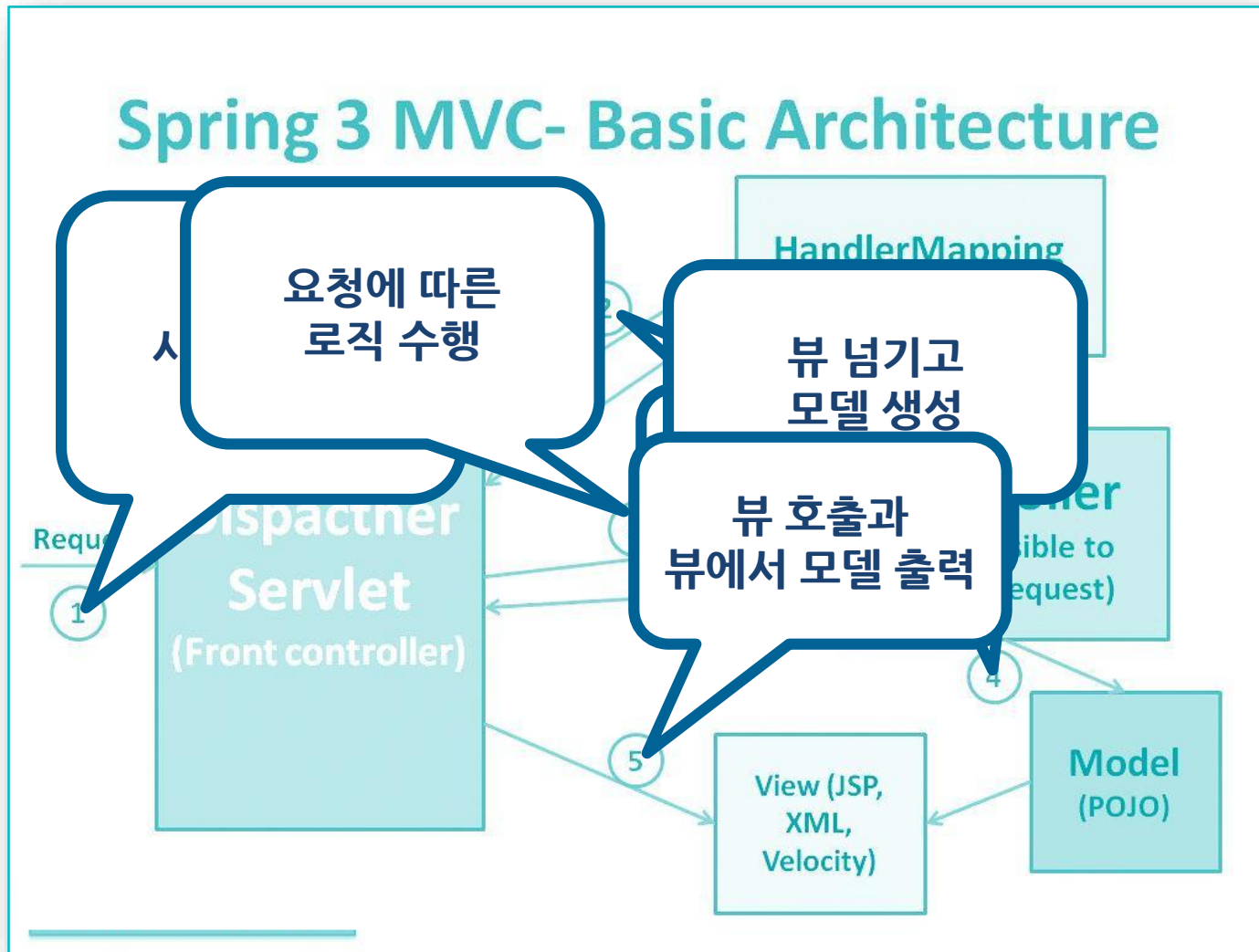


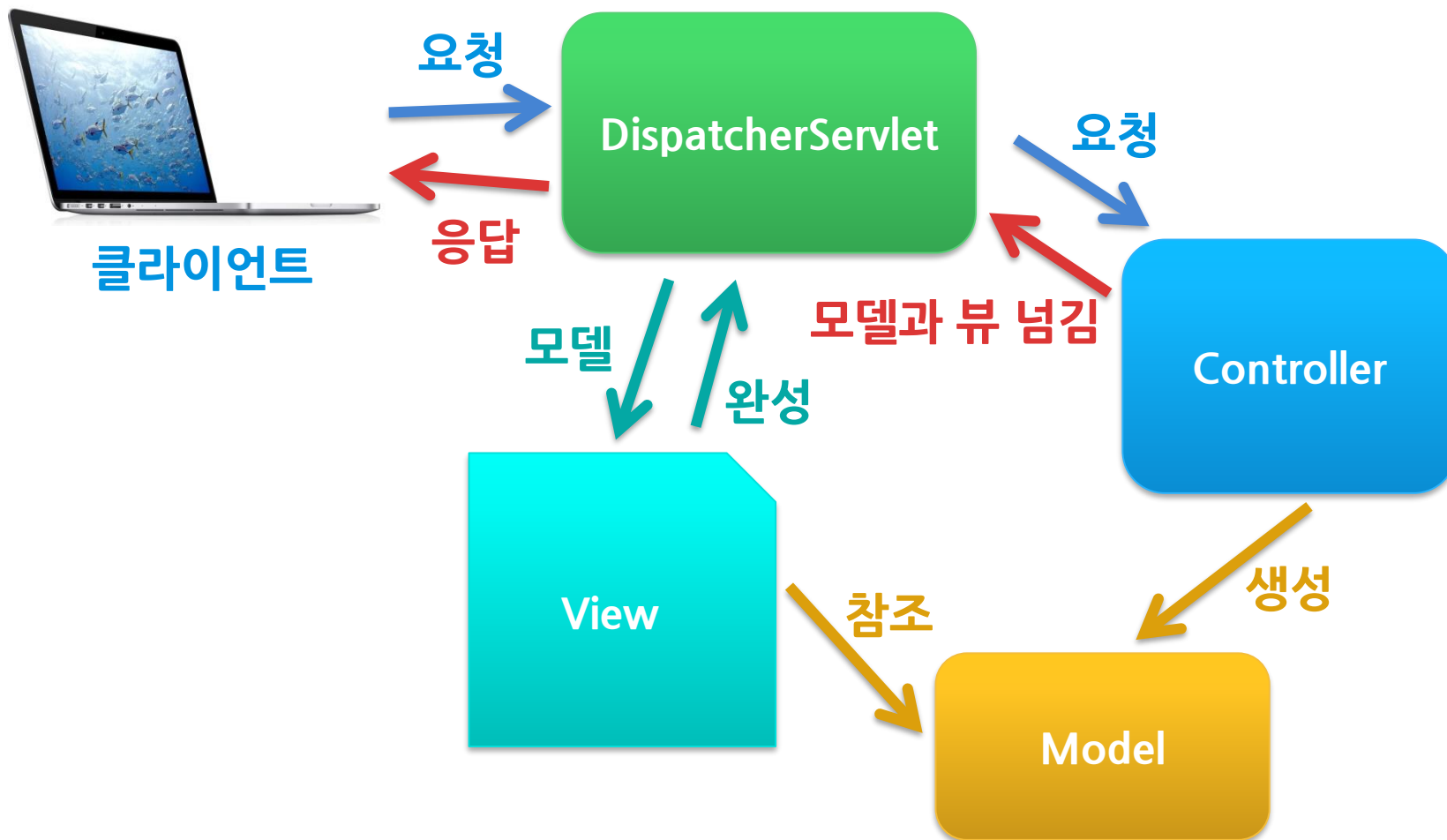
Spring MVC

programming

Spring MVC의 흐름



전체적인 흐름



DispatcherServlet

1. HTTP 요청 접수
2. 핸들러 맵핑에서 컨트롤러 선정
3. 컨트롤러에게 웹 요청을 위임
4. 컨트롤러에게 뷰와 모델을 리턴받음
5. 뷰 호출과 (뷰에서) 모델 참조

웹의 흐름



클라이언트

요청



서버

웹의 흐름

web.xml

```
<servlet>
  <servlet-name>board</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
  <servlet-name>board</servlet-name>
  <url-pattern>*.html</url-pattern>
</servlet-mapping>
```

웹의 흐름



웹의 흐름



HandlerMapping

- ✓ URL과 요청정보를 기준으로 컨트롤러 결정
- ✓ 하나이상의 핸들러 맵핑이 가능
- ✓ BeanNameUrlHandlerMapping
- ✓ RequestMappingHandlerMapping

