■ JUnit Test

-서비스 테스트하기

-DAO 테스트하기

```
@Component
public class TestService {

    @Autowired
    TestDAO dao

    public  ArrayList<String> test() {
        return dao.test();
    }

}

@Component
public class TestDAO {
    @Autowired
    DataSource ds

    public void select() {
        String sql ="select * from member_tbl_11"
        try {
            Connection con  =ds.getConnection();
            PreparedStatement  pst  = con.prepareStatement(sql);

        ResultSet rs  = pst.executeQuery();
        while( rs.next()) {
            System.out.print( rs.getString(1));
            System.out.print( rs.getString(2));
            System.out.print( rs.getString(3));
            System.out.println( rs.getString(4));
        }
           rs.close();
           pst.close();
           con.close();

        } catch (SQLException e) {
           e.printStackTrace();
        }

    }

    //테스트를 위한 매서드 작성
    public ArrayList<String> test() {

        ArrayList<String> testList = new ArrayList<>();
        testList.add("테스트항목1");
        testList.add("테스트항목2");

        // return null;   //  테스트시 실패
        return testList;  // 테스트시 성공
    }

}
```

■ TestService서비스 테스트하기

1. TestService 오른쪽 마우스클릭 -〉 new - 〉 Junit Test case -〉 Finish
2. @Test 로 test매서드 작성
   assert매서드 작성
3. Spring dependency 의존성 추가
4. 테스트를 위한 설정xml 작성하기

◎ Spring Test dependency

```
〈dependency〉
        〈groupId〉org.springframework〈/groupId〉
        〈artifactId〉spring-test〈/artifactId〉
        〈version〉${org.springframework-version}〈/version〉
〈/dependency〉

 〈dependency〉
         〈groupId〉junit〈/groupId〉
         〈artifactId〉junit〈/artifactId〉
         〈version〉4.12〈/version〉
 〈/dependency〉
```

```
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(locations = {"file:src/main/webapp/WEB-INF/spring/**/test.xml" , "file:src/main/webapp/WEB-INF/spring/**/test2.xml"} )
```

■ TestService 테스트 하기

```
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(locations = {"file:src/main/webapp/WEB-INF/spring/**/test.xml" ,
"file:src/main/webapp/WEB-INF/spring/**/test2.xml"} )

public class TestServiceTest {

        @Autowired
        TestService service

        @Test
        public void testTest() {
                assertTrue(service.test() != null);      //결과값이  null이 아니라면 성공
        }

}
```

■ TestDAO 테스트 하기

```
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(locations =
{"file:src/main/webapp/WEB-INF/spring/**/test.xml" , "file:src/main/webapp/WEB-INF/spring/**/test2.xml"} )
public class TestDAOTest {


   @Autowired
   TestDAO dao;

          @Test
          public void testSelect() throws SQLException {
          //         ApplicationContext  ac =
          //              new GenericXmlApplicationContext("file:src/main/webapp/WEB-INF/spring/**/test.xml" ,
          //                   "file:src/main/webapp/WEB-INF/spring/**/test2.xml");
          //         TestDAO dao  =ac.getBean(TestDAO.class);
                     assertTrue (dao.test() != null);
          }
}
```

test.xml

```
<?xml version="1.0" encoding="UTF-8"
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
https://www.springframework.org/schema/beans/spring-beans.xsd">

<!-- Root Context: defines shared resources visible to all other web components -->



        <bean id="d" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
                <property name="driverClassName" value="oracle.jdbc.driver.OracleDriver"></property>
                <property name="url" value="jdbc:oracle:thin:@localhost:1521:xe"></property>
                <property name="username" value="System"></property>
                <property name="password" value="1234"></property>
        </bean>



         <!--
        <bean id="dao"  class="com.acorn.db.TestDAO"></bean>
        <bean id="service"  class="com.acorn.db.TestService">
        <property name="dao" ref="dao"></property>
        </bean>
        -->

        <bean id="calculator"  class="com.acorn.db.di.Calculator"></bean>
        <bean id="program"  class="com.acorn.db.di.Program">
                <property name="calculator" ref="calculator"></property>
        </bean>

</beans>
```

test2.xml

```
<?xml version="1.0" encoding="UTF-8"
<beans:beans xmlns="http://www.springframework.org/schema/mvc"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:beans="http://www.springframework.org/schema/beans"
xmlns:context="http://www.springframework.org/schema/context"
xsi:schemaLocation="http://www.springframework.org/schema/mvc
https://www.springframework.org/schema/mvc/spring-mvc.xsd
http://www.springframework.org/schema/beans
https://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context
https://www.springframework.org/schema/context/spring-context.xsd">

<!-- DispatcherServlet Context: defines this servlet's request-processing infrastructure -->


<context:component-scan base-package="com.acorn.db" />


</beans:beans>
```