**springBoot란?**

스프링부트이니셜라이저: <https://start.spring.io/>

**graddle란?**

groovy언어 기반의 빌드 도구, maven과 ant의 장점을 모아놓았음

**maven과의 비교**

**Jar**: 자바 어플리케이션이 동작할 수 있도록 해주는 리소스 속성파일, 라이브러리 등이 포함된 압축파일/ JDK 와 JRE만 있어도 실행 가능

(JDK: 자바 개발 키트 JRE:

**War**: 서블릿, jsp 컨테이너에 배치하는 압축 포맷 파일, 웹 서버 또는 웹 컨테이너 필요

외부 서버를 이용하는 웹 프로젝트는 war파일로 패키징해야 하지만 스프링 부트는 내장 톰캣으로 돌려서 jar파일로 패키징 가능

**spring boot devtools란?**

- Property Defaults : Thymeleaf는 기본적으로 성능을 향상시키기 위해서 캐싱 기능을 사용한다. 하지만 개발하는 과정에서 캐싱 기능을 사용한다면 수정한 소스가 제대로 반영되지 않을 수 있기 때문에 cache의 기본값을 false로 설정할 수 있다.

- Automatic Restart: claapath에 있는 파일이 변경될 때마다 애플리케이션을 자동으로 재시작해준다. 개발자가 소스 수정 후 애플리케이션을 재실행하는 과정을 줄일 수 있으므로 생산성을 향상시킬 수 있다.

- Live Reload : 정적 자원(html, css, js) 수정 시 새로고침 없이 바로 적용할 수 있다.

쉽게 말하면 브라우저로 전송되는 내용들에 대한 코드가 변경되면, 자동으로 어플리케이션을 재시작하여 브라우저에도 업데이트를 해주는 역할을 한다.

관련 디펜던시:

developmentOnly 'org.springframework.boot:spring-boot-devtools'

**스프링과 DB연결**

- 관련 디펜던시 추가(오라클 사용)

implementation 'org.mybatis.spring.boot:mybatis-spring-boot-starter:2.1.4'

runtimeOnly 'com.oracle.database.jdbc:ojdbc8'

- 커넥트 풀 설정

spring.datasource.driver-class-name=oracle.jdbc.driver.OracleDriver

spring.datasource.url=jdbc:oracle:thin:@localhost:1521:XE

spring.datasource.username='계정명'

spring.datasource.password='계정 비밀번호'

참고: <https://azurealstn.tistory.com/83>

**참고:** [**https://bongra.tistory.com/185**](https://bongra.tistory.com/185)

**스프링 부트에서 오라클 사용하는 dependecy 와 커넥션 풀(application.properties)**

#build.gradle

runtimeOnly 'com.oracle.database.jdbc:ojdbc8'

마리아 DB 연동

마리아 DB 설치

설치사이트: <https://mariadb.org/download/?t=mariadb&p=mariadb&r=10.6.7>

참고: <https://jhnyang.tistory.com/257>

마리아 DB GUI인 Heidi SQL 설치 및 사용

참고: <https://www.codingfactory.net/12663>

마리아 DB 권한 부여

grant all privileges on 데이터베이스명.\* to '계정명'@'%' identified by '비번';

참고: <https://bskyvision.com/970>

마리아 DB 스프링 부트와 연동

implementation 'org.mybatis.spring.boot:mybatis-spring-boot-starter:2.2.2'

runtimeOnly 'org.mariadb.jdbc:mariadb-java-client'

