트랜잭션 - TransactionManager 활용

프로젝트명 : B08aTransactionManager

준비사항

- 의존설정 : Spring Web, Lombok, JDBC API, Oracle Driver, Mybatis Framework
- JSP 사용을 위한 설정을 한다.
- Refresh Gradle Project 를 눌러 적용한다.

트랜잭션(Transaction) 이란..??

- 데이터베이스의 상태를 변환시키는 하나의 논리적 기능을 수행하기 위한 작업의 단위 또는 한꺼번에 모두 수행되어야 할 일련의 연산들을 의미한다.
- 상태 변경이란 INSERT, SELECT, UPDATE, DELETE와 같은 CRUD 작업을 의미한다.
- 트랜잭션은 상황에 따라 여러 개가 만들어질 수 있다.
- 그 하나의 트랜잭션은 Commit(저장) 되거나 Rollback(철회)될 수 있다.

트랜잭션의 전형적인 예

- A → B 에게 이체

```
try {
    A 계좌 잔고 조회;
    if ( 이체 가능 ) {
        A 계좌 잔고 감소 업데이트;    // 데이터베이스 작업 수행 1
        B 계좌 잔고 증가 업데이트;    // 데이터베이스 작업 수행 2
    }
} catch (Exception e) {
    System.out.println("에러 발생");
}

// 거래 중 에러가 발생하면 롤백 시킨다.
```

트랜잭션 설정 및 처리

```
try {
    작업 1;
    balance.put(id, amount);    // 데이터베이스 작업 수행 1
    작업 2;
    order.put(id, count);       // 데이터베이스 작업 수행 2
    작업 3;
} catch (Exception e) {
    System.out.println("에러 발생");
}
```

- 작업의 범위가 위와 같은 경우
    - 데이터 베이스 작업 수행시 에러가 발생할 수 있음
    - 일반적인 비즈니스 로직에서 에러가 발생할 수 있음
- 작업 범위내에서 에러가 발생하면 이미 수행된 작업을 **롤백(Rollback)** 한다.
- 에러가 없다면 데이터베이스에 수행한 모든 작업을 **커밋(Commit)** 한다.

테이블 생성

```
--기존테이블 삭제
drop table transaction_pay;
drop table transaction_ticket;

--티켓 구매 금액을 입력하는 테이블
create table transaction_pay (
    userid varchar2(30) not null,
    amount number not null
);

--구매한 티켓의 갯수를 입력하는 테이블
create table transaction_ticket (
    userid varchar2(30) not null,
    t_count number(2) not null
        check(t_count<=5)
);

--데이터 입력 테스트1 (성공)
insert into transaction_pay values ('korea', 40000); --입력성공
insert into transaction_ticket values ('korea', 4); --입력성공

--데이터 입력 테스트2 (실패)
insert into transaction_pay values ('korea', 90000); --입력성공
insert into transaction_ticket values ('korea', 9); --check제약조건 위배로 입력불가

--확인 및 커밋
select * from transaction_pay;
select * from transaction_ticket;
commit;
```

여기까지 작성하세요.

구매페이지

DTO : com.edu.springboot.jdbc.PayDTO.java

```java
package com.edu.springboot.jdbc;

import lombok.Data;

@Data
public class PayDTO {
    private String userid;
    private int amount;
}
```

DTO : com.edu.springboot.jdbc.TicketDTO.java

```java
package com.edu.springboot.jdbc;

import lombok.Data;

@Data
public class TicketDTO {
    private String userid;
    private int t_count;
}
```

서비스 : com.edu.springboot.jdbc.ITicketService.java

```java
package com.edu.springboot.jdbc;

import org.apache.ibatis.annotations.Mapper;

@Mapper
public interface ITicketService {

    public int ticketInsert(TicketDTO ticketDTO);
    public int payInsert(PayDTO payDTO);
}
```

