**Mybatis-plus-taste**

发现自己一直逛的超市也叫taste，巧

这里作为taste过程记录的新位置，在这里直接贴图，贴代码都会比较方便，希望能够节省出更多的时间。

# **201903151053**

跑测试用例的时候，如果发现有相关Statement找不到的情况，可以检查一下target相应目录下面是否有XXXMapper.xml.若不存在，则需要手动复制一份对应的mapper.xml配置过去，才可以。

# **201903151345**

发现一个问题，将数据库连接相关的配置存放到属性文件中，mybatis-config.xml用properties标签指定了属性文件配置，调试跟踪源码之后，发现也确实有去读取属性文件内容，但是配置连接池信息处，引用了属性文件(jdbc.properties)中的属性，却没有得到正确的转换，比如driver，它在配置DataSource对象的时候，拿到对应driver的值变成了“${jdbc.driver}”，而不是”com.mysql.jdbc.Driver”。

于是来了兴趣，找了mybatis配置DataSource对象的相关源码，它也没有去读出properties文件内的key，直接将字面量赋值到DataSource中的相应属性。这会是什么原因？难道是先后顺序吗?

# **201903152330**

还是上面说的问题，对DataSource类成员变量设置值的代码有翻到了，觉得mybatis中实现中并没做相关处理——从属性文件中获取同名属性

# **201903171036**

版本号：d5f17395ca91e37d2aeba03144090b272e6d5051

现在在测试分页插件.测试代码挺简单的，测试结果也十分准确。

这次测试的背后知识点是：Mybatis下RowBounds分页实现,在分页拦截器中禁用RowBounds内存分页，使用拼接后带分页语法的sql进行查询。

有关RowBounds，我引用别人的文章过来提供参考。

【原文地址】https://blog.csdn.net/u010077905/article/details/38469653

RowBounds类，用于实现分页查询。RowBounds中有两个属性，offset和limit。



RowBounds在处理分页时，只是简单的把offset之前的数据都skip掉，超过limit之后的数据不取出(内存分页)，上图中的代码取自MyBatis中的DefaultResultSetHandler类。跳过offset之前的数据是由方法skipRows处理，判断数据是否超过了limit则是由shouldProcessMoreRows方法进行判断。简单点说道，就是先把数据全部查询到ResultSet，然后从ResultSet中取出offset和limit之间的数据，这就实现了分页查询。

小结论：内存分页是从数据库中查出所有数据后，mybatis框架排除掉范围外数据。从性能上来说，这种分页方式不是很好。

**201903171246**

分页插件相关的开发以及测试程序都已经完整了，接下来分支版本是补全test用例，同步一下官方的Readme,提交一次git.

完成了上面的事情，那么框架的版本号也跟着推进到 4b6a7d4d9e08e1c987cfec73d4d78f593a3d28a0