Spring Data JPA

# 프로젝트 : B21cSpringDataJPA2

## 준비사항

* 의존설정 : JDBC API, Lombok, Oracle Driver, Spring Web, Spring Data JPA
* JSP 사용을 위한 설정을 한다.
* Refresh Gradle Project 를 눌러 적용한다.

## JpaRepository

* Spring Data JPA에서 제공하는 인터페이스
* 기본적인 CRUD (Create, Read, Update, Delete) 기능을 자동으로 제공
* 메서드 명명 규칙에 따라 다양한 쿼리를 확장할 수 있음
* 형식 : JpaRepository<엔티티, PK타입>

### 

### 1. **기본 키워드**

* findBy : 조회 작업을 수행. 일반적으로 여러 결과를 반환하는 메서드를 정의할 때 사용.
* readBy : 조회 작업을 수행하며, 읽기 작업이라는 것을 강조하고 싶을 때 사용.
* queryBy : 쿼리 작업을 강조하여 데이터를 조회할 때 사용. findBy와 동일하게 동작.
* getBy : 단일 결과를 기대하는 조회 쿼리. 그러나 여러 결과를 반환하는 List로도 사용할 수 있음.
* countBy : 특정 조건에 맞는 엔티티의 개수를 반환.
* deleteBy : 조건에 맞는 데이터를 삭제.

### 2. 쿼리 확장을 위한 메서드 명명 규칙

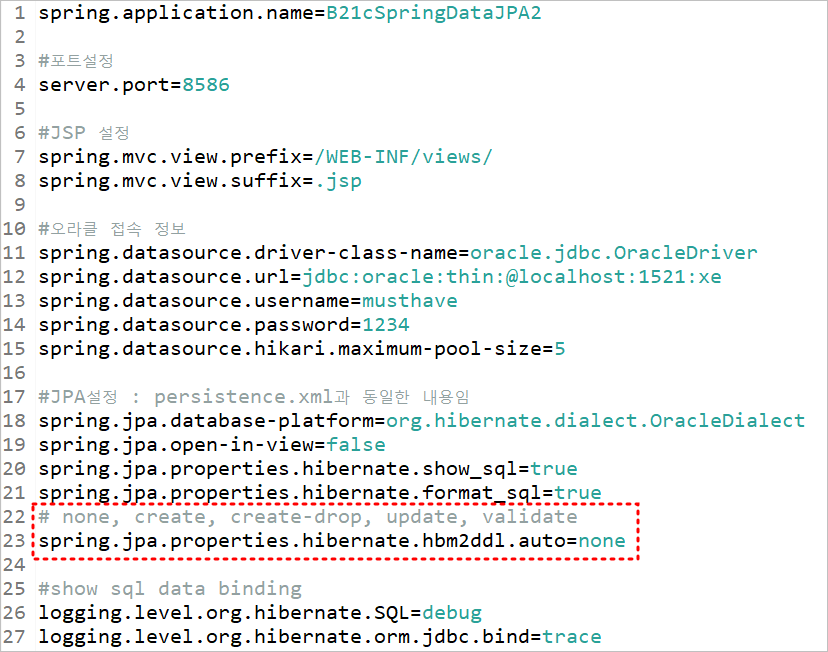
* Spring Data JPA는 메서드 이름을 분석하여 자동으로 쿼리를 생성
* 기본 키워드를 시작으로 뒤에 조건을 명시하여 확장
  + findBy : 조회 쿼리를 생성하는 기본 키워드
  + By : 조건을 지정할 때 사용되는 구분자
  + 컬럼명 : 엔티티의 필드명을 사용하여 조건을 설정.
  + 연산자 : And, Or, Like, LessThan, GreaterThan, IsNull, In, Between 등 추가 가능

### 3. 메서드 예시

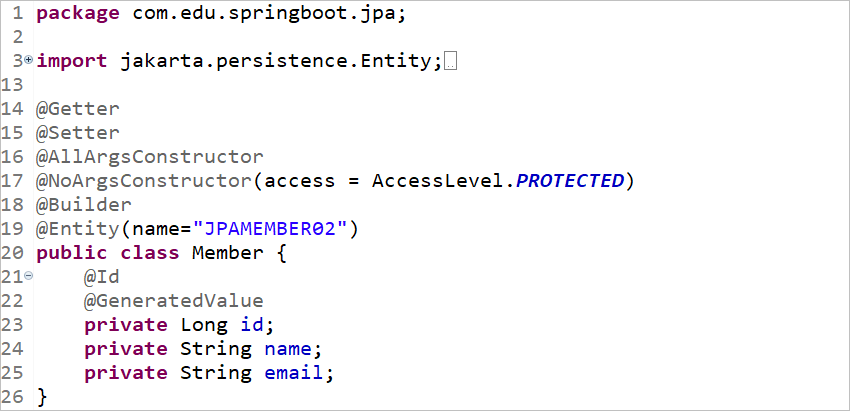
컬럼명은 name, email, userid 라고 가정했을때 작성할 수 있는 메서드 명

