

MYBATIS - CONFIGURATION XML

http://www.tutorialspoint.com/mybatis/mybatis_configuration_xml.htm

Copyright © tutorialspoint.com

In the previous chapter, we have seen how to install MyBatis. This chapter discusses how to configure MyBatis using XML file.

Since we are communicating with the database, we have to configure the details of the database. *Configuration XML* is the file used for the XML-based configuration. By using this file, you can configure various elements.

The following programming is a typical structure of MyBatis configuration file.

```
<configuration>

  <typeAliases>
    <typeAlias alias = "class_alias_Name" type = "absolute_clas_Name"/>
  </typeAliases>

  <environments default = "default_environment_name">
    <environment id = "environment_id">
      <transactionManager type = "JDBC/MANAGED"/>

      <dataSource type = "UNPOOLED/POOLED/JNDI">
        <property name = "driver" value = "database_driver_class_name"/>
        <property name = "url" value = "database_url"/>
        <property name = "username" value = "database_user_name"/>
        <property name = "password" value = "database_password"/>
      </dataSource>

    </environment>
  </environments>

  <mappers>
    <mapper resource = "path of the configuration XML file"/>
  </mappers>

</configuration>
```

Let us discuss the important elements *tags* of the configuration XML file one by one.

environments tag

Within the environments element, we configure the environment of the database that we use in our application. In MyBatis, you can connect to multiple databases by configuring multiple *environment* elements. To configure the environment, we are provided with two sub tags namely **transactionManager** and **dataSource**.

transactionManager tag

MyBatis supports two transaction managers namely **JDBC** and **MANAGED**

- If we use the transaction manager of type **JDBC**, the application is responsible for the transaction management operations, such as, commit, roll-back, etc...
- If we use the transaction manager of type **MANAGED**, the application server is responsible to manage the connection life cycle. It is generally used with the Web Applications.

dataSource tag

It is used to configure the connection properties of the database, such as driver-name, url, user-name, and password of the database that we want to connect. It is of three types namely –

- **UNPOOLED** – For the dataSource type UNPOOLED, MyBatis simply opens and closes a connection for every database operation. It is a bit slower and generally used for the simple

applications.

- **POOLED** – For the dataSource type POOLED, MyBatis will maintain a database connection pool. And, for every database operation, MyBatis uses one of these connections, and returns them to the pool after the completion of the operation. It reduces the initial connection and authentication time that required to create a new connection.
- **JNDI** – For the dataSource type JNDI, MyBatis will get the connection from the JNDI dataSource.

Here is how you can use an environment tag in practice –

```
<environments default = "development">
  <environment id = "development">
    <transactionManager type = "JDBC"/>
    <dataSource type = "POOLED">
      <property name = "driver" value = "com.mysql.jdbc.Driver"/>
      <property name = "url" value = "jdbc:mysql://localhost:3306/details"/>
      <property name = "username" value = "root"/>
      <property name = "password" value = "password"/>
    </dataSource>
  </environment>
</environments>
```

typeAliases tag

Instead of specifying the absolute class name everywhere, we can use typeAliases, a shorter name for a Java type. Suppose, we have a class Student in Student.java file within the package named *tutorials_point.com.mybatis_examples*, then the absolute class name will be *tutorials_point.com.mybatis_examples.Student*. Instead of using this name to address the class every time, you can declare an alias to that class as shown below –

```
<typeAliases>
  <typeAlias alias = "Student" type = "mybatis.Student"/>
</typeAliases>
```

mappers tag

Mapper XML file is the important file, which contains the mapped SQL statements. Mapper's element is used to configure the location of these mappers xml files in the configuration file of MyBatis *this element contains four attributes namely resources, url, class, and name*.

For example, the name of the mapper xml file is **Student.xml** and it resides in the package named as **mybatis**, then you can configure the mapper tag as shown below.

```
<mappers>
  <mapper resource = "mybatis/Student.xml"/>
</mappers>
```

- The attribute **resource** points to the classpath of the XML file.
- The attribute **url** points to the fully qualified path of the xml file.
- We can use mapper interfaces instead of xml file, the attribute **class** points to the class-path of the mapper interface.
- The attribute **name** points to the package name of the mapper interface. In the example provided in this chapter, we have specified the class path of the mapper XML using the resource attribute.

In addition to these, there are other elements that can be used in the configuration file of [MyBatis documentation](#). Refer MyBatis documentation for the complete details.

MyBatis with MySQL database

MySQL is one of the most popular open-source database systems available today. Let us create a **SqlMapConfig.xml** configuration file to connect to mysql database. The example given below are the dataSource properties *driver – name, url, user – name, and password* for MySQL database –

| Sr.No. | Property Name | value |
|--------|---------------|---|
| 1 | driver | com.mysql.jdbc.Driver |
| 2 | url | jdbc:mysql://localhost:3306/details <i>assumedatabaseis " details "</i> |
| 3 | username | root |
| 4 | password | password |

We use the transaction manager of type JDBC, means we have to perform the operations, such as commit and roll-back manually, within the application.

We use the dataSource of type UNPOOLED, which means new connection is created for each database operation. Therefore, it is recommended to close the connection manually after the completion of database operations.

SqlMapConfig.xml

Below given is the XML configuration for the examples used in this tutorial. Copy the content given below in a text file and save it as **SqlMapConfig.xml**. We are going to use this file in all the examples given in this tutorial.

```
<configuration>

    <environments default = "development">
        <environment id = "development">
            <transactionManager type = "JDBC"/>

            <dataSource type = "POOLED">
                <property name = "driver" value = "com.mysql.jdbc.Driver"/>
                <property name = "url" value = "jdbc:mysql://localhost:3306/details"/>
                <property name = "username" value = "root"/>
                <property name = "password" value = "password"/>
            </dataSource>

        </environment>
    </environments>

    <mappers>
        <mapper resource = "mybatis/Student.xml"/>
    </mappers>

</configuration>
```

Processing math: 100%