


(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

 MyBatis

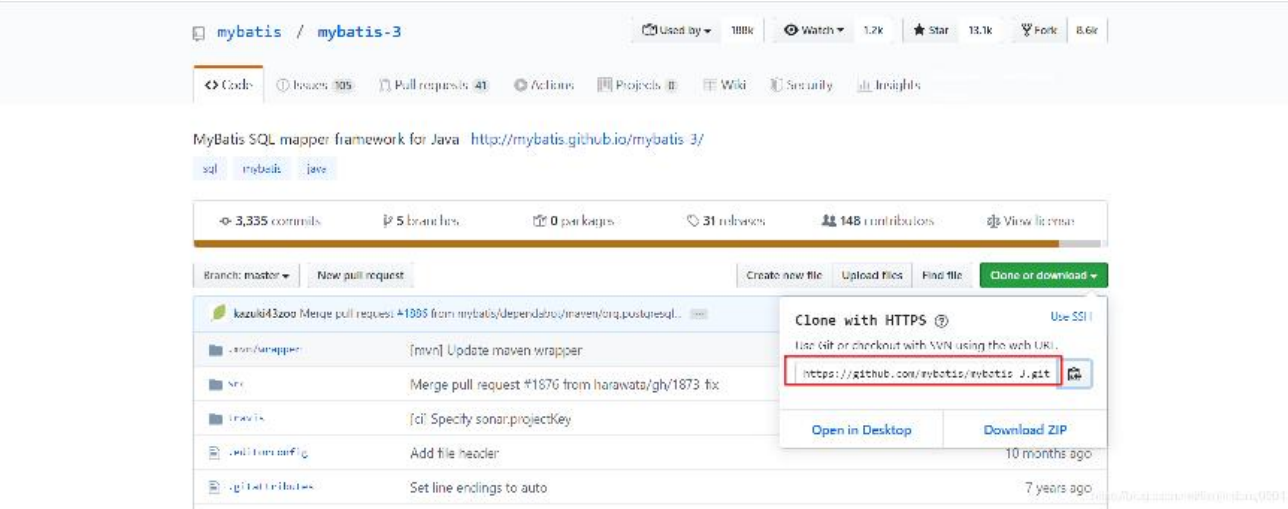
同时被 2 个专栏收录

1 订阅 12 篇文章

订阅专栏

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：

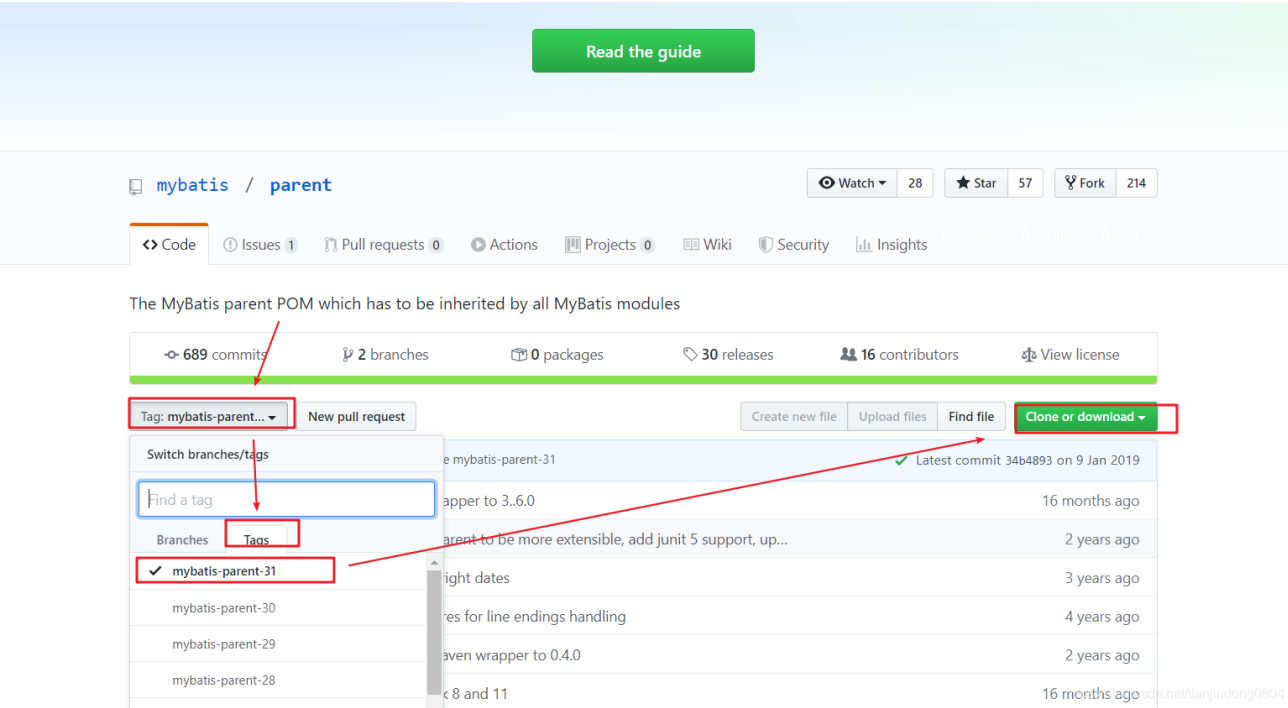
```
1 | git clone https://github.com/mybatis/mybatis-3.git mybatis-source
```

