

스프링 MVC

The logo consists of three overlapping squares: a yellow one at the top, a red one on the left, and a blue one at the bottom. A black crosshair is superimposed on the squares.

Spring MVC

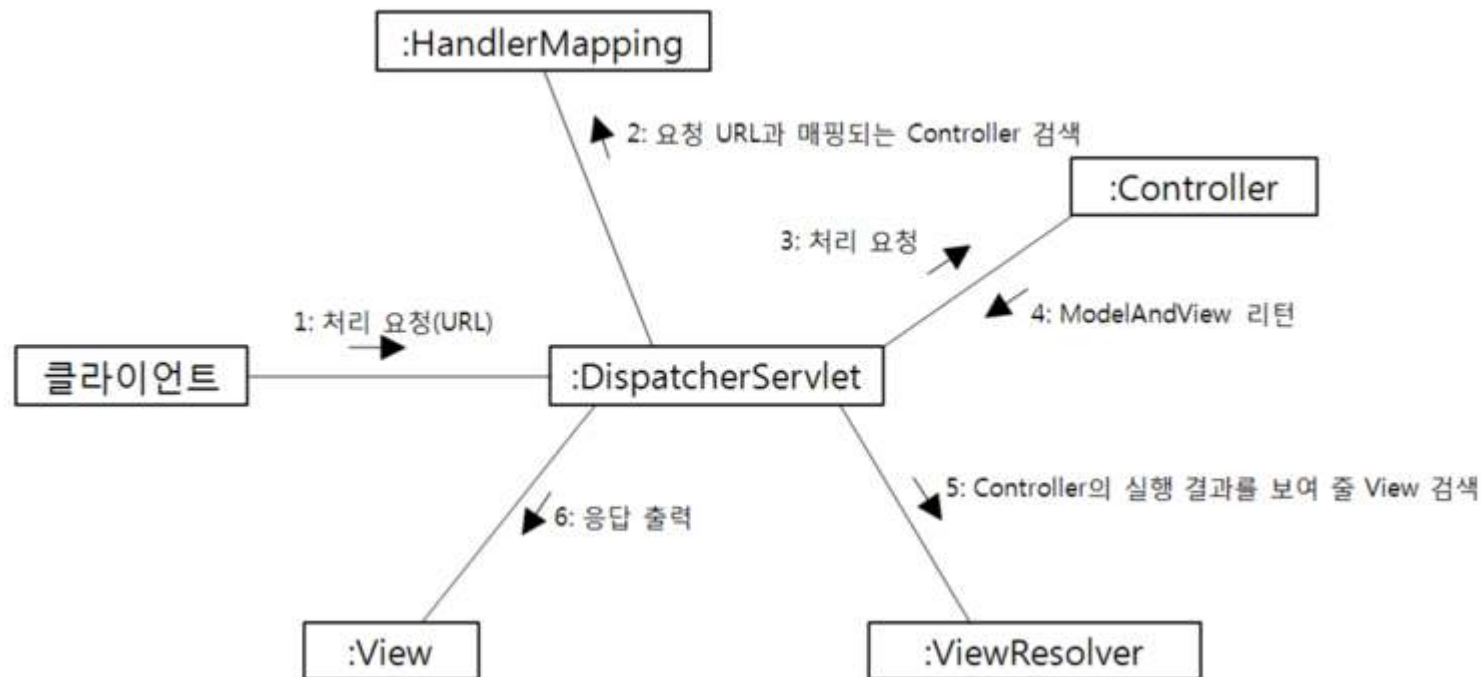
- 스프링이 제공하는 서블릿 기반의 MVC 프레임워크이다.
- 프론트 컨트롤러 역할을 하는 DispatcherServlet 을 사용한다.
- 스프링이 제공하는 AOP, 트랜잭션 처리, DI 등의 기능을 그대로 사용하면서 MVC 패턴에 기반하여 웹 어플리케이션을 개발할 수 있다.



