MyBatis

MyBatis란

기존 JDBC는 JDBC 연동 과정이 복잡하고 SQL문이 코드 내에 섞여 복잡해짐

장점

데이터 베이스에 접속하는 작업을 캡슐화 => JDBC코드 작성 불편함 제거

저장 프로시저(SQL문을 만들어 놓고 필요할 때마다 호출) 지원

객체 SQL 사이의 parameter들을 자동으로 Mapping

sql쿼리를 소스코드와 분리(XML에 저장)하여 코드의 간결성 유지보수성 향상

VO 없이 DTO ,MAP 등으로 매핑 가능

특징

퍼시스턴스(Persistence) 프레임워크

프로그램이 종료되어도 데이터가 사라지지 않음

SQL에 변경이 있을 때 마다 자바 코드를 수정하거나 컴파일하지 않아도 됨

데이터소스(DataSource)기능과 트랜잭션 처리 기능을 제공

데이터소스: 커넥션을 얻어오고 반납하는 작업 수행

커넥션 풀 관리 목적

트랜잭션 처리 : sql 연산의 논리적 단위



config file의 구성

MyBatis configuration file에서 트랜잭션 매니저 설정, 데이터소스 설정

MyBatis의 컴포넌트들

MyBatis configuration file

MyBatis3의 작업 설정을 설명하는 XML 파일

데이터베이스 연결대상<dataSource>, 매핑 파일 경로<mappers>

Spring과 통합하여 사용할 때는 지정 필요x

org.apache.ibatis.session.SqlSessionFactoryBuilder

MyBatis3 구성파일을 읽고 생성하는 SqlSessionFactory 구성요소

XML설정 파일의 설정으로 SqlSessionFactory 인스턴스를 빌드

org.apache.ibatis.session.SqlSessionFactory

SqlSession을 생성

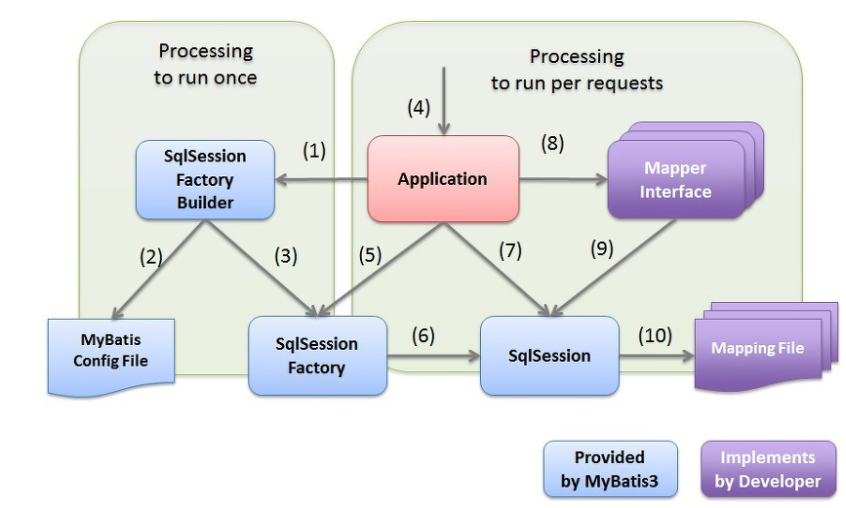
org.apache.ibatis.session.SqlSession

SQL 실행 및 트랜잭션 제어를 위한 API 제공

생성 SQL명령어를 실행하기 위한 메서드들을 가지고 있음

SqlSession을 통행 직접 쿼리문 실행

MyBatis 주요 구성요소가 DataBase Access하는 순서



1. 어플리케이션이 SqlSessionFactoryBuilder에 SqlSessionFactory 빌드하도록 요청
2. MyBatis configuration file의 구성 파일 읽음
3. 구성 파일의 정의에 따라 SqlSessionFactory 생성
4. 어플리케이션이 쿼리문 호출을 위해 SqlSession 객체를 생성을 요청
5. 빌드된 SqlSessionFactory를 통해 sqlSessionFactory 가져옴
6. SqlSession 애플리케이션 반환
7. SqlSession을 통해 mapper인터페이스 구현 개체가져옴
8. 애플리케이션이 mapper 인터페이스 메서드를 호출(나중에 나옴)
9. mapper 인터페이스 구현 개체가 SqlSession 호출하고 SQL실행 요청
10. SqlSession 매핑파일에서 실행할 SQL 가져와 SQL 실행