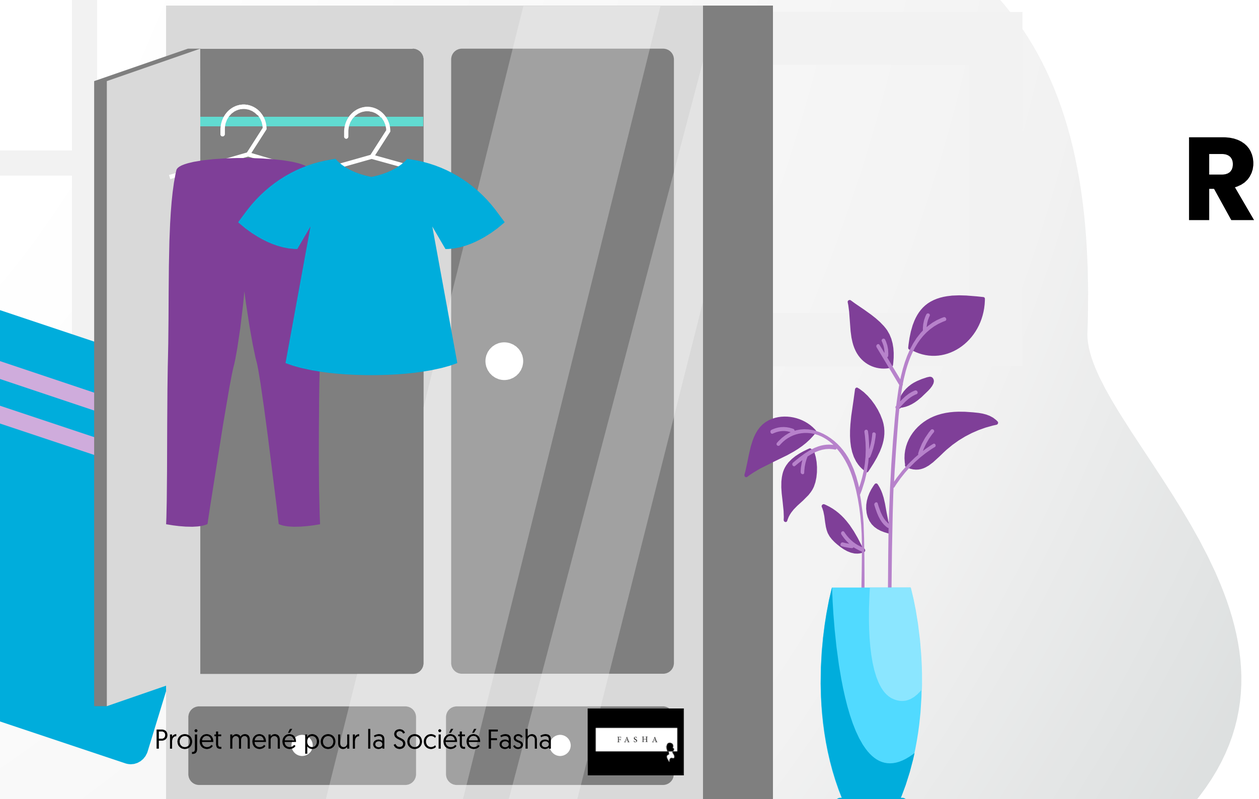
Magdalena Larmet
Salesforce
Developer

Fasha

OPTIMISER UN BACK END
APEX

Sommaire

Intro



Rappels du projet

Projet mené pour la Société Fasha

Rappel du projet

02-06-2023	02-07-2023	35 000€
Date démarrage	Date de fin	Coût du projet

BESOINS EXPRIMÉS

- Optimiser l'application pour le traitement des données en masse
- Corriger les bugs
- Réorganiser le code

PROBLEMES RENCONTRES SUR L'APPLICATION

- ✓ Code pas optimisé pour le traitement des données en Batch
- ✓ Problème de trigger (résultant en des erreurs)
- ✓ Mauvaises pratiques de Tests
- ✓ Nomenclature à revoir

Rappel du projet

JUIN

JUILLET

CORRIGER LES ERREURS DU TRIGGER

TRIGGER

Revoir le code pour corriger les erreurs rencontrées

OPTIMISATION POUR BATCH

BATCH & SCHEDULER

Revoir le code pour un traitement optimisé lors des mise à jour en Batch

NOMENCLATURE DES CLASSES

CLASSES

Revoir la nomenclature des classes pour qu'ells répondent aux normes en matière de nommage des classes et facilitent le travail en équipe.