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 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
5     <modelVersion>4.0.0</modelVersion>
6
7     <groupId>cn.tjd</groupId>
8     <artifactId>mybatis-source-study</artifactId>
9     <version>1.0-SNAPSHOT</version>
10
11     <properties>
12         <junit.version>4.12</junit.version>
13         <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     <!-- Log4j -->
19     <dependency>
20         <groupId>log4j</groupId>
21         <artifactId>log4j</artifactId>
22         <version>1.2.17</version>
23     </dependency>
24     <!-- Mybatis -->
25     <dependency>
26         <groupId>org.mybatis</groupId>
27         <artifactId>mybatis</artifactId>
28         <version>${mybatis.version}</version>
29     </dependency>
30     <!-- MySQL -->
31     <dependency>
32         <groupId>mysql</groupId>
33         <artifactId>mysql-connector-java</artifactId>
34         <version>${mysql.version}</version>
35     </dependency>
36     <!-- 连接池 -->
37     <dependency>
38         <groupId>com.alibaba</groupId>
39         <artifactId>druid</artifactId>
40         <version>${druid.version}</version>
41     </dependency>
42     <!-- Junit -->
43     <dependency>
44         <groupId>junit</groupId>
45         <artifactId>junit</artifactId>
46         <version>${junit.version}</version>
47         <scope>test</scope>
48     </dependency>
49 </dependencies>
50
51 <build>
52     <!-- 加载配置文件 -->
53     <resources>
54         <resource>
55             <directory>src/main/java</directory>
56             <includes>
57                 <include>/**/*.xml</include>
58             </includes>
59         </resource>
60     </resources>
```

```
60 </resources>
61 </build>
62
63 </project>
64
```

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：

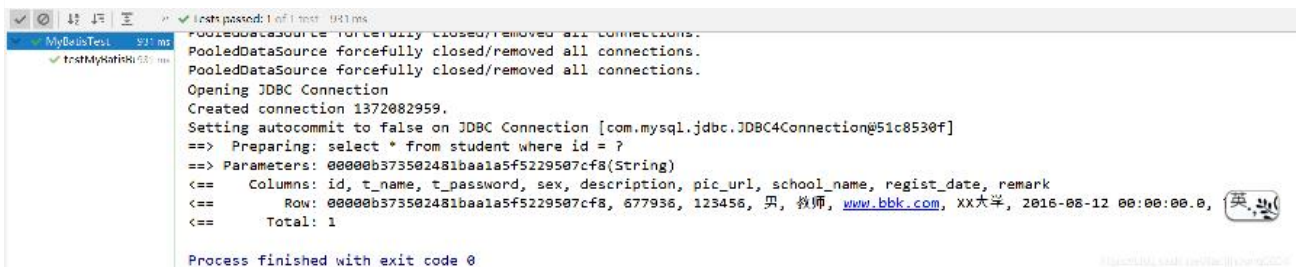
```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE configuration
3     PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
4     "http://mybatis.org/dtd/mybatis-3-config.dtd">
5 <configuration>
6     <!-- 开启日志输出 -->
7     <settings>
8         <setting name="logImpl" value="STDOUT_LOGGING" />
9     </settings>
10    <!-- 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Mapl
11    <typeAliases>
12        <package name="cn.tjd.pojo" />
13    </typeAliases>
14    <environments default="development">
15        <environment id="development">
16            <transactionManager type="JDBC"></transactionManager>
17            <dataSource type="POOLED">
18                <property name="driver" value="com.mysql.jdbc.Driver" />
19                <property name="username" value="root" />
20                <property name="password" value="123456" />
21                <property name="url" value="jdbc:mysql://localhost:3306/test?
22            </dataSource>
16    </environment>
```

```
23     </environments>
24     <mappers>
25         <package name="cn.tjd.mapper"/>
26     </mappers>
27
28 </configuration>
29
```

## 第四步：编写测试类测试

