

# 미니프로젝트 1 - 야구 관리 프로그램

1. DB 연결 방법
2. DB 사용법
3. 템플릿 엔진 사용법

위 3 가지를 아직 배우지 못했기 때문에, 자바 애플리케이션을 만들도록 하겠습니다.

모든 결과는 Console에 출력합니다. (웹 화면 대신)

모든 입력은 main 메서드가 있는 클래스에서 Scanner로 입력 받습니다. (Controller 대신)

## 1. DB 생성

### 1.1 야구장 테이블(총 3개 야구장)

- stadium

```
id (PK, 자동 번호 증가) (int)
name (varchar)
created_at (timestamp)
```

### 1.2 팀 테이블(총 3개 팀)

- team

```
id (PK, 자동 번호 증가)
stadium_id (int)
name (varchar)
created_at (timestamp)

stadium_id는 fk 입니다.
```

### 1.3 선수 테이블(각 9명)

- player

```
id (PK, 자동 번호 증가)
team_id (int)
name (varchar)
position (varchar)
created_at (timestamp)
```

team\_id와 position은 다중 칼럼 유니크 제약조건이 필요합니다.  
team\_id는 fk 입니다.

## 1.4 퇴출선수 테이블

- out\_player

```
id (PK, 자동 번호 증가)
player_id (int)
reason (varchar)
created_at (timestamp)
```

player\_id는 fk 입니다.

## 2. 자바 클래스 생성

### 2.1 BaseBallApp 생성

main 함수를 가지는 클래스

해당 클래스에서 Scanner로 입력을 받고, 입력받은 값을 토대로 Service의 메서드를 호출합니다.  
Service의 메서드를 호출한 뒤 return 받은 결과를 Console에 출력합니다.

콘솔 예시!!

1. 콘솔에 출력되는 질문

- 어떤 기능을 요청하시겠습니까?

2. 클라이언트가 입력하는 내용

- 선수등록?teamId=1&name=이대호&position=1루수

3. main 메서드에서 할 일

- 입력되는 값을 파싱한다.
- ? 앞의 내용이 선수등록이면 PlayerService의 선수등록() 메서드를 호출한다.
- 호출시에 인수로 1, 이대호, 1루수를 전달한다.
- PlayerService의 선수등록() 메서드에서 해당 값을 받아서 PlayerDao의 insert() 메서드를 호출한다.
- 값이 DB에 잘 들어갔다면, 결과가 1로 리턴될 것이다. 1이 리턴되면 Console에 성공이라는 메시지를 출력한다.

## 2.2 DBConnection 생성

DB 연결 Connection을 생성하는 클래스

## 2.3 Model 생성

모델은 테이블과 다르게 camel 표기법으로 변수 네이밍을 합니다.

- Stadium
- Team
- Player
- OutPlayer

## 2.4 Dao 생성

- StadiumDao.java
- TeamDao.java
- PlayerDao.java
- OutPlayerDao.java

## 2.5 Service 생성

- StadiumService.java
- TeamService.java
- PlayerService.java
- OutPlayerService.java

## 3. 구현할 기능

### 3.1 야구장 등록

요청 : 야구장등록?name=잠실야구장

응답 : 성공이라는 메시지를 출력한다.

```
insert
```

### 3.2 전체 야구장 목록보기

요청 : 야구장목록

응답 : 야구장 목록은 Model -> Stadium을 List에 담아서 출력한다.

```
select
```

### 3.3 팀 등록

요청 : 팀등록?stadiumId=1&name=NC

응답 : 성공이라는 메시지를 출력한다.

```
insert
```

### 3.4 전체 팀 목록

요청 : 팀목록

응답 : 팀 목록은 Stadium 정보를 조인해서 출력해야 된다. TeamRespDTO가 필요하다.

```
select
```

### 3.5 선수 등록

요청 : 선수등록?teamId=1&name=이대호&position=1루수

응답 : 성공이라는 메시지를 출력한다.

포지션은 아래와 같이 중복되지 않고 입력되어야 합니다. 같은 포지션에 두 명의 선수가 등록될 수 없습니다. player 테이블에 포지션 칼럼은 팀 별로 유일해야 합니다.

즉, player 테이블의 team\_id와 position은 다중 칼럼 유니크 제약조건이 필요합니다.

- 1루수
- 2루수
- 3루수
- 포수
- 투수
- 유격수
- 좌익수
- 중견수
- 우익수

```
insert
```

### 3.6 팀별 선수 목록

요청 : 선수목록?teamId=1

응답 : 선수 목록은 Model -> Player를 List에 담아서 출력한다. (team\_id는 출력하지 않아도 된다)

```
select
```

### 3.7 선수 퇴출 등록

요청 : 퇴출등록?playerId=1&reason=도박

응답 : 성공이라는 메시지를 출력합니다.

두 개 이상의 write 문이 실행되어야 합니다. 이때는 반드시 트랜잭션 관리가 Service에서 필요합니다.

out\_player에 퇴출 선수를 insert하고, player 테이블에서 해당 선수의 team\_id를 null로 변경합니다.

```
insert -> out_player
update -> player
```

### 3.8 선수 퇴출 목록

요청 : 퇴출목록

응답 : OutPlayerRespDTO에 담아서 출력합니다.

```
select
```

아래는 player 테이블과 out\_player 테이블이 join 된 결과입니다.

퇴출 된 선수는 옆에 표시가 되고, 퇴출 되지 않은 선수는 표시되지 않습니다. 이것은 OuterJoin에 대해서 이해하고 있어야 풀 수 있는 문제입니다.

해당 문제는 스칼라 서브쿼리로도 해결할 수 있습니다.

p.id	p.name	p.position	o.reason(이유)	o.day(퇴출일)
1	이대홍	1루수	집안일	2020.09.09
2	강민홍	포수	집안일	2020.02.01
3	홍길동	투수		
4	임격경	유격수		
.. 등등등				
12	원빈	내야수		

### 3.10 포지션별 팀 야구 선수 페이지

요청 : 포지션별목록

응답 : PositionRespDto 에 값을 담아서 콘솔에 출력합니다.

```
select
```

포지션은 테이블의 칼럼입니다. 그리고 롯데, NC, 기아, 해태는 값입니다. 값이 칼럼으로 올라와 있는 형태의 결과입니다. 실무를 하다 보면, 다양한 요구사항이 생기게 됩니다.

가령 회사의 다른 부서에서 아래의 같은 요구를 할 수 있습니다.

- 김대리 각팀에 1루수가 누군지, 2루수가 누군지 보고 싶어
- 그런데, 기존에 보여주던 방식은 세로로 너무 길게 보여서 보기가 너무 힘들어
- 가로로 좀 아래와 같이 보여줄 수 없겠나?

당연히 클라이언트는 전문 용어를 모르기 때문에 위와 같이 요구하게 됩니다. 이때 피벗을 사용해서 세로로 보이는 데이터들을 가로로 보여줄 수 있습니다.

포지션	롯데	NC	기아	해태
1루수	아무개1	아무개2	아무개3	아무개4
2루수	아무개5	아무개6	아무개7	아무개8