1. **SqlSessionFactoryBuilder.builder(inputStream)**

1传入一个输入流，构建一个XMLConfigBuilder对象，调用parse()对配置文件进行解析

2）XMLConfigBuilder. parse()方法里，调用parseConfiguration(),指定解析‘configuration‘标签里的内容

3）parseConfiguration (XNode root)里，有指定读取了’properties’,’ typeAliases’,’ mappers’，‘databaseIdProvider ‘等标签里的内容

4）XMLConfigBuilder.mapperElement(root.evalNode("mappers"));指定解析mappers标签,解析mappers，若配置了mapper则，继续去解析XXMapper.xml后，

5)解析XXMapper.xml入口在XMLConfigBuilder.mapperElement(root.evalNode("mappers"));

6）XMLConfigBuilder.mapperElement(root.evalNode("mappers"));

中调用XMLMapperBuilder.parse()，

1. Parse()里面调用XMLMapperBuilder.bindMapperForNamespace()

.bindMapperForNamespace()里调用configration.addMapper将么mapper的代理工厂对象添加到configration.knownMappers的map中

1. Parse()里面调用configurationElement(this.parser.evalNode("/mapper"));去解析mapper.xml, buildStatementFromContext()里创建XMLStatementBuilder，掉哦那个parseStatementNode，将解析每一段配置，并且将其放入mappedStatment里保存到

SqlSessionFactoryBuilder.builder(inputStream) 整体就是读取配置文件，将读取到的数据配置在一个全局变量Configuration上，然后生成一个sqlSessionFactory.

parsePendingResultMaps是干了什么

接下来应该去看解析这些标签之后，相应的对象存放在Configuration里的哪个属性

1. **SqlSessionFactory .openSession();**

通过全据变量获取到environment，在通过environment获取到事物对象，通过事务对象、执行器类型创建Executor对象，通过configration和executor创建一个defaultSqlsession（内涵一个configration，executor和名为dirty的boolean值）

1. **session.selectList("com.entity.User.selectAll");**

1）Default.selectList()通过（包名+id）这个唯一标识来获取到configuration. mappedStatements中的MappedStatement对象。（mappedStatements是个Map<String，MappedStatement>）

2）Executor. Query（）获取到MappedStatement后，将MappedStatement，parameter交由executor去执行

3）sqlsession做的事情，通过（包名+id）获取到MappedStatement（从mappedStatements这个hashMap中去，这个map用包名+id作为key，value为MappedStatement），并且解析parameter，然后将MappedStatement，parameter交由Executor去执行

4）SimpleExecutor.query()通过parameter，获取到MappedStatement的BoundSql，然后创建一个CacheKey，在调用query去查询

）SimpleExecutor.query() 去查询，**这里还可以去看下去，比如获取Connection，Preparement，ResultHandler等东西**

**此时可以去看一下MappedStatement，有哪些属性，是怎么生成的**

1. 阿斯顿
2. Asdasasdads
3. 十大