

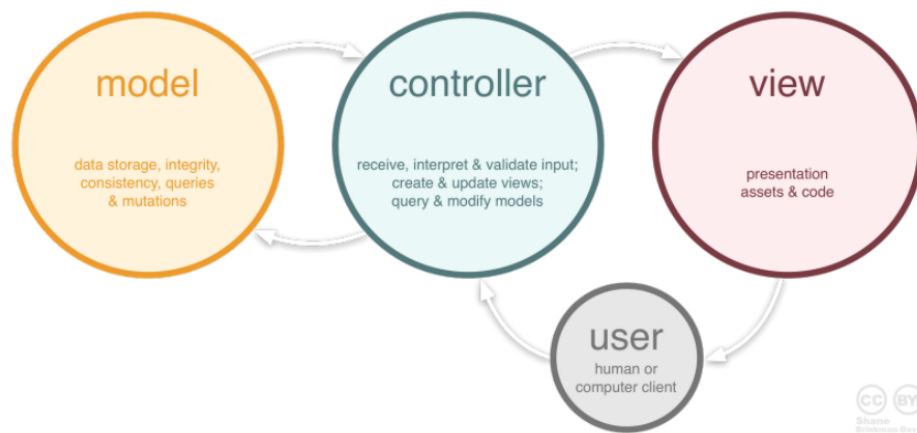


# [2주차] 스프링 핵심 개념

## 1. 스프링 MVC 구조

- MVC 패턴 개요
- MVC 패턴이란? 애플리케이션을 개발할 때 사용하는 디자인 패턴

### Model, View, Controller



#### 1) Model(모델)

: Spring MVC 기반의 웹 애플리케이션이 **클라이언트의 요청**을 전달 받아 **요청 사항을 처리**하기 위한 작업을 합니다. 처리한 작업의 결과 데이터를 클라이언트에게 응답으로 돌려주어야 하는데 이때 **클라이언트에게 응답으로 돌려주는 작업의 처리 결과 데이터를 Model** 이라고 합니다.

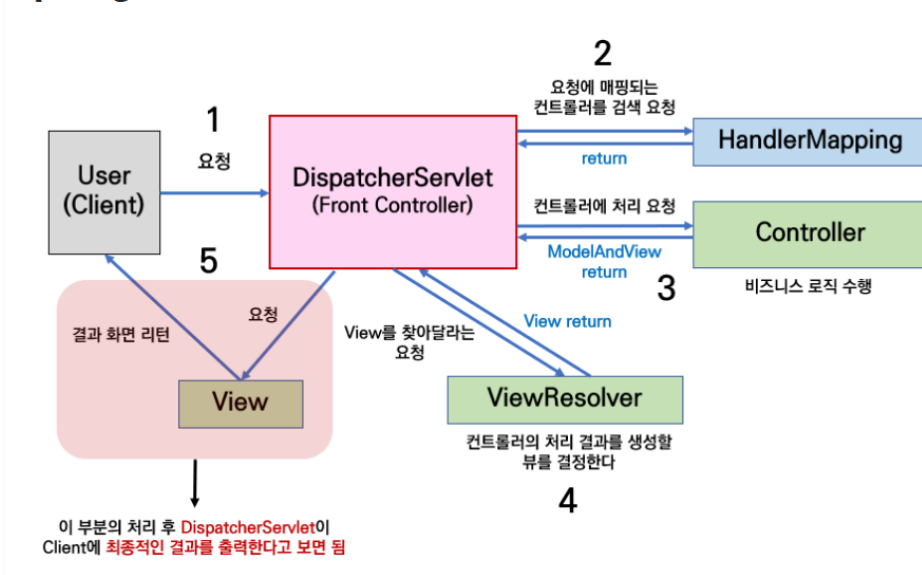
#### 2) View(뷰)

: Model을 이용하여 웹 브라우저와 같은 애플리케이션의 **화면에 보이는 Resource**를 제공하는 역할을 합니다. 예를 들어 Spring MVC에는 HTML 페이지 출력, XML/JSON 등 특정 형식의 포맷으로 변환하는 것입니다.

