global class LeadProcessor implements

Database.Batchable<sObject> {

// instance member to retain state across transactions

global Database.QueryLocator start(Database.BatchableContext bc) {

return Database.getQueryLocator(

'SELECT country=US,city=new York from lead'

);

}

global void execute(Database.BatchableContext bc, List<lead> scope){

// process each batch of records

list<lead> leads =new list<lead>();

for (lead lead : scope) {

lead.leadsource = 'web';

leads.add(lead);

}

update leads;

}

global void finish(Database.BatchableContext bc){

}

}

@isTest

private class LeadProcessorTest {

@testSetup

static void setup() {

List<lead> leads = new List<lead>();

// insert 10 accounts

for (Integer i=0;i<200;i++) {

leads.add(new lead(lastname='lead'+ i, company='test co', rating='hot'));

}

insert leads;

}

static testmethod void test() {

Test.startTest();

leadprocessor myBatchObject = new leadprocessor();

Id batchId = Database.executeBatch(myBatchObject);

Test.stopTest();

// after the testing stops, assert records were updated properly

System.assertEquals(200, [select count() from lead where leadsource = 'web']);

}

}