



• 스프링의 웹요청 처리 흐름 •

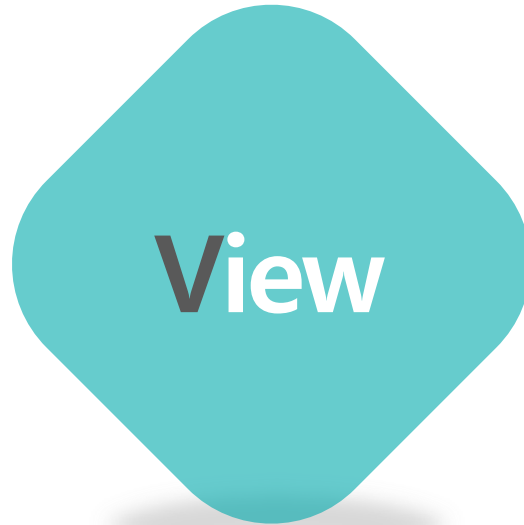
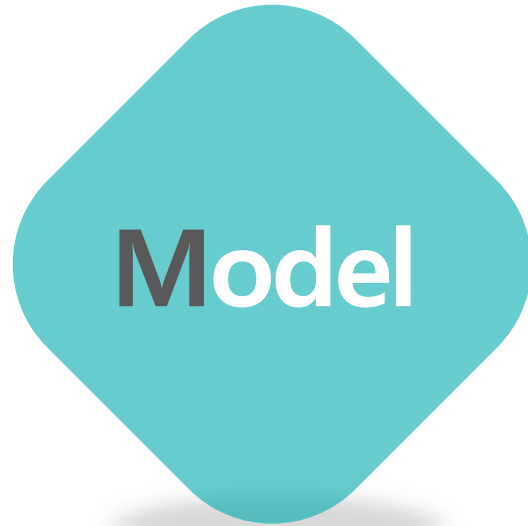
장윤정



1

MVC 패턴과 Front Controller 패턴

MVC 패턴



디자인패턴은 소프트웨어 개발방법을 공식화 한 것.

MVC는 Model. View. Controller의 약자로 어플리케이션을 세가지 역할로 구분한 개발방법론.

MVC 패턴의 목표는 사용자 인터페이스로부터 비즈니스 로직과 프레젠테이션 로직의 분리이며 분리를 통해 비즈니스 로직은 재사용이 가능하며 View는 쉽게 고칠 수 있는 장점이 있다.

Front Controller 패턴(1/2)

