

Case) List<Board>

1. dao.BoardDao.java

```
// BoardDao.java 파일에서 가장 먼저 실행되는 것이 public BoardDao { }  
    해당 함수(메서드)에서 필요한 객체를 호출하거나 public abstract List<Board> boardList();  
    메서드를 만들어 사용한다. public abstract Board getBoard(int no);  
public interface BoardDao {  
    public abstract List<Board> boardList();  
    public abstract Board getBoard(int no);  
    ...}
```

2. 문제인식/정의

Boarddao.java 인터페이스에 '게시판 목록' 쿼리 수행을 호출하는 메서드를 추가한다. 해당 메서드도 게시판 목록 데이터를 반환받아야 하기 때문에 리턴 타입은 List <BoardVO>다.

왜 List<Board> 인가?

우선, Board 는 VO 클래스 타입인 domain.Board.java 의 public class Board { } 객체를 의미한다. List 에 대한 정의를 다시 살펴보자.¹

3. 접근/해결

SELECT 결과 하나의 행에 대한 정보만 얻을 경우에는 BoardVO 클래스 타입으로 반환 받아야 한다. 하지만 목록 페이지의 경우 대부분 2개 이상의 행에 있는 정보를 받아야하므로 List 타입으로 반환받는다. List는 배열과 비슷한 자바의 자료형이다. 배열에 비해 여러 편리한점이 많지만 그중 가장 큰 장점은 크기가 가변적이라는 점이다. 배열의 경우 크기가 10개로 정했다면 10개를 초과하는 값들은 담을 수 없다. 더 많은 값을 담고 싶다면 새로운 더 큰 배열을 생성하여 사용하여야 한다. 하지만 List의 경우 10개의 크기로 생성하였더라도 11번째 값을 저장할 때 동적으로 그 크기가 커진다.

4. 정리

dao.BoardDaoImpl.java 에서 List<Board> boardList(); 를 살펴보자

우선 네임스페이스로 DB의 위치 경로를 불러온다. 반복적인 경로를 리팩토링으로 줄인 것
private final String NAME_SPACE = "com.springstudy.bbs.mapper.BoardMapper";

DB의 쿼리는 BoardMapper.xml 에서 boardList 를 호출시 아래의 쿼리가 저장된다.

```
<select id="boardList" resultType="Board" resultMap="boardResultMap">  
    SELECT * FROM springbbs ORDER BY no DESC  
</select>
```

결국 List<Board> boardList() 가 호출되면 DB의 쿼리의 쿼리 결과가 저장되는 셈이다.

@Override

```
public List<Board> boardList() {  
    return sqlSession.selectList(NAME_SPACE + ".boardList");}
```

쉽게 이해하자면 아래와 같다고 볼 수 있다.

```
List<Board> boardList() = SELECT * FROM springbbs ORDER BY no DESC
```

¹ List <BoardVO>에 대해 이해하고 싶으시다면 '컬렉션 프레임워크'와 '제너릭'에 대한 공부에 대한 필요하다. <https://kimvampa.tistory.com/165> 에서 추후 살펴보자.