BEST PRACTICES TESTS

TESTS

Appliquer les bonnes pratiques sur les classes Test

1

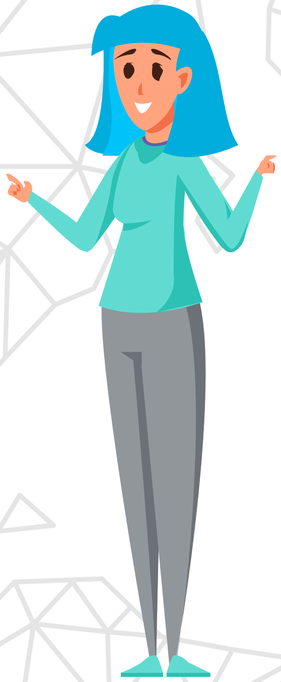
Le Trigger

Magdalena Larmet | DEVELOPPEUR **SALESFORCE**

Correction du Trigger

Mauvaise Pratique

2 trigger par objet



2 triggers

OpenClassrooms-Student-Center / **Optimisez-un-backend-Apex**

Type / to search

Code | Issues | Pull requests | Actions | Projects | Security | Insights

Code Blame 5 lines (4 loc) · 159 Bytes Raw

```
1   trigger CalculMontant on Order (before update) {
2
3           Order newOrder= trigger.new[0];
4           newOrder.NetAmount__c = newOrder.TotalAmount - newOrder.ShipmentCost__c;
5   }
```

Code Blame 12 lines (9 loc) · 410 Bytes Raw

```
1   trigger UpdateAccountCA on Order (after update) {
2
3       set<Id> setAccountIds = new set<Id>();
4
5       for(integer i=0; i< trigger.new.size(); i++){
6           Order newOrder= trigger.new[i];
7
8           Account acc = [SELECT Id, Chiffre_d_affaire__c FROM Account WHERE Id =:newOrder.AccountId ];
9           acc.Chiffre_d_affaire__c = acc.Chiffre_d_affaire__c + newOrder.TotalAmount;
10          update acc;
11      }
12  }
```

