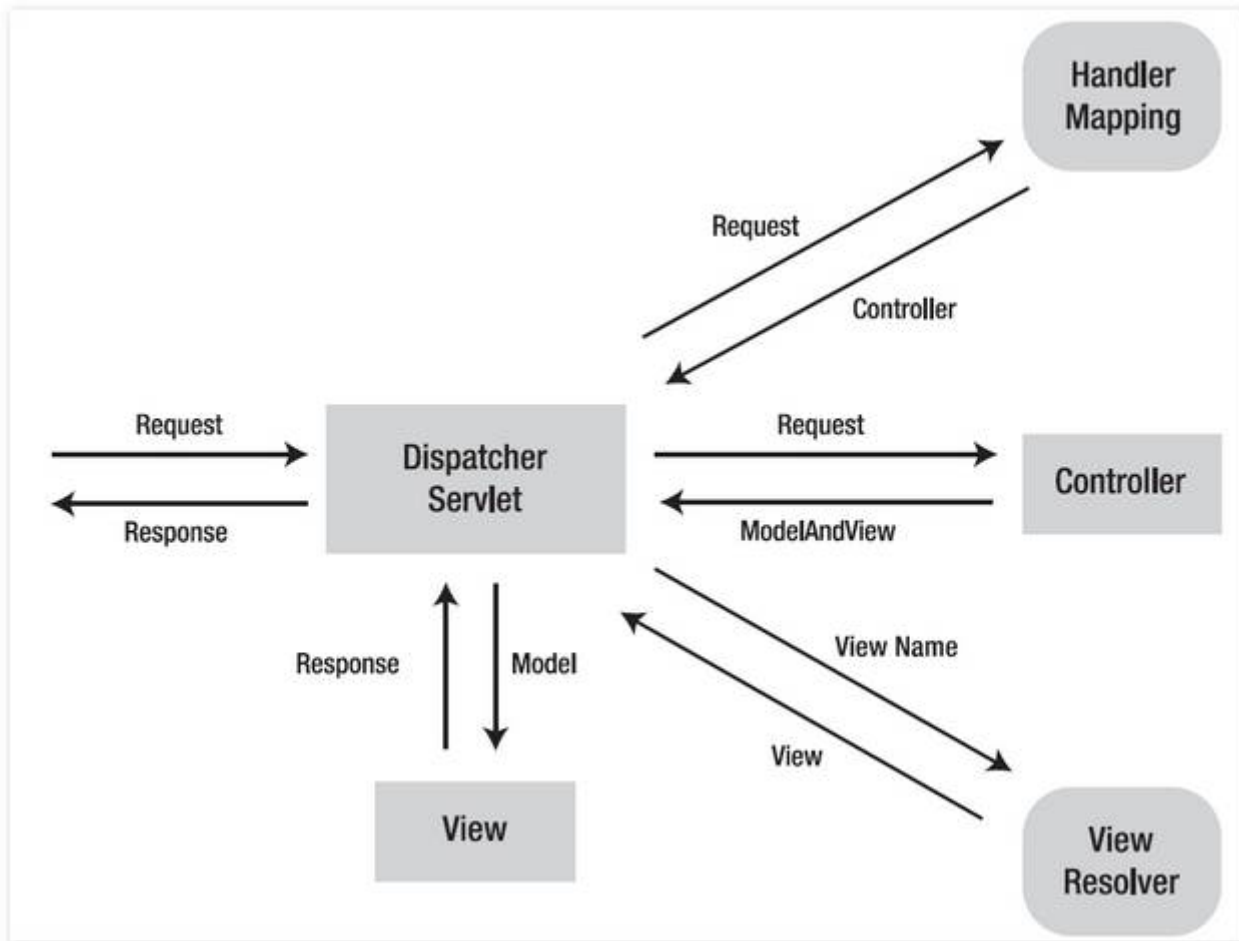


Spring MVC Framework

스프링 MVC 프레임워크가 동작하는 원리를 이해하고 있어야 한다



클라이언트가 서버에게 url을 통해 요청할 때 일어나는 스프링 프레임워크의 동작을 그림으로 표현한 것이다.

MVC 진행 과정

- 클라이언트가 url을 요청하면, 웹 브라우저에서 스프링으로 request가 보내진다.
- **Dispatcher Servlet**이 request를 받으면, **Handler Mapping**을 통해 해당 url을 담당하는 Controller를 탐색 후 찾아낸다.
- 찾아낸 **Controller**로 request를 보내주고, 보내주기 위해 필요한 Model을 구성한다.
- **Model**에서는 페이지 처리에 필요한 정보들을 Database에 접근하여 쿼리문을 통해 가져온다.
- 데이터를 통해 얻은 Model 정보를 Controller에게 response 해주면, Controller는 이를 받아 Model을 완성시켜 Dispatcher Servlet에게 전달해준다.
- Dispatcher Servlet은 **View Resolver**를 통해 request에 해당하는 view 파일을 탐색 후 받아낸다.

- 받아낸 View 페이지 파일에 Model을 보낸 후 클라이언트에게 보낼 페이지를 완성시켜 받아낸다.
- 완성된 View 파일을 클라이언트에 response하여 화면에 출력한다.

구성 요소

Dispatcher Servlet

모든 request를 처리하는 중심 컨트롤러라고 생각하면 된다. 서블릿 컨테이너에서 http 프로토콜을 통해 들어 오는 모든 request에 대해 제일 앞단에서 중앙집중식으로 처리해주는 핵심적인 역할을 한다.

기존에는 web.xml에 모두 등록해줘야 했지만, 디스패처 서블릿이 모든 request를 핸들링하면서 작업을 편리하게 할 수 있다.

Handler Mapping

클라이언트의 request url을 어떤 컨트롤러가 처리해야 할 지 찾아서 Dispatcher Servlet에게 전달해주는 역할을 담당한다.

컨트롤러 상에서 url을 매핑시키기 위해 `@RequestMapping`을 사용하는데, 핸들러가 이를 찾아주는 역할을 한다.

Controller

실질적인 요청을 처리하는 곳이다. Dispatcher Servlet이 프론트 컨트롤러라면, 이 곳은 백엔드 컨트롤러라고 볼 수 있다.

모델의 처리 결과를 담아 Dispatcher Servlet에게 반환해준다.

View Resolver

컨트롤러의 처리 결과를 만들 view를 결정해주는 역할을 담당한다. 다양한 종류가 있기 때문에 상황에 맞게 활용하면 된다.

[참고사항]

- [링크](#)
- [링크](#)