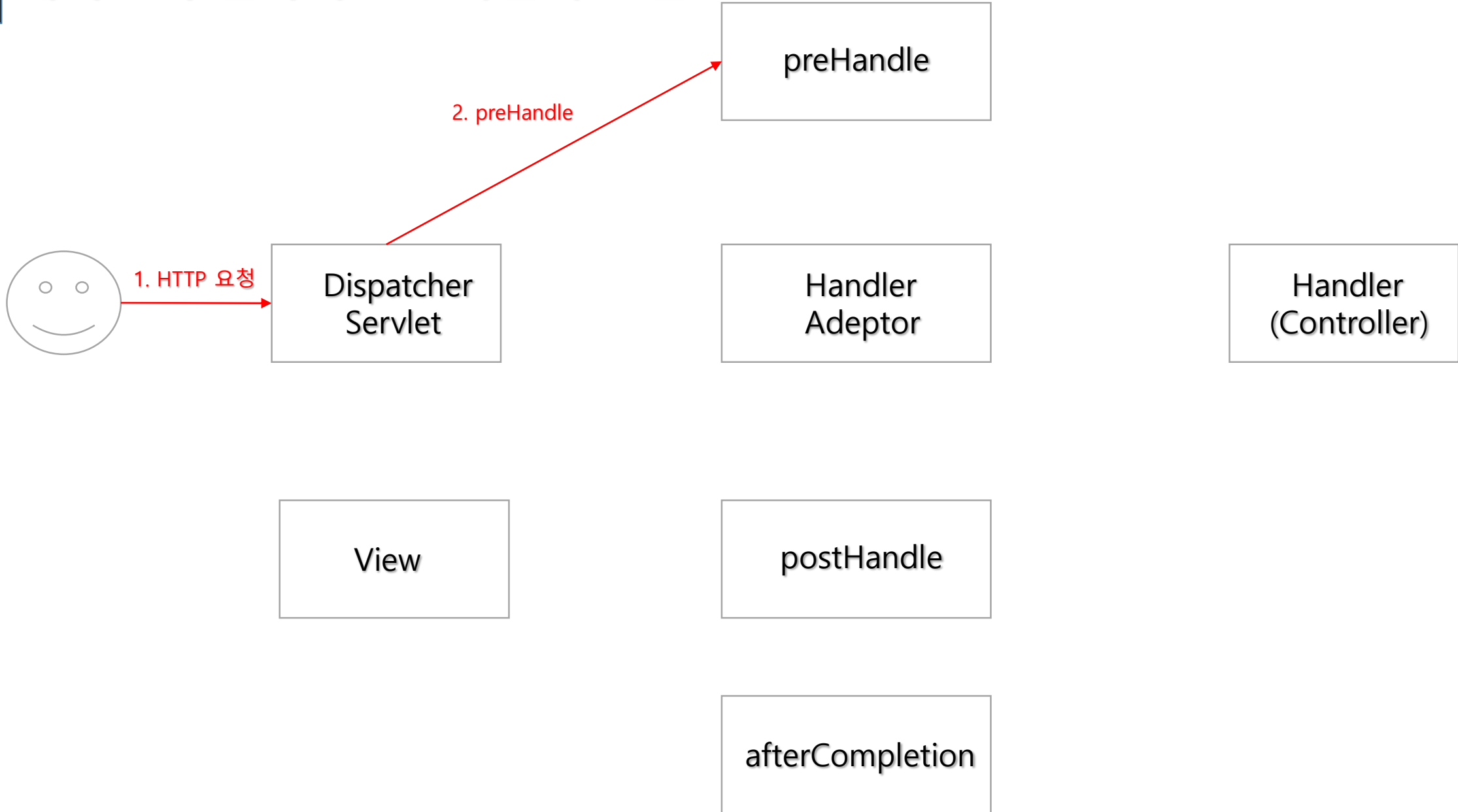
뷰가 렌더링 된 이후에 호출

게시판 구현하기 (인터셉터 흐름)