Projet mené pour la Société Fasha

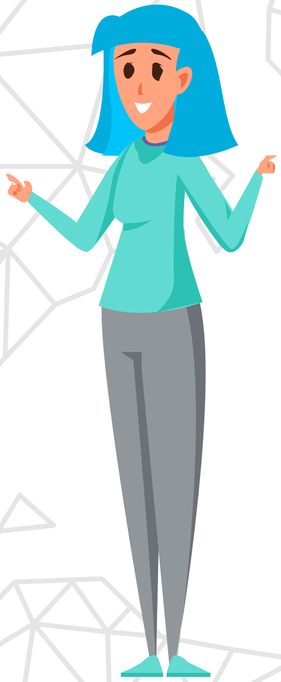
FASHA

Magdalena Larmet | DEVELOPPEUR **SALESFORCE**

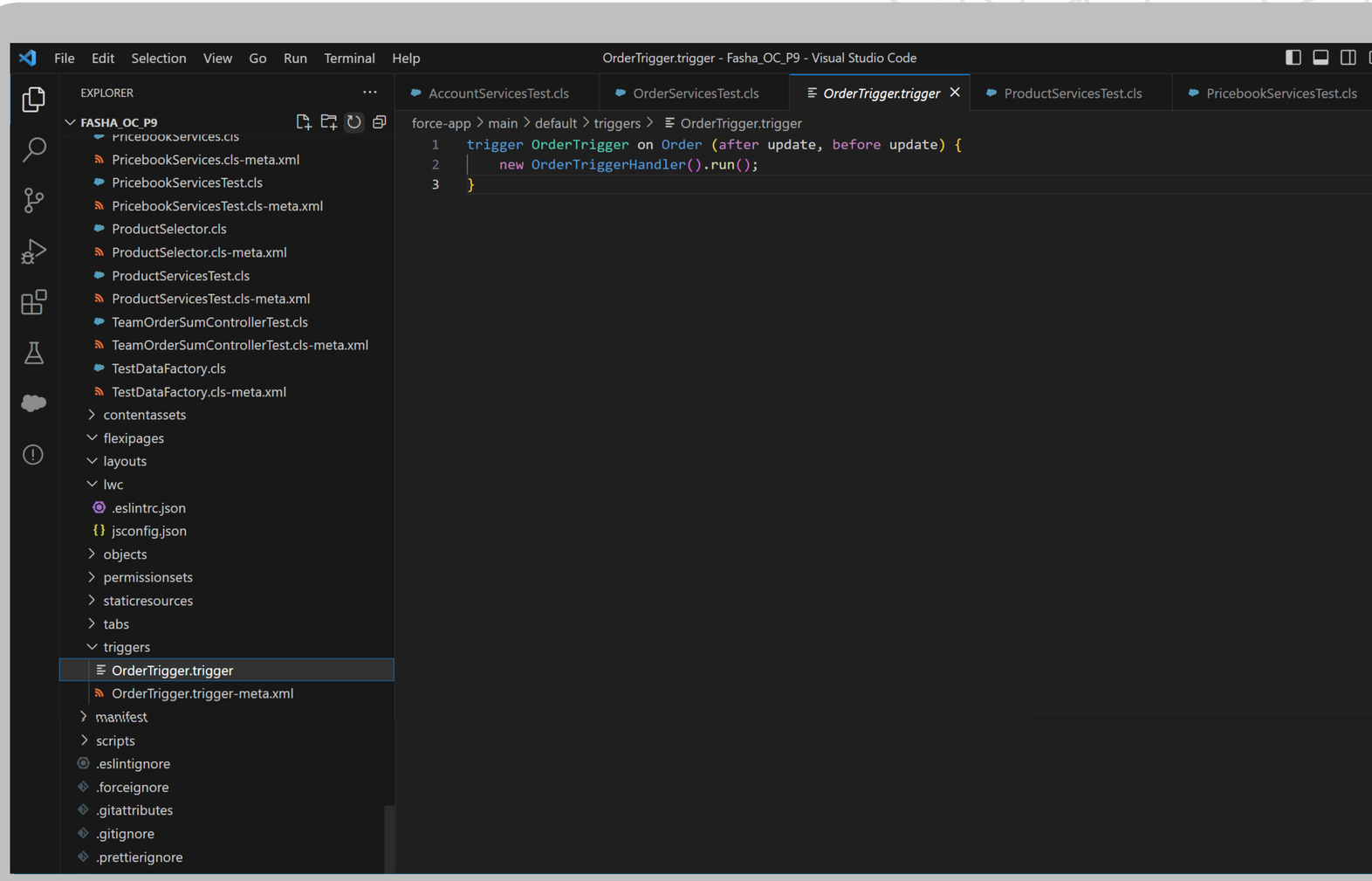
Correction du Trigger

Bonnes Pratiques

1 trigger par objet



Run()



