

스프링부트로 **RestFulAPI** 구현하기

6장 JPA 기본

- 영속성 컨텍스트

박명회

5장 JPA 속성

1. 1차 캐시

영속성컨텍스트는 내부에 **1차 캐시**를 가지고 있습니다. 이 때 캐시의 키는 엔티티의 **@Id** 애너테이션이 달린 기본키입니다.

2. 쓰기지연

쓰기지연은 트랜잭션을 커밋하기 전까지는 데이터베이스에 실제로 질의문을 보내지 않고 쿼리를 모았다가 트랜잭션을 커밋하면 모았던 쿼리를 한번에 실행하는 것을 의미합니다.

3. 변경감지

트랜잭션을 커밋하면 **1차 캐시**에 저장되어 있는 엔티티의 값과 현재 엔티티의 값을 비교해서 변경된 값이 있다면 변경 사항을 감지해 변경된 값을 데이터베이스에 자동으로 반영합니다.

4. 지연로딩

지연로딩은 쿼리로 요청한 데이터를 애플리케이션에 바로 로딩하는 것이 아니라 필요할때 날려 데이터를 조회하는것을 말합니다.

5장 JPA 속성

New 상태: 새로 생성된 객체, 아직 영속성 컨텍스트에 추가되지 않음.

Managed 상태: 영속성 컨텍스트에 포함되어 관리되는 객체.

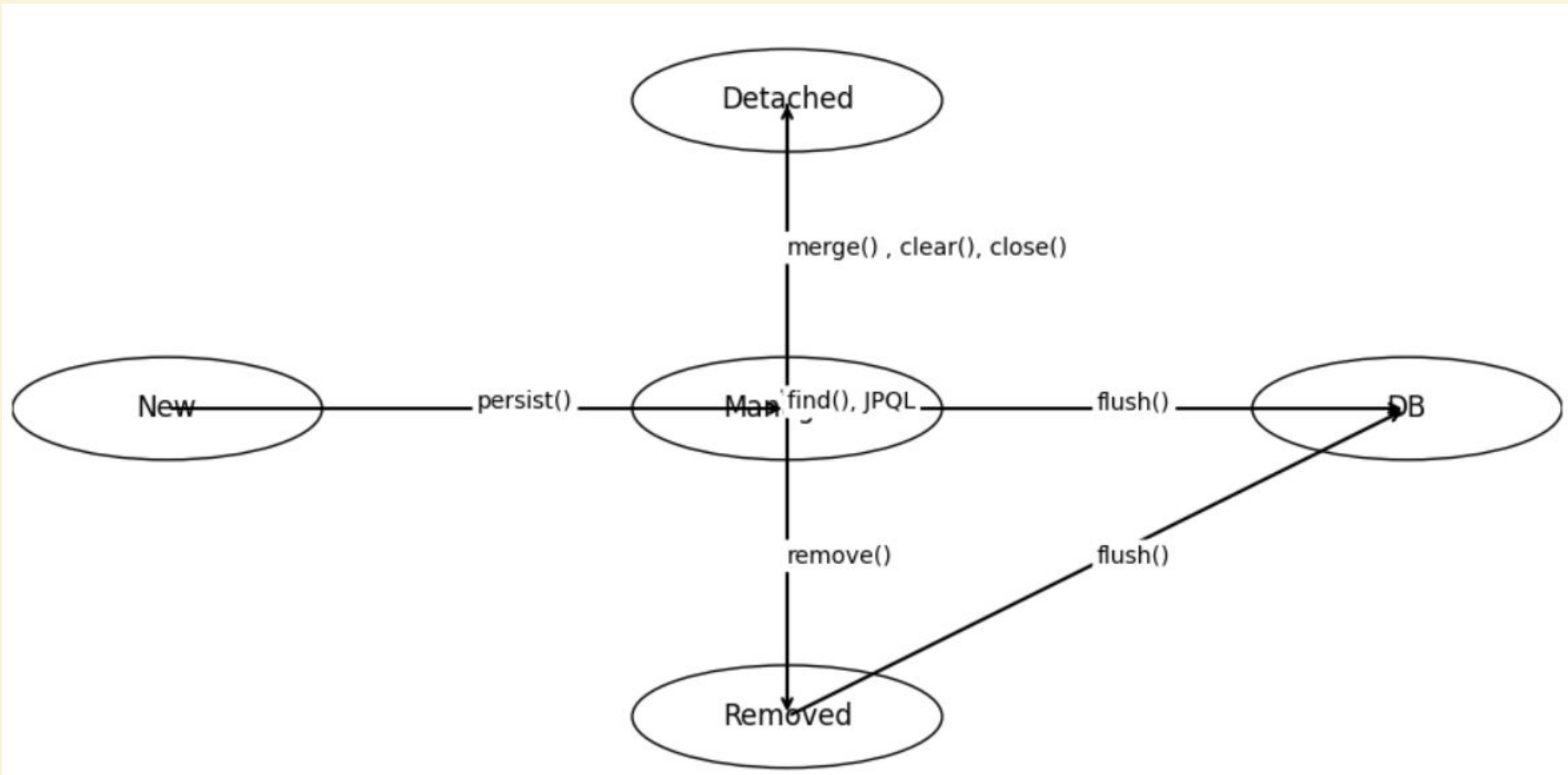
Detached 상태: 영속성 컨텍스트에서 분리된 객체.

Removed 상태: 삭제된 객체.

DB: 실제 데이터베이스.

5장 JPA 영속성 컨텍스트

영속성 컨텍스트



5장 @Transactional에 대해서

@Service

@RequiredArgsConstructor

public class UserTestService {

private final EntityManager entityManager;

private final UserRepository userRepository;

@Transactional

public void testUser(){

User user1 = User.builder().name("이길").email("ddd@naver.com").build();

User dbUser1 = userRepository.save(user1); // Persist (New -> Persist)

System.out.println(entityManager.contains(dbUser1));

System.out.println(entityManager.contains(user1));

dbUser1.setName("변경감지");

user1.setName("버겨겨경감지");

}

}

5장 JPA 날짜 입력

```
@EntityListeners(AuditingEntityListener.class)
```

```
public class User {
```

```
    @CreateDate
```

```
    @Column(updatable = false)
```

```
    private LocalDateTime createdAt;
```

```
    @LastModifiedDate
```

```
    @Column(updatable = true)
```

```
    private LocalDateTime modifiedAt;
```

```
...
```

```
@SpringBootApplication
```

```
@EnableJpaAuditing
```

```
public class Ex04Application {
```

5장 JPA 날짜 입력

AuditorAwareImpl.class

```
@Component  
public class AuditorAwareImpl implements AuditorAware<String> {  
    @Override  
    public Optional<String> getCurrentAuditor() {  
        return Optional.of("박명회");  
    }  
}
```

User.class

```
@EntityListeners(AuditingEntityListener.class)  
public class User {  
  
    @CreatedBy  
    private String createName;  
  
    @LastModifiedBy  
    private String modName;  
  
    ....  
  
}
```

5장 JPA 기본 함수 추가시...

@Repository

```
public interface UserRepository extends JpaRepository<User,Long> {  
    Optional<User> findByName(String name);  
    Optional<User> findByEmail(String email);  
}
```


감사합니다