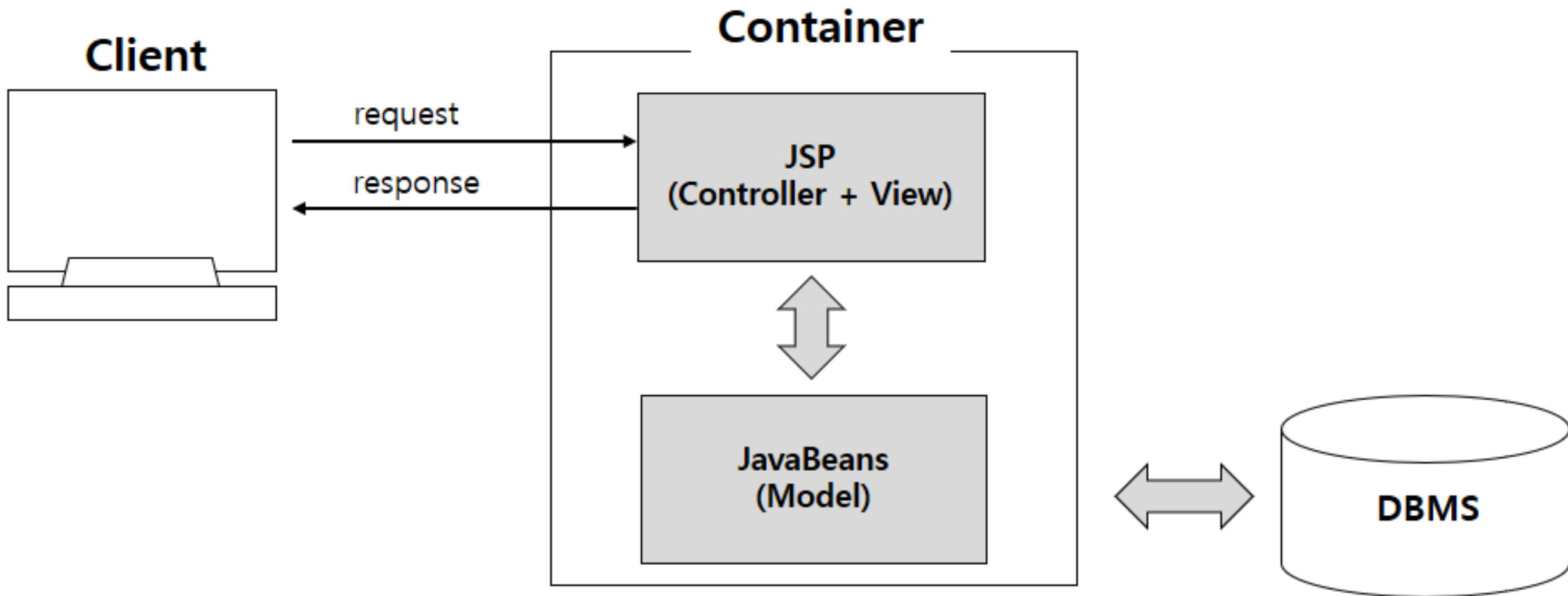


DAY - 03

Spring MVC

웹 개발 모델

- Model 1 아키텍처



- Control 로직

기 능

사용자 입력 정보 추출

DB 연동 처리

화면 내비게이션

사 용 예

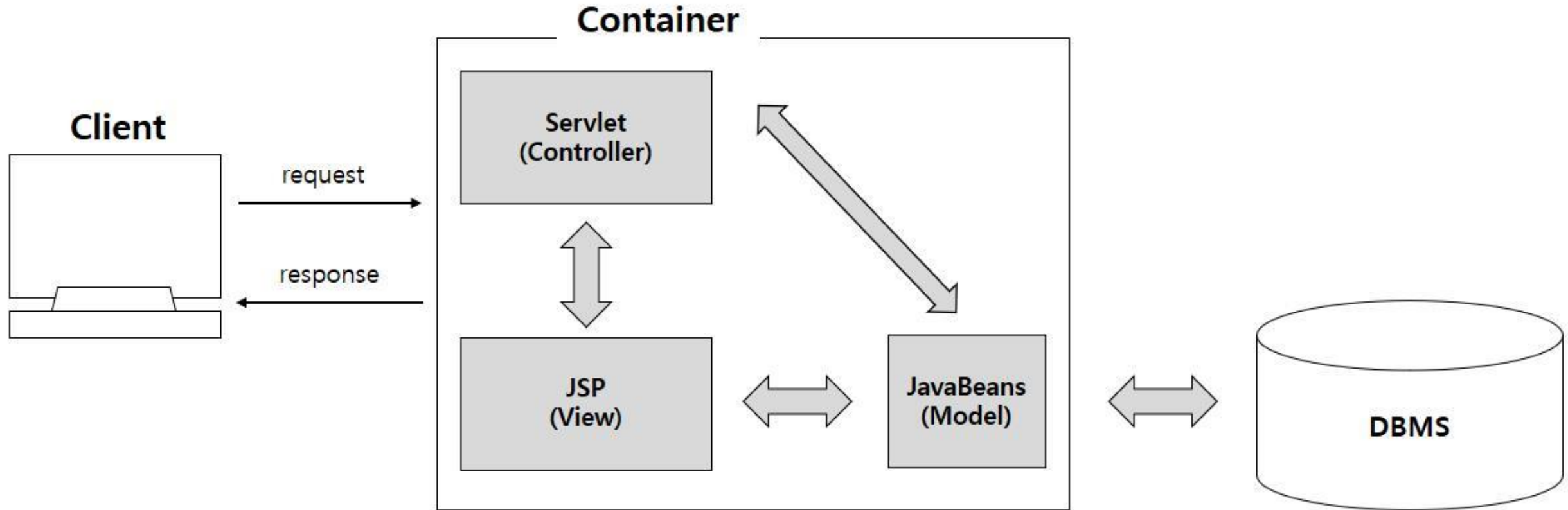
```
String id = request.getParameter("userId")
```

```
UserVO vo = new UserVO();  
vo.setId(id);
```

```