#### 3) Controller(컨트롤러)

: 클라이언트 측의 요청을 직접적으로 전달 받는 엔드 포인트로, **Model과 View의 중간에서 상호작용**을 해주는 역할을 합니다. 즉, 클라이언트 측의 요청을 전달 받아 비즈니스 로직을 거친 후 Model 데이터가 만들어지면 이 Model 데이터를 View로 전달하는 역할을 합니다.

## Spring MVC 동작 구조



### • DispatcherServlet의 역할과 동작 원리 + 요청 처리 흐름 상세 설명

1. 사용자가 서버에 요청을 하면 Front Controller인 DispatcherServlet 클래스가 요청을 받습니다.

**Spring은 프론트 컨트롤러 패턴을 따르고 이것을 DispatcherServlet이 담당합니다.**

### 💛 Front Controller 패턴이란?

Spring은 프론트 컨트롤러 패턴을 따르고 이것을 DispatcherServlet이 담당합니다.

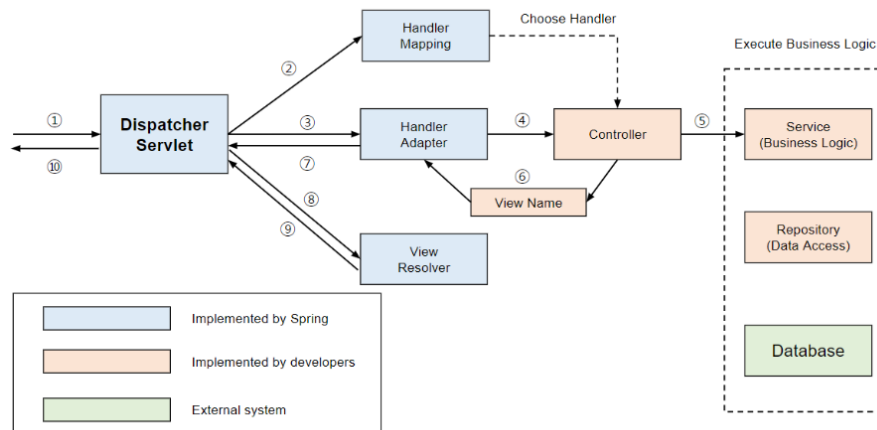
2. @RequestMapping을 통해 요청을 처리할 메소드로 이동합니다.
3. 컨트롤러는 해당 요청을 처리한 service를 받아 비즈니스 로직을 서비스에게 위임합니다. Service는 요청에 필요한 작업을 수행하고, 요청에 대한 DB에 접근해야 한다면 DAO에 요청하여 처리를 위임합니다. DAO는 DB 정보를 DTO를 통해 받아 서비스에게 전달합니다. 서비스는 전달받은 데이터를 컨트롤러에 전달합니다.
4. 컨트롤러는 Model 객체에게 요청에 맞는 View 정보를 담아 DispatcherServlet에게 전송합니다.

5. DispatcherServlet은 ViewResolver에게 전달받은 View 정보를 전달합니다. ViewResolver는 응답할 View에 대한 JSP를 찾아 DispatcherServlet에게 전달합니다.
6. 그 다음 DispatcherServlet은 응답할 뷰의 Render를 지시하고 뷰는 로직을 처리합니다. DispatcherServlet은 클라이언트에게 Rendering된 뷰를 응답하며 요청을 마칩니다.

- @Controller와 @RestController의 차이

- @Controller란?

: Spring MVC의 컨트롤러인 @Controller는 주로 View를 반환하기 위해 사용됩니다.



Controller는 요청을 처리한 후에 ViewName을 반환합니다. →

DispatcherServlet은 ViewResolver를 통해 ViewName에 해당하는 View를 찾아 사용자에게 반환합니다.

- @RestController란?

: 하지만 Spring MVC의 컨트롤러를 사용하면서 Data를 반환해야 하는 경우도 있습니다. 컨트롤러에서는 데이터를 반환하기 위해 @ResponseBody 어노테이션을 활용해주어야 합니다. 이를 통해 Controller도 Json 형태로 데이터를 반환할 수 있습니다. @RestController는 @Controller에 @ResponseBody가 추가된 것입니다. 따라서 @RestController를 사용하면 Json으로 사용자에게 반환할 수 있습니다.