# Summary

SqlLoader는 CCMS에서 사용하는 DB Framework입니다.

## System Architecure

SqlLoader는 xml으로 Query를 관리를 합니다. 시스템이 초기화될 때 모든 xml의 Query를 메모리에 로딩을 하고, Dao에서 ID을 사용해서 호출하는 구조입니다.

동적으로 쿼리가 바뀌는 Dynamic Query의 경우 메모리에 룰을 로딩해서 입력 값에 따라 룰이 판단하여 쿼리를 반환을 해줍니다.

Xml은 특정한 문법이 정해져 있습니다. 그리고ID는 쿼리를 찾는 키가 됨으로 중복이 되면 안됩니다.

# Query

SqlLoader에서 사용하는 xml의 문법은 간단합니다. Select, insert, update, delete 이렇게 기본적으로 4개를 제공을 하고 있고, Select는 내부적으로 rule 태그를 가질 수 있습니다.

## Select

Select 태그는 데이터베이스에서 데이터를 조회할 때 사용이 됩니다. 일반적인 Query부터, 입력에 따라 동적으로 Query가 변경이 되는 Dynamic Query 2가지 스타일이 있습니다.

또한 selectObject, selectInteger, selectLong등 Object 타입 외에 Integer나 Long을 반환이 가능합니다.

### **Default Select**

기본적인 select 사용방법은 다음과 같습니다.

|  |
| --- |
|  |

Select 태그 안에 실행되기를 원하는 Query문을 작성하면 됩니다.

이제 Dao에서 select1 쿼리를 실행하는 방법은 다음과 같습니다.

|  |
| --- |
|  |

Select 태그의 경우 단일 결과를 반환하는 경우 selectObject을 호출을 하고, 목록을 반환하는 경우 selectList을 호출하면 됩니다.

IRowmapper는 실제 데이터베이스에서 Query를 실행하고 얻은 ResultSet을 Object로 변경을 해주는 역할을 하게 됩니다.

만약 SelectObject였다면 IRowMapper가 한번만 수행이 되고, SelectList였다면 실행결과의 Row 수만큼 호출이 됩니다.

### **PrearedStatement**

실행되는 쿼리를 PreparedStatement을 사용하여 실행을 하고 싶은 경우에 대해 알아보도록 하겠습니다.

|  |
| --- |
|  |

쿼리에서는 “?” 을 사용하면 됩니다. 이 값은 JDBC의 PreparedStatement의 setString을 통해서 입력이 되어야 합니다. 이 값을 넣어주는 방법은 IParamMapper을 사용을 하면 됩니다.

IParamMapper의 사용방법은 다음과 같습니다.

|  |
| --- |
|  |

selectObject에 ParamMapper를 넘기면 실제 쿼리를 실행전에 IParamMapper가 한번 호출이 됩니다.

여기서 “?”의 개수만큼 PreparedStatement의 setString이나 setInt 등을 호출해서 값을 넣어주어야 정상적으로 쿼리가 실행이 됩니다.

### **Repaced Query**

쿼리내의 특정 문자열을 치환하는 기능에 대해 알아보도록 하겠습니다.

문자열을 치환하는 기능은 쿼리 실행 전에 쿼리를 수정을 해서 데이터베이스에서 실행되도록 하는 기능입니다. 이 기능은 특별히 제약이 있지는 않습니다. 유틸에서 제공하는 방법은 $바꿀문자$을 사용한 글자를 바꿔주는 기능을 제공합니다. 실제 사용한 예제를 보면 다음과 같습니다.

|  |
| --- |
|  |

위의 쿼리에는 $TS$라고 되어있는 부분이 있습니다. 이 부분을 쿼리 실행 전에 바꾸는 방법은 다음과 같습니다.

|  |
| --- |
|  |

IStringMapper을 보면 Hashmap을 생성해서 QueryStringUtil의 findAndReplace을 호출하고 그 결과를 다시 반환을 하고 있습니다.

즉 쿼리의 특정 문자를 바꾸는 방법은 $을 사용하는 것 외에 이 부분에서 임의의 방법으로 쿼리 자체를 변경이 가능합니다.

위의 소스대로 하면 $TS$는 BSS라는 글자로 변경이 되고, 이를 데이터베이스에서 실행이 됩니다.

## Insert

Insert는 테이블에 데이터를 추가할 때 사용이 됩니다.

### **Default Insert**

사용방법은 다음과 같습니다.

|  |
| --- |
|  |

## Update

Update는 테이블의 데이터를 수정을 할 때 사용이 됩니다.

