

6장 미니 MVC 프레임워크 만들기

6장 미니 MVC 프레임워크 만들기

- ✓ 스프링 MVC를 모방한 간단한 MVC 프레임워크
만들어 보기

6장 미니 MVC 프레임워크 만들기

- ✓ 스프링 MVC를 모방한 간단한 MVC 프레임워크 만들어 보기
- ✓ 스프링 프레임워크의 구동 원리와 내부 구조를 이해하기

6장 미니 MVC 프레임워크 만들기

- ✓ 스프링 MVC를 모방한 간단한 MVC 프레임워크 만들어 보기
- ✓ 스프링 프레임워크의 구동 원리와 내부 구조를 이해하기
- ✓ 디자인 패턴 적용하는 방법과 오픈 소스 라이브러리를 사용하는 방법을 경험하기

6장 미니 MVC 프레임워크 만들기

- ✓ 스프링 MVC를 모방한 간단한 MVC 프레임워크 만들어 보기
- ✓ 스프링 프레임워크의 구동 원리와 내부 구조를 이해하기
- ✓ 디자인 패턴 적용하는 방법과 오픈 소스 라이브러리를 사용하는 방법을 경험하기
- ✓ 리플렉션 API 사용법 익히기

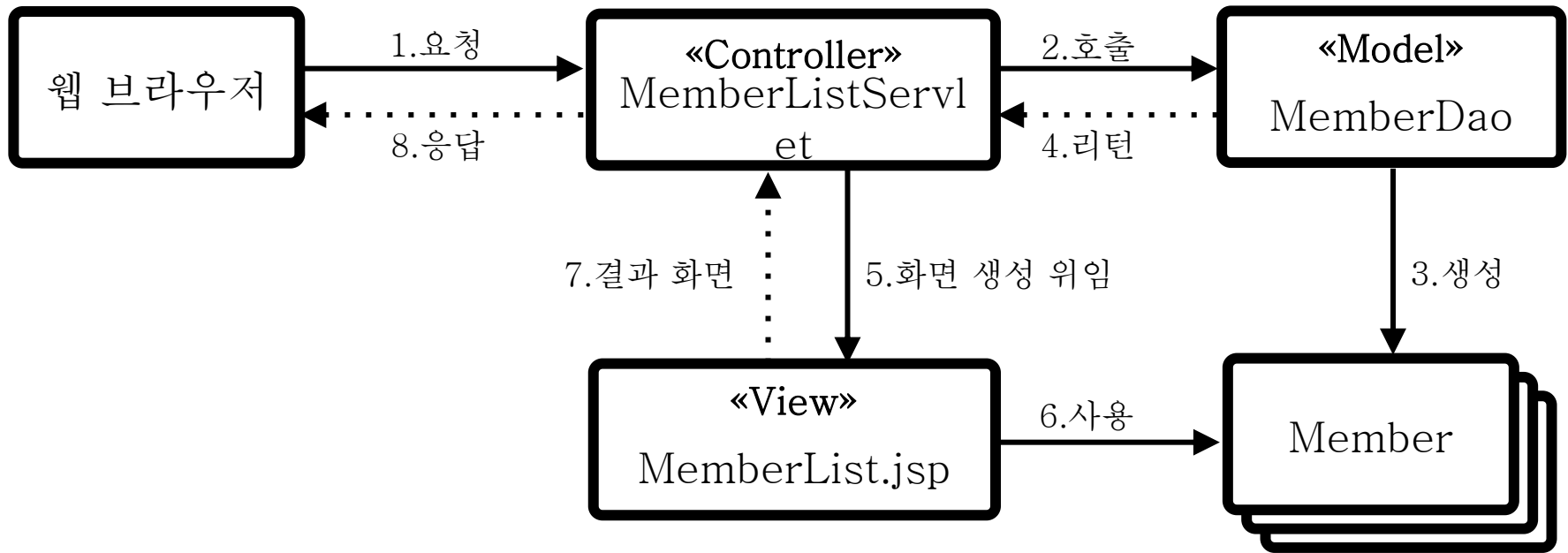
6장 미니 MVC 프레임워크 만들기

- ✓ 스프링 MVC를 모방한 간단한 MVC 프레임워크 만들어 보기
- ✓ 스프링 프레임워크의 구동 원리와 내부 구조를 이해하기
- ✓ 디자인 패턴 적용하는 방법과 오픈 소스 라이브러리를 사용하는 방법을 경험하기
- ✓ 리플렉션 API 사용법 익히기
- ✓ 애노테이션을 정의하고 사용하는 방법 익히기