모든 요청을 처리하는
하나의 대표 컨트롤러를 두는 패턴

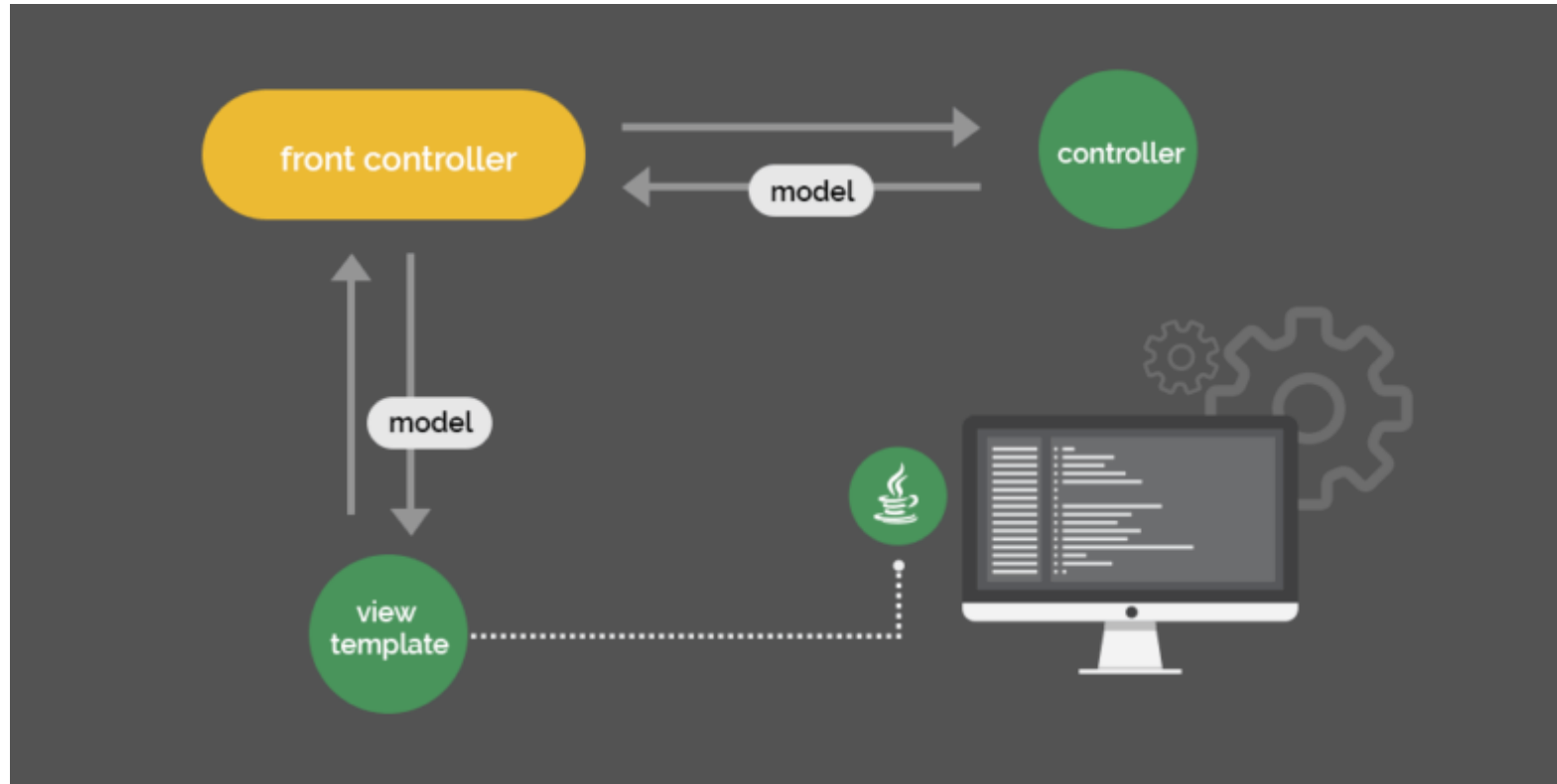
Front Controller

- ✓ 모든 요청의 진입점
- ✓ 모든 요청을 일괄적으로 처리 가능



DispatcherServlet

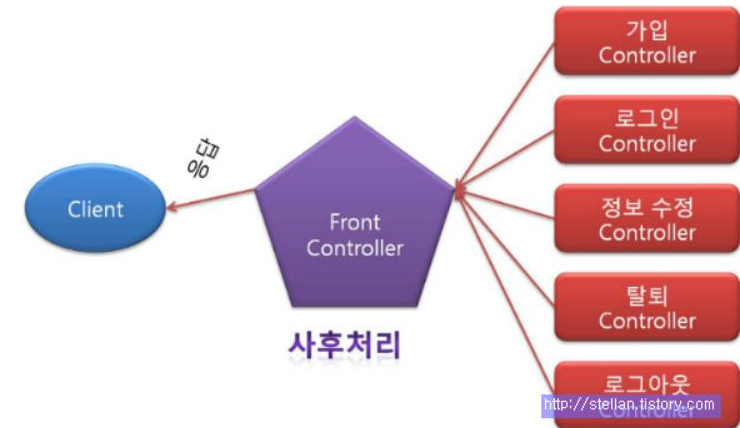
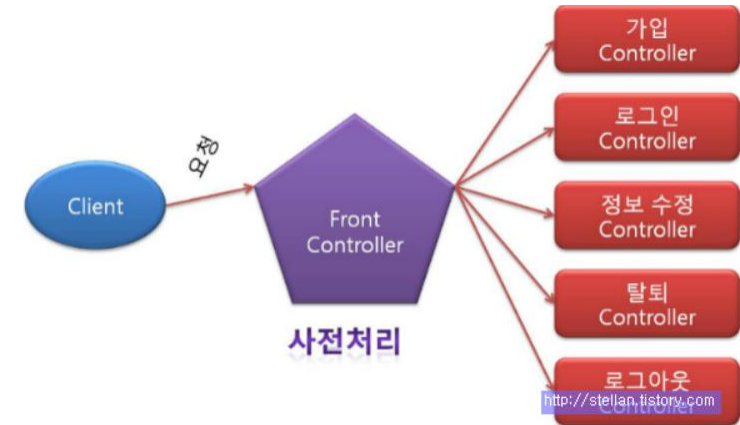
Front Controller 패턴(2/2)



