

System.UnexpectedException No more than one executeBatch can be called from within a testmethod. Please make sure the iterable returned from your start method matches the batch size, resulting in one executeBatch invocation.

July 2, 2014 [bartoszbrowiec](#) [Leave a comment](#)

This exception occurs when *execute* method from salesforce batch is executed more than one time in unit test.

Please consider following code:

```
batch
1  global class POC_MultiStepException_Batch implements Database.BatchableInterface {
2      String query = 'SELECT id, Index__c from Batch_Step__c c';
3      global POC_MultiStepException_Batch() {}
4  }
5
6      global Database.QueryLocator start(Database.BatchableContext BC) {
7          return Database.getQueryLocator( query );
8      }
9
10     global void execute(Database.BatchableContext BC, List<SObject> batch) {}
11
12
13     global void finish(Database.BatchableContext BC) {}
14
15 }
```

```
test
1  @isTest
2  private class MultiStepsBatch_Test {
3
4      //this method should not throw an exeption
5      @isTest
6      static void MultiStepsBatchTestPass() {
7
8          Integer numberOfInteration = 1;
9          createData( numberOfInteration );
10
11          Test.startTest();
12
13          POC_MultiStepException_Batch pocMultiStepExceptionBatch = new POC_MultiStepException_Batch();
14          Database.executeBatch( pocMultiStepExceptionBatch );
15
16          Test.stopTest();
17          System.assert(true);
18      }
19
20      //this method will throw an exeption because in test method we call executeBatch more than once
21      //System.UnexpectedException: No more than one executeBatch can be called from within a test method. Please make sure the iterable returned from your start method matches the batch size, resulting in one executeBatch invocation.
22      @isTest
23      static void MultiStepsBatchTestFail() {
24
25          Integer numberOfInteration = 2;
26          createData( numberOfInteration );
27          try{
28
29              Test.startTest();
30
31              POC_MultiStepException_Batch pocMultiStepExceptionBatch = new POC_MultiStepException_Batch();
32              Database.executeBatch( pocMultiStepExceptionBatch );
33
34              Test.stopTest();
35
36              System.assert(false); // exception should be thrown in this line
37          } catch (System.UnexpectedException theException){ // here we catch the exception
38              System.assertEquals('No more than one executeBatch can be called from within a test method. Please make sure the iterable returned from your start method matches the batch size, resulting in one executeBatch invocation.', theException.getMessage());
39          }
40      }
41  }
42
43
44  private static void createData(Integer numberOfInteration) {
45      List<Batch_Step__c> stepCounts = new List<Batch_Step__c>();
46      for(Integer i = 0; i < numberOfInteration; i++){
47          Batch_Step__c theStepCount = new Batch_Step__c();
48          theStepCount.Index__c = i;
49          theStepCount.Name = 'Step ' + i;
50          stepCounts.add( theStepCount );
51      }
52      insert stepCounts;
53  }
54
55 }
```

As you can see **System.UnexpectedException** is thrown when **execute** method is executed more than one time. Compare MultiStepsBatchTestPass and MultiStepsBatchTestFail.

TIP!!
System.UnexpectedException is thrown when code execution reaches *Test.stopTest()*, but cannot be caught by *try catch* block.