6.1 프런트 컨트롤러의 도입

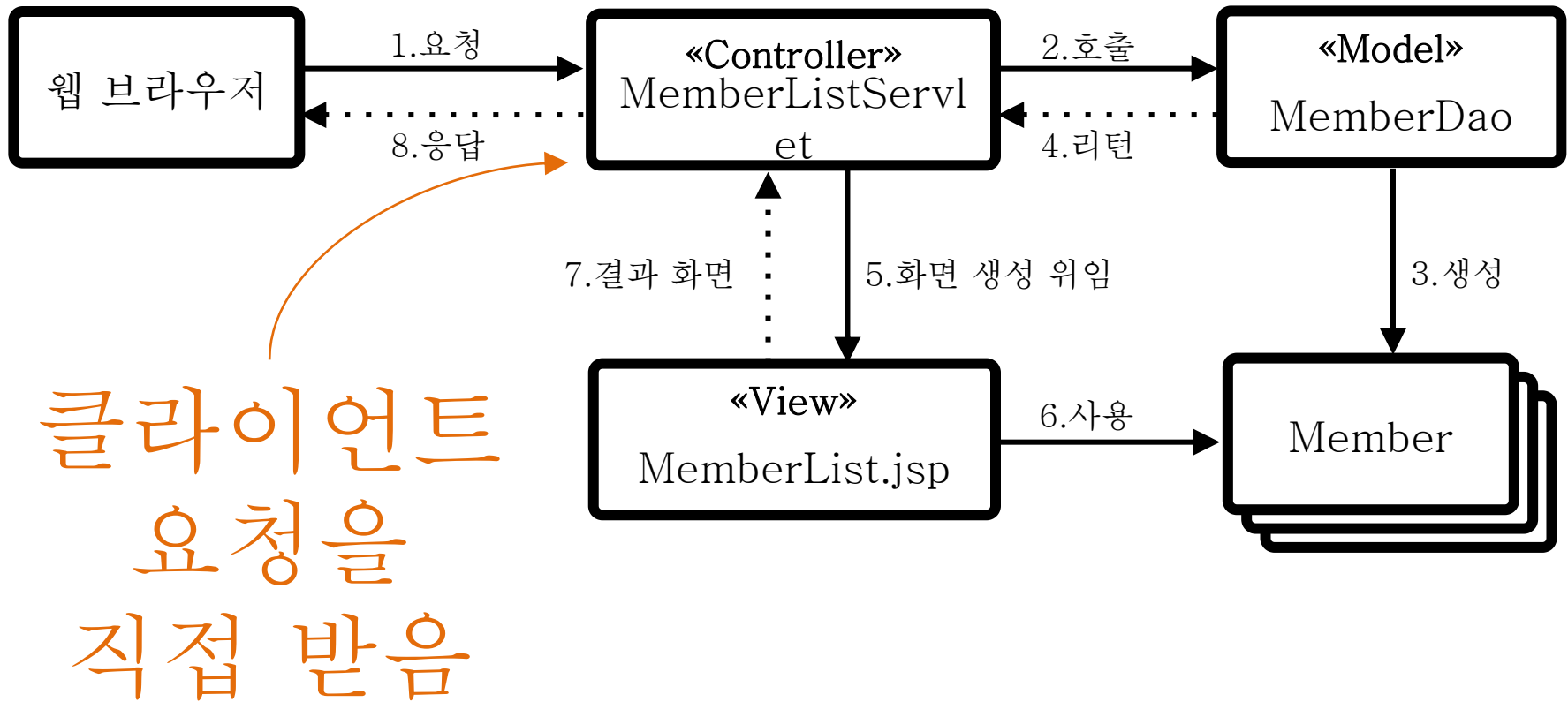
6.1 프런트 컨트롤러의 도입

프런트 컨트롤러 도입 전



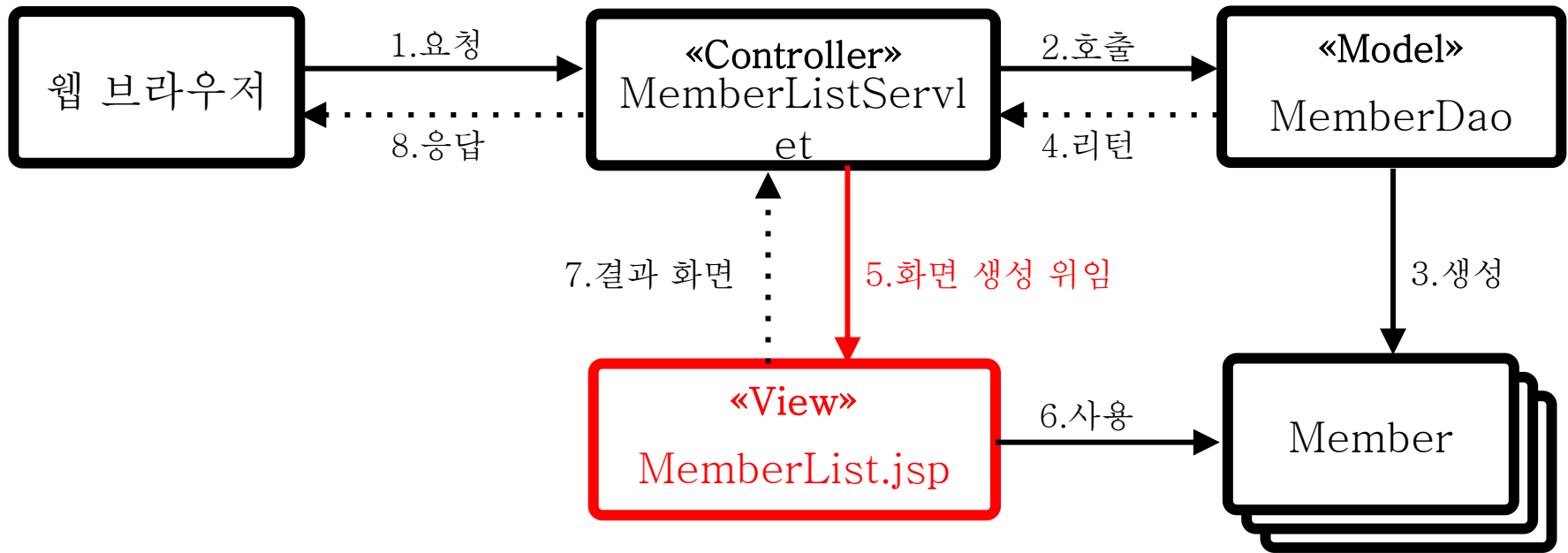
6.1 프런트 컨트롤러의 도입

프런트 컨트롤러 도입 전



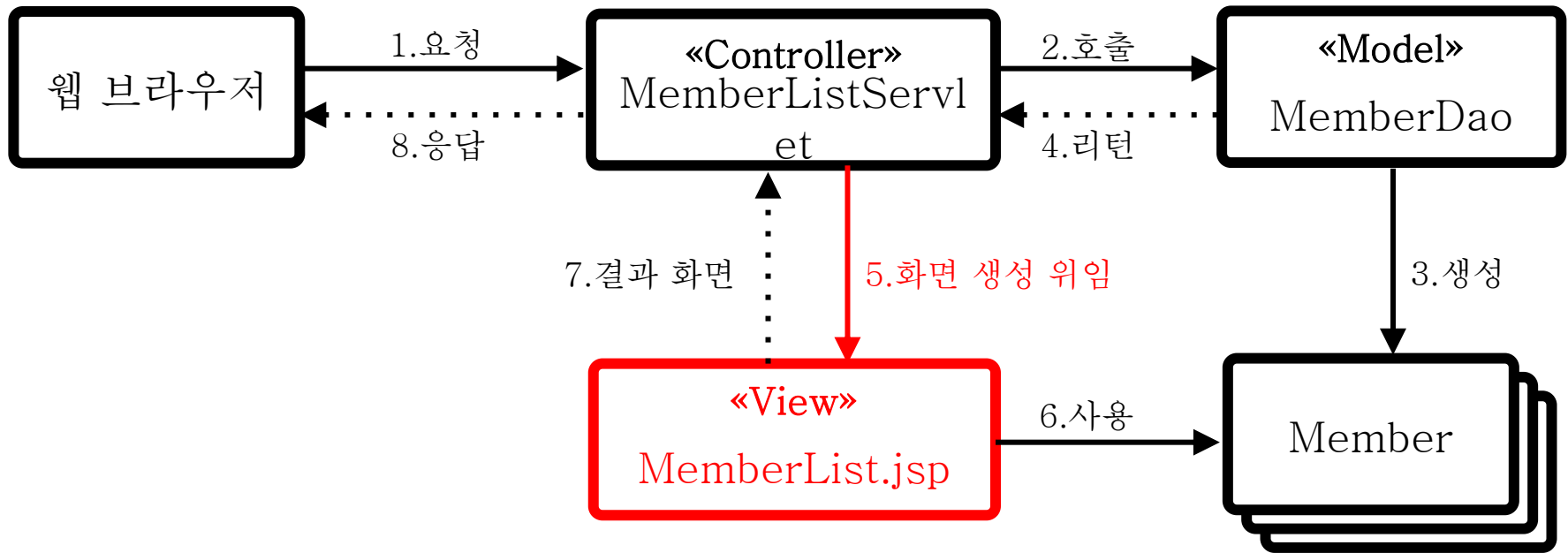
6.1 프런트 컨트롤러의 도입

각각의 컨트롤러에서
화면 생성을 위임하는 코드를 작성함



6.1 프런트 컨트롤러의 도입

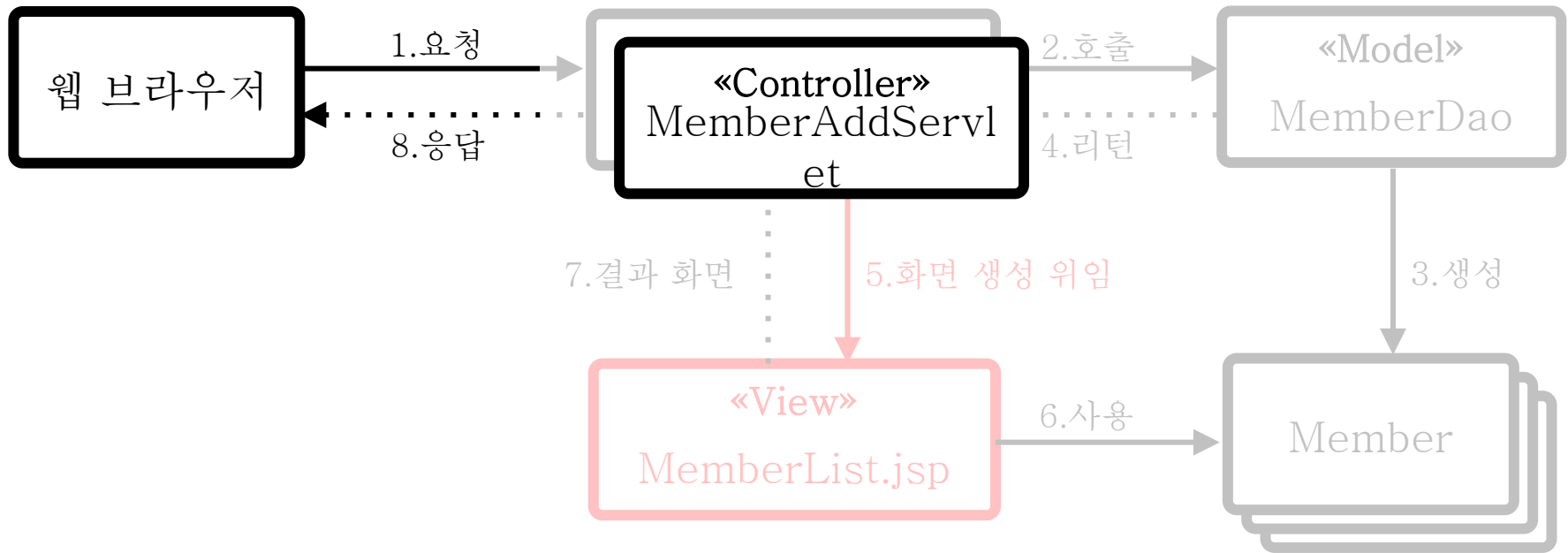
각각의 컨트롤러에서
화면 생성을 위임하는 코드를 작성함



```
RequestDispatcher rd = request.getRequestDispatcher("/member/MemberList.jsp");  
rd.include(request, response);
```