요청의 진입점이 되는 하나의 대표 컨트롤러를 둔다

Front Controller

- 서블릿 컨테이너의 제일 앞에서 서버로 들어오는 클라이언트의 모든 요청을 받아 처리하는 Controller
- 모든 요청을 한 곳에 집중시키고, Controller의 공통 로직을 앞단에서 처리
 - ➔ Tracking(추적)이나 Security(보안) 적용 시 편하게 구현이 가능하고, URL 구성이 간편
- 이러한 특징 때문에 Front Controller는 MVC 패턴과 함께 많이 사용

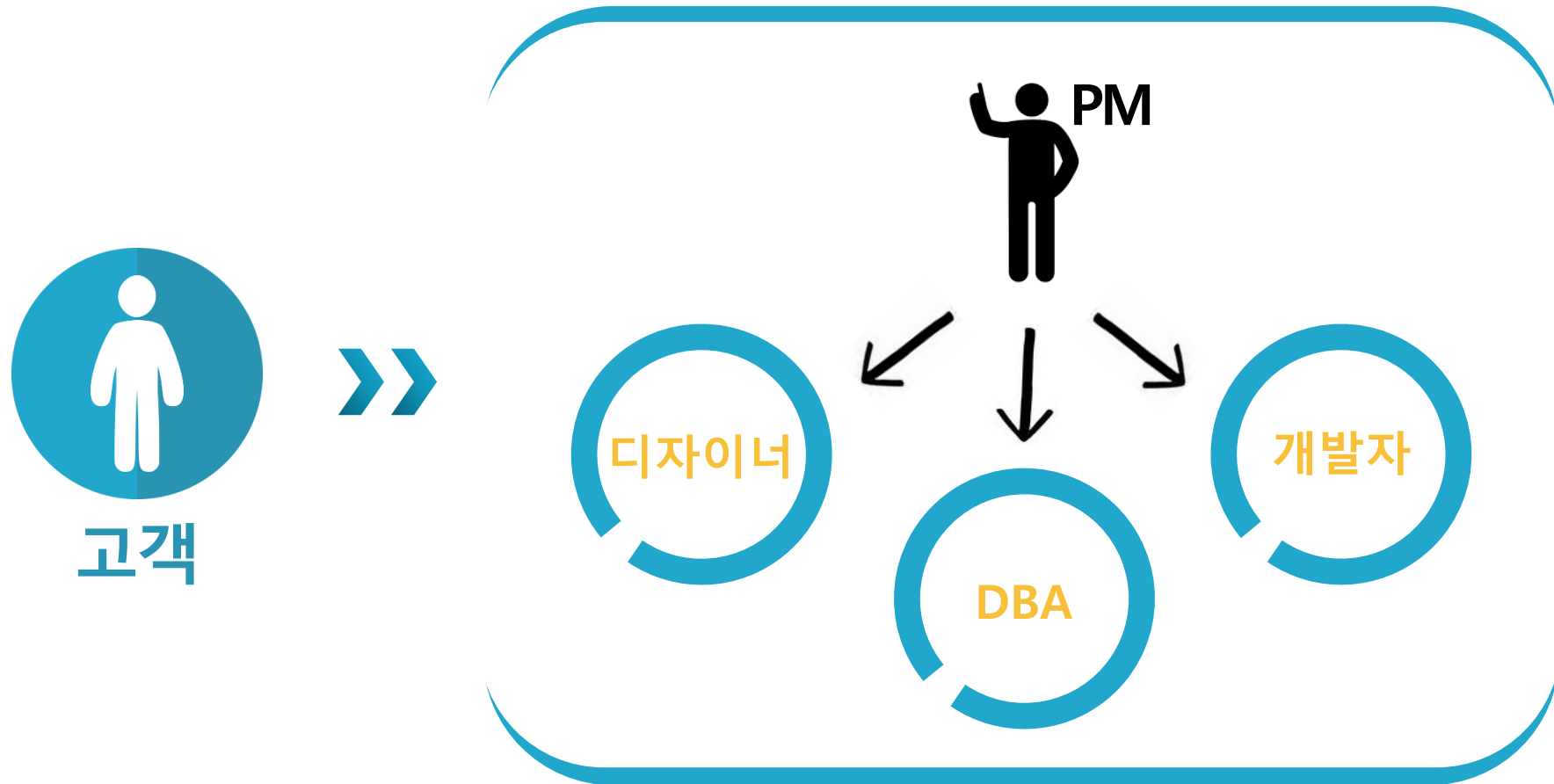


DispatcherServlet

서버로 들어오는 모든 요청을
제일 앞에서 처리하는 Front Controller를
Spring에서 정의하였고,
이를 DispatcherServlet 이라고 한다.