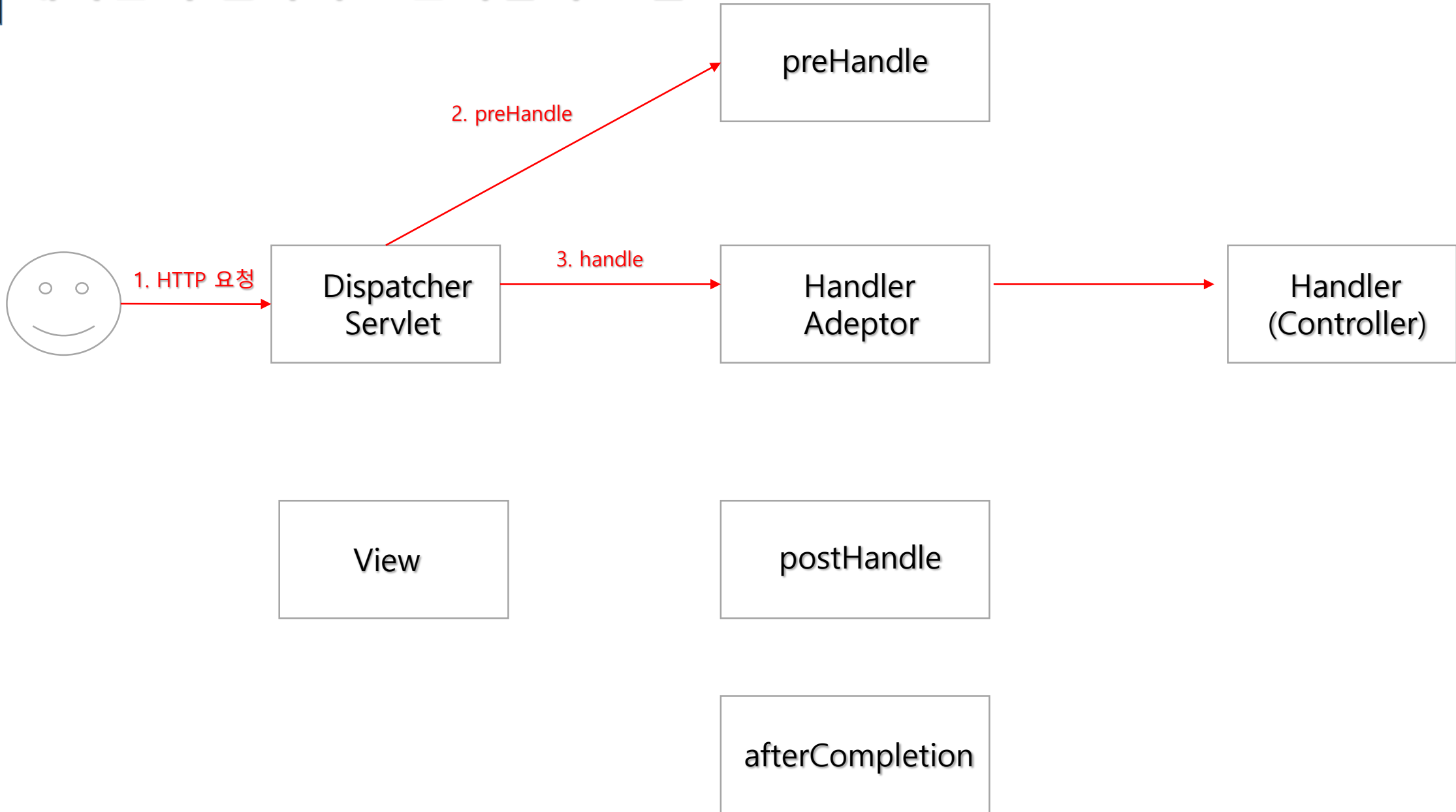
게시판 구현하기 - 인터셉터 흐름



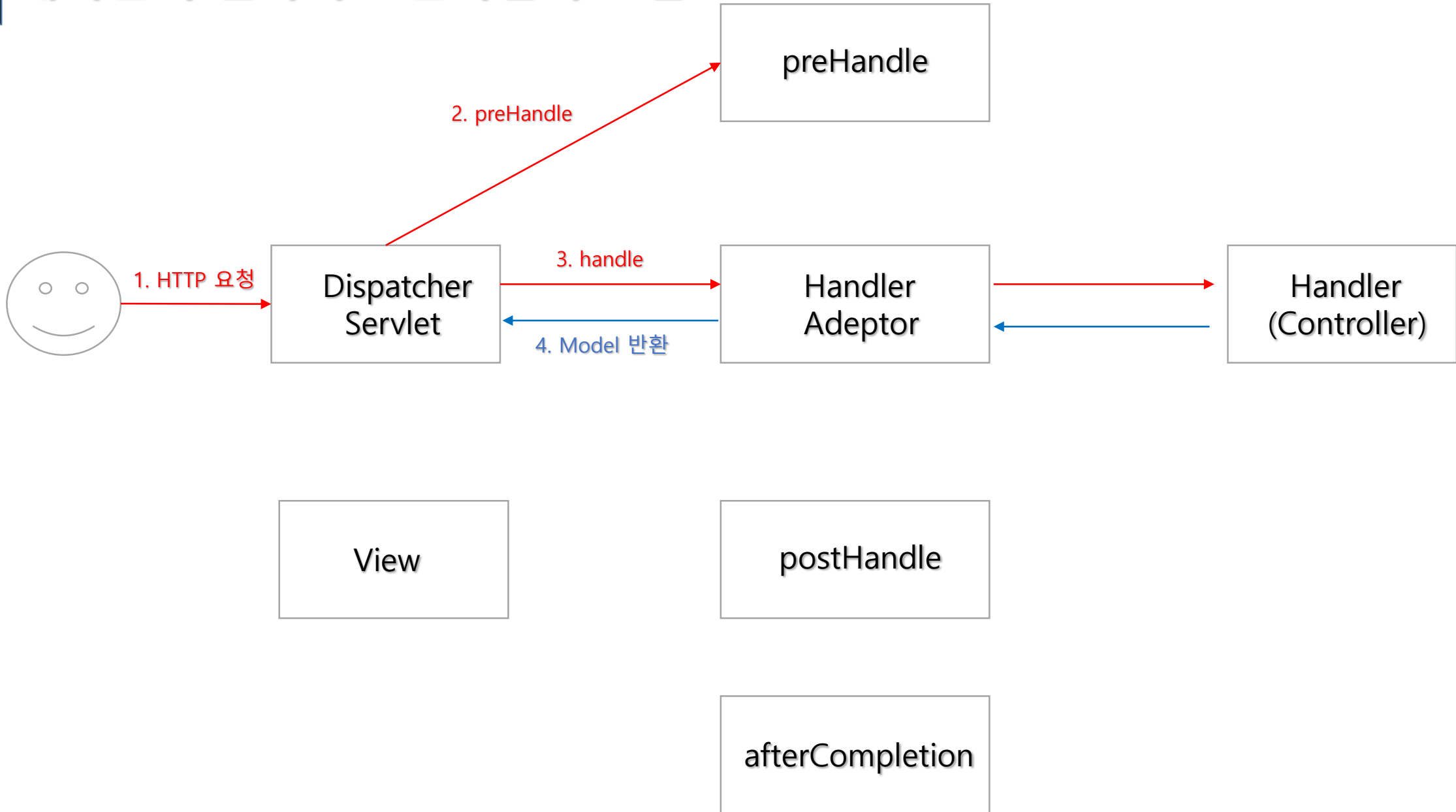
게시판 구현하기 - 인터셉터 흐름



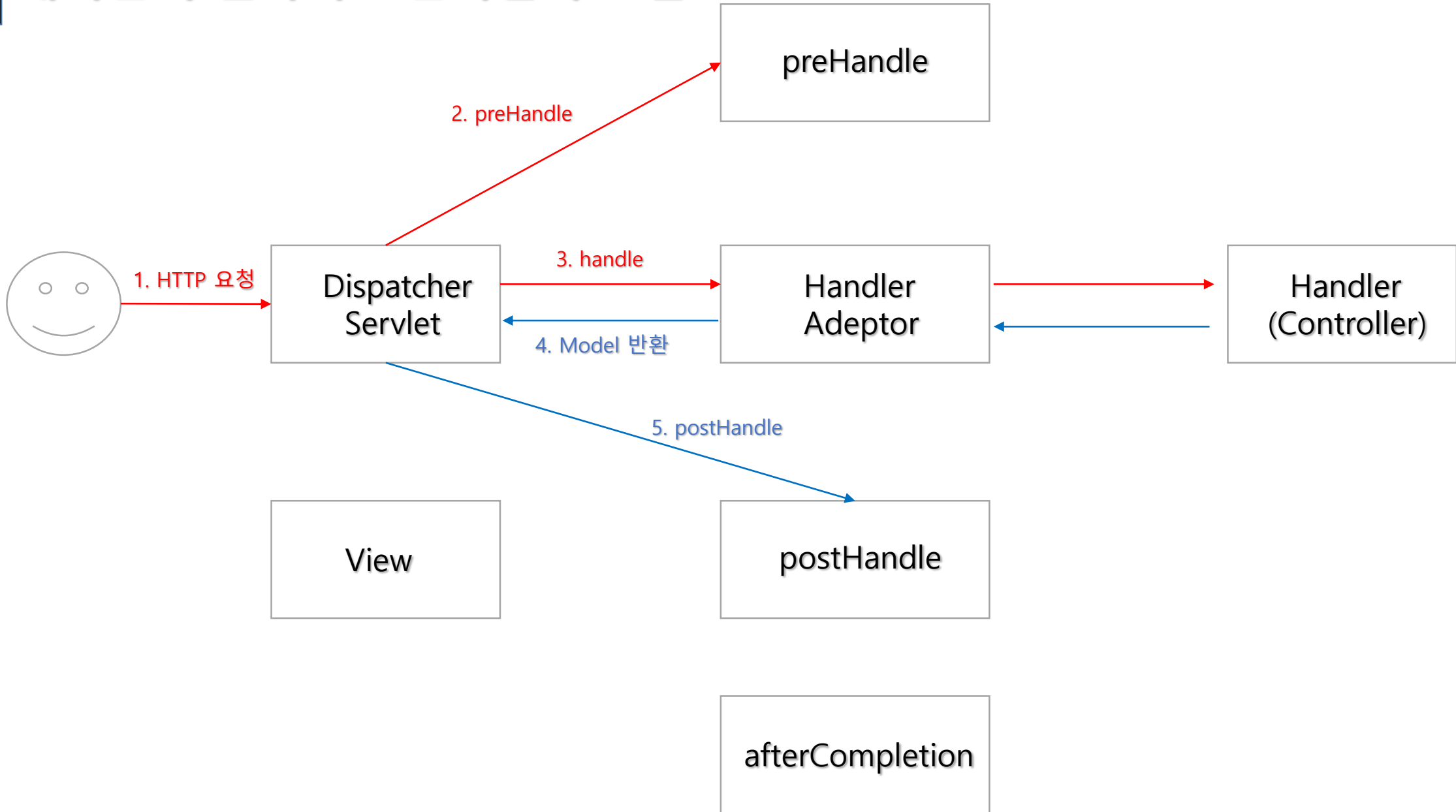
게시판 구현하기 - 인터셉터 흐름



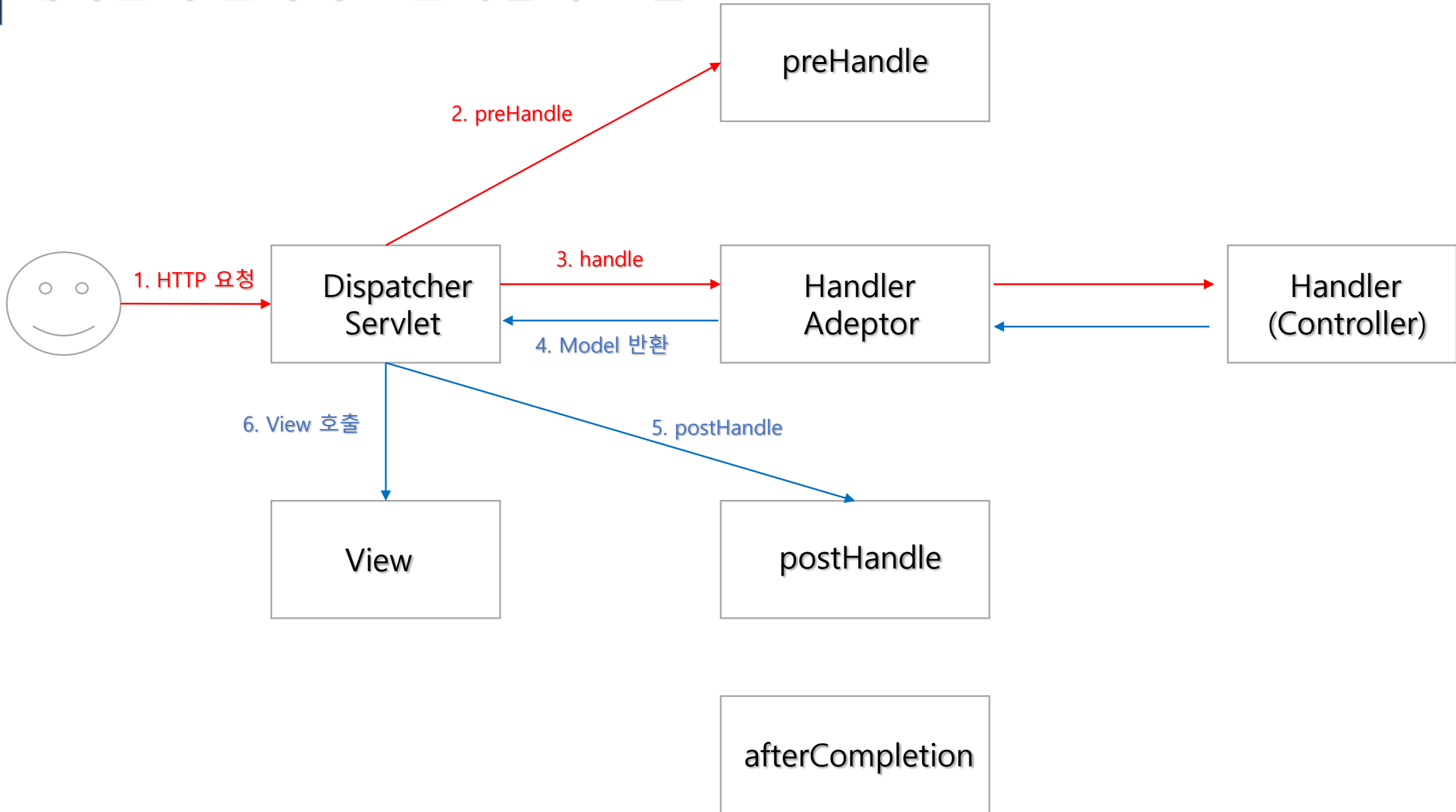
게시판 구현하기 - 인터셉터 흐름



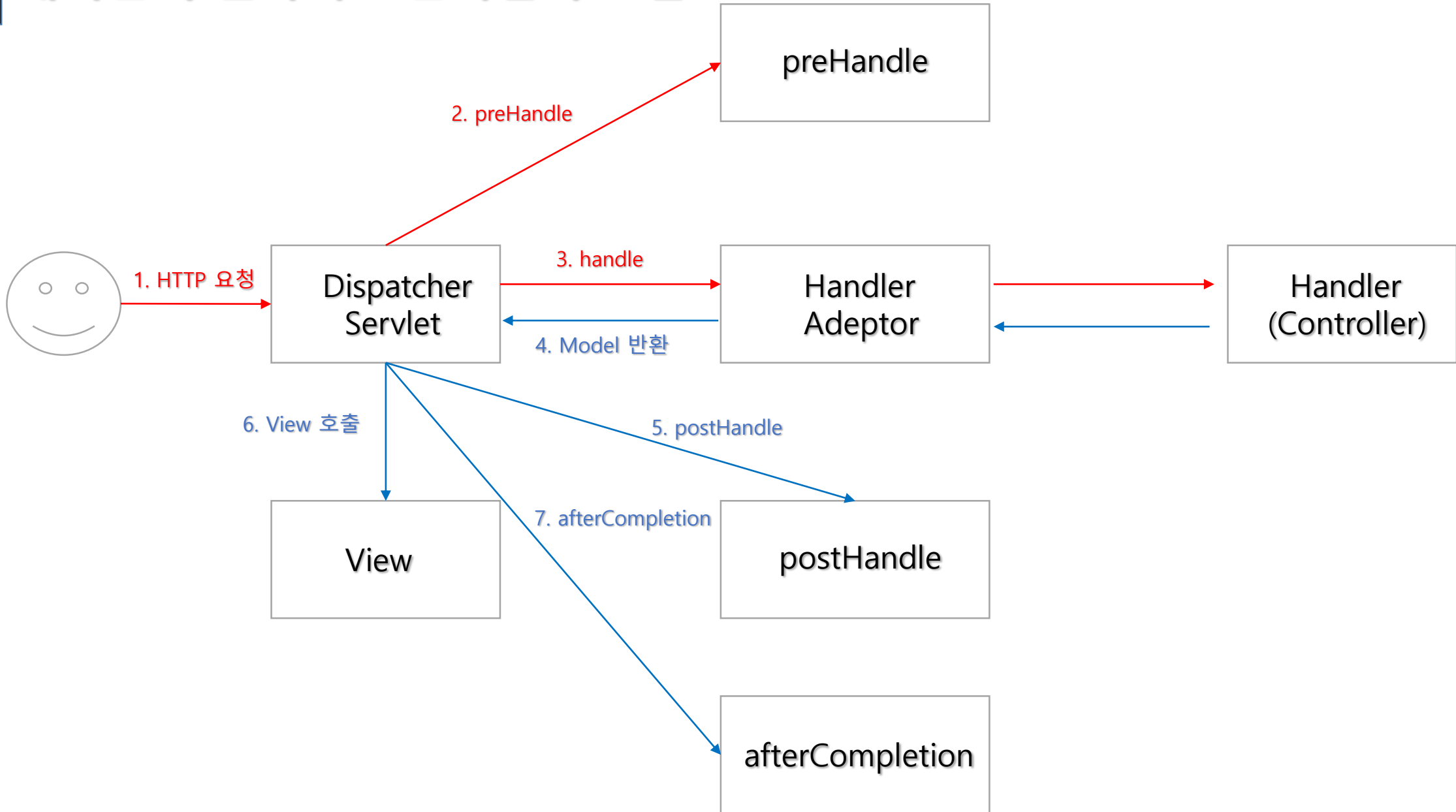
게시판 구현하기 - 인터셉터 흐름



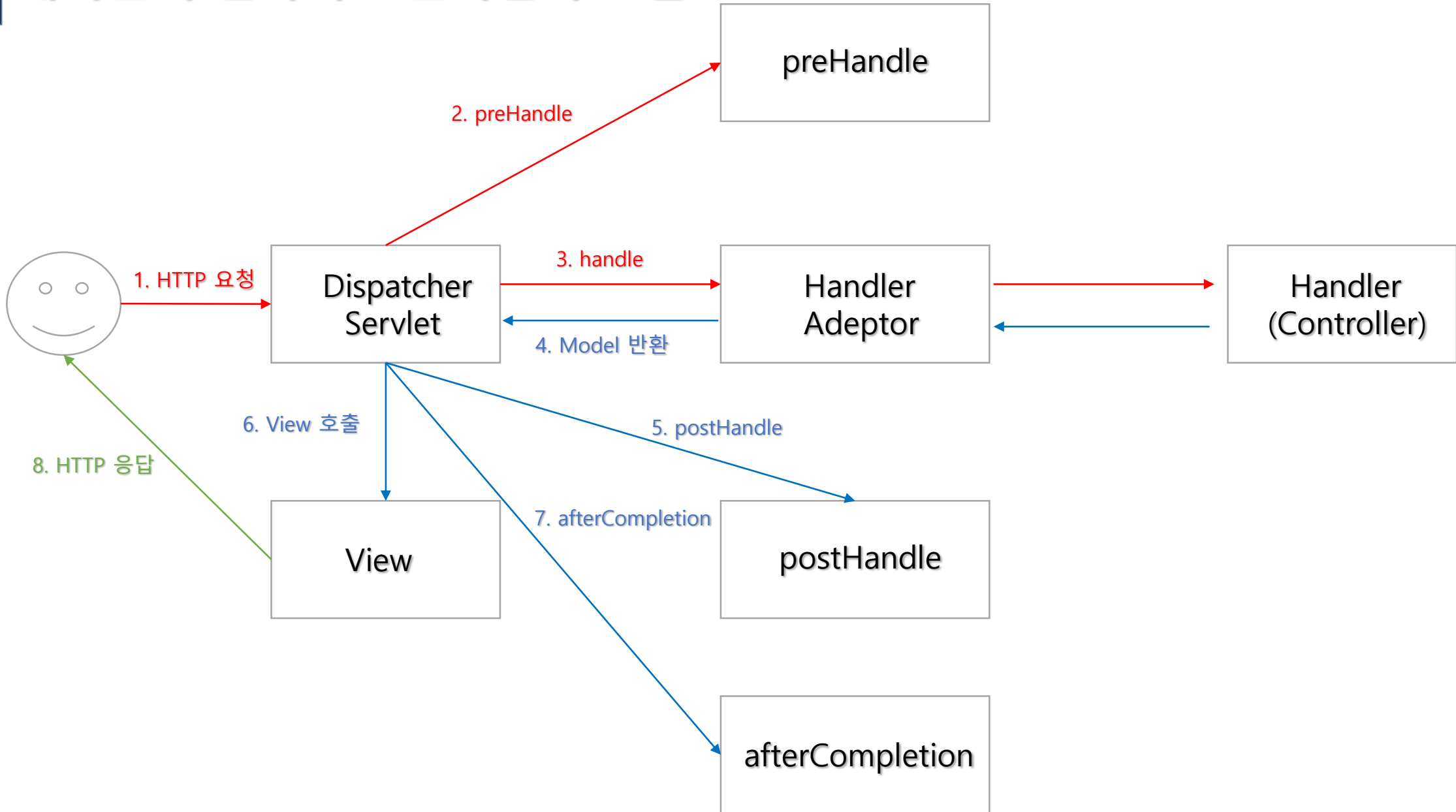
게시판 구현하기 - 인터셉터 흐름



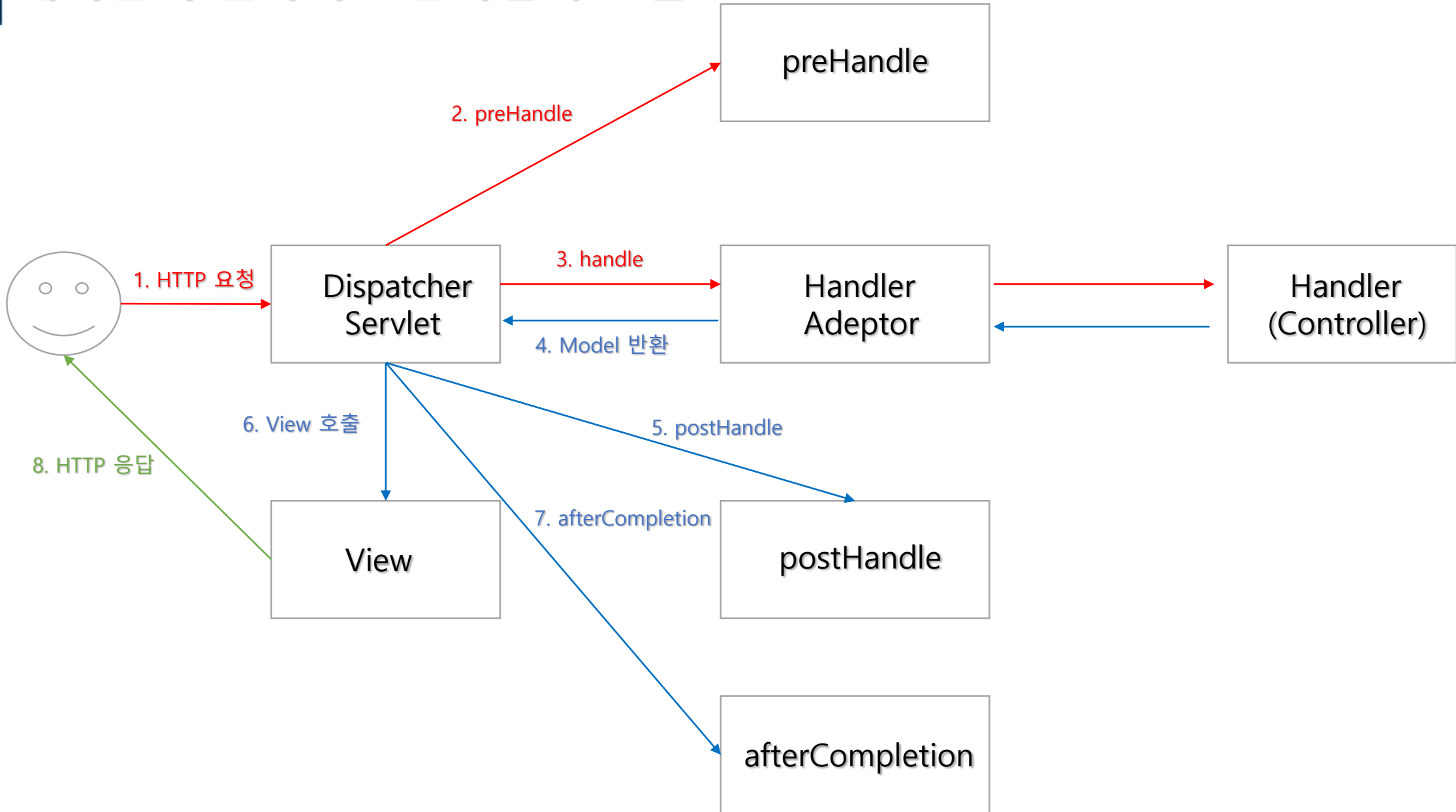
게시판 구현하기 - 인터셉터 흐름



게시판 구현하기 - 인터셉터 흐름



게시판 구현하기 - 인터셉터 흐름



게시판 구현하기 (인터셉터 설정)