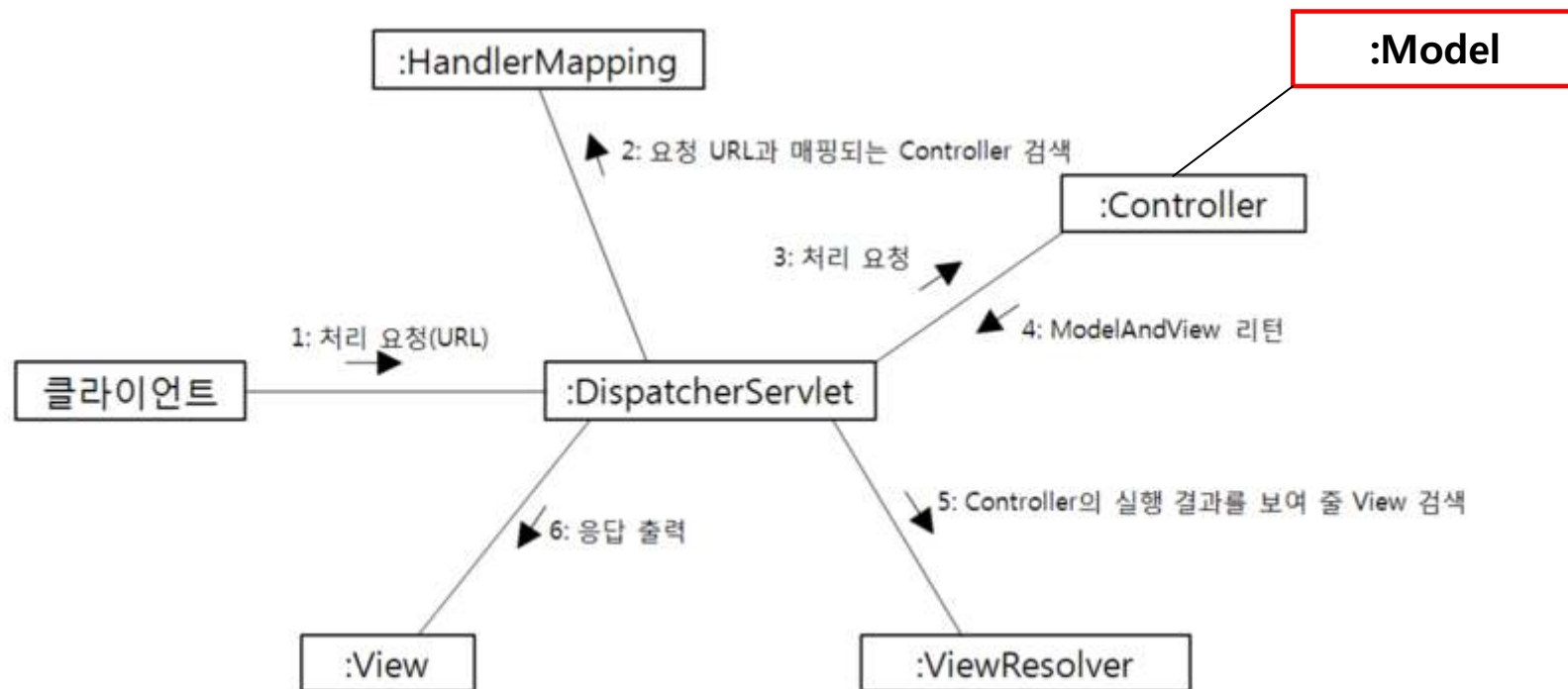
Spring MVC 구성요소

| 구성요소 | 설명 |
|--------------------------|---|
| DispatcherServlet | 모든 클라이언트의 요청을 전달받는다. Controller에게 클라이언트의 요청을 전달하고, Controller가 리턴한 결과값을 View에 전달하여 알맞은 응답을 생성하도록 한다. |
| HandlerMapping | 클라이언트의 요청 URL을 어떤 Controller가 처리할지를 결정한다. |
| Controller | 클라이언트의 요청을 처리한 뒤, 그 결과를 DispatcherServlet에 알려준다. |
| ViewResolver | Controller의 처리 결과를 보여줄 View를 결정한다. |
| View | Controller의 처리 결과를 보여줄 응답화면을 생성한다 |

Spring MVC 흐름도



Spring MVC 흐름도





Spring MVC 흐름도

- 클라이언트의 요청이 DispatcherServlet에 전달된다.
- DispatcherServlet은 HandlerMapping을 사용하여 클라이언트의 요청이 전달될 Controller 객체를 구한다.
- DispatcherServlet은 Controller 의 `handleRequest()` 메소드를 호출하여 클라이언트의 요청을 처리한다.
- `Controller.handleRequest()` 메소드는 처리 결과 정보를 담은 **ModelAndView** 객체를 리턴한다.
- DispatcherServlet은 ViewResolver로부터 처리 결과를 보여줄 View를 구한다.
- View는 클라이언트에 전송할 응답으로 화면을 생성한다.



Spring MVC 코드 작성순서

1. 클라이언트의 요청이 전달 될 DispatcherServlet을 web.xml에 설정한다.
2. HandlerMapping을 이용하여 요청 url과 맵핑되는 Controller를 설정한다.
3. 클라이언트의 요청을 처리할 Controller를 작성한다.
4. Controller의 결과화면을 담을 view를 결정하는 ViewResolver를 설정한다.
5. jsp 등을 이용하여 뷰 영역의 코드를 작성한다.