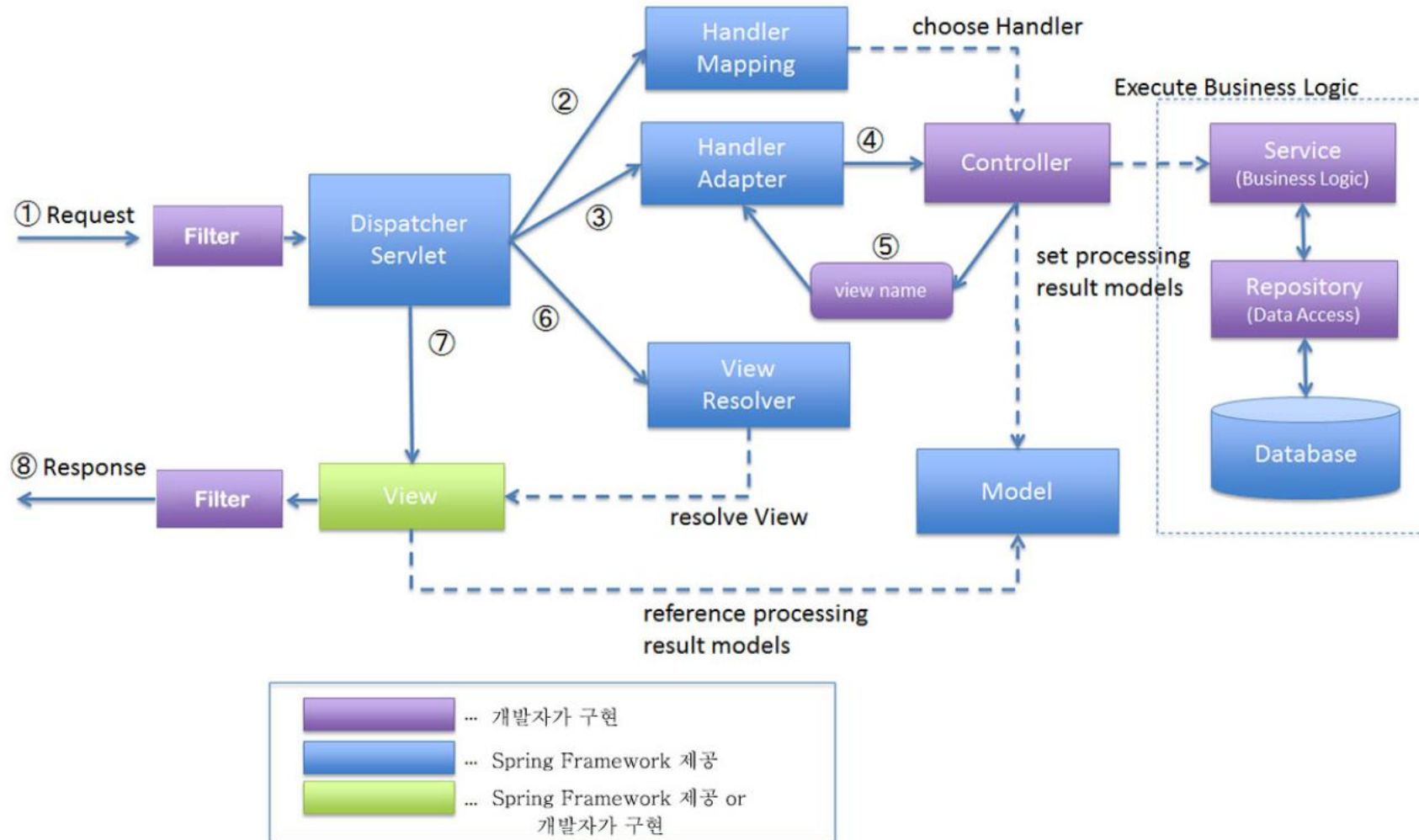
다음 중 **Front Controller**의 역할을 하는 사람은?

