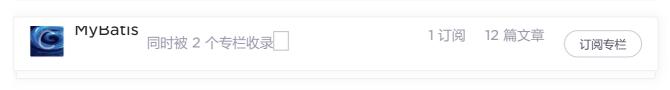
kilakila

(5条消息) 使用IDEA导入MyBatis源码进行调试_听 到微笑的博客-CSDN博客_idea调试mybatis源码

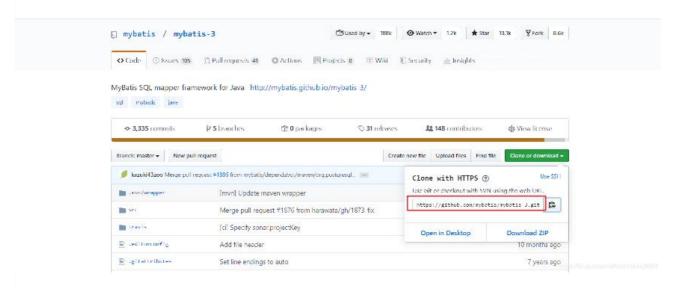
blog.csdn.net 已更新2022年8月16日

使用IDEA导入MyBatis源码进行调试



一. 下载源码

GitHub 地址: https://github.com/mybatis/mybatis-3



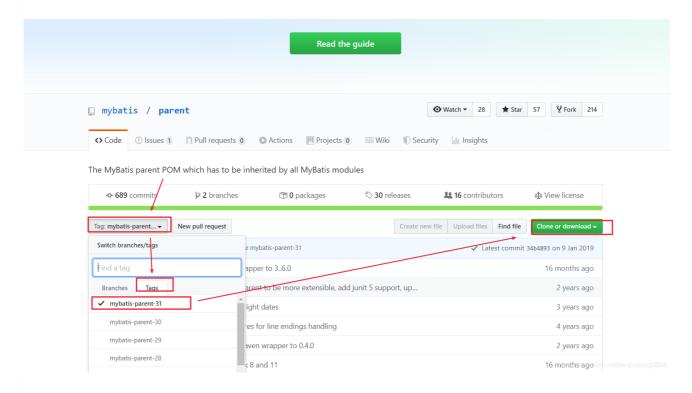
复制上面的地址执行下列命令:

1 git clone https://github.com/mybatis/mybatis-3.git mybatis-source

注: GitHub可能会很慢, 但是可以借助国内的Gitee进行"下载加速" 参考链接

打开 mybatis -source下pom.xml文件,会发现Mybatis源码还依赖于一个父工程:

去Github上下载对应的父工程,**下载的mybatis-parent版本要和mybatis源码中pom文件指定的父工程版本一致**。



二. 搭建MyBatis测试环境

注意: 这里我们只分析MyBatis源码, 所以并没有与Spring集成。

第一步:新建Maven,加入MyBatis相应依赖

```
1
    2
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 3
           xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.a
4
       <modelVersion>4.0.0</modelVersion>
 5
6
       <groupId>cn.tjd
7
       <artifactId>mybatis-source-study</artifactId>
8
       <version>1.0-SNAPSHOT
9
10
       cproperties>
11
          <junit.version>4.12</junit.version>
12
          <mybatis.version>3.2.8</mybatis.version>
```

```
13
            <mysql.version>5.1.32</mysql.version>
14
            <druid.version>1.0.9</druid.version>
15
        </properties>
16
17
        <dependencies>
18
19
            <dependency>
20
                <groupId>log4j
21
                <artifactId>log4j</artifactId>
22
                <version>1.2.17
23
            </dependency>
24
25
            <dependency>
26
                <groupId>org.mybatis
27
                <artifactId>mybatis</artifactId>
28
                <version>${mybatis.version}</version>
            </dependency>
29
30
            <dependency>
31
                <groupId>mysql</groupId>
32
                <artifactId>mysql-connector-java</artifactId>
33
                <version>${mysql.version}</version>
34
            </dependency>
35
36
            <dependency>
37
                <groupId>com.alibaba/groupId>
38
                <artifactId>druid</artifactId>
39
                <version>${druid.version}</version>
40
            </dependency>
41
42
            <dependency>
43
                <groupId>junit
44
                <artifactId>junit</artifactId>
45
                <version>${junit.version}</version>
46
                <scope>test</scope>
47
            </dependency>
48
49
        </dependencies>
50
51
        <build>
52
            <!-- 加载配置文件 -->
53
            <resources>
54
                <resource>
                    <directory>src/main/java</directory>
55
                    <includes>
56
                        <include>**/*.xml</include>
57
                    </includes>
58
                </resource>
59
```