* name 컬럼을 기준으로 하는 메서드
  + findByName(String name) : name 으로 조회
  + findByNameLike(String name) : name 으로 LIKE 조회
  + findByNameStartingWith(String prefix) : name이 특정 문자열로 시작하는 경우 조회
  + findByNameEndingWith(String suffix) : name이 특정 문자열로 끝나는 경우 조회
  + findByNameContaining(String part) : name에 특정 문자열이 포함된 경우 조회
  + findByNameIgnoreCase(String name) : 대소문자를 구분하지 않고 name으로 조회
* email 컬럼을 기준으로 하는 메서드
  + findByEmailIsNull(): email 값이 NULL인 레코드 조회
  + findByEmailIsNotNull(): email 값이 NULL이 아닌 레코드 조회
* 두개 이상의 컬럼을 사용하는 메서드
  + findByNameAndEmail(String name, String email) : name과 email 값이 모두 일치하는 경우 조회
  + findByNameOrEmail(String name, String email) : name 또는 email 중 하나가 일치하는 경우 조회
  + findByUseridAndName(String userid, String name): userid와 name이 모두 일치하는 경우 조회.
  + findByUseridOrEmail(String userid, String email): userid 또는 email이 일치하는 경우 조회.
* 그 외 연산자 및 정렬
  + findByUseridIn(List<String> userids) : 주어진 리스트 중 하나라도 일치하는 경우 조회
  + findByEmailIsNull() 혹은 findByEmailIsNotNull() : email이 null 혹은 null 이 아닌 경우 조회
  + findByNameOrderByUseridAsc(String name): name이 일치하는 결과를 userid 기준으로 오름차순 정렬
  + findByNameOrderByUseridDesc(String name): name이 일치하는 결과를 userid 기준으로 내림차순 정렬
* 페이징 적용
  + findByName(String name, Pageable pageable): name이 일치하는 결과를 페이지 단위로 반환