UserDAO userDAO = new UserDAO();  
UserVO user = userDAO.getUser(vo);
```

```
if(user != null) { // 로그인 성공  
    response.sendRedirect("getBoardList.jsp");  
} else {          // 로그인 실패  
    response.sendRedirect("login.jsp");  
}
```

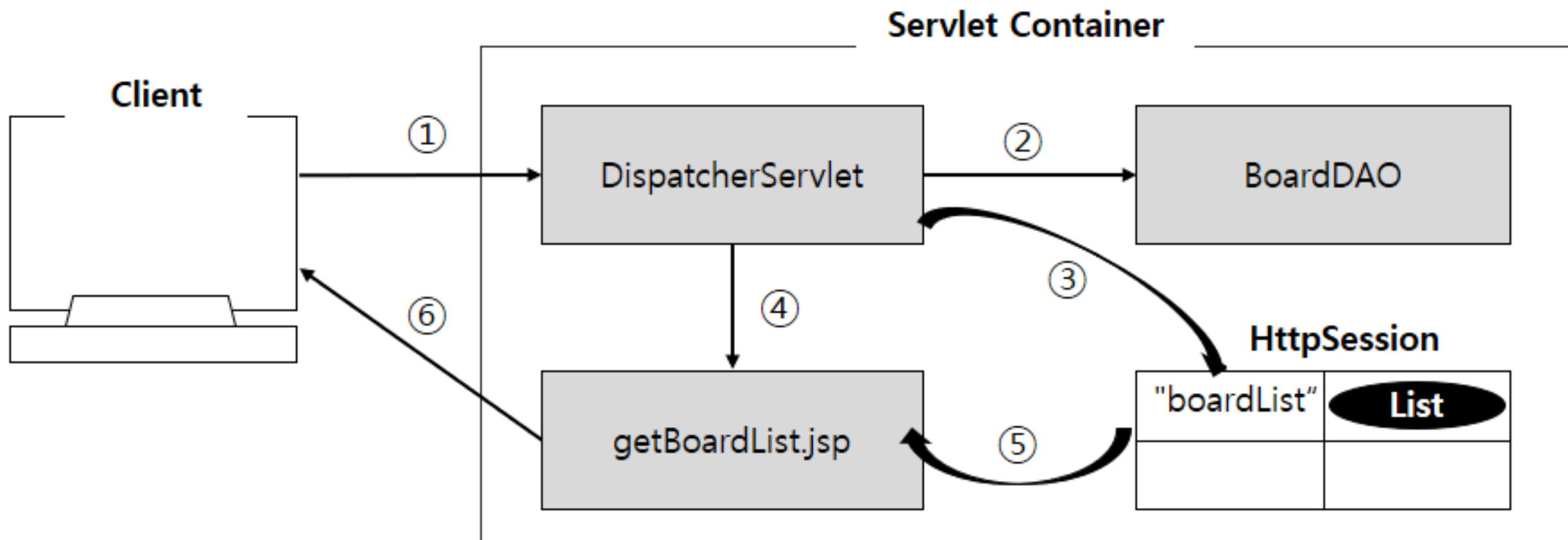
- Model 2 아키텍처 (MVC)



