

Rapport d'exécution de test - Couverture de code du backend Apex

- ❖ **NOMBRE DE TESTS EXÉCUTÉS : 6**
- ❖ **NOMBRE DE TESTS RÉUSSIS : 6**
- ❖ **COVERTURES DE CODE TOTALE : 85%**
- ❖ **NOMBRE DE TESTS ÉCHOUÉS : 0**

➤ **DETAILS DE LA COUVERTURE DE CODE :**

- **CLASS 1:** CREATEREMINDERBATCHSHLEDULETEST CONTIENT UNE SEULE METHODE DE TEST COUVRAGE À 80%.

The screenshot shows the Salesforce IDE with the code for `CreateReminderBatchShledule.apxc` and the test results table. The code is as follows:

```
1 global class CreateReminderBatchShledule implements Schedulable {
2
3     // cette méthode est appelée par le planificateur pour exécuter la tâche planifiée
4     global void execute(SchedulableContext ctx) {
5         try {
6             // Créez une instance de la classe CreateReminderBatch
7             CreateReminderBatch batchTasks = new CreateReminderBatch();
8
9             // Exécutez la tâche en tant que batch
10            Database.executeBatch(batchTasks);
11        }
12        catch (Exception ex) {
13            // En cas d'erreur, afficher le message d'erreur dans la console de débogage
14            System.debug(ex.getMessage());
15        }
16    }
17 }
```

The test results table shows the following data:

Status	Test Run	Enqueued Time	Duration	Failures	Total
✓	TestRun @ 6:04:04 pm			0	1
✓	CreateReminderBatchShleduleTest			0	1
✓	ReminderBatchShleduleTest		0:05	0	1
✓	TestRun @ 6:05:55 pm			0	2
✓	CreateReminderBatchTest			0	2
✓	TestCreateReminderBatchWithNoOrders		0:03	0	2

The overall code coverage is 85%.

- **CLASS 2:** CREATEREMINDERBATCHTEST CONTIENT DEUX METHODES DE TEST LE COUVRAGE À 85%.

The screenshot shows the Salesforce IDE with the code for `CreateReminderBatch.apxc` and the test results table. The code is as follows:

```
1 global class CreateReminderBatch implements Database.Batchable<Object> {
2     // La méthode start renvoie un QueryLocator pour le lot à traiter
3     global Database.QueryLocator start(Database.BatchableContext info) {
4         // Interroger les comptes avec leurs commandes et appeler les tâches
5         return Database.getQueryLocator([
6             SELECT Id, ownerId, (SELECT Id FROM Orders), (SELECT Id FROM Tasks WHERE Subject='Call
7         ]);
8     }
9
10    // La méthode Execute traite chaque lot d'enregistrements renvoyés par la méthode start
11    global void execute(Database.BatchableContext bc, List<Account> scope) {
12        try{
13            CreateReminderBatchClass.processBatchCreateReminder(scope);
14        } catch (exception ex){
15            // Capturez toutes les exceptions et enregistrez-les dans le journal de débogage
16            System.debug(ex.getMessage());
17        }
18    }
19 }
```

The test results table shows the following data:

Status	Test Run	Enqueued Time	Duration	Failures	Total
✓	TestRun @ 6:04:04 pm			0	1
✓	CreateReminderBatchShleduleTest			0	1
✓	ReminderBatchShleduleTest		0:05	0	1
✓	TestRun @ 6:05:55 pm			0	2
✓	CreateReminderBatchTest			0	2
✓	TestCreateReminderBatchWithNoOrders		0:03	0	2

The overall code coverage is 85%.

- **CLASS 3: ORDERTRIGGERSHANDLERTEST** CONTIENT TROIS METHODES DE TESTS LE COUVRAGE À 83%.

The screenshot shows the Salesforce Developer Console with the Apex code for the `OrderTriggersHandler` class. The code is a trigger on the `Order` object, triggered before updates, after inserts, and after deletes. It calls the `handleBeforeUpdate` and `handlerAfterDelete` methods of the `OrderTriggersHandler` class.

The test results table shows the following data:

Status	Test Run	Enqueued Time	Duration	Failures	Total
✓	TestRun @ 6:04:04 pm			0	1
✓	CreateReminderBatchShleduleTest			0	1
✓	ReminderBatchShleduleTest		0:05	0	2
✓	TestRun @ 6:05:55 pm			0	2
✓	CreateReminderBatchTest		0:03	0	2
✓	TestCreateReminderBatchWithNoOrders		0:03	0	2

The overall code coverage is 93%.

The screenshot shows the Salesforce Developer Console with the Apex code for the `OrderTriggersHandler` class. The code is a trigger on the `Order` object, triggered before updates, after inserts, and after deletes. It calls the `handleBeforeUpdate` and `handlerAfterDelete` methods of the `OrderTriggersHandler` class.

The test results table shows the following data:

Status	Test Run	Enqueued Time	Duration	Failures	Total
✓	TestRun @ 6:04:04 pm			0	1
✓	CreateReminderBatchShleduleTest			0	1
✓	ReminderBatchShleduleTest		0:05	0	2
✓	TestRun @ 6:05:55 pm			0	2
✓	CreateReminderBatchTest		0:03	0	2
✓	TestCreateReminderBatchWithNoOrders		0:03	0	2
✓	TestCreateReminderBatchWithOrders		0:05	0	2
✓	TestRun @ 6:08:02 pm			0	3
✓	orderTriggersHandlerTest			0	3
✓	handleBeforeUpdateTest		0:04	0	3
✓	handlerAfterDeleteWithNoproductTest		0:02	0	3
✓	handlerAfterDeleteWithproductTest		0:04	0	3

The overall code coverage is 93%.

Résumé : L'exécution des tests a montré une couverture de code globalement satisfaisante du backend Apex.