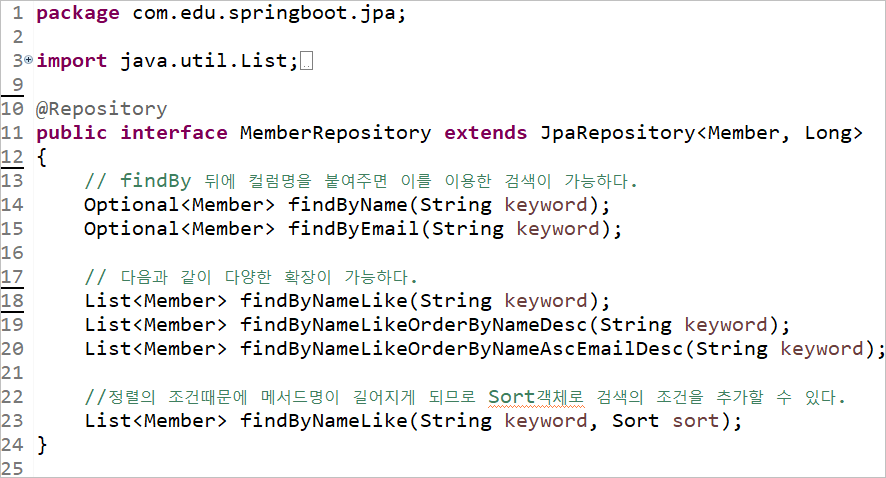
### 프로퍼티 : application.properties



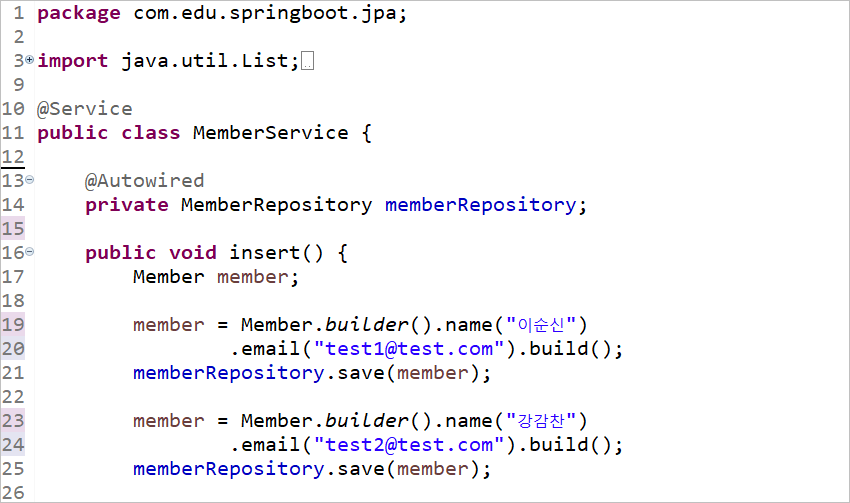
### 엔티티 : Member.java



### 레포지토리 : edu/spring/boot/jpa/MemberRepository.java

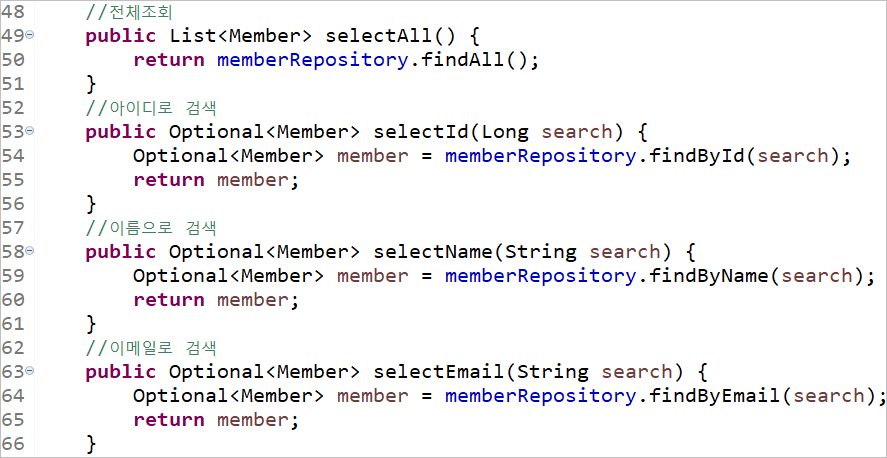


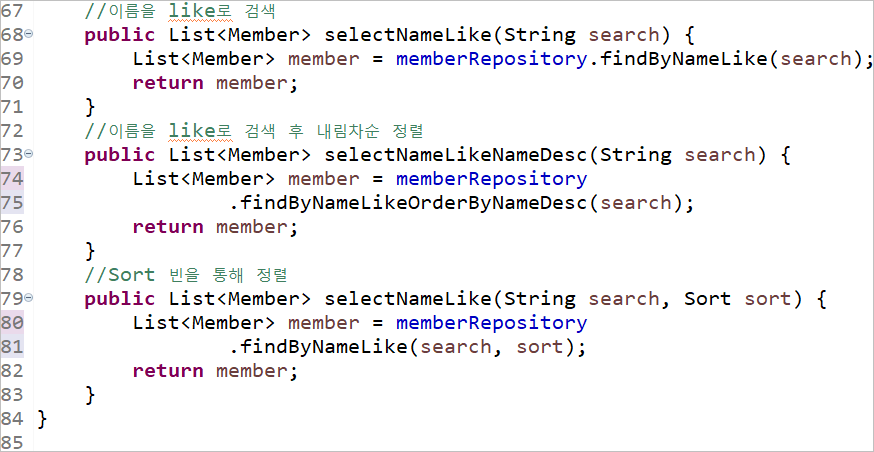
### 서비스 : /B21bSpringDataJPA1/src/main/java/edu/spring/boot/jpa/MemberService.java