게시판 구현하기 - 인터셉터 설정

- servlet-context.xml

스프링 설정에서 아래와 같이 코드 작성

```
<beans:bean id="authenticationInterceptor" class="kr.co.greenart.common.interceptor.LoginInterceptor">
</beans:bean>

<interceptors>
  <interceptor>
    <mapping path="/**"/>
    <exclude-mapping path="/member/**"/>
    <exclude-mapping path="/**/*.*css"/>
    <exclude-mapping path="/**/*.*js"/>
    <beans:ref bean="authenticationInterceptor"/>
  </interceptor>
</interceptors>

<context:component-scan base-package="kr.co.greenart" />
```

인터셉터를 생성할 클래스 경로



인터셉터를 적용할 경로



인터셉터를 제외할 경로



게시판 구현하기 – 인터셉터 설정

- **kr.co.greenart.common.interceptor.LoginInterceptor**

인터셉터의 preHandle에 로그인한 유저인지 확인할 코드를 작성

```
public class LoginInterceptor implements HandlerInterceptor {  
  
    @Override  
    public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler) throws Exception {  
        HttpSession session = request.getSession();  
        Integer memberId = (Integer) session.getAttribute("memberId");  
  
        if (memberId == null) {  
            // 로그인되지 않은 사용자이므로 다른 페이지로 리다이렉트  
            response.sendRedirect(request.getContextPath() + "/member/redirect.do");  
            return false;  
        }  
  
        return true;  
    }  
  
    @Override  
    public void postHandle(HttpServletRequest request, HttpServletResponse response, Object handler, ModelAndView modelAndView) throws Exception {  
        // 이후 처리가 필요한 경우 작성  
    }  
  
    @Override  