컨트롤러 : com.edu.springboot.MainController.java

```java
package com.edu.springboot;

import javax.servlet.http.HttpServletRequest;

@Controller
public class MainController {

    @Autowired
    ITicketService dao;
```

```java
    @Autowired
    PlatformTransactionManager transactionManager;

    @Autowired
    TransactionDefinition definition;

    @RequestMapping("/")
    public String home() {
        return "home";
    }

    //티켓구매
    @RequestMapping(value="/buyTicket.do", method=RequestMethod.GET)
    public String buy1() {
        return "buy";
    }
```

Home : webapp/WEB-INF/views/home.jsp

```jsp
<body>
    <h2>스프링 부트 프로젝트</h2>
    <ul>
        <li><a href="/">루트</a></li>
    </ul>

    <h2>TransactionManager로 트랜잭션 처리</h2>
    <ul>
        <li><a href="./buyTicket.do">티켓구매</a></li>
    </ul>
</body>
```

뷰 : webapp/WEB-INF/views/buy.jsp

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="jakarta.tags.core" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
    <h2>티켓 구매하기</h2>
    <form action="buyTicket.do" method="post">
    <table border="1">
        <tr>
            <th>아이디</th>
            <td><input type="text" name="userid" value="" /></td>
        </tr>
```

```
        <tr>
            <th>수량</th>
            <td>
                <select name="t_count">
                <c:forEach begin="1" end="10" step="1" var="num">
                    <option value="${num }">${num }</option>
                </c:forEach>
                </select>
            </td>
        </tr>
```

```
        <tr>
            <th>에러발생</th>
            <td>
                <input type="checkbox" name="err_flag" value="1" />
                체크하면 예외가 발생합니다.
            </td>
        </tr>
    </table>
    <input type="submit" value="전송하기" />
    </form>
</body>
</html>
```

여기까지 작성해서 구매페이지로 진입되는지 확인해주세요.

구매처리하기

컨트롤러 : com.edu.springboot.MainController.java

```java
    @RequestMapping(value="/buyTicket.do", method=RequestMethod.POST)
    public String buy2(TicketDTO ticketDTO, PayDTO payDTO,
            HttpServletRequest req, Model model) {

        String viewPath = "success";
        TransactionStatus status = transactionManager.getTransaction(definition);
        try {
            //DB처리1
            payDTO.setAmount(ticketDTO.getT_count() * 10000);
            int result1 = dao.payInsert(payDTO);
            if(result1==1) System.out.println("transaction_pay 입력성공");

            //비즈니스로직 처리(의도적인 에러발생)
            String errFlag = req.getParameter("err_flag");
            if(errFlag!=null) {
                int money = Integer.parseInt("100원");
            }

            //DB처리2
            int result2 = dao.ticketInsert(ticketDTO);
            if(result2==1) System.out.println("transaction_ticket 입력성공");

            model.addAttribute("ticketDTO", ticketDTO);
            model.addAttribute("payDTO", payDTO);

            transactionManager.commit(status);
        }
        catch (Exception e) {
            e.printStackTrace();
            viewPath = "error";
            transactionManager.rollback(status);
        }
        return viewPath;
    }
}
```

Mapper : src/main/resources/mybatis/mapper/TicketPayDAO.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE mapper
    PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

```

```xml
<mapper namespace="com.edu.springboot.jdbc.ITicketService">

    <insert id="payInsert"
        parameterType="com.edu.springboot.jdbc.PayDTO">
            insert into transaction_pay (userid, amount)
                values (#{userid}, #{amount})
    </insert>

    <insert id="ticketInsert"
        parameterType="com.edu.springboot.jdbc.TicketDTO">
            insert into transaction_ticket (userid, t_count)
                values (#{userid}, #{t_count})
    </insert>
</mapper>
```

뷰 : webapp/WEB-INF/views/error.jsp

```jsp
<body>
    <h2>티켓 구매 실패</h2>

    <a href="buyTicket.do">티켓구매</a>
</body>
```

뷰 : webapp/WEB-INF/views/success.jsp

```jsp
</head>
<body>
    <h2>티켓 구매 성공</h2>
    아이디 : ${ticketDTO.userid }<br />
    구매수 : ${ticketDTO.t_count }<br />
    결제금액 : ${payDTO.amount }<br />
    <a href="buyTicket.do">티켓구매</a>
</body>
</html>
```