### **Default Update**

사용방법은 다음과 같습니다.

|  |
| --- |
|  |

## Delete

Delete 태그는 DB의 테이블의 데이터를 삭제할 때 사용이 됩니다.

### **Default Delete**

사용방법은 다음과 같습니다.

|  |
| --- |
|  |

# Dynamic Query

Dynamic Query는 입력에 따라 실행이 되는 쿼리가 동적으로 바뀌는 경우에 사용을 합니다.

## IsNotNull condition

IsNotNull은 입력으로 넘어온 오브젝트의 특정 값이 null이 아니면 isNotNull이 가진 쿼리를 추가하게 하는 태그이다.

|  |
| --- |
|  |

위의 태그를 예를 들어 보면, 입력으로 특정 오브젝트를 받으면 getName을 호출하고 그 값이 null이 아니면 select \* from company where id=1이 되고, 만약 그 값이 null 이라면 select \* from company 이 최종 쿼리가 된다.

실제 호출하는 Dao을 보면 다음과 같다.

|  |
| --- |
|  |

기존의 selectObject 대신에 dynamicSelectObject을 호출한다. dynamicSelectObject의 경우 입력으로 IRowMapper와 select 태그에서 사용이 될 company 를 넘기게 된다.

dynamicSelectObject에서 company의 getName을 호출하면 test라는 값을 받게 되고, 실제 데이터베이스에 select \* from company where id=1을 수행하고, IRowMapper을 호출하게 된다.

## IsNotEmpty condition

IsNotEmpty은 입력으로 넘어온 오브젝트의 특정 값이 null이 아니고, 숫자타입이면 0이 아닌 경우, 리스트의 경우 size가 0이 아닌 경우, isNotEmpty이 가진 쿼리를 추가하게 하는 태그이다.

|  |
| --- |
|  |

## IsEmpty condition

IsEmpty은 입력으로 넘어온 오브젝트의 특정 값이 null이거나, 숫자타입이면 0이, 리스트의 경우 size가 0인 경우, isEmpty이 가진 쿼리를 추가하게 하는 태그이다.

|  |
| --- |
|  |

## IsSame condition

IsSame은 특정 값인지를 체크해서 같은 경우에 수행이 된다. 특히 문자열은 대소문자를 무시하고 비교를 할지, 아니면 대소문자도 같은지 구분을 할지 옵션을 추가할 수 있다.

Sensitive는 옵션이다.

|  |
| --- |
|  |

위의 결과를 보면 Company에 name에 Test1를 넣었다면, select \* from company where id=1이 된다.

첫번째 isSame은 sensitvie가 n이여서 대소문자를 가리지 않는다.

만약 Test2를 넣었다면 첫번째 isSame은 test1과 Test2는 틀리기 때문에 무시되고, 두번째 isSame에서 대소문자 포함해서 비교를 하여, select \* from company where id=2라는 결과가 나오게 된다.

## IsNotSame condition

IsNotSame은 은 IsSame의 반대이다.

|  |
| --- |
|  |

## IsTrue condition

IsTrue는 무조건 쿼리를 더하는 경우에 사용이 된다.

|  |
| --- |
|  |

위의 경우 조건에 상관없이 쿼리는 select \* from company where id=2 가 된다.

## In condition

InCondition은 쿼리의 in절을 사용하고 싶을 때 사용이 된다.

|  |
| --- |
|  |

In의 경우는 입력 받은 개체의 getRoles를 해서 얻은 결과를 name in (1,2,3) 으로 변경을 한다.

Dao를 보면 다음과 같다.

|  |
| --- |
|  |

String2개 admin과 user를 List로 넘기게 되면, 조금 전의 쿼리는 select \* from ccc where 1=1 and name in (‘admin’, ‘user’)로 변경이 된다.

만약 문자열이 아닌 오브젝트라면 다음과 같이 Field 속성을 주면 된다.

|  |
| --- |
|  |

Dao는 다음과 같다.

|  |
| --- |
|  |

String list가 아닌 SimpleRole의 list를 넘겨주게 되면 in절에서 SimpleRole의 어떤 값을 사용할지를 알려줘야 하는데 이때 field가 사용이 된다.

For condition

for는 조건식을 여러 번 호출을 할 때 사용이 된다. 단 Input으로 넘어가는 오브젝트는 for 가 넘겨받은 개체가 아니라, 루프를 돌 때 사용되는 개체를 서브 조건식에 넘겨주게 된다.

|  |
| --- |
|  |

For는 property 값 하나만 파라미터로 받고, isNotNull은 getDepts를 해서 얻은 부서 정보를 IsNotNull로 넘겨주게 된다.

|  |
| --- |
|  |