    public void afterCompletion(HttpServletRequest request, HttpServletResponse response, Object handler, Exception exception) throws Exception {  
        // 이후 처리가 필요한 경우 작성  
    }  
}
```


게시판 구현하기 – 인터셉터 설정

- **kr.co.greenart.common.interceptor.LoginInterceptor**

인터셉터의 preHandle에 로그인한 유저인지 확인할 코드를 작성

```
public class LoginInterceptor implements HandlerInterceptor {  
  
    @Override  
    public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler) throws Exception {  
        HttpSession session = request.getSession();  
        Integer memberId = (Integer) session.getAttribute("memberId");  
  
        if (memberId == null) {  
            // 로그인되지 않은 사용자이므로 다른 페이지로 리다이렉트  
            response.sendRedirect(request.getContextPath() + "/member/redirect.do");  
            return false;  
        }  
  
        return true;  
    }  
  
    @Override  
    public void postHandle(HttpServletRequest request, HttpServletResponse response, Object handler, ModelAndView modelAndView) throws Exception {  
        // 이후 처리가 필요한 경우 작성  
    }  
  
    @Override  
    public void afterCompletion(HttpServletRequest request, HttpServletResponse response, Object handler, Exception ex) throws Exception {  
        // 이후 처리가 필요한 경우 작성  
    }  
}
```

게시판 구현하기 – 인터셉터 설정

- **kr.co.greenart.member.controller.MemberController**

인터셉터에서 GET 요청을 받아 리다이렉트할 메소드 생성

```
@GetMapping("/redirect.do")  
public String redirectIndex() {  
    return "member/login";  
}
```

게시판 구현하기 – 인터셉터 설정

- 인터셉터 동작 확인하기

기존에 만들었던 Login한 유저인지 확인하던 코드를 주석처리한 후 인터셉터가 제대로 동작하는지 확인