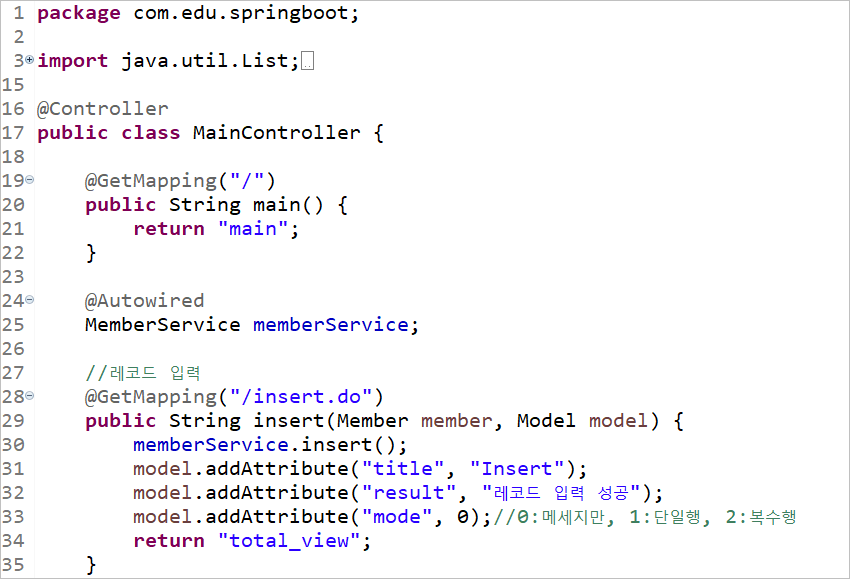
이와 같은 패턴으로 10개정도 입력합니다.

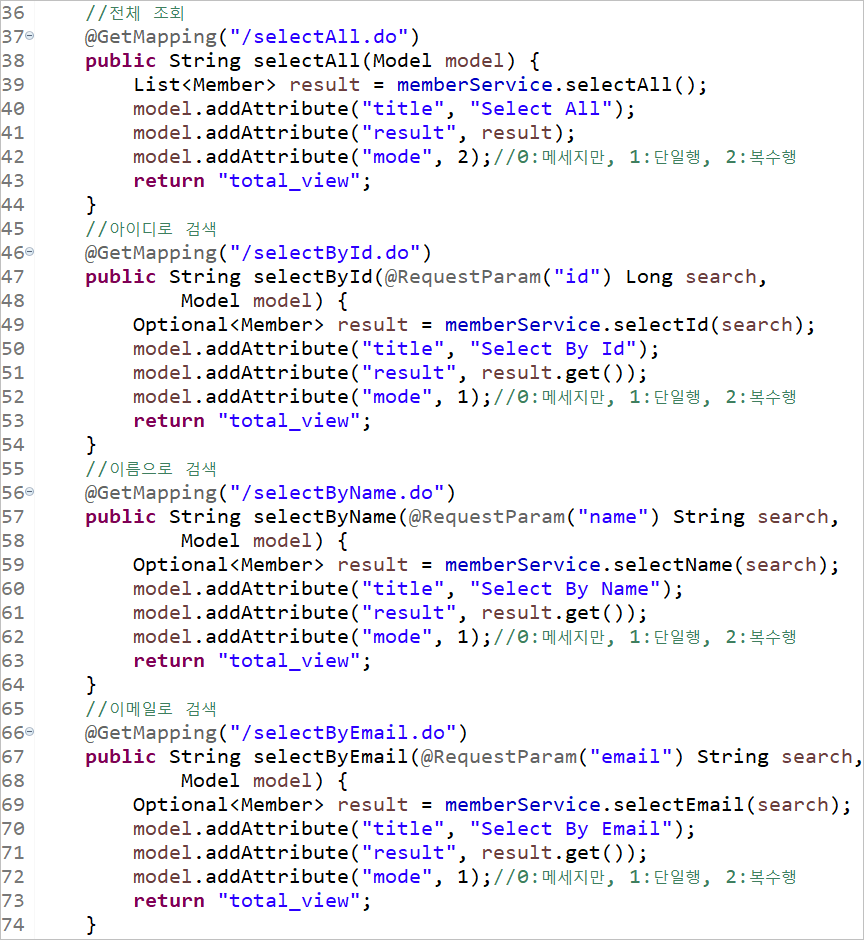
을지문덕, 계백, 김유신, 연개소문, 양만춘, 김종서, 최영, 선덕여왕, 신사임당, 성삼문, 이방원 등