- MVC 주체

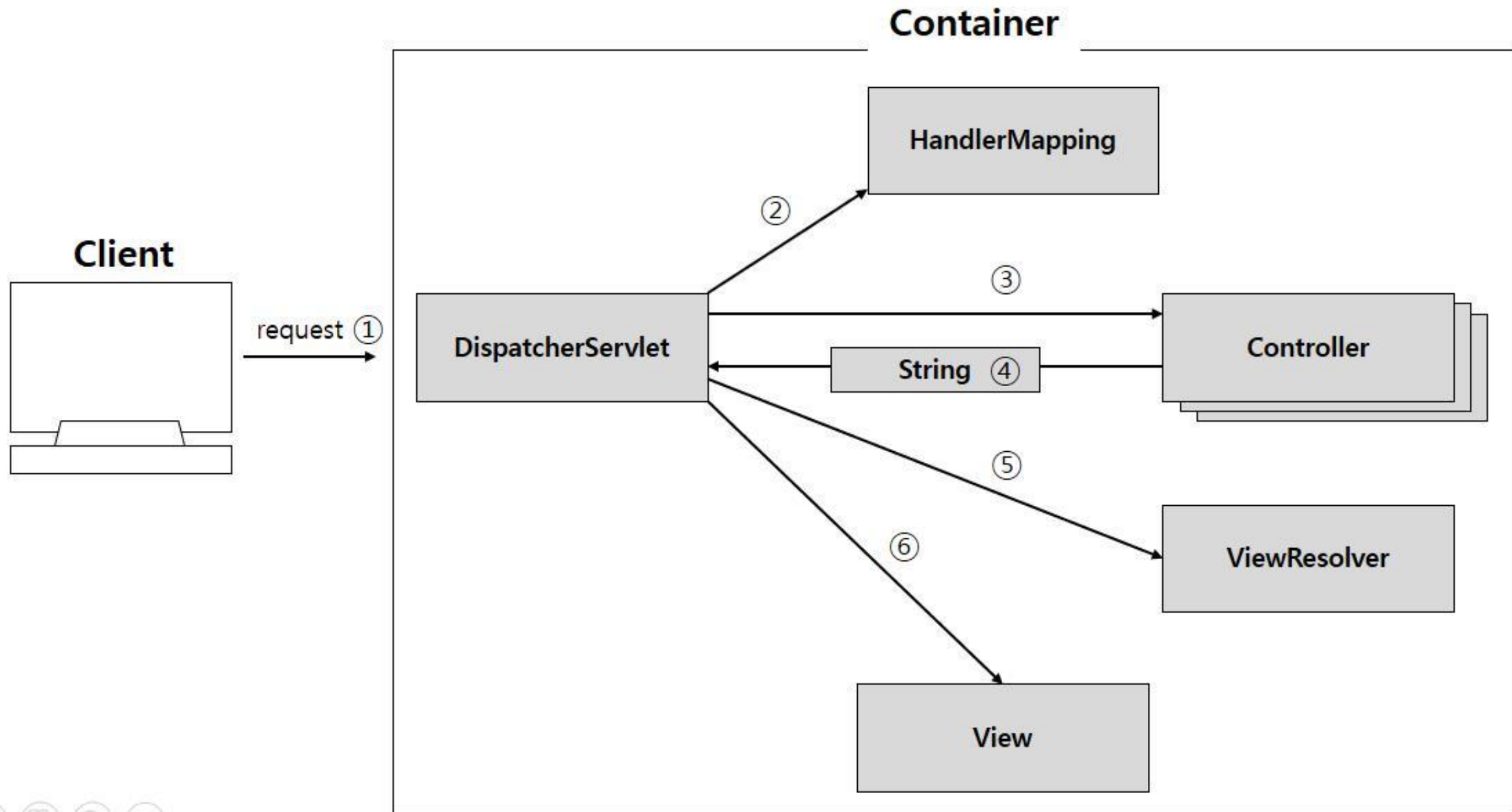
기 능	구성 요소	개발 주체
Model	VO, DAO 클래스	자바 개발자
View	JSP 페이지	웹 디자이너
Controller	Servlet 클래스	자바 개발자 또는 MVC 프레임워크

