常用类

Resources ：获取资源的类，方便把类路径或网络的文件的流

-》也可以使用Class类的对应方法代替

SqlSessionFactoryBuilder：用来创建SqlSessionFactory 的类

SqlSessionFactory ：用来创建SqlSession的类

SqlSession：mybatis操作sql的入口类（增加，修改，删除，查询，事务...）

获取Sqlsession的步骤（单用mybatis，如果集成了spring，方式不一样）

String resource = "org/mybatis/builder/mybatis-config.xml";

InputStream inputStream = Resources.getResourceAsStream(resource);

SqlSessionFactoryBuilder builder = new SqlSessionFactoryBuilder();

SqlSessionFactory factory = builder.build(inputStream);

**获取SqlSession**

factory.openSession() ：获取开启事务的sessoin，需求手动commit和rollback操作事务

factory.openSession(boolean autoCommit)：false手动提交（会开启事务），true则自动提交。

factory.openSession(ExecutorType execType)

--------------------

**ExecutorType.SIMPLE: 这个执行器类型不做特殊的事情。它为每个语句的执行创建一个新的预处理语句Statement。**

**ExecutorType.REUSE: 这个执行器类型会【复用】预处理语句Statement。**

**ExecutorType.BATCH: 这个执行器会批量执行所有更新语句。果 SELECT 在它们中间执行还会标定它们是 必须的,来保证一个简单并易于理解的行为。**

Sqlsession的操作

**----------------基本的-----------**

T> T selectOne(String statement, Object parameter) 查询一条数据，没有数据或多余一条数据报错。

<E> List<E> selectList(String statement, Object parameter) 查询0到多条数据

int insert(String statement, Object parameter) 插入数据

int update(String statement, Object parameter) 更新数据

int delete(String statement, Object parameter) 删除数据

<E> List<E> selectList (String statement, Object parameter, RowBounds rowBounds) 查询0到多条数据，带分页的。

**----------------事务处理-----------**

commit 提交事务

rollback 回滚事务

**----------------扩展的----------**

void select (String statement, Object parameter, ResultHandler<T> handler)

查询，手动把结果数据进行处理（一般不需要，因为mybatis已经处理的很好了-通过resultmap等）

---批量执行

a）获取带批量执行功能的session，factory.openSession(ExecutorType.BATCH:)

b）用session批量执行代码（insert或update...）

c）刷新批量语句flushStatements()