

JSP/Servlet

MyBatis - SQL처리해도 됨.

MyBatis는 자바 언어를 위한 오픈 소스 ORM(Object-Relational Mapping) 프레임워크입니다. 기존의 JDBC 코드에서 발생하는 반복적인 작업과 SQL 매핑을 단순화하고, 데이터베이스와의 상호작용을 더 쉽게 만들어줍니다.

MyBatis의 주요 특징과 기능은 다음과 같습니다:

SQL 매핑: MyBatis는 XML 파일이나 어노테이션을 사용하여 SQL 문을 매핑할 수 있습니다. 이를 통해 개발자는 별도의 SQL 코드를 작성할 필요 없이 매핑 파일을 통해 SQL 문을 관리할 수 있습니다.

파라미터 바인딩: MyBatis는 SQL 문의 파라미터를 자동으로 매핑해줍니다. 파라미터를 자동으로 바인딩하여 JDBC 코드에서 발생하는 반복적인 작업을 줄여줍니다.

결과 매핑: MyBatis는 쿼리 실행 결과를 자바 객체로 매핑해줍니다. SQL 결과를 자동으로 객체로 변환하여 개발자가 편리하게 사용할 수 있습니다. 매핑 방식은 XML 파일이나 어노테이션을 사용하여 정의할 수 있습니다.

동적 SQL: MyBatis는 동적 SQL 문을 작성할 수 있습니다. 조건에 따라 동적으로 SQL 문을 생성하고 실행할 수 있으며, 복잡한 쿼리 작성을 간소화할 수 있습니다.

성능 최적화: MyBatis는 캐시 기능을 제공하여 데이터베이스 액세스 성능을 향상시킬 수 있습니다. 결과를 캐시하여 동일한 쿼리의 반복 실행을 피하고, 성능을 개선할 수 있습니다.

MyBatis는 XML 기반의 설정 파일과 SQL 매핑 파일을 사용하여 데이터베이스와의 상호작용을 정의합니다. MyBatis를 사용하기 위해서는 설정 파일에 데이터베이스 연결 정보와 매핑 파일의 경로를 지정해야 하며, 필요한 곳에서 SQL 세션을 생성하여 사용할 수 있습니다.

MyBatis는 자유도가 높은 SQL 매핑과 성능 최적화를 위한 다양한 기능을 제공하여 개발자들이 데이터베이스와 효과적으로 상호작용할 수 있도록 도와줍니다.

스프링 - JDBC 템플릿

JdbcTemplate은 스프링 프레임워크에서 제공하는 JDBC(Java Database Connectivity)를 사용하기 위한 간편한 도구입니다. JdbcTemplate은 JDBC 작업을 수행하는 데 필요한 많은 반복적인 코드를 줄여주고, 예외 처리, 리소스 관리 등을 자동으로 처리해줍니다.

JdbcTemplate은 데이터베이스와의 연결, SQL 문 실행, 결과 처리 등을 단순화하고 추상화합니다. 주요 기능으로는 다음과 같습니다:

연결 관리: JdbcTemplate은 데이터베이스 연결을 관리하고 재사용할 수 있도록 지원합니다. 연결 획득, 반환, 예외 처리 등을 자동으로 처리해줍니다.

SQL 문 실행: JdbcTemplate은 SQL 문 실행을 간단하게 처리할 수 있도록 도와줍니다. PreparedStatement를 생성하고 실행하며, 파라미터 바인딩, 쿼리 실행 결과를 ResultSet으로 변환하는 작업을 수행합니다.

예외 처리: JdbcTemplate은 JDBC 작업 중 발생하는 예외를 처리합니다. SQLException을 스프링의 DataAccessException 계층 구조로 변환하여 일관된 예외 처리를 제공합니다.

결과 처리: JdbcTemplate은 쿼리 실행 결과를 ResultSet으로 받아와서 결과를 처리합니다. ResultSet을 Java 객체로 매핑하거나, 결과를 List, Map 등의 컬렉션 형태로 반환할 수 있습니다.

JdbcTemplate은 스프링의 JDBC 지원 모듈에 포함되어 있으며, 스프링 프레임워크의 일부로 제공됩니다. JdbcTemplate을 사용하려면 스프링 컨테이너에 등록하고 필요한 곳에서 DI(Dependency Injection)을 통해 사용하면 됩니다.

<GSON>

Maven Repository - GSON - 2.8.5.jar 다운로드 - lib 복사

Gson은 Google에서 개발한 Java 기반의 JSON 라이브러리입니다. Gson은 JSON 데이터와 Java 객체 간의 직렬화 및 역직렬화를 처리하는데 사용됩니다. JSON 형식의 데이터를 Java 객체로 변환하거나, Java 객체를 JSON 형식으로 변환하는 작업을 간편하게 수행할 수 있습니다.

Gson은 다양한 기능을 제공합니다. JSON 데이터를 Java 객체로 변환할 때는 JSON 키와 Java 객체의 필드를 매핑하여 자동으로 매핑 작업을 수행합니다. 또한, 복잡한 JSON 구조를 가진 데이터도 처리할 수 있으며, 필요에 따라 직접 변환 규칙을 설정할 수도 있습니다.

역직렬화 작업에서는 Java 객체를 JSON 데이터로 변환할 수 있습니다. 이때 Gson은 Java 객체의 필드 값을 기반으로 JSON 데이터를 생성합니다. 필요한 경우 직렬화 대상에서 제외할 필드를 설정할 수도 있습니다.

Gson은 Maven이나 Gradle과 같은 의존성 관리 도구를 통해 프로젝트에 추가할 수 있습니다. Gson을 사용하기 위해서는 Gson 라이브러리를 프로젝트에 포함시킨 후에 필요한 클래스와 메서드를 import하여 사용하면 됩니다.