**mybatis源码阅读**

* 通过mybatis做一次查询内部的流程
  + 1.加载配置文件，得到输入流
  + 2.创建SqlsessionFactory
    - 1.通过SqlsessionFactoryBuilder类创建，类中存在多个build的重载方法，的最主要的两个方法一个是通过传入Reader一个是通过传入InputStrem作为参数传入到XMLConfigBuilder来解析配置文件的。
    - 2.XMLConfigBuilder通过传入的出传入的IO（Reader,InputStream）对象解析生成一个Configuration对象
      * XMLConfigBuilder对象：通过XPathParser来解析流存放的xml数据
        + 1.解析流程

1.创建XPathParser对象：调用公共构造函数初始化EntityResolver及其他成员变量和xpath

2.调用createDocument方法创建document对象，通过初始化的EntityResolver类型对象（实际是实现了EntityResolver接口的XMLMapperEntityResolver对象）实现xml约束文件本地化，不需要到互联网上取DTD文件了。EntityResolver类型对象中存放了外部的dtd地址和本地dtd的映射关系，所以即使使用的外部地址，也会被替换为使用本地的dtd.

3.XMLConfigBuilder通过步骤1,2获取到parser进行具体的解析。如果已经解析过了就会抛出异常报错。根节点为configuration，其节点下的其他子节点分别都有一个方法对应解析，具体每个方法的解析还有待进一步研究分析。解析的结果全部都封装到了Configration对象中。

1.解析mappers（具体这样的解析逻辑，等把整体框架看完之后再回头研究，留个位置）

2.解析环境environments

* + - 3.在通过2中创建的Configuration对象调用最后一个重载的build方法创建一个DefaultSqlSessionFactory对象返回
    - 4.主要代码截图：
  + 3.创建Sqlsession
    - 1.通过2中返回的DefaultSqlSessionFactory工厂对象调用opensession()方法获取Sqlsession对象。DefaultSqlSessionFactory类中存在多个opensession()重载方法，但是最后都会汇总到调用openSessionFromDataSource，openSessionFromConnection这两个其中一个方法来返回一个DefaultSqlSession
  + 4.通过Sqlsession执行Mappper中定义的sql

4.主要代码截图：





