

Batch Apex Class

global class LeadProcessor implements

Database.Batchable<sObject>, Database.Stateful {

global Integer recordsProcessed = 0;

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

return Database.getQueryLocator('SELECT Id, LeadSource FROM Lead');

}

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

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

for (Lead lead : scope) {

lead.LeadSource = 'Dreamforce';

recordsProcessed = recordsProcessed + 1;

}

update leads;

}

global void finish(Database.BatchableContext bc){

System.debug(recordsProcessed + ' records processed. Shazam!');

}

}

Apex Test Class

@isTest

```
public class LeadProcessorTest {
```

```
    @testSetup
```

```
    static void setup() {
```

```
        List<Lead> leads = new List<Lead>();
```

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

```
            leads.add(new Lead(LastName='Lead '+i,
```

```
                Company='Lead', Status='Open - Not Contacted'));
```

```
        }
```

```
        insert leads;
```

```
    }
```

```
    static testmethod void test() {
```

```
        Test.startTest();
```

```
        LeadProcessor lp = new LeadProcessor();
```

```
        Id batchId = Database.executeBatch(lp, 200);
```

```
        Test.stopTest();
```

```
        System.assertEquals(200, [select count() from lead where LeadSource = 'Dreamforce']);
```

```
    }
```

```
}
```