

chapter08

Spring Boot

1. Transaction 처리

01 Transaction 처리

■ Spring Transaction 설정 (AOP 설정)

- 데이터베이스, 코드 기반, 선언적 트랜잭션 처리를 지원
- 설정 방법은 XML 설정, 애너테이션 설정, AOP 설정
- DBConfiguration 클래스에 @EnableTransactionManagement와 transactionManager 메서드를 추가

```
package com.board.configuration;

import javax.sql.DataSource;

@PropertySource("classpath:/application.properties")
@EnableTransactionManagement
@Configuration
public class DBConfiguration {

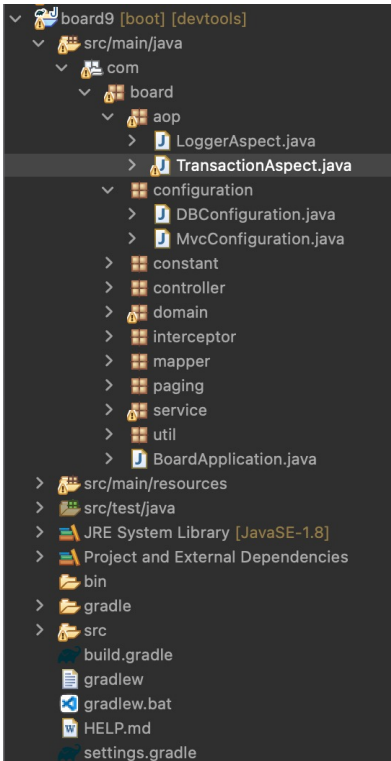
    @Autowired
    private ApplicationContext applicationContext;

    @Bean
    public PlatformTransactionManager transactionManager() {
        return new DataSourceTransactionManager(dataSource());
    }
}
```

01 Transaction 처리

■ Spring Transaction 설정 (AOP 설정)

- aop 패키지에 TransactionAspect 클래스를 추가



```
package com.board.aop;

import java.util.Collections;

@Configuration
public class TransactionAspect {

    @Autowired
    private PlatformTransactionManager transactionManager;

    private static final String EXPRESSION = "execution(* com.board..service.*Impl.*(..))";

    @Bean
    public TransactionInterceptor transactionAdvice() {

        List<RollbackRuleAttribute> rollbackRules = Collections.singletonList(new RollbackRuleAttribute(Exception.class));

        RuleBasedTransactionAttribute transactionAttribute = new RuleBasedTransactionAttribute();
        transactionAttribute.setRollbackRules(rollbackRules);
        transactionAttribute.setName("*");

        MatchAlwaysTransactionAttributeSource attributeSource = new MatchAlwaysTransactionAttributeSource();
        attributeSource.setTransactionAttribute(transactionAttribute);

        return new TransactionInterceptor(transactionManager, attributeSource);
    }

    @Bean
    public Advisor transactionAdvisor() {

        AspectJExpressionPointcut pointcut = new AspectJExpressionPointcut();
        pointcut.setExpression(EXPRESSION);

        return new DefaultPointcutAdvisor(pointcut, transactionAdvice());
    }
}
```

01 Transaction 처리

■ Spring Transaction 설정 (AOP 설정)

- TransactionAspect 클래스 설명

구성 요소	설명
transactionManager	DBConfiguration 클래스에 빈(Bean)으로 등록한 PlatformTransactionManager 객체
EXPRESSION	포인트컷!! 비즈니스 로직을 수행하는 모든 ServiceImpl 클래스의 모든 메서드를 의미
rollbackRules	트랜잭션에서 롤백을 수행하는 규칙(Rule) RollbackRuleAttribute 생성자의 인자로 Exception 클래스를 지정하였는데, 자바에서 모든 예외는 Exception 클래스를 상속받기 때문에 어떠한 예외가 발생하던 무조건 롤백이 수행됨을 의미
setName 메서드	트랜잭션의 이름을 설정 함 트랜잭션 모니터에서 트랜잭션의 이름으로 확인이 가능 함
pointcut	AOP의 포인트컷(pointcut)을 설정 함을 의미 EXPRESSION에 지정한 ServiceImpl 클래스의 모든 메서드를 대상으로 설정

01 Transaction 처리

■ Spring Transaction 설정 (AOP 설정)

- BoardServiceImpl 클래스의 registerBoard 메서드 수정

```
@Override
public boolean registerBoard(BoardDTO params) {
    int queryResult = 0;

    if (params.getIdx() == null) {
        queryResult = boardMapper.insertBoard(params);
    } else {
        queryResult = boardMapper.updateBoard(params);
    }

    BoardDTO board = null;
    System.out.println(board.getTitle());

    return (queryResult == 1) ? true : false;
}
```

01 Transaction 처리

■ Run App

- 애플리케이션 실행하고 게시글 등록 수행 후 콘솔 내용 확인
- BoardServiceImpl 클래스의 registerBoard 메서드 수정 (원본으로)