EASYMOCK - EASYMOCKSUPPORT

EasyMockSupport is a utility or helper class for test classes. It provides the following functionalities:

- **replayAll** – Registers all the created mocks in one batch.
- **verifyAll** – Verifies all the mock operations in one batch.
- **resetAll** – Resets all the mock operations in one batch.

Example

Step 1: Create an interface called CalculatorService to provide mathematical functions

File: CalculatorService.java

```
public interface CalculatorService {
   public double add(double input1, double input2);
   public double subtract(double input1, double input2);
   public double multiply(double input1, double input2);
   public double divide(double input1, double input2);
}
```

Step 2: Create a JAVA class to represent MathApplication

File: MathApplication.java

```
public class MathApplication {
   private CalculatorService calcService;

   public void setCalculatorService(CalculatorService calcService){
      this.calcService = calcService;
   }

   public double add(double input1, double input2){
      return calcService.add(input1, input2);
   }

   public double subtract(double input1, double input2){
      return calcService.subtract(input1, input2);
   }

   public double multiply(double input1, double input2){
      return calcService.multiply(input1, input2);
   }

   public double divide(double input1, double input2){
      return calcService.divide(input1, input2);
   }
}
```

Step 3: Test the MathApplication class

Let's test the MathApplication class, by injecting in it a mock of calculatorService. Mock will be created by EasyMock.

File: MathApplicationTester.java

```
import org.easymock.EasyMockRunner;
import org.easymock.EasyMockSupport;

import org.junit.Assert;
import org.junit.Before;
import org.junit.Test;
```

```
import org.junit.runner.RunWith;

@RunWith(EasyMockRunner.class)
public class MathApplicationTester extends EasyMockSupport {

   private MathApplication mathApplication1;
   private MathApplication mathApplication2;

   private CalculatorService calcService1;
   private CalculatorService calcService2;

   @Before
   public void setUp(){
      mathApplication1 = new MathApplication();
      mathApplication2 = new MathApplication();
      calcService1 = createNiceMock(CalculatorService.class);
      calcService2 = createNiceMock(CalculatorService.class);
      mathApplication1.setCalculatorService(calcService1);
      mathApplication2.setCalculatorService(calcService2);
   }

   @Test
   public void testCalcService(){

      //activate all mocks
      replayAll();

      //test the add functionality
      Assert.assertEquals(mathApplication1.add(20.0, 10.0),0.0,0);

      //test the subtract functionality
      Assert.assertEquals(mathApplication1.subtract(20.0, 10.0),0.0,0);

      //test the multiply functionality
      Assert.assertEquals(mathApplication1.divide(20.0, 10.0),0.0,0);
      //test the divide functionality
      Assert.assertEquals(mathApplication1.multiply(20.0, 10.0),0.0,0);

      //test the add functionality
      Assert.assertEquals(mathApplication2.add(20.0, 10.0),0.0,0);

      //test the subtract functionality
      Assert.assertEquals(mathApplication2.subtract(20.0, 10.0),0.0,0);

      //test the multiply functionality
      Assert.assertEquals(mathApplication2.divide(20.0, 10.0),0.0,0);

      //test the divide functionality
      Assert.assertEquals(mathApplication2.multiply(20.0, 10.0),0.0,0);

      //verify all the mocks
      verifyAll();
   }
}
```

Step 4: Execute test cases

Create a java class file named TestRunner in **C:\> EasyMock_WORKSPACE** to execute Test cases .

File: TestRunner.java

```
import org.junit.runner.JUnitCore;
import org.junit.runner.Result;
import org.junit.runner.notification.Failure;

public class TestRunner {
   public static void main(String[] args) {
      Result result = JUnitCore.runClasses(MathApplicationTester.class);
```

```
      for (Failure failure : result.getFailures()) {
         System.out.println(failure.toString());
      }

      System.out.println(result.wasSuccessful());
   }
}
```

Step 5: Verify the Result

Compile the classes using javac compiler as follows:

```
C:\EasyMock_WORKSPACE>javac MathApplicationTester.java
```

Now run the Test Runner to see the result:

```
C:\EasyMock_WORKSPACE>java TestRunner
```

Verify the output.

```
true
```