## 普通批量操作

在获取sqlSession时，指定为批量操作，之后批量操作时，被执行的sql只会被预编译1次，会被传参数n次，数据库提交1次

SqlSessionFactory sqlSessionFactory = getSqlSessionFactory();

// 设置执行方式为批量操作

SqlSession openSession = sqlSessionFactory.

openSession(ExecutorType.***BATCH***);

EmployeeMapper mapper = openSession.

getMapper(EmployeeMapper.**class**);

**for** (**int** i = 0; i < 10000; i++) {

String id = UUID.*randomUUID*().toString().replace("-", "");

mapper.addEmp(**new** Employee(id.substring(0, 5), "0", "a"));

}

openSession.commit();

openSession.close();

## 整合spring批量操作

1. 在Spring配置文件中专门配置一个sqlSession用来批量处理

<!-- 配置专门批量处理的sqlSession对象 -->

<bean id=*"batchSqlSession"* class=*"org.mybatis.spring.SqlSessionTemplate"*>

<constructor-arg name=*"sqlSessionFactory"* ref=*"sqlSessionFactory"*></constructor-arg>

<constructor-arg name=*"executorType"* value=*"BATCH"*></constructor-arg>

</bean>

1. 在service设置自动装配

@Autowired

**private** SqlSession batchSqlSession;

1. 使用

EmployeeDao mapper = batchSqlSession.getMapper(EmployeeDao.

**class**);

// 执行批量操作

//mapper.XXX