= DispatcherServlet



스프링 MVC 패턴

- 스프링 MVC 패턴은 FrontController 패턴과 함께 사용





흐름의 중심

= DispatcherServlet



2

스프링의 웹요청 처리 흐름

스프링 웹 요청 처리 흐름(1/2)

- DispatcherServlet이 Controller에 전달하는 과정

Dispatcher
Servlet

STEP1

DispatcherServlet은 web.xml에 정의된 URL 패턴에 맞는 요청을 받고 URL과 컨트롤러의 맵핑 작업을 HandlerMapping에 요청한다.

컨트롤러

Handler
Mapping

STEP2

HandlerMapping은 요청정보(URL)를 기준으로 어떤 컨트롤러를 사용할 지 결정하며, 결과는 HandlerExecution Chain 객체에 담아서 리턴하는데, 요청에 해당하는 Interceptor가 있을 경우에는 함께 담아서 준다.

인터셉터

Handler
Adapter

STEP3

HandlerAdapter는 컨트롤러의 메소드를 호출하는 역할을 하는데 실행될 Interceptor가 있을 때는 Interceptor의 preHandle() 메소드를 실행한 다음 컨트롤러의 메소드를 호출하여 요청을 처리한다.

Controller

STEP4

컨트롤러는 요청을 처리한 뒤 처리한 결과 및 뷰 정보(ModelAndView)를 DispatcherServlet에 전달한다.

스프링 웹 요청 처리 흐름(2/2)

- ViewResolver로 실제 뷰 객체 찾기

Dispatcher
Servlet

● STEP5

DispatcherServlet은 컨트롤러에서 전달받은 View 이름과 매칭되는 실제 View 파일을 찾기 위해 ViewResolver에게 요청한다.

View
Resolver

● STEP6

ViewResolver는 컨트롤러가 처리한 결과를 보여줄 뷰를 결정한다. 컨트롤러에서 전달받은 View 이름의 앞뒤로 prefix, suffix 프로퍼티를 추가한 값이 실제 사용할 뷰 파일의 경로가 된다. ViewResolver는 맵핑되는 View 객체를 DispatcherServlet에 전달한다.



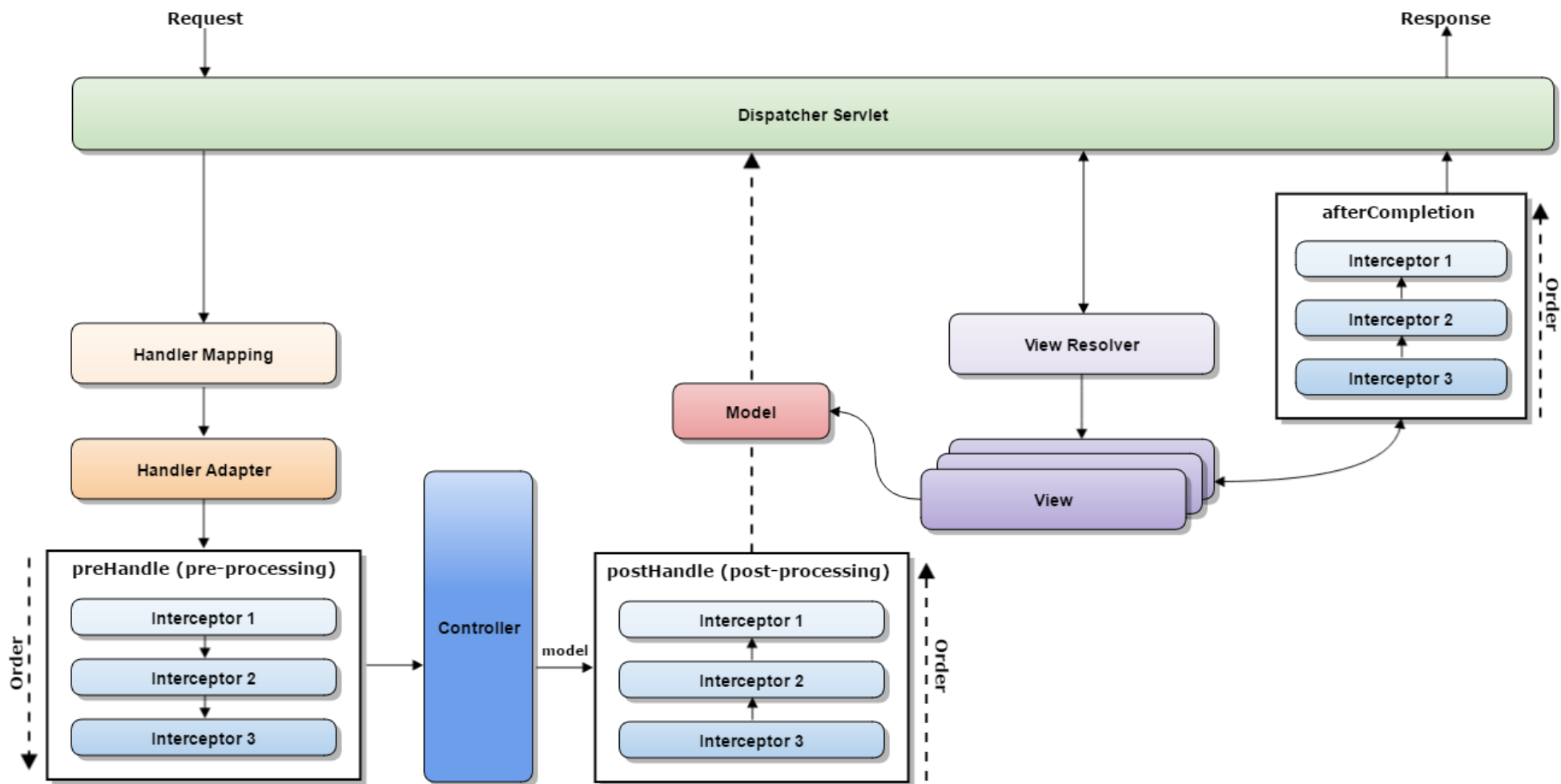
우리가 많이 쓰는 Tiles도 ViewResolver의 한 종류!