- MVC Framework 직접 구현



MVC Framework 개발

- MVC Framework 구조



- MVC 각 요소 별 역할

클 래 스	기 능
DispatcherServlet	유일한 서블릿 클래스로서 모든 클라이언트의 요청을 가장 먼저 처리하는 Front Controller
HandlerMapping	클라이언트의 요청을 처리할 Controller 매핑
Controller	실질적인 클라이언트의 요청 처리
ViewResolver	Controller가 반환한 View 이름으로 실행될 JSP 경로 완성

Spring MVC 적용

- DispatcherServlet 등록 (web.xml)

```
<servlet>
```

```
    <servlet-name>action</servlet-name>
```

```
    <servlet-class>
```

```
        org.springframework.web.servlet.DispatcherServlet
```

```
    </servlet-class>
```

```
    <init-param>
```

```
        <param-name>contextConfigLocation</param-name>
```

```
        <param-value>/WEB-INF/config/presentation-layer.xml</param-value>
```

```
    </init-param>
```

```
</servlet>
```

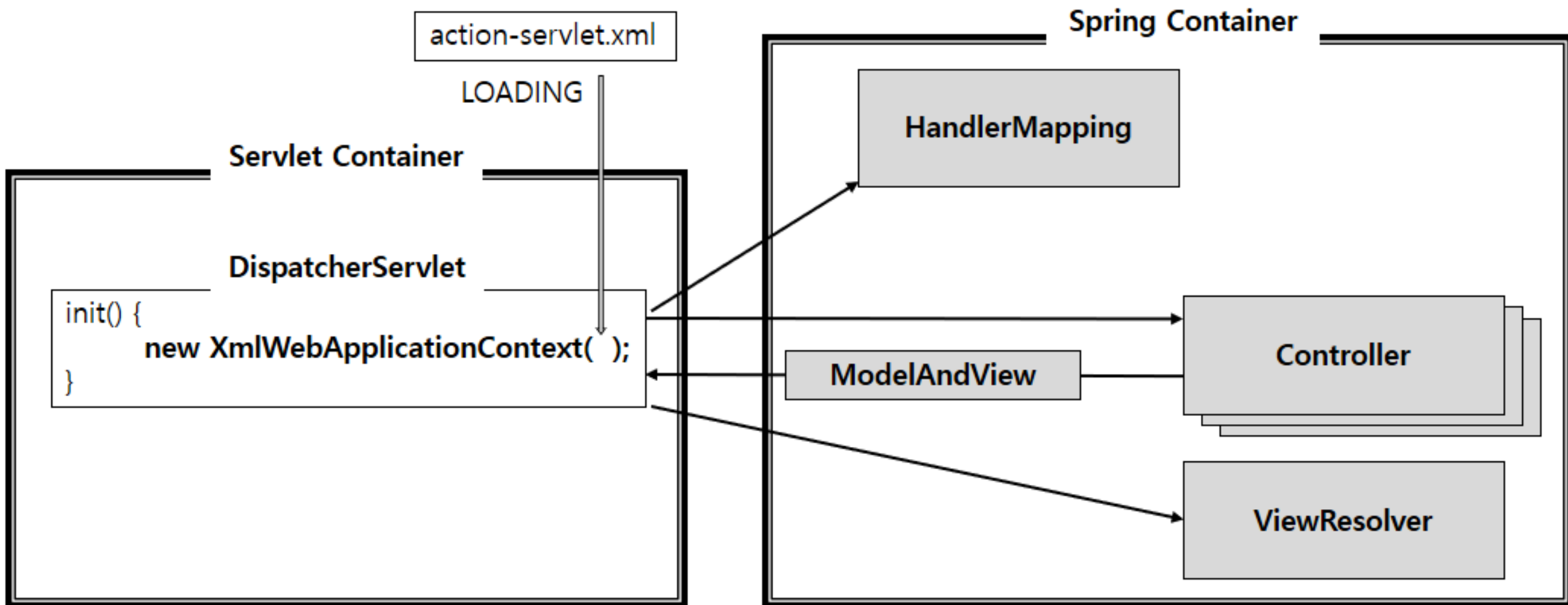
```
<servlet-mapping>
```

```
    <servlet-name>action</servlet-name>
```

```
    <url-pattern>*.do</url-pattern>
```

```
</servlet-mapping>
```