```
1 public class MyBatisTest {
2     @Test
3     public void testMyBatisBuild() throws IOException {
4         InputStream input = Resources.getResourceAsStream("SqlSessionConfig.xml");
5         SqlSessionFactory sessionFactory = new SqlSessionFactoryBuilder().build(input);
6         SqlSession sqlSession = sessionFactory.openSession();
7         TestMapper mapper = sqlSession.getMapper(TestMapper.class);
8         Map<String, String> student = mapper.getStudentById("00000b373502481b");
9     }
10 }
```

测试成功即可进行下一步。

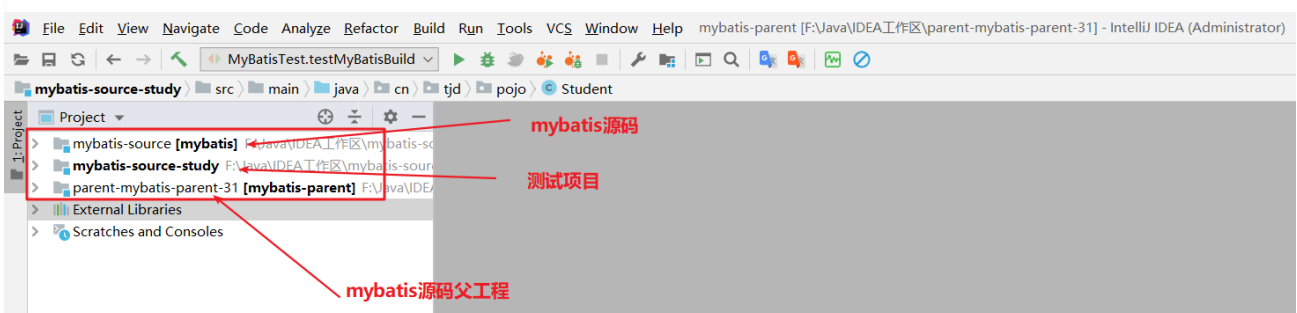


注：之所以要在源码整合之前搭建好测试环境，是为了保证测试环境的可靠，因为后序的源码整合可能会发生一些问题，这样可以排除是因为测试环境错误所导致的。

## 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：

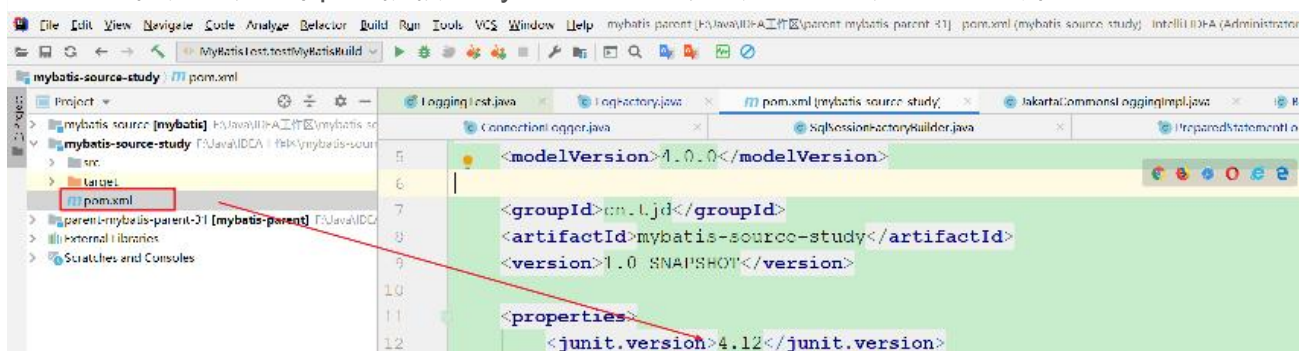
```
[INFO] Tests are skipped.
[INFO]
[INFO] --- maven-pdf-plugin:1.3:pdf (pdf) @ mybatis ---
[INFO] Skipped report generation.
[INFO]
[INFO] BUILD FAILURE
[INFO]
[INFO] Total time: 25.626 s
[INFO] Finished at: 2018-03-30T09:28:19+08:00