6.1 프런트 컨트롤러의 도입

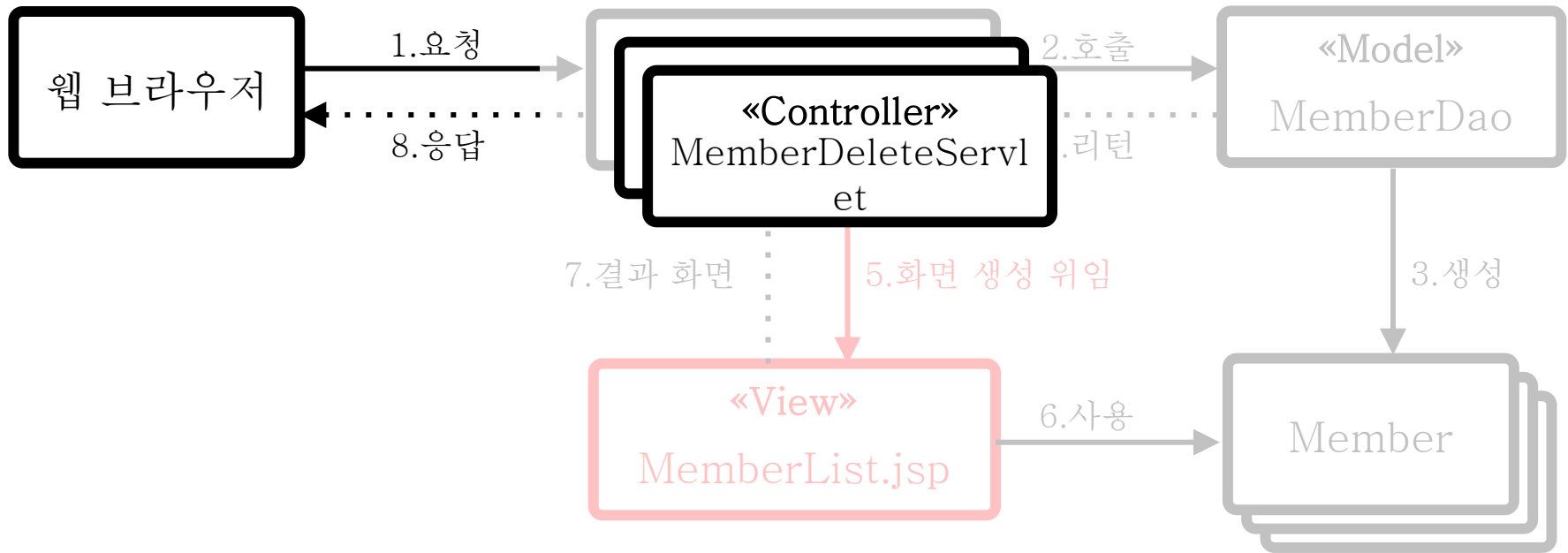
화면을 생성 하지 않는 경우,



```
RequestDispatcher rd = request.getRequestDispatcher("/member/MemberList.jsp");  
rd.include(request, response);
```

6.1 프런트 컨트롤러의 도입

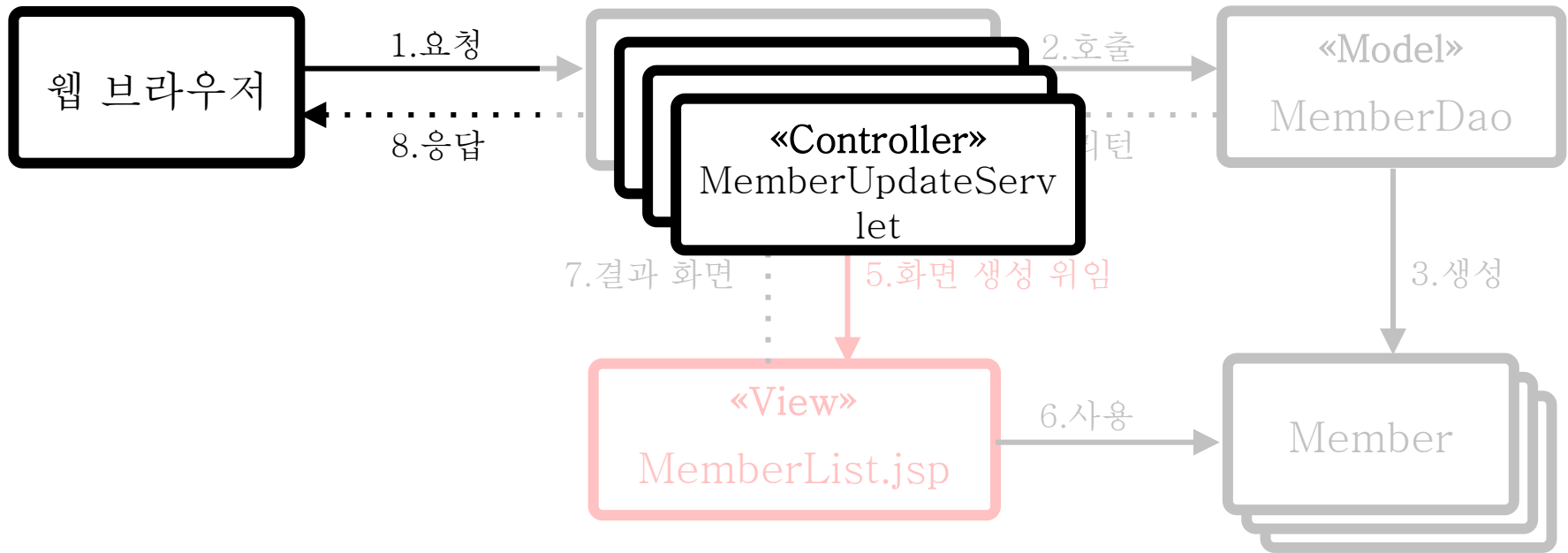
화면을 생성 하지 않는 경우,



```
RequestDispatcher rd = request.getRequestDispatcher("/member/MemberList.jsp");  
rd.include(request, response);
```