#mariadb \uC124\uC815

spring.datasource.driver-class-name=org.mariadb.jdbc.Driver

spring.datasource.url=jdbc:mariadb://localhost:3306/test

spring.datasource.username=root

spring.datasource.password=root

# mapper.xml mapper location setting

mybatis.mapper-locations:classpath:mappers/\*\*/\*.xml

# model camelcase setting

mybatis.configuration.map-underscore-to-camel-case=true

#alias setting

mybatis.type-aliases-package=vo패키지

# mapper logging level setting

logging.level.com.prototype.domain.repository=TRACE

#server setting

server.port=8900

참고: <https://www.devkcj.com/gatsby-springMVC-2/>

**스프링 부트 필요 어노테이션**

@Entity: 데이터 베이스 연동을 위한 모델 클래스라는 것을 알려주는 어노테이션

\*PK를 나타내기 위해 @Id 어노테이션을 사용하며, 생성 방법을 정의하기 위해 @GeneratedValue 를 사용한다

@Id: 기본키라는 것을 알려주는 어노테이션

@GeneratedValue(strategy = identity, auto. sequence(DB에 따로 시퀀스 생성해야). table): 증가 값을 자동적으로 적용

@Mapping : 매퍼 인터페이스임을 명시(Repository, 매퍼는 직접 DB에 접촉) 매퍼.xml파일 namespace에 매퍼 파일의 경로명

@Column

@CreatedDate

@LastModifiedDate

@EnableJpaAuditing : Application 메인 클래스에 넣어 audit 활성화 해주는 어노테이션

@ENtityListeners(AuiditingEntityListener.class): vo에 임포트

참고: <https://webcoding-start.tistory.com/53> baseTImeEntity클래스 이용 audit(게시판 여러 개일때)

참고: <https://compunication.tistory.com/27> vo 컬럼에 직접 넣음(게시판 하나일 때 사용 가능)

@DynamicInsert : jpa로 인서트 시 DB에서 디폴트 값이 null로 들어가는데, 이 어노테이션으로 DB에서 디폴트 값을 사용할 수 있다.

Cf. Auditing 기능이란? Spring data jpa에서 시간에 대해 자동으로 값을 넣어주는 기능

@Data 는 안 쓴다. Setter 를 해주지 말아야할 필드도 있기 때문에

@Builder

**타임리프란?**

Tymelleaf는 웹 환경과 독립 실행형 환경 모두를 위한 현대적인 서버 측 자바 템플릿 엔진입니다.

Tymelleaf의 주요 목표는 브라우저에 올바르게 표시될 수 있고 정적 프로토타입으로 작동하여 개발 팀에서 더 강력한 협업을 가능하게 하는 HTML인 우아한 자연 템플릿을 개발 워크플로우에 가져오는 것입니다.

Spring Framework용 모듈, 좋아하는 도구와의 많은 통합, 그리고 자신의 기능을 연결할 수 있는 기능을 갖춘 Tymelleaf는 더 많은 것을 할 수 있지만 현대의 HTML5 JVM 웹 개발에 이상적입니다.

적용 시 필용한 태그: <html xmlns:th="http://www.thymeleaf.org">

<https://yeonyeon.tistory.com/153>

타임리프 th:onclick 이용법

th:onclick=*"'location.href = \'' + @{/board/list} + '\''"*

th:onclick=*"'location.href = \'' + @{/board/edit/{boardNo}(boardNo=${boardContent.boardNo})} + '\''"*

타임리프 반복문

<th:block th:each=*"i: ${#numbers.sequence(시작 값, 끝 값)}"*>

<button th:if=*"${response.page == i}"* class=*"active"*>Dummy</button>

<button th:unless=*"${response.page == i}"*>Dummy</button>

</th:block>

**스프링 시큐리티란?**

**참고:** [**https://wildeveloperetrain.tistory.com/50**](https://wildeveloperetrain.tistory.com/50)

**시큐리티 적용법**

**참고:**[**https://spring.io/guides/gs/securing-web/**](https://spring.io/guides/gs/securing-web/)

시큐리티 관련 디펜던시 추가

implementation 'org.springframework.boot:spring-boot-starter-security'

기능 구현

참고: <https://spring.io/guides>

Entity(setter 지양) vs DTO vs VO

참고: <https://gmlwjd9405.github.io/2018/12/25/difference-dao-dto-entity.html>