11 JDBC와 SpringDAO

SpringDAO 기초

- SpringDAO
  - Spring에서 제공하는 DAO(Data Access Object) 모듈
  - 일반적으로 Data Access Framework이라 부름
  - Framework들의 기본 원리는 Template Method Pattern을 기본으로 구성됨

- Template Method Pattern
  - 어떤 작업 알고리즘의 골격을 정의함
  - 알고리즘의 구조는 그대로 유지하면서 특정 작업 부분만 변경할 수 있음

- JDBC
  - JDBC를 이용해서 Database에 비종속적인 DB 연동 프로그램을 구현할 수 있음
  - JDBC 프로그램 작성 순서

Step 1	Driver 로딩	Step 2	Connection 연결	Step 3	Statement 생성
Step 4	SQL 명령어 전송	Step 5	결과값 처리하기	Step 6	Connection 연결 해제

SpringDAO 설정

- DataSource 설정
  - SpringDAO만을 위한 설정은 아니지만 SpringDAO를 이용하기 위한 가장 기본적인 설정임
    → DataSource 설정을 통해 Connnection 연결을 획득할 수 있기 때문임
  - DataSource를 이용할 수 있는 방법은 여러 가지가 있음
    - Apache 그룹의 Common 라이브러리를 이용해서 설정하는 방법
    - Spring Framework가 제공하는 DataSource 구현 클래스를 이용하는 방법
    - WAS에서 제공하는 DataSource를 이용하는 방법

- JdbcTemplate
  - Template Method Pattern을 적용하여 Template Method를 제공해 주는 클래스
  - JdbcTemplate 객체를 획득하는 방법
    - UserDAOSpring 클래스를 상속받아 작성하는 방법
    - JdbcDaoSupport라는 클래스를 상속받아 작성하는 방법

JdbcTemplate API 사용

JdbcTemplate API	설명
queryForInt() 메소드	▪ SELECT문 실행결과로 리턴 되는 하나의 정수 값을 받기 위해서 사용함 ▪ SELECT 구문에서 파라미터 데이터가 포함되어 있는 경우, 이를 매핑하기 위한 방법도 제공함
queryForObject() 메소드	SQL문 수행 결과가 하나의 Object로 리턴되는 경우에 이용할 수 있는 메소드임
query() 메소드	SELECT문의 실행결과가 여러 목록으로 리턴되는 경우에 사용할 수 있음
RowMapper 이용	RowMapper 인터페이스를 구현한 클래스 내에 결과 값의 매핑정보를 담고 그것을 JdbcTemplate 메소드의 매개변수로 넘겨주면 됨
update() 메소드	INSERT, UPDATE, DELETE SQL 문을 수행하기 위해서 사용됨
PreparedStatementSetter 이용	SQL 문에 파라미터를 매핑할때 PreparedStatementSetter를 사용하여 파라미터의 이름으로 명확하게 매핑할 수 있음

Transaction 관리