6.1 프런트 컨트롤러의 도입

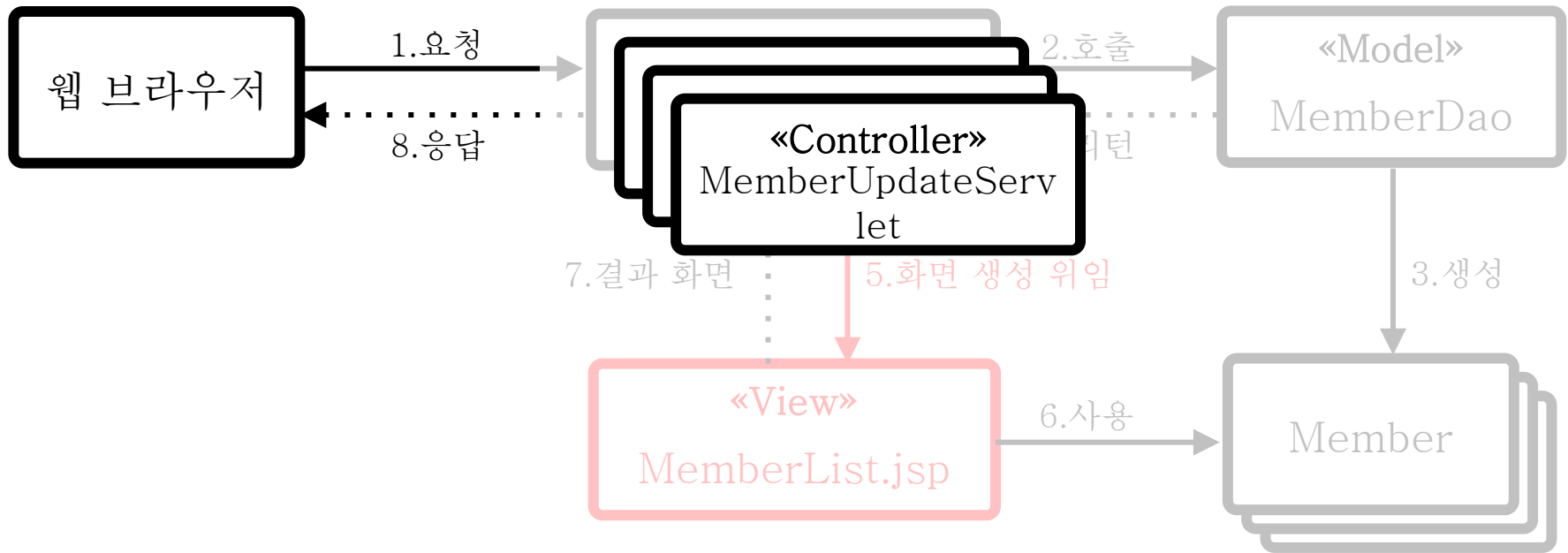
화면을 생성 하지 않는 경우,



```
RequestDispatcher rd = request.getRequestDispatcher("/member/MemberList.jsp");  
rd.include(request, response);
```

6.1 프런트 컨트롤러의 도입

화면을 생성 하지 않는 경우,
다른 페이지로 보내기 위해 리다이렉트로
처리함

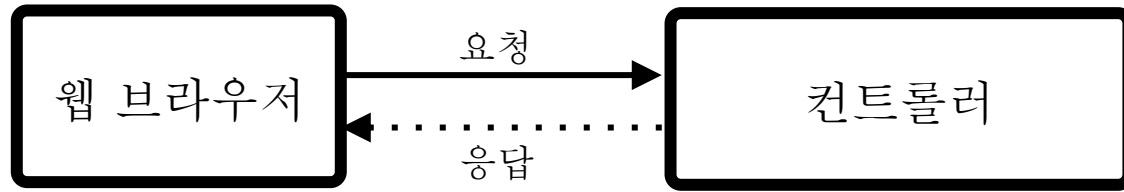


```
RequestDispatcher rd = request.getRequestDispatcher("/member/MemberList.jsp");
rd.include(request, response);
```

```
response.sendRedirect("list");
```

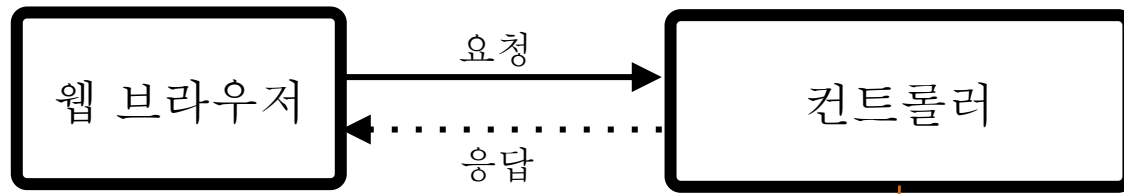
6.1 프런트 컨트롤러의 도입

6장 1절의 학습 목표



6.1 프런트 컨트롤러의 도입

컨트롤러가 하는 일 중에서
공통 또는 반복적으로 하는 작업을
추출하여

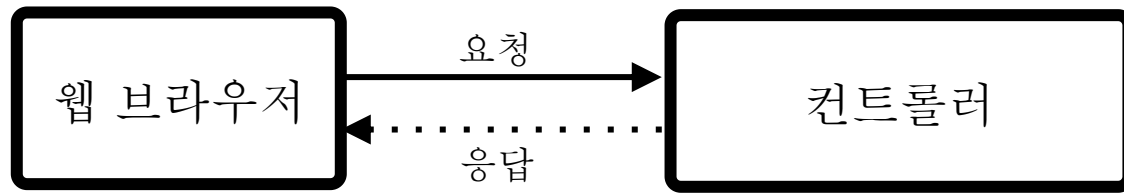


컨트롤러가
공통으로
하는 작업

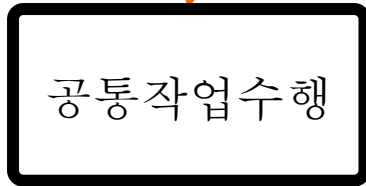
- JSP로 실행을 위임하는 코드
- 다른 페이지로 리다이렉트 시키는 코드
- 파라미터를 값 객체로 만드는 코드
- ...

6.1 프런트 컨트롤러의 도입

컨트롤러가 하는 일 중에서
공통 또는 반복적으로 하는 작업을
추출하여
클래스로 정의하고



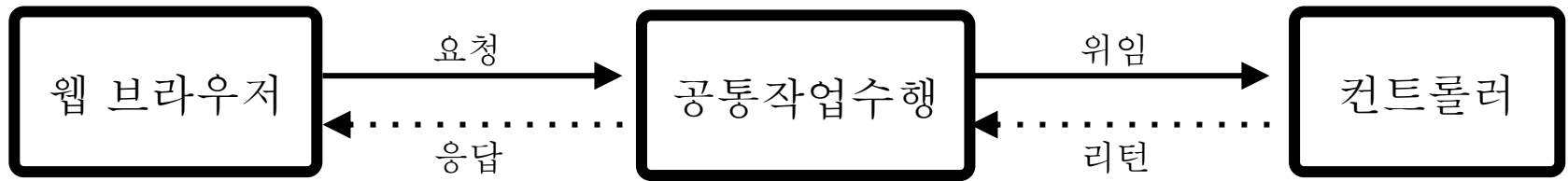
클래스로
정의



- JSP로 실행을 위임하는 코드
- 다른 페이지로 리다이렉트 시키는 코드
- 파라미터를 값 객체로 만드는 코드
- ...

