**Sample Source that Uses both EasyMock and Untils**

The following is the entire sample source that uses both EasyMock and Unitils.

**Test Class**

package egovframework.guideprogram.test.testcase.mock;

import static org.easymock.EasyMock.expect;

import static org.junit.Assert.assertNotNull;

import static org.unitils.reflectionassert.ReflectionAssert.assertPropertyLenientEquals;

import java.util.Arrays;

import java.util.List;

import org.junit.Before;

import org.junit.Test;

import org.junit.runner.RunWith;

import org.unitils.UnitilsJUnit4TestClassRunner;

import org.unitils.easymock.EasyMockUnitils;

import org.unitils.easymock.annotation.Mock;

import org.unitils.inject.annotation.InjectIntoByType;

import org.unitils.inject.annotation.TestedObject;

import egovframework.guideprogram.test.target.mock.BoardVO;

import egovframework.guideprogram.test.target.mock.CollaboratorDao;

import egovframework.guideprogram.test.target.mock.EasyMockService;

@RunWith(UnitilsJUnit4TestClassRunner.class)

public class EasyMockTest {

/\*\*

\* Mock Object for EgovDao

\* @Mock : create Mock

\* @InjectIntoByType : Dao is injected into Service

\*/

@Mock

@InjectIntoByType

private CollaboratorDao mockDao;

/\*\*

\* Service for testing

\* Target to which the Mock object is injected

\*/

@TestedObject

private EasyMockService service;

/\*\*

\* Prepare test fixture

\*/

@Before

public void setUp() {

service = new EasyMockService();

}

/\*\*

\* Example showing that the method to mock has the return value

\*/

@Test

public void testSelectList() throws [Exception](http://www.google.com/search?hl=en&q=allinurl%3Aexception+java.sun.com&btnI=I%27m%20Feeling%20Lucky) {

assertNotNull("Check if the test target has properly been produced", service);

assertNotNull("Check if the Mock Object has properly been produced", mockDao);

// Use EasyMock + Unitils to add and verify the method to mock.

expect(mockDao.selectList()).andReturn([Arrays](http://www.google.com/search?hl=en&q=allinurl%3Aarrays+java.sun.com&btnI=I%27m%20Feeling%20Lucky).asList(new BoardVO(101), new BoardVO(102)));

EasyMockUnitils.replay();

// Execute the test target method.

List<BoardVO> selectList = service.selectList();

// Check the results.

assertNotNull("Check if the test target has properly been produced", selectList);

assertPropertyLenientEquals("id", [Arrays](http://www.google.com/search?hl=en&q=allinurl%3Aarrays+java.sun.com&btnI=I%27m%20Feeling%20Lucky).asList(101, 102), selectList);

}

/\*\*

\* Example showing that the method to mock has the return value

\*/

@Test

public void testInsert() throws [Exception](http://www.google.com/search?hl=en&q=allinurl%3Aexception+java.sun.com&btnI=I%27m%20Feeling%20Lucky) {

assertNotNull("Check if the test target has properly been produced", service);

assertNotNull("Check if the Mock Object has properly been produced", mockDao);

BoardVO board = new BoardVO(103);

// Use EasyMock + Unitils to add and verify the method to mock.

mockDao.insert(board);

EasyMockUnitils.replay();

// Execute the test target method.

service.insert(board);

}

}

**Target Class**

Test target class(EasyMockService), associate object(CollaboratorDao) and VO class.

package egovframework.guideprogram.test.target.mock;

import java.util.List;

import javax.annotation.Resource;

import org.springframework.stereotype.Service;

@Service("easyMockService")

public class EasyMockService {

@Resource(name="collaboratorDao")

private CollaboratorDao collaboratorDao;

public List<BoardVO> selectList() throws [Exception](http://www.google.com/search?hl=en&q=allinurl%3Aexception+java.sun.com&btnI=I%27m%20Feeling%20Lucky) {

List<BoardVO> result = collaboratorDao.selectList();

return result;

}

public void insert(BoardVO vo) throws [Exception](http://www.google.com/search?hl=en&q=allinurl%3Aexception+java.sun.com&btnI=I%27m%20Feeling%20Lucky) {

collaboratorDao.insert(vo);

}

}

package egovframework.guideprogram.test.target.mock;

import java.util.List;

public interface CollaboratorDao {

List<BoardVO> selectList();

void insert(BoardVO vo);

}

package egovframework.guideprogram.test.target.mock;

public class BoardVO {

/\*\* id \*/

private int id;

/\*\* userName \*/

private [String](http://www.google.com/search?hl=en&q=allinurl%3Astring+java.sun.com&btnI=I%27m%20Feeling%20Lucky) userName;

/\*\* content \*/

private [String](http://www.google.com/search?hl=en&q=allinurl%3Astring+java.sun.com&btnI=I%27m%20Feeling%20Lucky) content;

/\*\*

\* Constructor

\*/

public BoardVO() {

super();

}

/\*\*

\* Constructor

\*

\* @param id

\*/

public BoardVO(int id) {

super();

this.id = id;

}

/\*\*

\* Constructor

\*

\* @param id

\* @param userName

\* @param content

\*/

public BoardVO(int id, [String](http://www.google.com/search?hl=en&q=allinurl%3Astring+java.sun.com&btnI=I%27m%20Feeling%20Lucky) guestName, [String](http://www.google.com/search?hl=en&q=allinurl%3Astring+java.sun.com&btnI=I%27m%20Feeling%20Lucky) content) {

super();

this.id = id;

this.userName = guestName;

this.content = content;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public [String](http://www.google.com/search?hl=en&q=allinurl%3Astring+java.sun.com&btnI=I%27m%20Feeling%20Lucky) getUserName() {

return userName;

}

public void setUserName([String](http://www.google.com/search?hl=en&q=allinurl%3Astring+java.sun.com&btnI=I%27m%20Feeling%20Lucky) guestName) {

this.userName = guestName;

}

public [String](http://www.google.com/search?hl=en&q=allinurl%3Astring+java.sun.com&btnI=I%27m%20Feeling%20Lucky) getContent() {

return content;

}

public void setContent([String](http://www.google.com/search?hl=en&q=allinurl%3Astring+java.sun.com&btnI=I%27m%20Feeling%20Lucky) content) {

this.content = content;

}

}