

A photograph of Tim Cook, CEO of Apple, smiling and making a peace sign with his right hand. He is wearing glasses and a dark polo shirt. The background is slightly blurred, showing other people in a crowd. The image has a dark, semi-transparent overlay.

Tim Cooki

TEAM



BE 박영규



BE 김지민



FE 조배경



FE 김가연

01 진행사항

02 기술발표

진행사항

FE

- ✓ 프로젝트 폴더구조 설계
- ✓ 컴포넌트 분리
- ✓ 리액트 스터디

BE

- ✓ ERD 설계
- ✓ 스프링 시큐리티 및 JPA 등 스터디
- 👁 API 명세 작성

TimCooki

- ✓ 깃허브 프로젝트 보드 생성

Entity의 기본 생성자와 AccessLevel



HIBERNATE

@Entity

&

@NoArgsConstructor(access = AccessLevel.PRIVATE)



```
@Entity
@NoArgsConstructor(access = AccessLevel.PRIVATE)
public class Member {

    @Builder
    private Member(){
        ...
    }
    ...
}
```

같이 쓰면?



```
> Class 'Member' should have [public, protected] no-arg constructor
```


왜 에어가 뜨지?

JSR - 338

Java Specification Request - 자바 스펙 명세서. JDK구현시 반영되는 명세.

2.1 The Entity Class

The entity class must be annotated with the `Entity` annotation or denoted in the XML descriptor as an entity.

The **entity class must have a no-arg constructor**. The entity class may have other constructors as well. The **no-arg constructor must be public or protected**.

The entity class must be a top-level class. An enum or interface must not be designated as an entity.

The entity class must not be final. No methods or persistent instance variables of the entity class may be final.

왜 JSR에 저렇게 명세되었을까?

JPA의 지연로딩과 Proxy객체

1:N 관계로 매핑해놓은 엔티티에 지연로딩 설정을 해놓는다면, Proxy 객체를 통해 엔티티의 정보를 가져온다.

엔티티를 상속받은Proxy객체를 생성하기 위해서 @Entity로 설정한 클래스는 기본생성자 접근권한을 Protected
, Public으로 제한해놓은게 아닐까?

기술 발표

테스트를 해boja

Protected + 지연로딩



```
@Entity
@NoArgsConstructor(access = AccessLevel.PROTECTED)
public class Member {

    @Builder
    private Member() {
        ...
    }
    ...
}
```



```
@Entity
@Slf4j
@Getter
@NoArgsConstructor(access = AccessLevel.PROTECTED)
public class Board {

    ...

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "MEMBER_ID")
    private Member member;

}
```



```
insert into member values(1, '2023-05-23', 'test@naver.com', 'name', 'password');  
insert into board values(1, 'content', 'title', 1);
```

```
@Test
@Transactional
public void proxyTest(){

    Board board = em.find(Board.class, 1L); // Board_id 값이 1L인 데이터를 찾는다.
    System.out.println("-----");
    System.out.println("Board ID : " + board.getId());
    System.out.println("Board Title : " + board.getTitle());
    System.out.println("클래스 타입 : " + board.getMember().getClass());
    System.out.println("-----");
    System.out.println("Board의 (FK : Member) 참조 데이터");
    System.out.println("Member ID : " + board.getMember().getId());
    System.out.println("Member Name : " + board.getMember().getName());
    System.out.println("-----");
    em.close();
}
```


기술 발표

테스트 결과



```
-----  
Board ID : 1  
Board Title : title  
클래스 타입 : class com.ygpark.noticeService.domain.Member$HibernateProxy$y8gj6EhN  
-----
```

Board의 (FK : Member) 참조 데이터

Member ID : 1

Hibernate:

```
    select  
        member0_.member_id as member_i1_1_0_,  
        member0_.member_createdby as member_c2_1_0_,  
        member0_.member_email as member_e3_1_0_,  
        member0_.member_name as member_n4_1_0_,  
        member0_.member_password as member_p5_1_0_  
    from  
        member member0_  
    where  
        member0_.member_id=?
```

Member Name : name

```
-----
```



```
class com.ygpark.noticeService.domain.Member$HibernateProxy$y8gj6EhN
```

Private + 지연로딩



```
@Entity
@NoArgsConstructor(access = AccessLevel.PRIVATE)
public class Member {

    @Builder
    private Member(){
        ...
    }
    ...
}
```


그 외 조건은 이전 테스트와 동일

기술 발표

테스트 결과

기술 발표



```
HHH000143: Bytecode enhancement failed because no public, protected or package-private default  
constructor was found for entity: com.ygpark.noticeService.domain.Member. Private constructors don't  
work with runtime proxies!
```

Private + 즉시로딩



```
@Entity
@NoArgsConstructor(access = AccessLevel.PRIVATE)
public class Member {

    @Builder
    private Member(){
        ...
    }
    ...
}
```




```
@NoArgsConstructor(access = AccessLevel.PROTECTED)
@Entity
public class Board {

    ...

    @ManyToOne(fetch = FetchType.EAGER)
    @JoinColumn(name = "MEMBER_ID")
    private Member member;
}
```

기술 발표

테스트 결과



```
-----  
Board ID : 1  
Board Title : title  
클래스 타입 : class com.ygpark.noticeService.domain.Member
```

```
-----  
Board의 (FK : Member) 참조 데이터  
Member ID : 1  
Member Name : name  
-----
```

결론

엔티티 클래스 기본생성자의 접근 Level을 Private로 하면
지연로딩을 통한 객체 참조할 때,
프록시 객체 생성이 안되기 때문에 에러가 발생한다.