6.1 프런트 컨트롤러의 도입

컨트롤러가 작업을 수행하기 전에
공통 작업을 먼저 수행



6.1 프런트 컨트롤러의 도입

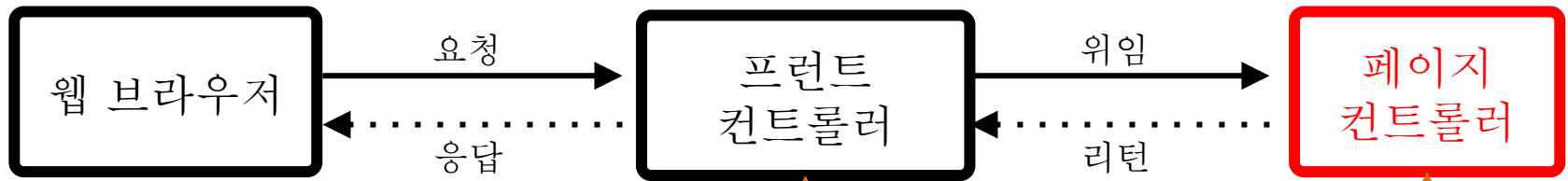
컨트롤러가 작업을 수행하기 전에
공통 작업을 먼저 수행



쪽에서 공통 작업을
수행

6.1 프런트 컨트롤러의 도입

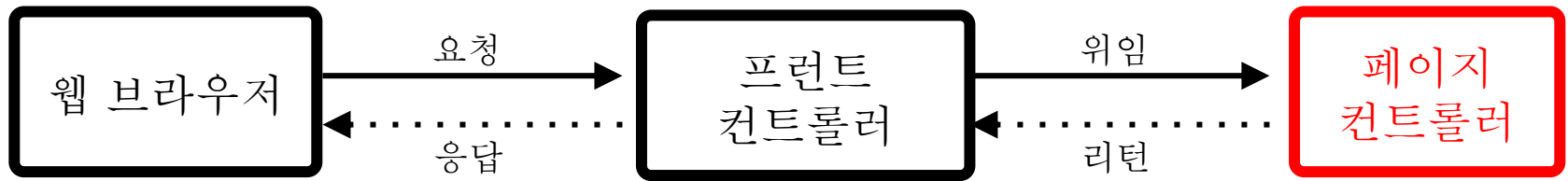
컨트롤러가 작업을 수행하기 전에
공통 작업을 먼저 수행



쪽에서 공통 작업을
수행
특정 페이지만을 위한
작업 수행

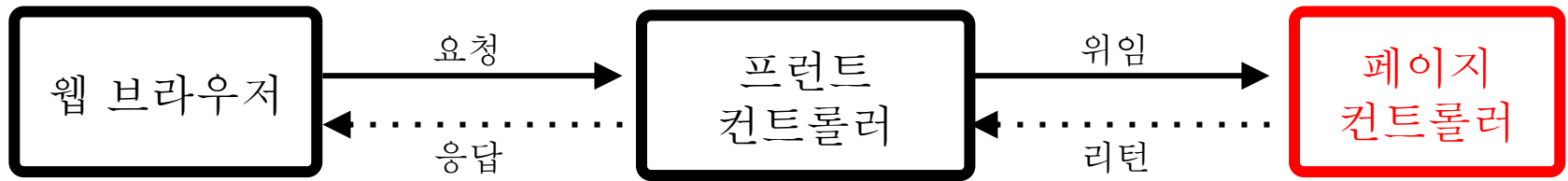
6.1 프런트 컨트롤러의 도입

프런트 컨트롤러 도입하면,



6.1 프런트 컨트롤러의 도입

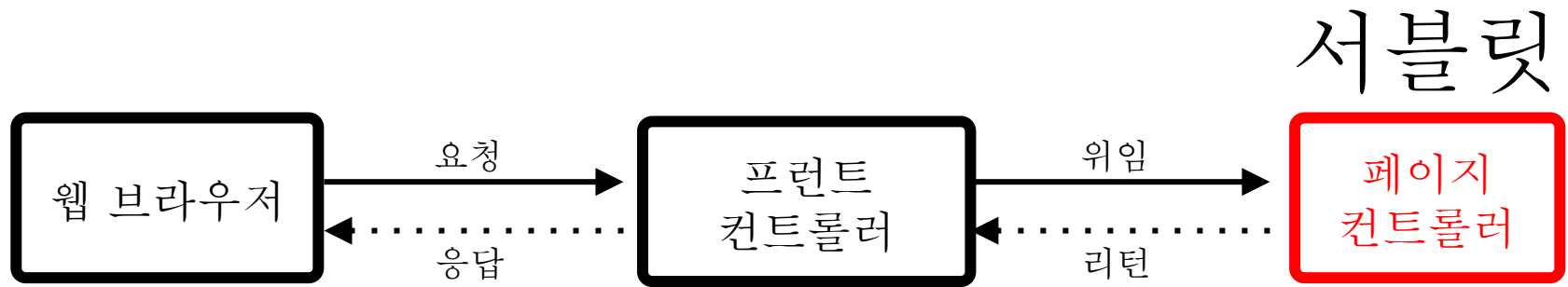
프런트 컨트롤러 도입하면,



- 페이지 컨트롤러의 코드가 간결해짐

6.1 프런트 컨트롤러의 도입

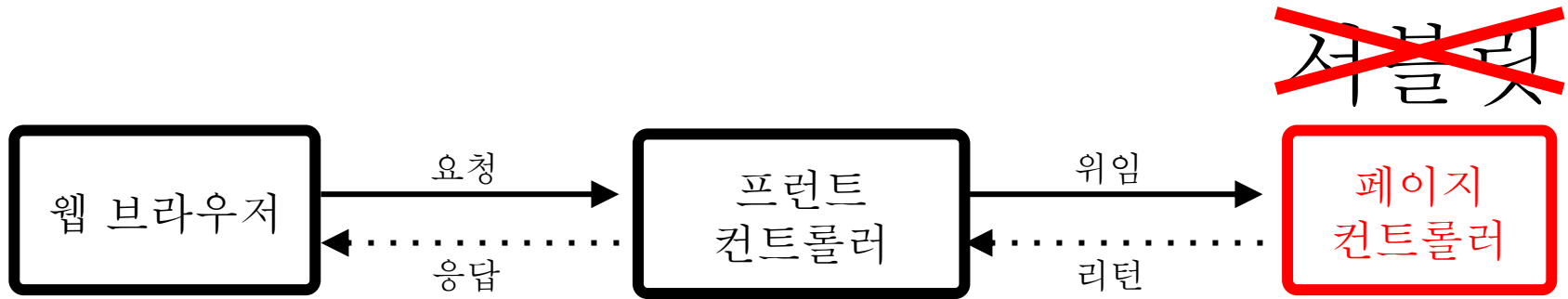
프런트 컨트롤러 도입하면,



- 페이지 컨트롤러의 코드가 간결해짐

6.1 프런트 컨트롤러의 도입

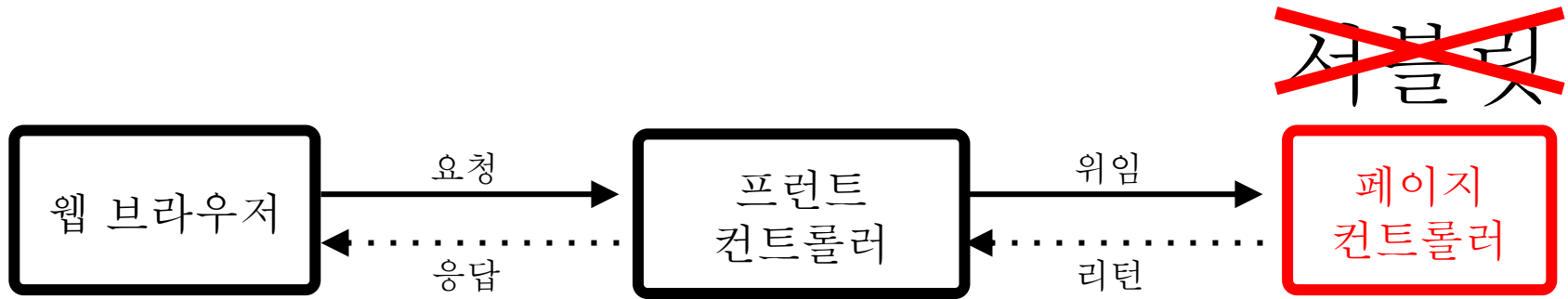
프런트 컨트롤러 도입하면,



- 페이지 컨트롤러의 코드가 간결해짐
- 페이지 컨트롤러는 서블릿일 필요가 없음

6.1 프런트 컨트롤러의 도입

프런트 컨트롤러 도입하면,



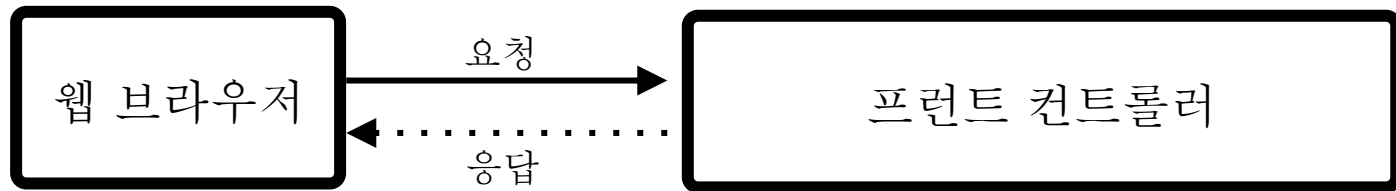
- 페이지 컨트롤러의 코드가 간결해짐
- 페이지 컨트롤러는 서블릿일 필요가 없음(6장2절)