[INFO] Final Memory: 44M/577M
[INFO]
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-pdf-plugin:1.3:pdf (pdf) on project mybatis: Error d
t 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 \uffff (position: COMMENT seen ...rning permission
limitations under the License.\n\n-->\n\u00ff... @18:2) -> [Help 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] 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:
https://blog.csdn.net/tianjindong0804
```

打开pom.xml 文件注释掉 maven-pdf-plugin 插件:

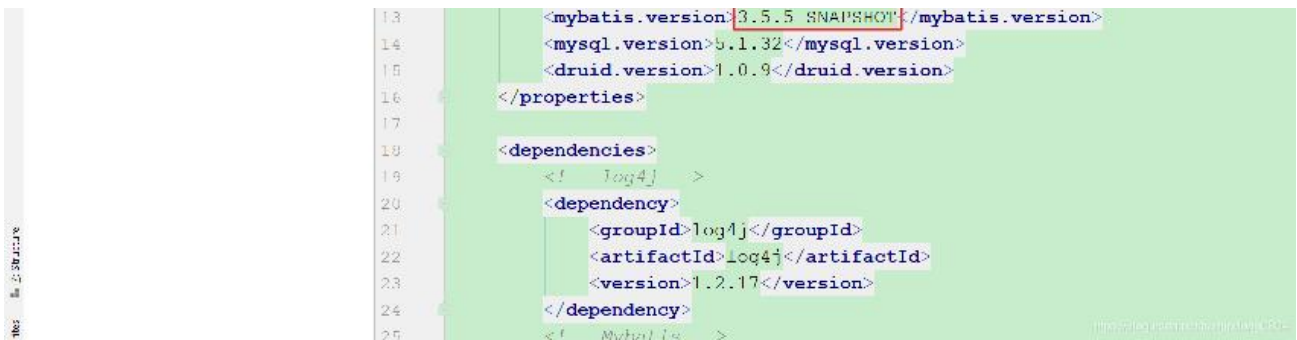
```
1 | <!--
2 |
3 | <plugin>
4 |   <groupId>org.apache.maven.plugins</groupId>
5 |   <artifactId>maven-pdf-plugin</artifactId>
6 | </plugin>
7 | -->
```

## 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：



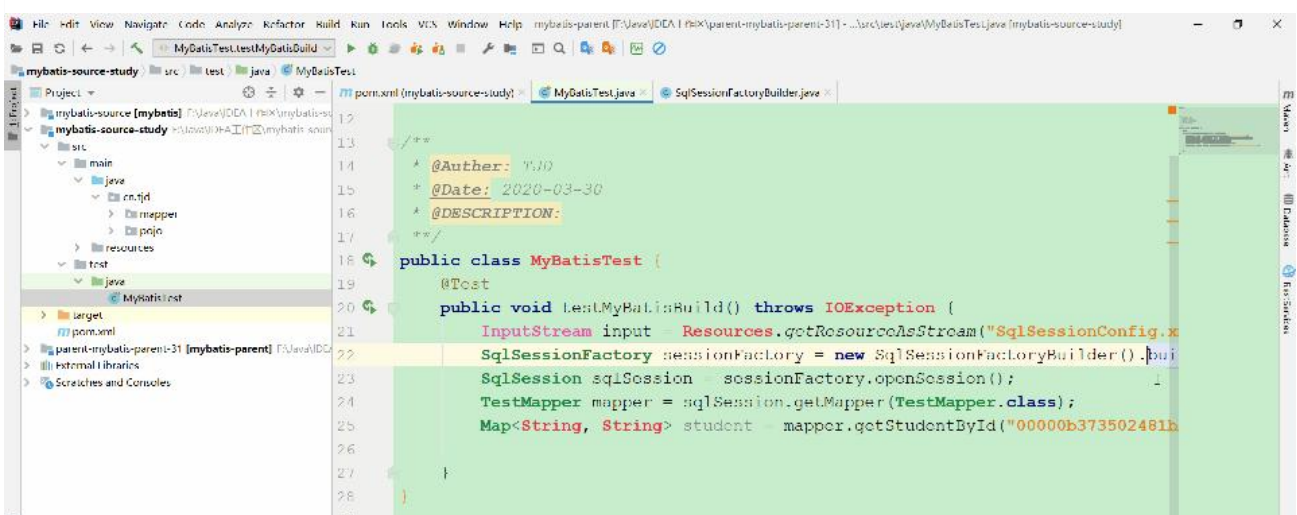
我们在pom文件中加入：

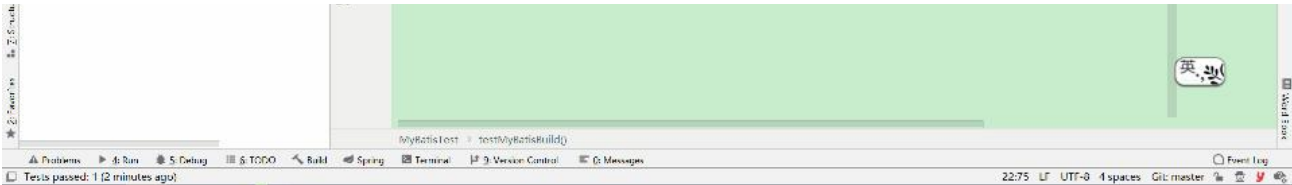


具体加坐标需要去[中央仓库](https://mvnrepository.com/artifact/org.javassist/javassist)搜索。

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