View

● STEP7

DispatcherServlet은 ViewResolver에 전달받은 View에 Model을 넘겨서 클라이언트에게 보여줄 화면을 생성한다.

다시보는 스프링 MVC 패턴



HandlerMapping



Handler
Mapping

- Handler는 컨트롤러를 의미
즉 HandlerMapping → ControllerMapping

HandlerMapping은 이름처럼 DispatcherServlet에 클라이언트로부터 요청이 들어오면 처리를 담당할 컨트롤러를 맵핑해주는 역할

servlet.xml에서 component-scan 시
@Controller에 있는 URL정보를 메모리에 올려두고
요청이 들어온 URL과 비교하여 처리할 컨트롤러를 찾아준다.

HandlerMapping의 종류

종류	설명
BeanNameUrlHandlerMapping	Bean 정의 태그에서 name에 선언된 URL과 class에 정의된 Controller를 맵핑
ControllerClassNameHandlerMapping	Bean 정의된 class 이름 중 suffix인 Controller를 제거한 나머지 이름의 소문자로 URL을 맵핑
SimpleUrlHandlerMapping	Ant-Style 패턴 매칭을 지원하며, 하나의 Controller에 여러 URL을 맵핑 가능 Property의 key에 URL 패턴을 지정하고 value에 Controller의 id나 이름을 지정
RequestMappingHandlerMapping	컴포넌트 스캔 할 패키지를 지정해주면 패키지의 @Controller 중에 @RequestMapping에 선언된 URL과 해당 @Controller 클래스의 메소드와 맵핑
SimpleUrlAnnotationHandlerMapping	DefaultAnnotationHandlerMapping에 인터셉터를 등록하면 모든 @Controller에 인터셉터가 적용되는 문제점 때문에 @Controller 사용시 URL 단위로 인터셉터를 적용하기 위해 개발

- HandlerMapping에 대한 정의가 없다면 기본 HandlerMapping(BeanNameUrlHandlerMapping)을 사용한다.
jdk1.5 이상의 실행환경일 때, Spring 3.1 이후 버전이면(egov 3.0부터) RequestMappingHandlerMapping가 기본 HandlerMapping이고
Spring 3.1이전 버전이면(egov 3.0이전 버전) DefaultAnnotationHandlerMapping가 기본 HandlerMapping이다.
DefaultAnnotationHandlerMapping은 3.1부터 사용되지 않고 RequestMappingHandlerMapping으로 대체된다.

