**Mybatis基本配置和搭建**

<https://blog.csdn.net/zdtao/article/details/49474557>

2015年10月28日 21:22:18

* 标签：
* [mybatis](http://so.csdn.net/so/search/s.do?q=mybatis&t=blog)
* 11673

今天，主要向大家分享下如何从头搭建一个最简单的mybatis项目

#### ****下载地址****

Mybatis3 最新下载地址：<https://github.com/mybatis/mybatis-3/releases>

### ****一, 创建配置文件****

在新建的project的src目录下，新建mybatis-config.xml

<?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>

<settings>

<setting name="mapUnderscoreToCamelCase" value="true"/>

<setting name="lazyLoadingEnabled" value="true"/>

</settings>

<typeAliases>

<typeAlias type="com.test.entity.Employee" alias="Employee"/>

<typeAlias type="com.test.entity.Customer" alias="Customer"/>

</typeAliases>

<environments default="local">

<environment id="local">

<transactionManager type="JDBC"></transactionManager>

<dataSource type="POOLED">

<property name="driver" value="com.mysql.jdbc.Driver"/>

<property name="url" value="jdbc:mysql://localhost:3306/orcl"/>

<property name="username" value="username"/>

<property name="password" value="password"/>

</dataSource>

</environment>

</environments>

<mappers>

<mapper resource="com/test/data/employee-mapper.xml"/>

<mapper resource="com/test/data/customer-mapper.xml"/>

</mappers>

</configuration>

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34

接下来为大家解释下该配置文件中的内容

1. settings   
定义mybatis的一些全局设置，这里 配置的两个属性

mapUnderscoreToCamelCase: 在映射database column名字和entity属性名时，自动将带下划线column名转化为常见的java驼峰形式属性名   
lazyLoadingEnabled: 延迟加载entity的关联属性

2. typeAlias   
定义java类型的别名，比如这里分别将com.yun.entity.Employee和com.yun.entity.Customer设置别名为Employee，Customer。这样在别处配置文件中使用 它们时，就不必再指明带package全名。

3. environments   
主要用于配置数据源   
可以配置多个environment，以用于不同的产品环境，这里只配置一个用于测试，并定义id为“local”

transactionManager: 有两种类型   
1, JDBC : 使用从数据源返回的连接管理commit和rollback   
2, MANAGED : 依靠容器来管理transaction

dataSource: 有3种类型   
1, UNPOOLED :每次请求新打开连接，用完后关闭连接   
2, POOLED : 使用连接池管理连接   
3, JNDI ：使用容器管理datasource时使用

4. mappers   
简而言之，mapper文件用于定义sql语句，以及与其对应的entity   
以employee-mapper.xml为例，以下是其简单实现：

<!DOCTYPE mapper

PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"

"http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.test.data.EmployeeMapper">

<select id="selectAllEmployee" resultType="Employee">

select \* from employee

</select>

<select id="selectEmployeeById" parameterType="long" resultType="Employee">

select \* from employee where employee\_id=#{id}

</select>

</mapper>

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13

这里，我们定义了2句sql statement，分别是

selectAllEmployee: 查找所有employee   
selectEmployeeById：根据id查找特定employee

返回结果类型均为Employee(看！这里typeAlias就起到作用了，不需要指明class全名)，区别在于前者是个List。

这里大家可能会对selectEmployeeById 中的**#{id}**感到疑惑，别急，下面会为您介绍。

**NOTE:**配置文件中的顺序不能乱来，对于例子中的几个配置，要按照顺序来定义：settings -> typeAliases -> environments -> mappers.因为mybatis是按照一个固定顺序来解析这个配置文件的，如果顺序不对，加载时会报错。

### ****二，获得SqlSession****

不赘述，直接上代码

public class MySqlSession {

private static String mybatisResource = "mybatis-config.xml";

private static SqlSessionFactory sqlSessionFactory;

public static SqlSessionFactory getSqlSessionFactory() {

try {

if (sqlSessionFactory == null) {

InputStream inputStream = Resources.getResourceAsStream(mybatisResource);

sqlSessionFactory = new SqlSessionFactoryBuilder().build(inputStream);

}

return sqlSessionFactory;

} catch (IOException e) {

throw new RuntimeException(e);

}

}

public static SqlSession newSqlSession() {

SqlSession session = getSqlSessionFactory().openSession();

return session;

}

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23

通过调用MySqlSession.newSqlSession()即可获得一个SQLSession对象，这也是mybatis中最核心的类了，负责各种select,update,delete接口等，这里就不详细解释了。

### ****三，DAO****

马上就到最后一步啦！   
还是直接上代码~

public class BatisEmployeeDao {

public List<Employee> getAllEmployees() {

SqlSession session = MySqlSession.newSqlSession();

try {

List<Employee> ret = session.selectList("selectAllEmployee");

return ret;

} finally {

session.close();

}

}

public Employee getEmployeeById(Long id) {

SqlSession session = MySqlSession.newSqlSession();

try {

Employee employee = session.selectOne("selectEmployeeById", id);

return employee;

} finally {

session.close();

}

}

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21

看到这里，想必各位看官也知道前面**#{id}**的value是哪里来的了,是的,它就是 getEmployeeById(Long id)中的参数，但两者名字不要求一致哦。

好了，到这里，一个最简单的mybatis配置和实现已经结束了，希望对您有所帮助。   
更多细节参看mybatis官方文档：<http://mybatis.github.io/mybatis-3/>

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/zdtao/article/details/49474557

[](https://blog.csdn.net/westbrook1998)

#### [westbrook1998](https://blog.csdn.net/westbrook1998)

2018-02-11 13:24#3楼

[回复](https://blog.csdn.net/zdtao/article/details/49474557#reply)

* 专门登录给楼主点个赞

[](https://blog.csdn.net/MarkerDiane)

#### [MarkerDiane](https://blog.csdn.net/MarkerDiane)

2018-02-06 17:00#2楼

[回复](https://blog.csdn.net/zdtao/article/details/49474557#reply)

* 从头到尾，还是挺详细的

[](https://blog.csdn.net/u012308971)

#### [u012308971](https://blog.csdn.net/u012308971)

2017-08-23 21:11#1楼

[回复](https://blog.csdn.net/zdtao/article/details/49474557#reply)

* 学习了。