- Servlet Container와 Spring Container의 관계



- 인코딩 설정 (web.xml)

```
<filter>
```

```
    <filter-name>characterEncoding</filter-name>
```

```
    <filter-class>
```

```
        org.springframework.web.filter.CharacterEncodingFilter
```

```
    </filter-class>
```

```
    <init-param>
```

```
        <param-name>encoding</param-name>
```

```
        <param-value>EUC-KR</param-value>
```

```
    </init-param>
```

```
</filter>
```

```
<filter-mapping>
```

```
    <filter-name>characterEncoding</filter-name>
```

```
    <url-pattern>*.do</url-pattern>
```

```
</filter-mapping>
```

- Spring 설정파일 (presentation-layer.xml)

<!-- HandlerMapping 등록 -->

```
<bean class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
```

```
    <property name="mappings">
```

```
        <props>
```

```
            <prop key="/login.do">login</prop>
```

```
        </props>
```

```
    </property>
```

```
</bean>
```

<!-- Controller 등록 -->

```
<bean id="login" class="com.springbook.view.user.LoginController"> </bean>
```



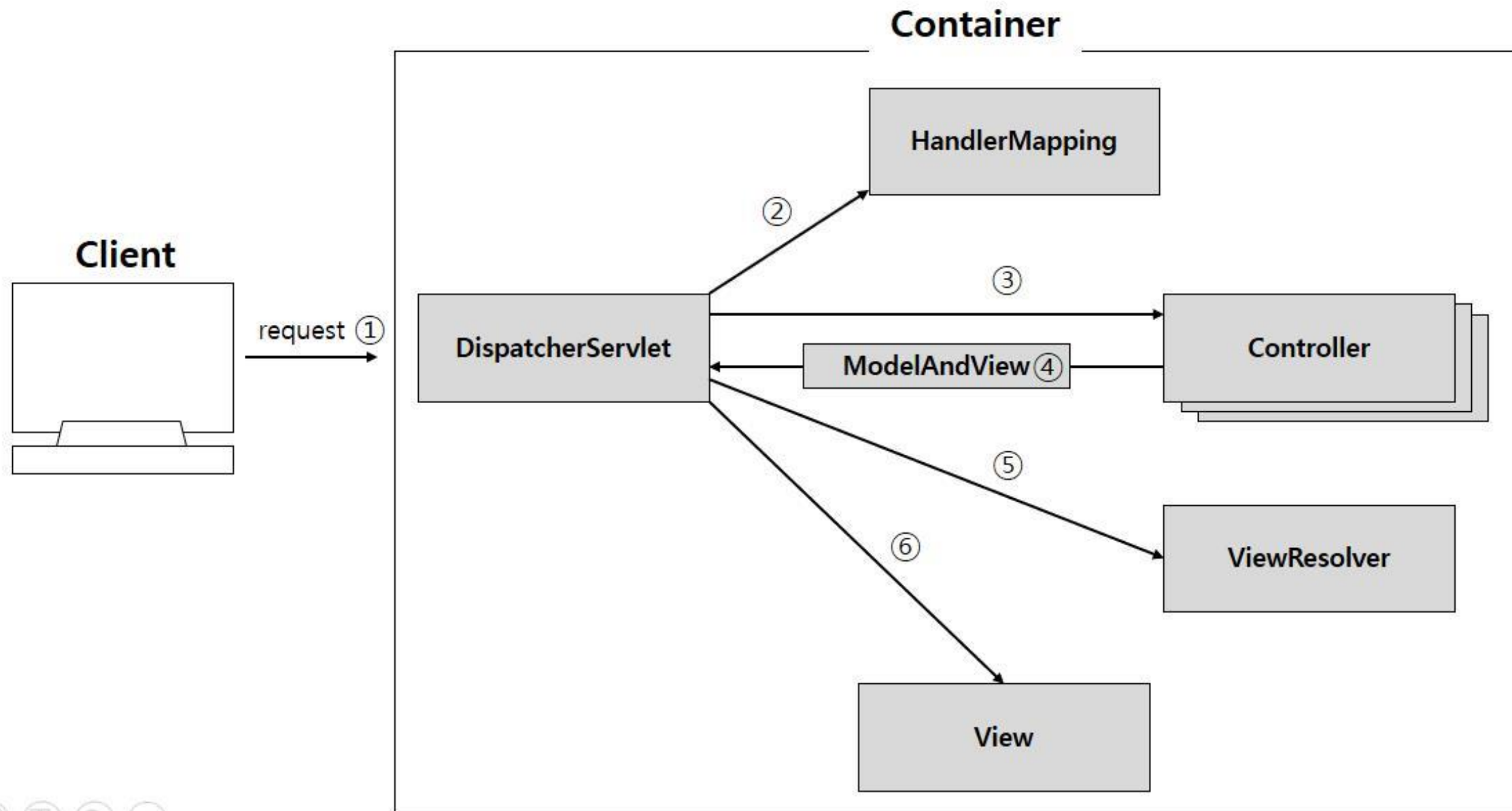
- Controller 클래스 구현

```
public class LoginController implements Controller {  
    @Override  
    public ModelAndView handleRequest(HttpServletRequest request,  
                                     HttpServletResponse response) {  
        // 1. 사용자 입력 정보 추출  
  
        // 2. DB 연동 처리  
  
        // 3. 화면 네비게이션  
        ModelAndView mav = new ModelAndView();  
        return mav;  
    }  
}
```

