MyBatis Generator Generated Java Client Objects

MyBatis Generator (MBG) generates Java client objects of several types. Java client objects are used to make interaction with the generated XML elements much easier. For each table in the configuration, MBG generates one or more Java client objects. For MyBatis 3, these are mapper interfaces. For iBATIS 2.x, these are DAO interfaces and implementation classes. Generating Java client objects is optional, and is controlled by the `<javaClientGenerator>` configuration element. MBGr can generate clients of the following types:

- For MyBatis 3.x:
  - XMLMAPPER - for use with the MyBatis 3.x mapper support
- For iBATIS 2.x:
  - IBATIS - for use with the iBATIS DAO Framework
  - SPRING - for use with the Spring Framework
  - GENERIC-CI - with no dependencies beyond the iBATIS Data Mapper
  - GENERIC-SI - also with no dependencies beyond the iBATIS Data Mapper

Every field and method generated includes the non-standard JavaDoc tag `@mbggenerated`. When run from the Eclipse plugin, on subsequent runs every field and method that includes this JavaDoc tag will be deleted and replaced. Any other field or method in the class will be untouched. With this in mind, you can add other fields and methods to the classes without fear of losing them in subsequent runs - simply DO NOT include the `@mbggenerated` JavaDoc tag on anything that you add to the class.

Outside of the Eclipse plugin, Java files need to be merged by hand, but you can use the `@mbggenerated` JavaDoc tag to know what is safe to delete from a prior version of a file.

Note: in the following descriptions, the term "BLOB" is used to refer to any column with a data type of BLOB, CLOB, LONGVARCHAR, or LONGVARBINARY.

Methods Common to All DAO Types 所有 DAO 类型的公共方法列表

Depending on the specifics of the table, and the configuration options, the Java client generator will generate these methods:

- countByExample
- deleteByPrimaryKey
- deleteByExample
- insert
- insertSelective
- selectByPrimaryKey
- selectByExample
- selectByExampleWithBLOBs
- updateByPrimaryKey (with an override to specify whether or not to update BLOB columns)
- updateByPrimaryKeySelective (will only update non-null fields in the parameter class)
- updateByExample (with an override to specify whether or not to update BLOB columns)
- updateByExampleSelective (will only update non-null fields in the parameter class)

MBG attempts to make it easier to deal with tables that contain BLOBs by generating different objects and methods so that you can use the BLOB fields, or ignore them, depending on the situation.

See the Example Class Usage page for an example of using the `selectByExample` method.

XMLMAPPER Clients (MyBatis 3.x)

XMLMAPPER clients are interfaces that will map to methods in the generated XML mapper files. For example, suppos that MBG generated an interface called `MyTableMapper`. You can use the interface as follows:

```
SqlSession sqlSession = sqlSessionFactory.openSession();

try {
  MyTableMapper mapper = sqlSession.getMapper(MyTableMapper.class);
  List<MyTable> allRecords = mapper.selectByExample(null);
} finally {
  sqlSession.close();
}
```

See the standard MyBatis documentation for details on how to create the instance of `sqlSessionFactory`.

IBATIS DAOs (iBATIS 2.x)

iBATIS DAOs depend on the iBATIS DAO framework (an optional part of iBATIS - now deprecated). They extend the SqlMapDaoTemplate class and are constructed with an instance of the DAOManager object, and call methods in their super class to execute the different statements.

MBG does not update the "dao.xml" file for you - you must add the appropriate entries manually.

The iBATIS DAO framework is a very elementary IoC container and can be useful if you are not already using something like Spring or PicoContainer to manage dependencies. However, the framework is now deprecated and we suggest that you move to Spring.

SPRING DAOs (iBATIS 2.x)

SPRING DAOs depend on the Spring framework. They extend Spring's SqlMapClientDaoSupport class, and are constructed by the Spring container.

GENERIC-CI DAOs (iBATIS 2.x)

GENERIC-CI DAOs call methods in iBATIS' SqlMapClient interface directly. An instance of the interface is supplied through constructor injection.

GENERIC-SI DAOs (iBATIS 2.x)

GENERIC-SI DAOs call methods in iBATIS' SqlMapClient interface directly. An instance of the interface is supplied through setter injection.