HandlerMapping의 종류

- ✓ **BeanNameUrlHandlerMapping**
- ✓ **RequestMappingHandlerMapping**
- ✓ **SimpleUrlHandlerMapping**
- ✓ **ControllerClassNameHandlerMapping**
- ✓ **ControllerBeanNameHandlerMapping**

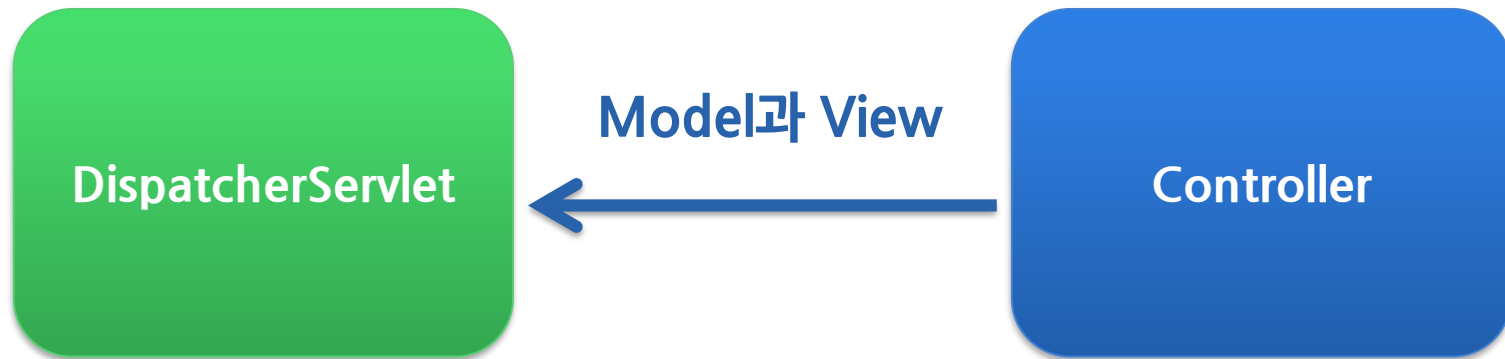
HandlerMapping의 종류

HandlerMapping	특징
BeanNameUrlHandlerMapping	bean설정시 name 속성으로 (기본)
RequestMappingHandlerMapping	컨트롤러에 어노테이션을 부여하여 맵핑 (기본)
SimpleUrlHandlerMapping	설정파일에 한번에 모아서 설정
ControllerClassNameHandlerMapping	클래스이름과 메서드명으로 맵핑
ControllerBeanNameHandlerMapping	빈의 이름으로 맵핑

HandlerAdapter

- ✓ 컨트롤러의 메서드를 호출할 때 사용
- ✓ `HttpRequestHandlerAdapter`
- ✓ `SimpleControllerHandlerAdapter`
- ✓ `RequestMappingHanlderAdapter`

웹의 흐름



View의 종류

- ✓ **InternalResourceView(forward)**
- ✓ **RedirectView(Redirect)**
- ✓ **MarshallingView(XML)**
- ✓ **MappingJacksonJsonView(Json)**

ViewResolver

- ✓ 컨트롤러가 리턴한 뷰 이름을 참고하여 뷰 찾기
- ✓ InternalResourceViewResolver

ViewResolver의 종류

- ✓ InternalResourceViewResolver
- ✓ ContentNegotiatingViewResolver
- ✓ ResourceBundleViewResolver
- ✓ XmlViewResolver