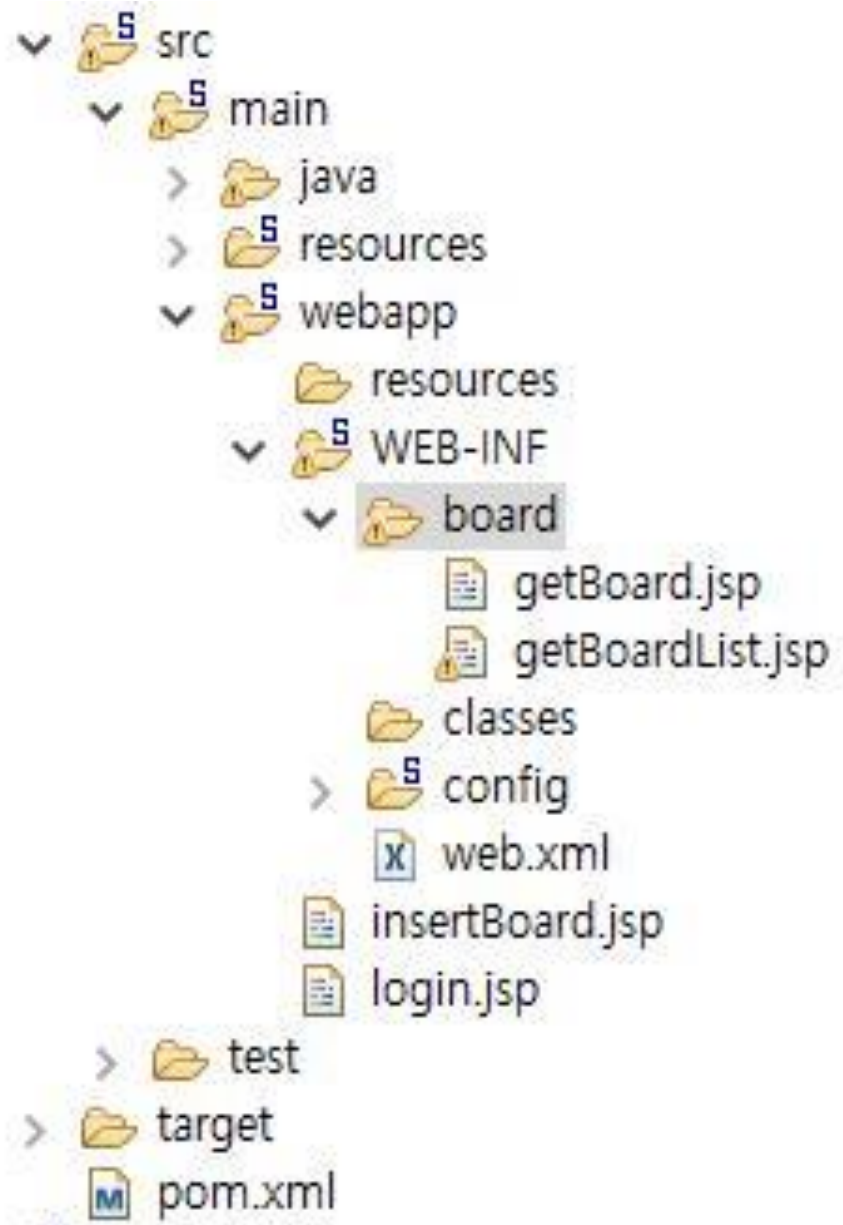

- 글 목록 검색 기능

```
public class GetBoardListController implements Controller {  
    @Override  
    public ModelAndView handleRequest(HttpServletRequest request,  
                                     HttpServletResponse response) {  
        // 1. DB 연동 처리  
        BoardVO vo = new BoardVO();  
        BoardDAO boardDAO = new BoardDAO();  
        List<BoardVO> boardList = boardDAO.getBoardList(vo);  
  
        // 2. 검색 결과와 화면 정보를 ModelAndView에 저장하여 리턴  
        ModelAndView mav = new ModelAndView();  
mav.addObject("boardList", boardList); // Model 정보 저장  
mav.setViewName("getBoardList.jsp"); // View 정보 저장  
return mav;  
    }  
}
```

- Spring MVC 흐름



- 직접적인 JSP 접근 차단



- ViewResolver 등록 (presentation-layer.xml)

```
<!-- ViewResolver 등록 -->
```

```
<bean id="viewResolver"
```

```
    class="org.springframework.web.servlet.view.InternalResourceViewResolver">
```

```
    <property name="prefix" value="/WEB-INF/board/"> </property>
```

```
    <property name="suffix" value=".jsp"> </property>
```

```
</bean>
```

- View 이름 지정

```
// 3. 화면 네비게이션
```

```
ModelAndView mav = new ModelAndView();  
if(user != null) {  
    mav.setViewName("redirect:getBoardList.do");  
} else {  
    mav.setViewName("login");  
}  
  
return mav;
```

- xxx.do로 이동할 때는 ViewResolver를 적용하기 않기 위해 "redirect:"을 View 이름 앞에 붙인다.
- xxx.jsp로 이동할 때는 확장자 ".jsp"를 제거한다.