HandlerAdapter



Handler
Adapter

- Adapter의 사전적 의미

- 다른 전기나 기계 장치를 서로 연결해서 작동할 수 있도록 만들어주는 결합 도구

즉 HandlerAdapter → ControllerAdapter

HandlerAdapter는 이름에서 유추할 수 있듯이 DispatcherServlet을 HandlerMapping으로 찾은 Controller와 연결시켜주는 역할을 하며 DispatcherServlet이 Controller를 호출할 때 사용

HandlerAdapter의 종류

컨트롤러	종류	설명
Servlet	SimpleServletHandlerAdapter	Servlet 클래스를 Spring MVC의 Controller로 사용할 수 있음
HttpRequestHandler	HttpRequestHandlerAdapter	Servlet처럼 동작하는 Controller를 만들기 위해 사용하며 Model과 View를 사용하지 않는 low level 서비스를 개발할 때 사용
Controller	SimpleControllerHandlerAdapter	디폴트 핸들러 어댑터이며 가장 대표적인 Spring MVC Controller
컨트롤러타입제한없음	AnnotationMethodHandlerAdapter	클래스와 메소드에 붙은 어노테이션 정보와 메소드 이름, 파라미터, 리턴 타입에 대한 규칙 등을 종합적으로 분석해서 컨트롤러를 호출

- 디폴트로 설정되어있는 HandlerAdapter는
HttpRequestMethodHandlerAdapter , SimpleControllerHandlerAdapter, AnnotationMethodHandlerAdapter 이다.
AnnotationMethodHandlerAdapter를 도입하면서
URL 매핑을 컨트롤러 단위가 아닌 메서드 단위로 처리하며, 한 컨트롤러가 하나 이상의 URL에 매핑 될 수 있다.

ViewResolver



View
Resolver

- Resolver의 뜻 → 해결자

ViewResolver는 Controller에서 받은
View 이름으로 실제 사용할 View객체(JSP)를 찾아주는 역할

View의 이름만으로 실제 View객체 찾는 일을 해결해 준다고 이해하자!

Tiles(타일즈)



Tiles

- Tiles(타일즈)는 우리가 자주 쓰는 ViewResolver의 한 종류이며 header와 footer같은 공통적으로 사용하는 부분의 소스 중복을 피하고, 유지보수를 용이하게 하기 위해 **Layout을 나누어 붙일 수 있는 기능**



Tiles가 적용된 소스를 확인해보자 !

ViewResolver의 종류

종류	설명
BeanNameViewResolver	논리적 뷰 이름과 동일한 ID를 갖는 view의 구현체를 찾는다.
ContentNegotiatingViewResolver	요청되는 콘텐츠 형식에 기반을 두어 선택한 하나 이상의 다른 ViewResolver에 위임한다.
FreeMarkerViewResolver	FreeMarker 기반의 템플릿을 찾는다.
InternalResourceViewResolver	웹어플리케이션의 WAR 파일 내에 포함된 뷰 템플릿을 찾는다.
JasperReportsViewResolver	JasperReports 리포트 파일로 정의된 뷰를 찾는다.
TilesViewResolver	Tiles 템플릿으로 정의된 뷰를 찾는다.
UrlbasedViewResolver	ViewResolver의 구현체로 특별한 �핑정보 없이 view 이름을 URL로 사용한다.
ResourceBundleViewResolver	ViewResolver의 구현체로 리소스 파일을 사용하며 views.properties를 기본 리소스 파일로 사용한다.
VelocityLayoutViewResolver	VelocityViewResolver의 서브클래스로, 스프링의 VelocityLayoutView를 통해 페이지 구성을 지원한다.
VelocityViewResolver	Velocity 기반의 뷰를 찾는다.
XmlViewResolver	ViewResolver의 구현체로 XML 파일을 사용하며 /WEB-INF/views.xml을 기본 설정파일로 사용한다.
XsltViewResolver	XSLT 기반의 뷰를 찾는다.

오늘 알아본 것 정리...

- MVC패턴과 FrontController패턴
- DispatcherServlet의 역할
- 스프링 웹요청 처리 흐름
- HandlerMapping
- HandlerAdapter
- ViewResolver(+Tiles)



다음시간에 알아볼 것

- web.xml
- ContextLoaderListener
- Filter / Interceptor / AOP
- @(annotation)



• • •
Thank you

오타 및 잘못된 내용 발견 시 알려주시면 감사하겠습니다 ☺