**DEV #13 – 2020.07.09**

>> 이클립스에 Lombok을 설치하고 Lombok 어노테이션을 조사하시오.

**@AllArgsConstructor / @NoArgsConstructor** – 생성자를 생선하는 어노테이션

Allargs~ 는 모든 멤버를 초기화하는 생성자 Noargs~는 기본 빈 생성자를 생성해줌

**@Getter / @Setter** – 자동으로 Getter 와 Setter 를 생성해주는 어노테이션

**@NonNull** – 런타임시 Null Check 를 하여 NullPointerException 발생(멤버에 직접사용)

**@ToString** – toString() 메서드 생성 상호 참조하는 객체의 경우 Stack Overflow발생할 수

있음 @ToString(excluded={“propertyName”})  를 사용하여 한쪽 객체를 다른쪽 객체에 대하여 제외해야함

**@EqualsAndHashCode** - equals(), hashCode() 메소드의 코드를 생성  
**@Data** – Class에 정의된 모든 필드에 대한 getter,setter,toString,equls,hashCode, final 이나 @NonNull 로 명시된 필드에 대한 값을 받는 생성자 메소드 코드를 생성해줌

**@Value** - @Data 와 유사하나 모든 필드를 기본적으로 Private 및 Final로 하고 ,Setter 함수를 생성하지 않고,Class 를 Final 로 지정함

**@Cleanup** – Local 변수에 붙일 시 Cleanup code 가 현재 code가 종료될 떄 자동으로 호출해줌

**@Synchronized** – static , instance 용 lock 오브젝트를 자동생성해주고 Annotation 을 메서드에 적용할 경우에는 메서드 Body를 synchronized 로 감싸준다

**@SneakyThrows** – checked exceptions를 처리해준다

**Val** – Local 변수선언시 val을 이용하면 변수타입을 유추하여 final로 생성해줌(js 의 var 과 같은 기능을 수행함)

**@Log** – 자동으로 logging 을 위한 필드인 private static final Logger log 를 추가해줌

**@Builder** – Class에 대한 복잡한 Build API들을 자동으로 생성해줌

>> spring-boot-start-\* 의 개념을 조사하고 숙지하시오.

* 스타터는 최소한의 설정으로 많은 라이브러리를 자동으로 관리하여 새로운 모듈을 쉽게 등록하거나 제거 할 수 있게 해주는 것
* 스프링이 제공하는 특정 모듈을 사용할 수 있도록 관련 라이브러리 의존성을 해결해줌

>> @SpingBootApplication 을 조사하시오. (내부구성)

@Configuration, @EnableAutoConfiguration, @ComponentScan

을 디폴트로 가지고있는 Annotation

>> Template 엔진의 종류를 조사하시오

* Mustache Template Engine 사용법을 익히시오.
* Spring Boot 에서 jsp 사용방법을 찾아보시오. ( 기본적으로 jsp 사용불가 )
  + Embed tomcat 의존성을 pom.xml 파일에 추가해주면 사용가능

<**dependency**>

<**groupId**>org.apache.tomcat.embed</**groupId**>

<**artifactId**>tomcat-embed-jasper</**artifactId**>

</**dependency**>

>> 스프링부트 환경프로퍼티 파일 (application.properties)

환경프로퍼티의 YAML형식으로의 변환후 프로퍼티 지정법을 익히시오.

* 프로퍼티 값을 불러오는 @Value, @ConfigurationProperties 사용법을 익히시오.

>> @RestController 에 대해 조사하시오.

-@Controller 은 View page 를 반환해주고

-@RestController 은 결과값을 반환해줌

- spring 3.x 에서는 @ResponseBody 를 통해 REST 방식을 처리했지만 spring 4.x붜 @Restcontroll 방식을 사용하여 REST 방식을 처리해줌

* ResponseEntity 에 대해 조사하시오.
  + @RestController는 별도의 뷰를 제공하지 않는 형태로 서비스를 실행하기 때문에,리턴 데이터가 예외적인 상황에서 문제발생할 수 있기에 개발자가 직접 결과 데이터와 HTTP 상탴코드를 제어할 수 있게 해주는 클래스이다