6.1 프런트 컨트롤러의 도입

프런트 컨트롤러 만들기

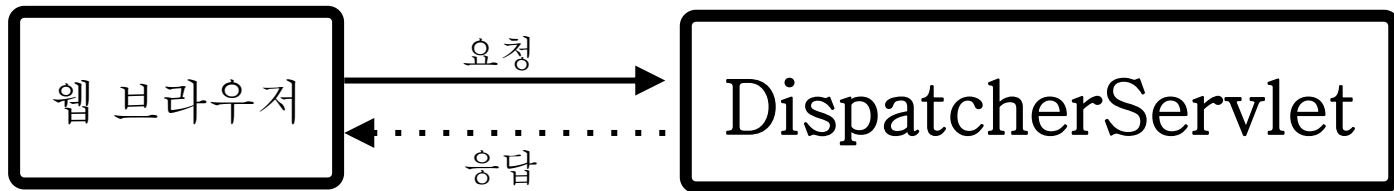
6.1 프런트 컨트롤러의 도입

프런트 컨트롤러를 만들어 보자



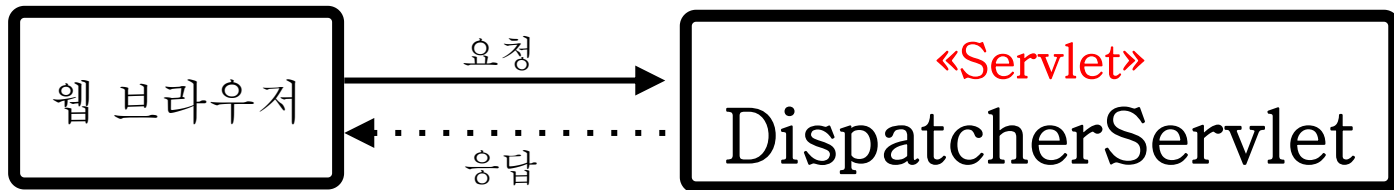
6.1 프런트 컨트롤러의 도입

프런트 컨트롤러의 클래스 이름은?



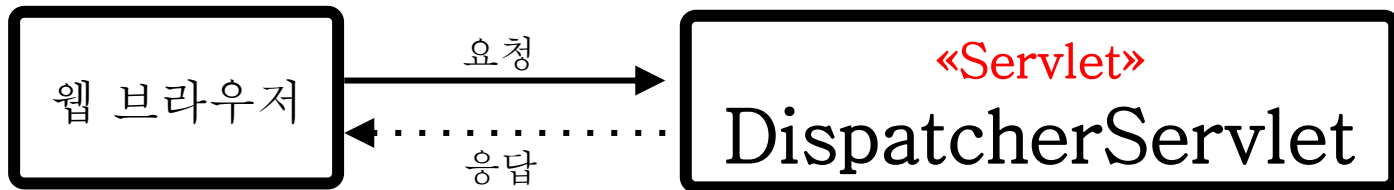
6.1 프런트 컨트롤러의 도입

클라이언트 요청을 받아야 하기 때문에
서블릿이어야 함



6.1 프런트 컨트롤러의 도입

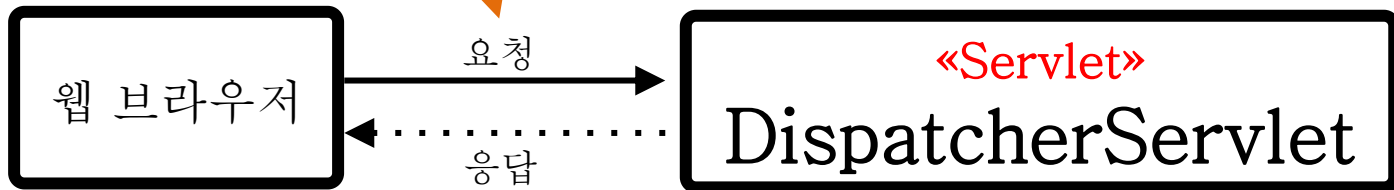
페이지 컨트롤러를 실행하기 전에,
공통 작업을 처리해야 하기 때문에,



6.1 프런트 컨트롤러의 도입

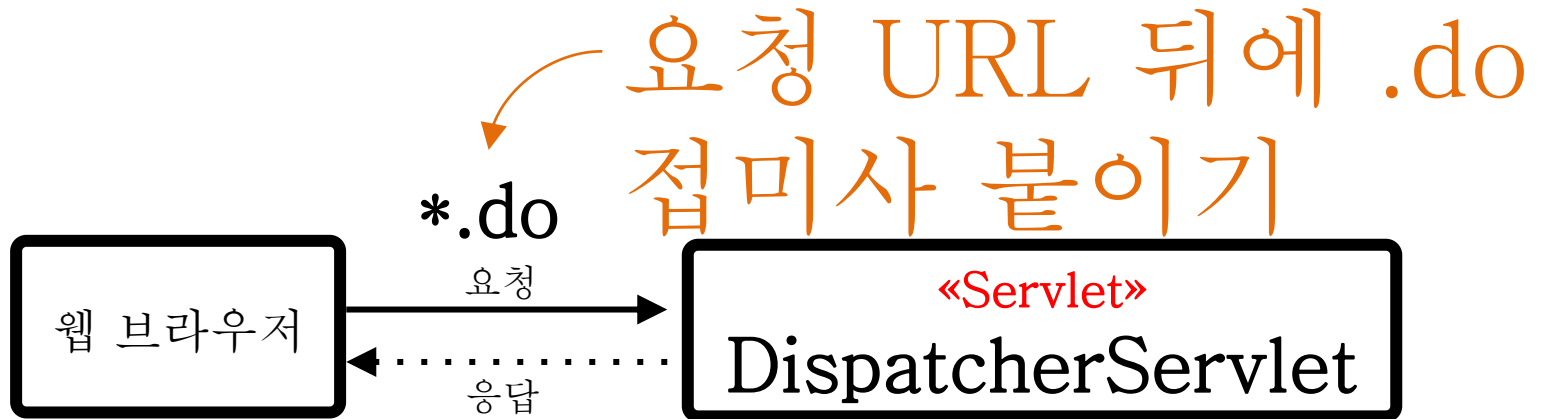
요청 URL의 규칙을 정의할 필요 있음

요청에 대한 규칙이
필요



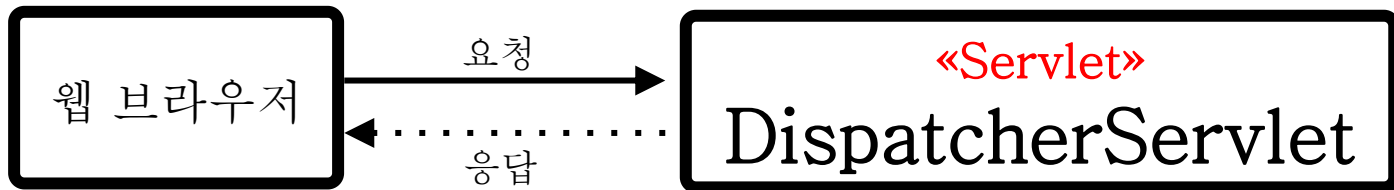
6.1 프런트 컨트롤러의 도입

즉 .do 접미사가 붙은 요청이 들어오면
DispatcherServlet에게 보냄



6.1 프런트 컨트롤러의 도입

즉 .do 접미사가 붙은 요청이 들어오면
DispatcherServlet에게 보냄

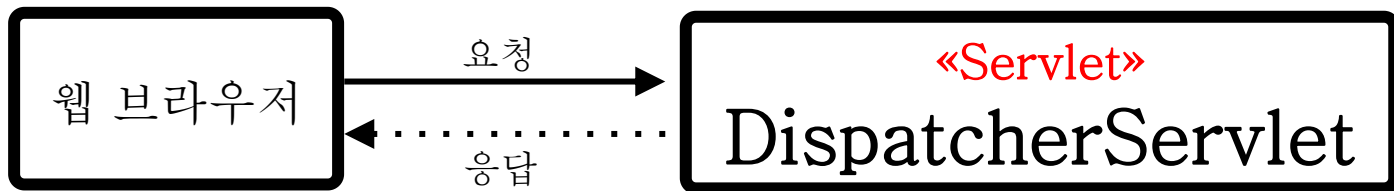


기존의 회원 목록 요청 URL

<http://localhost:9999/web05/member/list>

6.1 프런트 컨트롤러의 도입

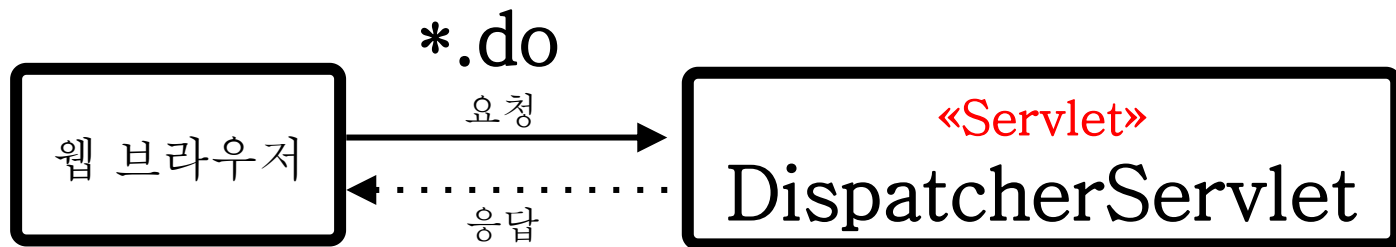
즉 .do 접미사가 붙은 요청이 들어오면
DispatcherServlet에게 보냄



