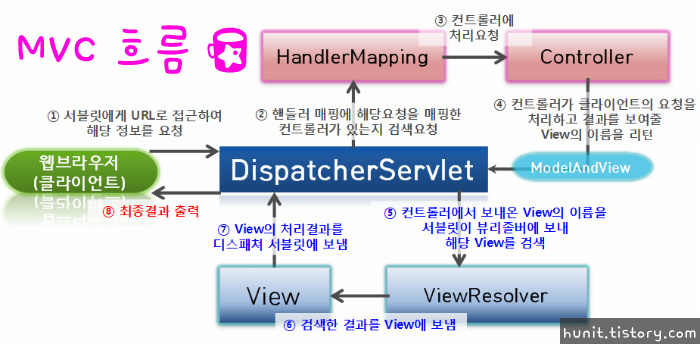
**DEV #10 – 2020.05.22**

>> Spring Web MVC 흐름을 조사하시오.

* Client 요청에서 부터 Client 응답까지의 흐름.
* 

>> Spring DI(Dependency Inject,의존성 주입) 방식을 조사하시오.

Fileld Injection

-Bean 으로 등록된 객체를 사용하고자 하는 클래스에 Field로 선언한 뒤 @Autowired 키워드를 붙여주면 됨

-동일한 Class의 Bean이 여러 개 존재하는 경우는 @Qualifier(“Field 명”)을 붙여주어

Bean 이름을 지정하여 주입받는 것이 가능

--의존성을 주입하기 쉬움으로 Constructor의 파라미터가 많아짐과 동시에 하나의 클래스가 많은 책임을 떠안기에 리팩토링 비용이 많이 발생할 수 있다

--클래스가 어떤 의존성을 책임지지 않을 떄, 메서드나 생성자를 통해 확실히 커뮤니케이션이 되어야하지만 Fileld Injection은 숨은 의존성만 제공해준다

--Field Injection 사용시 필요한 의존성을 가진 클래스를 곧바로 인스턴스화 시킬 수 없다.

--Field Injection은 final을 선언할 수 없어서 객체가 변할 수 있다.

Setter Injection

-setter 메소드에 @Autowired 를 붙여서 DI를 구현하는 방식

Constructor Injection

-현재 가장 권장되고 있는 방식으로 하나의 생성자가 존재시 Field Injection의 거이 모든 단점을 극복해낸 패턴

>> Spring Bean의 개념을 조사하시오.

-자주 사용하는 객체를 singleton으로 만드어 놓고 어디서든 불러쓸 수 있도록 한 것

>> Maven Project를 생성하고 Spring web-mvc를 적용하시오.

* Spring web mvc dependency 추가 (5.2.4.RELEASE)
* logback dependency 추가 (1.2.3)
* javax-servlet-api dependency 추가 (3.1.0)
* maven-compiler-plugin 설정 (jdk version 1.8)
* web.xml 생성 – spring 전용 서블릿 / 서블릿 매핑
* handler mapping, view resolver, bean등록
* componet-scan, annotation 사용 설정
* Controller Bean 등록 및 View 생성
* Annotation 방식 Controller지정

>> Spring Controller에 정의된 메소드의 매개변수로 올수 있는 type을 조사하시오.

1. (Http)ServletRequest,(Http)ServletReponse
2. WebRequest
3. MultipartRequest
4. HttpSession
5. TimeZon,ZoneId
6. 이외에 어노테이션을 사용하여 그에 맞는 적당한 타입으로 변환가능

PathVariable,RequestParam,ModelAttribute 등을 사용하여 변환가능

>> Spring Controller에 정의된 메소드의 리턴Type별로 적용되는 View를 조사하시오.

1. ModelAndView - localhost:8080/hello Request 시 hello.jsp 가 view가 됨
2. String – 리턴값이 view의 이름으로 사용됨
3. Void - **RequestToViewNameResolver**를 통해 자동생성되는 View 이름이 사용된다.
4. Object - **RequestToViewNameResolver**를 통해 자동생성되는 View 이름이 사용된다.  
   >> JSP 파일 include 방식을 조사하시오.