

myBatis_HelloWorld

一、myBatis 简介:

MyBatis 是支持定制化 SQL、存储过程以及高级映射的优秀的持久层框架。MyBatis 避免了几乎所有的 JDBC 代码和手工设置参数以及抽取结果集。MyBatis 使用简单的 XML 或 注解 来配置和映射, 将 接口 和 Java 的 POJOs(Plain Old Java Objects, 普通的 Java 对象)映射成数据库中的记录。

二、HelloWorld 的基本步骤:

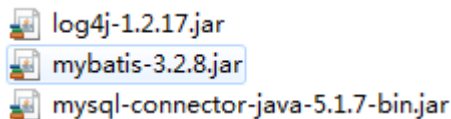
1. 加入 jar 包
2. 新建持久化类及数据表
3. 加入 MyBatis 配置文件: mybatis-config.xml
4. 加入 MyBatis 映射文件
5. 通过 MyBatis API 操作数据表

三、加入 jar 包:

日志包: log4j-1.2.17.jar

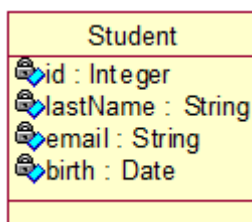
mybatis 核心包: mybatis-3.2.8.jar

数据库驱动: mysql-connector-java-5.1.7-bin.jar



四、新建持久化类及数据表

1. 持久化类:



```
public class Student {

    private Integer id;
    private String lastName;

    private String email;
    private Date birth;

    //setter and getter method
```

```
}
```

2. 数据表

students		
<u>id</u>	<u>int(11)</u>	<u><pk></u>
last_name	varchar(30)	
email	varchar(30)	
birth	date	

```
DROP TABLE IF EXISTS `students`;
```

```
CREATE TABLE `students` (  
    `id` int(11) NOT NULL auto_increment,  
    `last_name` varchar(30) default NULL,  
    `email` varchar(30) default NULL,  
    `birth` date default NULL,  
    PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=78 DEFAULT CHARSET=gb2312;
```

五、加入 MyBatis 配置文件

1. 打开 <http://mybatis.github.io/mybatis-3/zh/getting-started.html#> 复制以下部分，到 **src** 下的 **mybatis-config.xml** 文件中。

```
<?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>
  <environments default="development">
    <environment id="development">
      <transactionManager type="JDBC"/>
      <dataSource type="POOLED">
        <property name="driver" value="${driver}"/>
        <property name="url" value="${url}"/>
        <property name="username" value="${username}"/>
        <property name="password" value="${password}"/>
      </dataSource>
    </environment>
  </environments>
  <mappers>
    <mapper resource="org/mybatis/example/BlogMapper.xml"/>
  </mappers>
</configuration>
```

2. 修改连接数据库的部分:
3. 修改关联的 Mapper 文件:
4. 修改后的文件:

```
<?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>
  <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:///mybatis" />
        <property name="username" value="root" />
        <property name="password" value="1230" />
      </dataSource>
    </environment>
  </environments>
  <mappers>
    <mapper resource="com/atguigu/mybatis/helloworld/StudentMapper.xml" />
  </mappers>
</configuration>
```

连接数据库的基本信息

关联的 Mapper 文件

六、加入 MyBatis 映射文件：

1. 打开 <http://mybatis.github.io/mybatis-3/zh/getting-started.html#> 复制以下部分到 src 下的 com/atguigu/mybatis/helloworld/StudentMapper.xml 文件中

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper
  PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
  "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="org.mybatis.example.BlogMapper">
  <select id="selectBlog" resultType="Blog">
    select * from Blog where id = #{id}
  </select>
</mapper>
```

2. 修改 namespace 部分
3. 修改 select 节点部分
4. 修改后的文件：

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

<mapper namespace="com.atguigu.mybatis.helloworld.StudentMapper">

  <select id="selectStudent" resultType="com.atguigu.mybatis.helloworld.Student">
    select id, last_name, email, birth
    from students where id = #{id}
  </select>

</mapper>
```

七、通过 MyBatis API 操作数据表：

1. 在 com.atguigu.mybatis.helloworld 包下新建 Main 类，在其中新建 main 方法
2. 打开 <http://mybatis.github.io/mybatis-3/zh/getting-started.html#> 复制如下部分到 main 方法中

```
String resource = "org/mybatis/example/mybatis-config.xml";
InputStream inputStream = Resources.getResourceAsStream(resource);
sqlSessionFactory = new SqlSessionFactoryBuilder().build(inputStream);
```

```
SqlSession session = sqlSessionFactory.openSession();
try {
    Blog blog = (Blog) session.selectOne("org.mybatis.example.BlogMapper.selectBlog", 101);
} finally {
    session.close();
}
```

3. 做如下修改:

```
String resource = "mybatis-config.xml";
InputStream inputStream = Resources.getResourceAsStream(resource);
SqlSessionFactory sqlSessionFactory = new SqlSessionFactoryBuilder()
    .build(inputStream);

//SqlSession 表示 mybatis 应用程序和数据库 的一次会话。
//实际上对于 mybatis 的任何操作，都始于 SqlSession 对象。就像 hibernate 的 Session 一样
SqlSession session = sqlSessionFactory.openSession();
try {
    //selectOne:
    //第一个参数: namespace + "." + id
    //第二个参数: 实际传入的参数
    Student student = session.selectOne(
        "com.atguigu.mybatis.helloworld.StudentMapper.selectStudent", 50);
    System.out.println(student);
} finally {
    session.close();
}
```