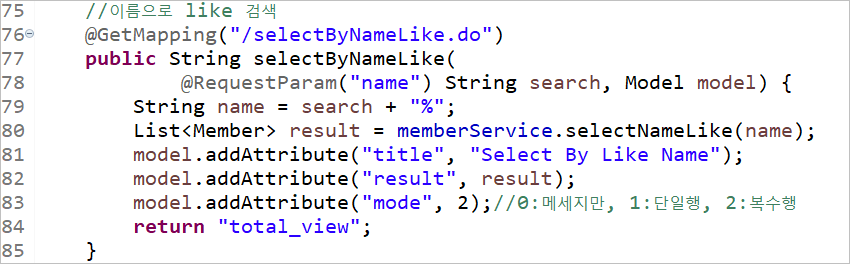


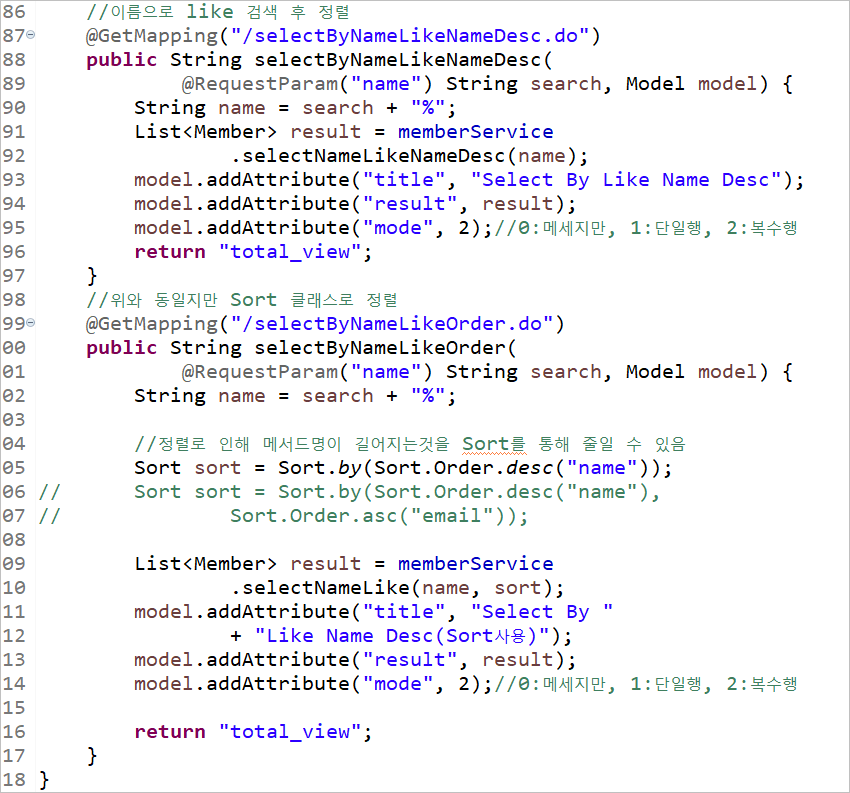


### 컨트롤러 : /B21bSpringDataJPA1/src/main/java/edu/spring/boot/MainController.java

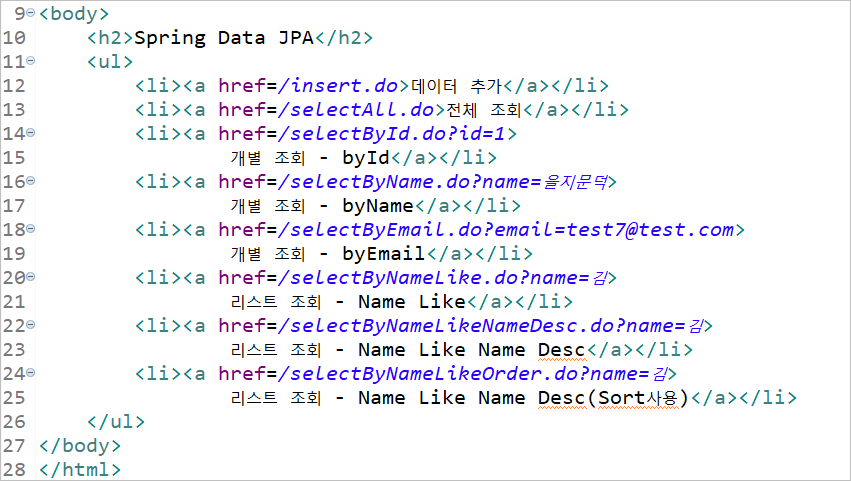








### 뷰 : /B21bSpringDataJPA1/src/main/webapp/WEB-INF/views/main.jsp



### 뷰 : /B21cSpringDataJPA2/src/main/webapp/WEB-INF/views/total\_view.jsp