회원 목록 요청 URL

<http://localhost:9999/web05/member/list.do>

6.1 프런트 컨트롤러의 도입
.do 로 끝나는 URL 요청을 처리하게
하려면,
서블릿 선언에 URL 매핑 정보를 설정해야
함



@WebServlet("*.do")

public class DispatcherServlet extends HttpServlet {

...
}

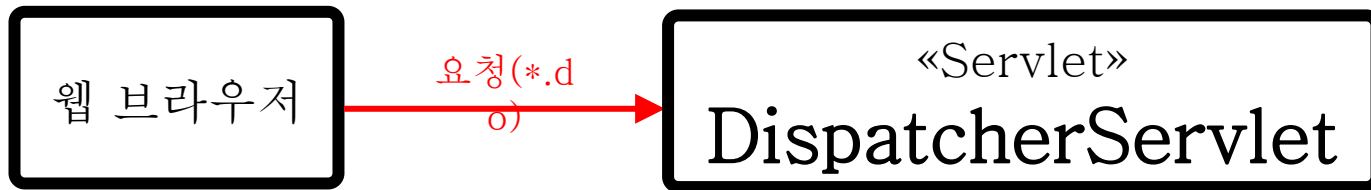
URL 매핑
지정

6.1 프런트 컨트롤러의 도입

프런트 컨트롤러가
요청을 처리하는 과정

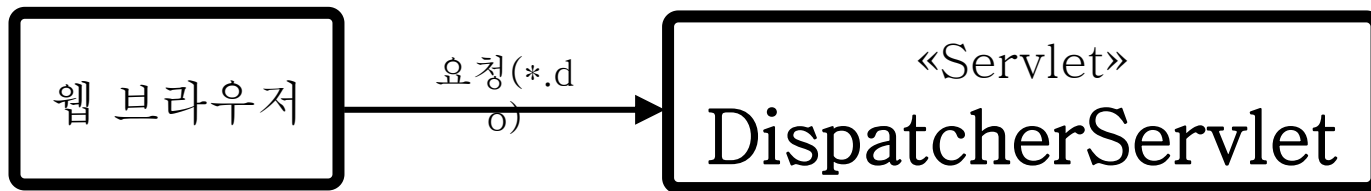
6.1 프런트 컨트롤러의 도입

웹브라우저로부터 *.do 요청이
들어오면,



6.1 프런트 컨트롤러의 도입

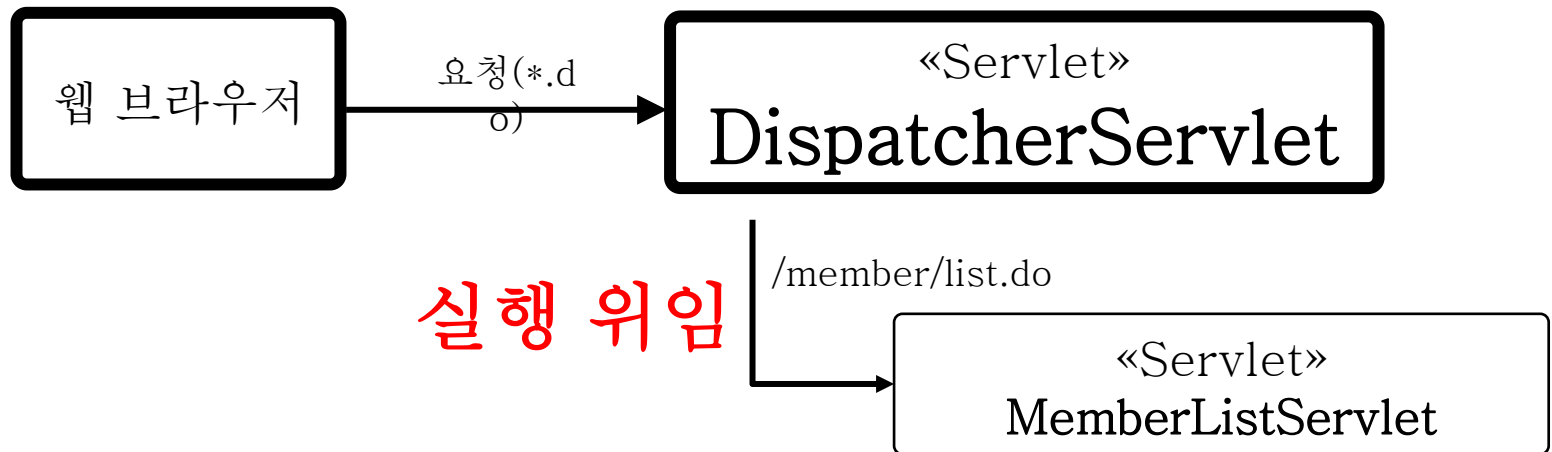
웹브라우저로부터 *.do 요청이 들어오면,
URL에서 서블릿 경로를 추출함



```
String servletPath = request.getServletPath();
```