Method 파라미터의 종류

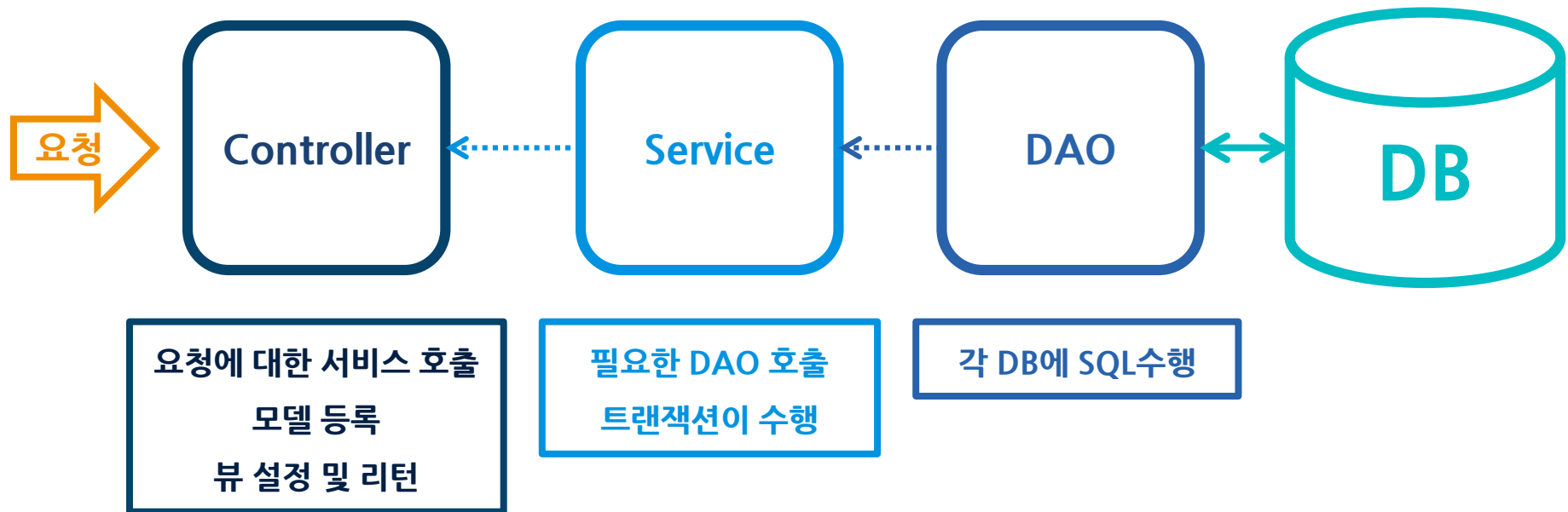
- ✓ `HttpServletRequest`, `HttpServletResponse`
- ✓ `HttpSession`
- ✓ `InputStream`, `Reader`, `OutputStream`, `Writer`
- ✓ `@PathVariable`, `@RequestParam`
- ✓ `@RequestHeader`, `Map`, `Model`, `BindingResult`
- ✓ `@ModelAttribute`
- ✓ `SessionStatus`

리턴타입의 종류

- ✓ @ModelAttribute, Model, Map 등은 자동으로
- ✓ ModelAndView
- ✓ String
- ✓ void (InternalResourceViewResolver)
- ✓ Model 오브젝트
- ✓ View
- ✓ @ResponseBody

Spring MVC 흐름

컨트롤러 / 서비스 / DAO



컨트롤러 / 뷰 설정시 고려사항

- ✓ 요청에 대한 view가 필요한가?
- ✓ 필요하면 forward
- ✓ 불필요하면 다른 요청으로 redirect
- ✓ 기본 뷰는 forward
- ✓ 'WEB-INF/view/(특정이름).jsp'

편리한 forward 뷰 지정

```
<mvc:view-resolvers>  
    <mvc:jsp prefix="/WEB-INF/view/" suffix=".jsp"/>  
</mvc:view-resolvers>
```

- ✓ WEB-INF/view 와 .jsp 생략가능
- ✓ redirect:list.html 등으로 redirect 지정가능

Controller Bean 등록

- ✓ xml에 설정
- ✓ 어노테이션 이용
- ✓ Controller는 xml에 설정하는 것이 좋음
- ✓ 이름-servlet.xml 에 설정

모델이 필요없는 요청

```
<mvc:annotation-driven/>
```

```
<mvc:view-controller path="/hello.html"/>
```

우리가 직접 Controller를 생성할 필요없음

필요에 따라 메서드 설정

- ✓ 어노테이션 `@RequestMapping` 지정
- ✓ GET / POST 방식 다르게 지정 가능
- ✓ `@RequestMapping(method=RequestMethod.POST)`
- ✓ `@RequestMapping(method=RequestMethod.GET)`

파라미터 편리하게 받기

- ✓ @ModelAttribute 어노테이션 이용
- ✓ 파라미터와 VO의 프로토타입 일치시 자동으로 세팅