5、Mybatis独立测试

一、编写持久层AcoountDao的映射文件AccountDao.xml

注意：映射文件要放在resources里，与src包中accountdao.java相同的路径

id要和方法名相同

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper
    PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.qiu.dao.IAccountDao">

<select id="findAll" resultType="com.qiu.domain.Account">
    select * from account
</select>

<insert id="saveAccount" parameterType="com.qiu.domain.Account">
    insert into account(name,money) values(#{name},#{money});
</insert>

</mapper>
```

二、编写 Mybatis配置文件SqlMapConfig .xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE configuration
PUBLIC "-//mybatis.org//DTD Config 3.0//EN"  "http://mybatis.org/dtd/mybatis-3-config.dtd">
<configuration>
  <properties resource="jdbcConfig.properties"></properties>

  <environments default="mysql">
    <environment id="mysql">
      <transactionManager type="JDBC"></transactionManager>
      <dataSource type="pooled">
        <property name="driver" value="${jdbc.driver}"/>
        <property name="url" value="${jdbc.url}"/>
        <property name="username" value="${jdbc.username}"/>
        <property name="password" value="${jdbc.password}"/>
      </dataSource>
```

```
        </environment>
    </environments>

    <mappers>
        <mapper resource="com/itheima/dao/AccountDao.xml"/>
    </mappers>
</configuration>
```

三、编写jdbcConfig.properties配置文件

```
jdbc.driver=com.mysql.cj.jdbc.Driver
jdbc.url=jdbc:mysql://127.0.0.1:3306/ssm?
useUnicode=true&characterEncoding=utf8&serverTimezone=GMT
jdbc.username=root
jdbc.password=qrj15521026074
```

四、测试

```
public class Test02MyBatis {
    @Test
    public void testFindAll() throws Exception{
        InputStream in = Resources.getResourceAsStream("SqlMapConfig.xml");
        SqlSessionFactory factory = new SqlSessionFactoryBuilder().build(in);
        SqlSession session= factory.openSession();
        IAccountDao aDao = session.getMapper(IAccountDao.class);
        List<Account> list = aDao.findAll();
        System.out.println(list);
        session.close();
        in.close();
        }

    @Test
    public void testSave() throws Exception {
        Account account = new Account();
        account.setName("test");
        account.setMoney(5000f);
        InputStream in = Resources.getResourceAsStream("SqlMapConfig.xml");
        SqlSessionFactory factory = new SqlSessionFactoryBuilder().build(in);
        SqlSession session= factory.openSession();
        IAccountDao aDao = session.getMapper(IAccountDao.class);
        aDao.save(account);
        session.commit();
        session.close();
        in.close();
    }
}
```