注意,此时这里引入的MyBatis依赖都是中央仓库中的,并不是我们下载的源码。

第二步:新建Mapper接口和对应的映射文件



第三步:新建MyBatis配置文件

在resource目录下创建 SqlSessionConfig.xml 文件:

```
1
 2
    <!DOCTYPE configuration</pre>
            PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
 3
            "http://mybatis.org/dtd/mybatis-3-config.dtd">
 4
    <configuration>
 5
 6
        <settings>
 7
            <setting name="logImpl" value="STDOUT_LOGGING" />
 8
        </settings>
 9
        <!--配置类别名,配置后在Mapper配置文件(通常我们将编写SQL语句的配置文件成为Mapp
10
        <typeAliases>
11
            <package name="cn.tjd.pojo"/>
12
        </typeAliases>
13
        <environments default="development">
14
            <environment id="development">
15
                <transactionManager type="JDBC"></transactionManager>
16
                <dataSource type="P00LED">
17
                    cproperty name="driver" value="com.mysql.jdbc.Driver"/>
18
                    cproperty name="username" value="root"/>
19
                    cproperty name="password" value="123456"/>
20
                    cproperty name="url" value="jdbc:mysql://localhost:3306/test?
21
                </dataSource>
            </environment>
```

第四步:编写测试类测试

```
public class MyBatisTest {
 1
 2
        @Test
 3
         public void testMyBatisBuild() throws IOException {
             InputStream input = Resources.getResourceAsStream("SqlSessionConfig.x
 4
             SqlSessionFactory sessionFactory = new SqlSessionFactoryBuilder().bui
 5
             SqlSession sqlSession = sessionFactory.openSession();
 6
             TestMapper mapper = sqlSession.getMapper(TestMapper.class);
 7
             Map<String, String> student = mapper.getStudentById("00000b373502481t
 8
 9
10
     ∢
                                                                                    \blacktriangleright
```

测试成功即可进行下一步。

```
▼ MyBeisTess Syrms

▼ MyBeisTess Syrms

▼ trstMySatistics:

▼ trstMySatistics:

▼ trstMySatistics:

▼ trstMySatistics:

▼ trstMySatistics:

▼ coledDataSource forcefully closed/removed all connections.

PooledDataSource forcefully closed/removed all connections.

Opening JDBC Connection

Created connection 1372082959.

Setting autocommit to false on JDBC Connection [com.mysql.jdbc.JDBC4Connection@51c8530f]

==> Preparing: select * from student where id = ?

=> Parameters: @0000b373502481baalasf5229507cf8(String)

<== Columns: id, t_name, t_password, sex, description, pic_url, school_name, regist_date, remark

<== Row: @0000b373502481baalasf5229507cf8, 677936, 123456, 贝, 教师, www.bbk.com, XX大学, 2016-08-12 00:00:00.0, 英类

▼ Total: 1

Process finished with exit code 0
```

注: 之所以要在源码整合之前搭建好测试环境,是为了保证测试环境的可靠,因为后序的源码整合可能会发生一些问题,这样可以排除是因为测试环境错误所导致的。

三. 源码整合

第一步: 将源码项目 (两个) 导入当前测试项目所在的界面中

导入后的效果如图:



至于IDEA如何在一个界面中导入多个项目请参考:一个IDEA界面如何同时打开多个项目

第二步:将源码安装至本地仓库中(参考至: mybatis 源码导入IDEA)

