

JPA persist, flush clear

☰ 속성	
☷ 태그	

[Spring] JPA, persist, flush, clear

14 Nov 2019

spring boot에서 사용하는 persist, flush, clear 정의 및 이해

```
@Repository
public class MemberRepository {

    @PersistenceContext
    private EntityManager em;

    public Long save(Member member) {
        em.persist(member);
        em.flush();
        em.clear();
        return member.getId();
    }
}
```

@PersistenceContext는 @Autowired랑 비슷하다고 한다. 이 포스트의 메인은 아니다.

EntityManager에서 세가지 동작을 할 수 있다. persist, flush, clearDB에는 트랜잭션 (Transaction)이 있다. 예를 들어 insert, update, delete와 같은 쿼리를 날렸다고 가정하자. 그러면 현재 트랜잭션이 시작되고 진행중이다. 트랜잭션이 시작되고 끝나기 직전까지 내가 날린 데이터는 영속성 컨텍스트에 저장된다. 캐시라고 보면 된다. 쿼리를 날려서 영속성 컨텍스트에 저장하는 명령어 = em.persist(member); 이다. member라는 객체가 persist를 통해 영속성 컨텍스트에 보관중이다. 트랜잭션을 끝내고 싶다면 commit을 통해 실제 DB에 저장, 수정, 삭제를 하던가, rollback을 통해서 DB에 저장하지 않고 영속성 컨텍스트 데이터를 날리면 된다.

- 영속성 컨텍스트에 쿼리를 저장 = persist()

```
public Long save(Member member) {
    em.persist(member); // 영속성 컨텍스트에 저장
    em.flush();
    em.clear();
    return member.getId();
}
```

flush는 영속성 컨텍스트에 있는 데이터를 DB까지 접근시킨다. commit이 되는건 아니다. 여기엔 이해를 위해 뇌피셜이 첨가하면 트랜잭션이 시작되고 persist를 해서 영속성 컨텍스트

에 저장되는 데이터는 쿼리를 날리는게 아니라 날릴 준비를 완료하는 것.commit을 하면 쿼리를 날리고 DB에 저장까지 함.flush는 쿼리를 날리지만 commit이 하는 저장 작업이 없어서 DB에 반영되지 않는다.flush를 하면 내가 jpa로 구성한 데이터가 DB에 저장되지는 않지만 어떤식으로 실제 쿼리를 날리는지 확인할 수 있음. persist로는 불가능함.

- DB에 실제 쿼리를 날림. DB에 반영안됨. = flush()

```
public Long save(Member member) {  
    em.persist(member); // 영속성 컨텍스트에 저장  
    em.flush(); // 영속성 컨텍스트에 있는 데이터를 DB로 쿼리 전송  
    em.clear();  
    return member.getId();  
}
```

clear는 영속성 컨텍스트에 저장한 데이터를 지워버린다. persist로 영속성 컨텍스트에 데이터가 저장되었을때 clear를 하면 데이터가 지워지는 것이다. 실제 DB에 저장된 데이터를 지우는게 아니다.

- 영속성 컨텍스트에 데이터를 제거함.

```
public Long save(Member member) {  
    em.persist(member); // 영속성 컨텍스트에 저장  
    em.flush(); // 영속성 컨텍스트에 있는 데이터를 DB로 쿼리 전송  
    em.clear(); // 영속성 컨텍스트에 있는 데이터를 제거  
    return member.getId();  
}
```