

MyBatis Configuration XML File

A MyBatis XML configuration file features a variety of settings and properties. You can take a brief look at how you can build the codes and configure options in this Section.

Configuration XML

A typical MyBatis XML comprises properties, settings, typeAliases, typeHandlers, objectFactory, plugins, environments, databaseIdProvider and mappers, described as follows:

Sample Configuration

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-config.dtd">

<configuration>

    <properties resource="org/mybatis/example/config.properties">
        <property name="username" value="dev_user"/>
        <property name="password" value="F2Fa3!33TYyg"/>
    </properties>

    <settings>
        <setting name="cacheEnabled" value="true"/>
        <setting name="lazyLoadingEnabled" value="true"/>
        <setting name="multipleResultSetsEnabled" value="true"/>
    </settings>

    <typeHandlers>
        <typeHandler
handler="egovframework.rte.psl.dataaccess.typehandler.CalendarMapperTypeHandler" />
    </typeHandlers>

    <typeAliases>
        <typeAlias alias="deptVO" type="egovframework.rte.psl.dataaccess.vo.DeptVO" />
        <typeAlias alias="empVO" type="egovframework.rte.psl.dataaccess.vo.EmpVO" />
        .
        .
        .
    </typeAliases>

    .
    .
    .
    .

</configuration>
```

- **properties** : Allows you to connect to the standard java properties (key=value) file and refer to the actual value that has been externalized in the form of property ({key}) (being drivers, urls, login credentials, etc.). You can also manually designate the resources for any valid URL by working classpath and url properties via the resource property.

- settings : Allows you to optimize MyBatis activities. Refer to the following for which settings are available to you and the default values thereof:

Setting	Description	Available Options		Default
cacheEnabled	You can enable/disable global use of caches set for each mappers.	true / false	TRUE	
lazyLoadingEnabled	You can enable/disable lazyLoading. Disable this for immediate loading.	true / false	TRUE	
aggressiveLazyLoading	You can enable/disable aggressiveLazyLoading. Enable this to keep the object with lazy loading properties from being called. The individual properties are kept from being loaded unless you call them.	true / false	TRUE	
multipleResultSetsEnabled	You can enable/disable allowing multiple ResultSets in a single syntax (available only if the driver supports the corresponding function).	true / false	TRUE	
useColumnLabel	You can use the column label instead of column name. As function may vary by driver, you need to work the test to make sure it functions as intended.	true / false	TRUE	
useGeneratedKeys	You can permit JDBC for keys generated. Driver is required. Set TRUE if you want to force key generation. Some drivers (Derby, etc.) ignore useGeneratedKeys.	true / false	FALSE	
autoMappingBehavior	You can enable auto-mapping of the columns in field/property by MyBatis. Set PARTIAL for auto-mapping, leaving the result unprocessed. Set FULL to process every auto-mapping.	NONE, PARTIAL, FULL	PARTIAL	
defaultExecutorType	You can set the default executor type. While SIMPLE features no distinct function, REUSE allows reuse of PreparedStatement. Set BATCH to reuse the	SIMPLE REUSE BATCH	SIMPLE	

	syntaxes and batch-process the amendments.		
defaultStatementTimeout	You can set the timeout for database response.	Positive	Null
safeRowBoundsEnabled	You can permit RowBound in the syntax overlap.	true / false	FALSE
mapUnderscoreToCamelCase	You can auto-map the conventional A_COLUMN into the Java property aColumn, the CamelCase type property.	true / false	FALSE
localCacheScope	MyBatis uses local cache to prevent circular references and speed up repeated nested queries. By default (SESSION) all queries executed during a session are cached. If localCacheScope=STATEMENT local session will be used just for statement execution, no data will be shared between two different calls to the same SqlSession.	SESSION / STATEMENT	SESSION
jdbcTypeForNull	Specifies the JDBC type for null values when no specific JDBC type was provided for the parameter. Some drivers require specifying the column JDBC type but others work with generic values like NULL, VARCHAR or OTHER.	JdbcType enumeration. Most common are: NULL, VARCHAR and OTHER	OTHER
lazyLoadTriggerMethods	Specifies which Object's methods trigger a lazy load	A method name list separated by commas	equals,clone,hashCode,toString
defaultScriptingLanguage	Specifies the language used by default for dynamic SQL generation.	A type alias or fully qualified class name.	org.apache.ibatis.scripting.xmltags.XMLDynamicLanguageDriver
callSettersOnNulls	Specifies if setters or map's put method will be called when a retrieved value is null. It is useful when you rely on Map.keySet() or null value initialization. Note primitives such as (int,boolean,etc.) will not be set to null.	true / false	FALSE
logPrefix	Specifies the prefix string that MyBatis will	Any String	Not set

add to the logger names.

logImpl	Specifies which logging implementation MyBatis should use. If this setting is not present logging implementation will be autodiscovered.	
	SLF4J / LOG4J / LOG4J2 / JDK_LOGGING / COMMONS_LOGGING / STDOUT_LOGGING / NO_LOGGING	Not set
proxyFactory	Specifies the proxy tool that MyBatis will use for creating lazy loading capable objects.	
	CGLIB / JAVASSIST	CGLIB

- typeAliases : You can make the Java type name short. Available only on XML configuration. Preferably used to ease typing.
- typeHandler : You can load the proper Java-type value everytime MyBatis sets parameters in PreparedStatement and loads corresponding values from ResultSet. You can register a typeHandler implementor.
- objectFactory : MyBatis uses ObjectFactory to generate the object instance.
- mappers: Defines SQL syntaxes mapped. The mapper file is designated to assign resources to either the classpath or URL.

A variety of miscellaneous settings such as plugins (intercepts calls when the mapping syntax is implemented. MyBatis permits plug-in to intercept the method calls), environments (for variation of environments), databaseIdProvider (execution of varied syntax available for database) are available to vary the configuration to fit the given environment and moment.