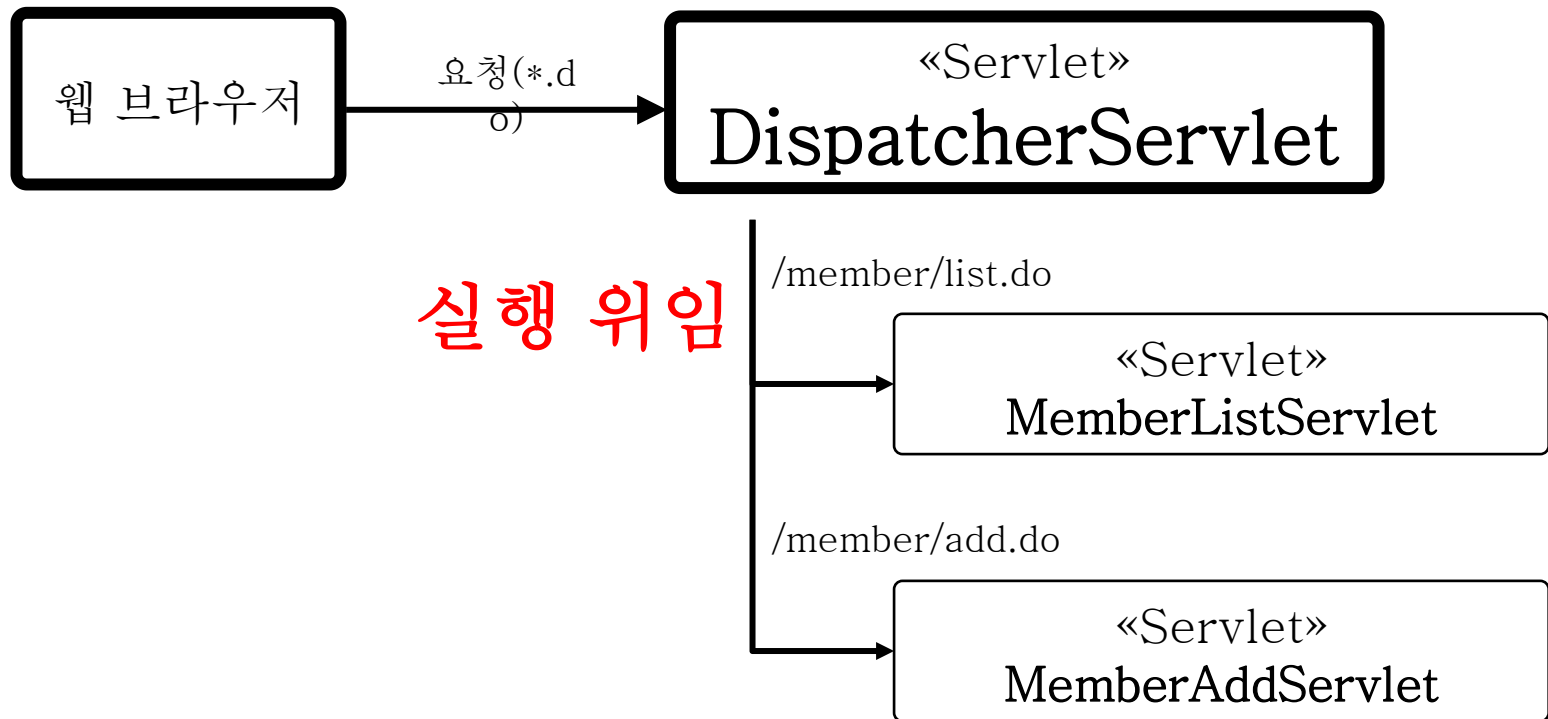
6.1 프런트 컨트롤러의 도입

서블릿 경로에 해당하는
페이지 컨트롤러를 찾아 실행을 위임함



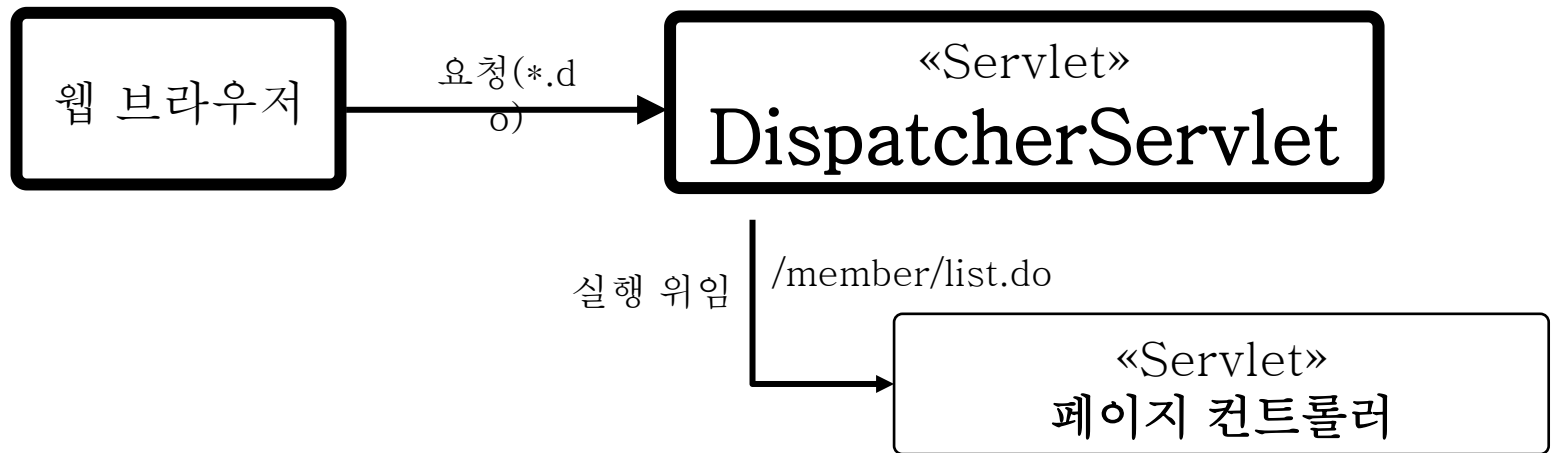
6.1 프런트 컨트롤러의 도입

서블릿 경로에 해당하는
페이지 컨트롤러를 찾아 실행을 위임함



6.1 프런트 컨트롤러의 도입

페이지 컨트롤러는 작업을 마친 후,
JSP URL 또는 리다이렉트 URL을 리턴함

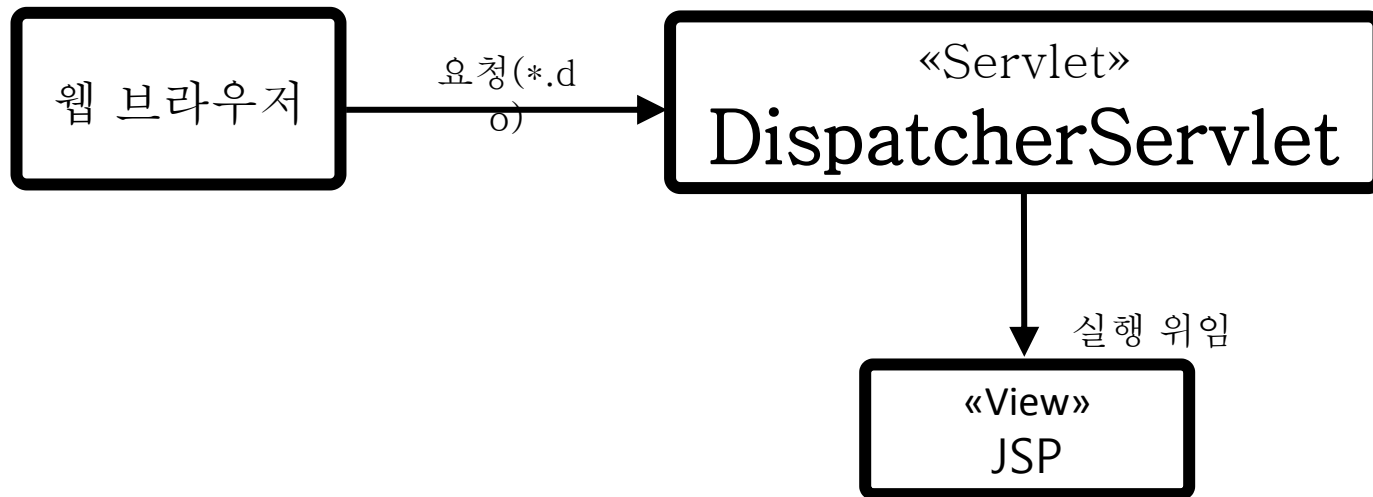


```
request.setAttribute("viewUrl", "/member/MemberList.jsp");
```

```
request.setAttribute("viewUrl", "redirect:list.do");
```

6.1 프런트 컨트롤러의 도입

프런트 컨트롤러는 페이지 컨트롤러가
알려준 JSP로 실행을 위임함

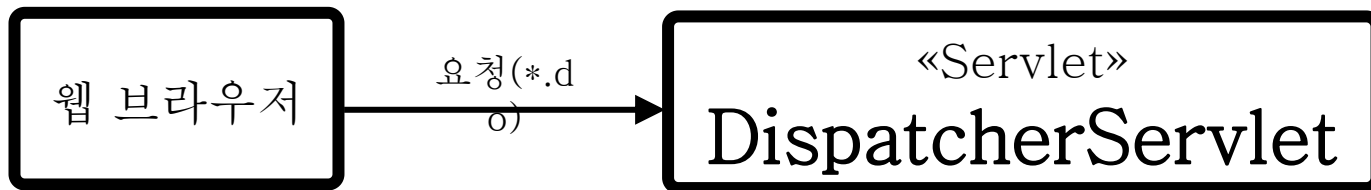


```
String viewUrl = (String) request.getAttribute("viewUrl");
```

```
rd = request.getRequestDispatcher(viewUrl);  
rd.include(request, response);
```

6.1 프런트 컨트롤러의 도입

만약 페이지 컨트롤러가 리턴한 것이
리다이렉트 URL이라면,
리다이렉트로 응답을 처리함



```
if (viewUrl.startsWith("redirect:")) {  
    response.sendRedirect(viewUrl.substring(9));  
    return;  
}
```