切换到你下载的mybatis-parent目录:

```
1 | mvn clean install -Dmaven.test.skip=true
```

切换到你下载的mybatis源码目录:

```
1 | mvn clean install -Dmaven.test.skip=true
```

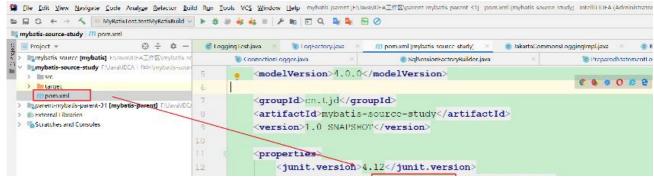
如果出现如下错误:

```
Tests are skipped.
INFO
              - maven-pdf-plugin:1.3:pdf (pdf) @ mybatis ---
INF01
INFO] Skipped report generation.
INFO]
INFO]
         BUILD FAILURE
[INFO]
INFO]
         Total time: 25.626 s
INFO] Finished at: 2018-03-30T09:28:19+08:00
INFO] Final Memory: 44M/577M
INF0]
ERROR] Failed to execute goal org.apache.maven.plugins:maven-pdf-plugin:1.3:pdf (pdf) on project mybatis: Error du
generation: Error parsing H:\mybatis\mybatis-3-mybatis-3.4.6\target\pdf\site.tmp\xdoc\getting-started.xml: Error
model: only whitespace content allowed before start tag and not \ufeff (position: COMMENT seen ...rning permission:
limitations under the License.\n\n-->\n\ufeff... @18:2) -> [Help 1]
ERROR]
           To see the full stack trace of the errors, re-run Maven with the -e switch. Re-run Maven using the -X switch to enable full debug logging.
ERROR]
ERROR
           For more information about the errors and possible solutions, please read the following articles:
```

打开pom.xml 文件注释掉 maven-pdf-plugin 插件:

第三步: 修改测试项目中pom文件依赖的MyBatis版本

我们需要将测试项目中pom依赖的MyBatis的版本修改为刚刚安装至本地仓库的版本:



第四步:运行测试类,看是否能够执行成功

第一次执行肯定会遇到很多问题,例如博主就遇到了缺少包的问题:

```
    MyBatisTest.testMyBatisBuild

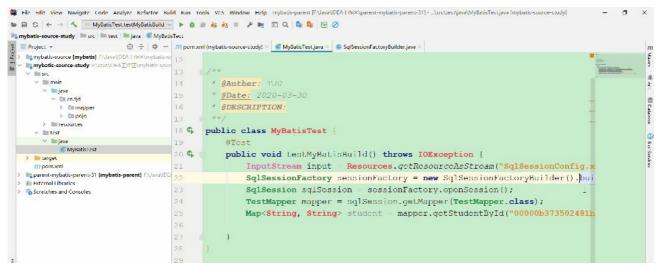
▶ ∨ Ø 15 17 E
                           9 10sts failed: 1 of 1 test 138.m
    ∨ ⑤ MyGatisTest 133ms D:\]AVA_Development\jdk_1.8\bin\java.exe ...
            testfylyBatisBi 13
0
                              org.apache.ibatis.exceptions.PersistenceException:
                                                                                                                                                                                                         48
                              ### Error building SqlSession
                              ### Cause: java.lang.IllegalStateException: Cannot enable lazy loading because Javassist is not available. Add Javassist to your
                                    at org.apache.ibatis.exceptions.ExceptionFactory.wrapException(ExceptionFactory.java:30) at org.apache.ibatis.session.SqlSessionFactoryBuilder.build(SqlSessionFactoryBuilder.jav
                                                                                                                                                                                                         =
                                                                                                                                                                                                         :4
                                   at org.apache.ibatis.session.SqlSessionFactoryBuilder.build(SqlSessionFactoryBuilder.java:64) at MyBatisTest.testMyBatisBuild(MyBatisTest.java:22) <19 internal calls> at com.intellij.rt.junit.IdeaTestRunner$Repeater.startRunnerWithArgs(IdeaTestRunner.java:33)
=
                                                                                                                                                                                                         0
```

我们在pom文件中加入:

具体加坐标需要去中央仓库搜索。

再次执行,会发现依然缺少依赖,按照相同步骤解决即可。

把所有的问题解决后,我们通过debug可以发现能够进入到mybatis源码项目中去,大功告成!







印象笔记, 让记忆永存

服务条款 隐私政策