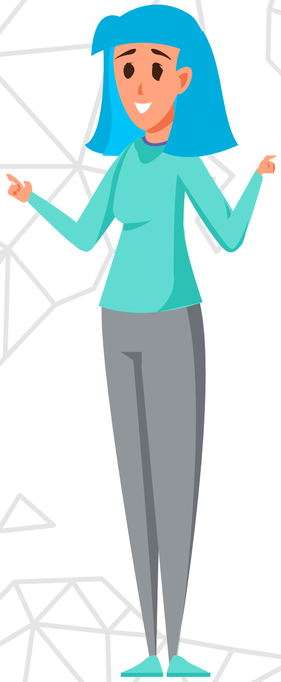
Projet mené pour la Société Fasha

Magdalena Larmet | DEVELOPPEUR **SALESFORCE**

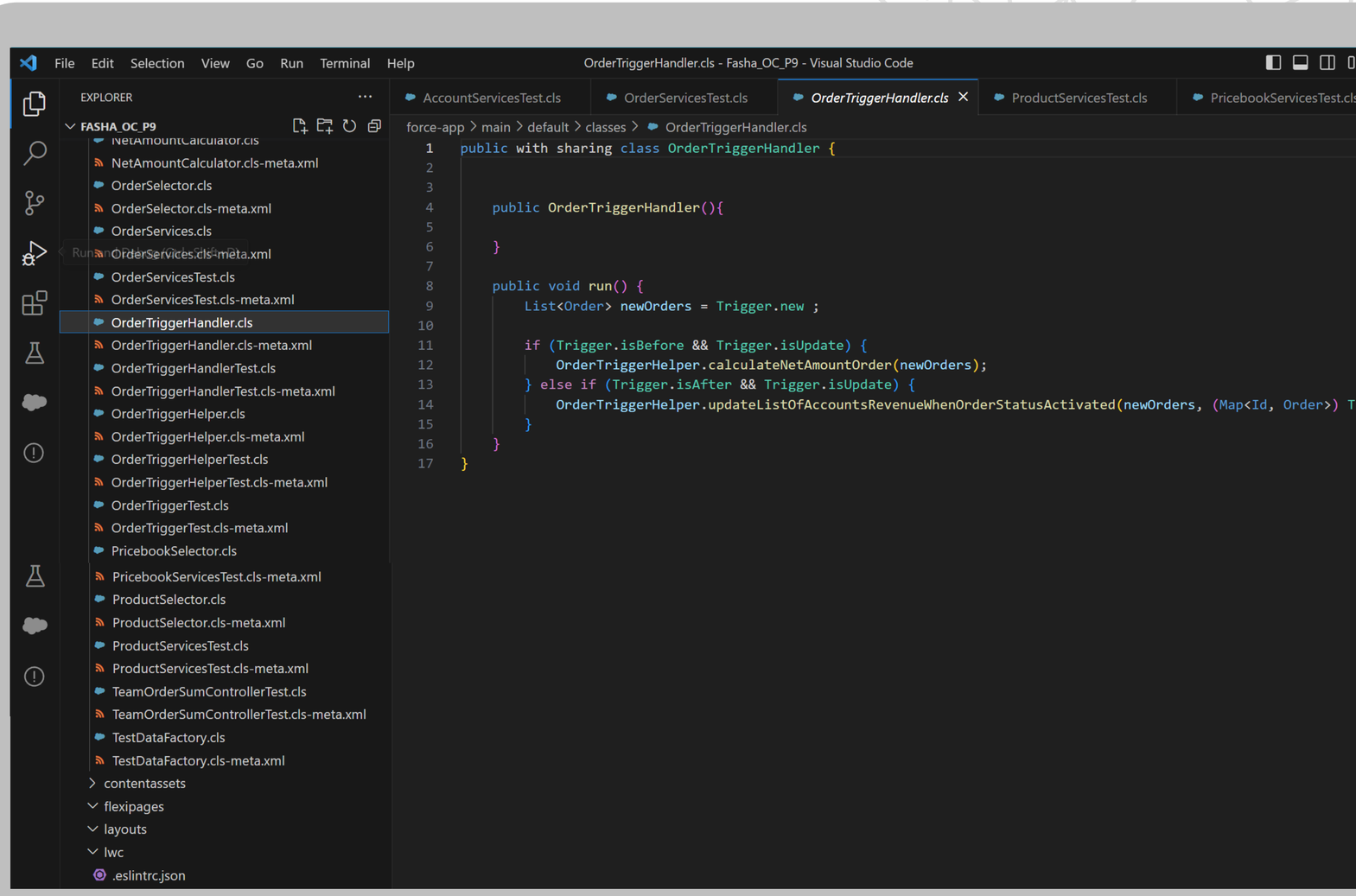
Correction du Trigger

Bonnes Pratiques

1 trigger par objet



TriggerHandler



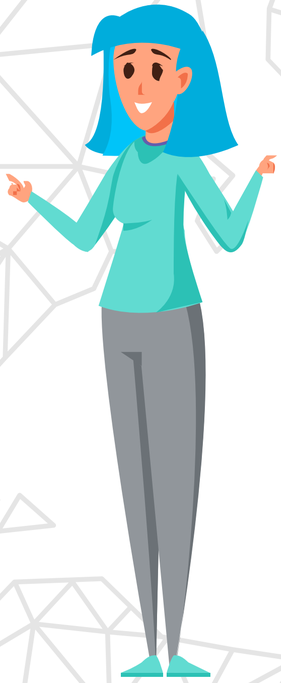
Projet mené pour la Société Fasha

Magdalena Larmet | DEVELOPPEUR **SALESFORCE**

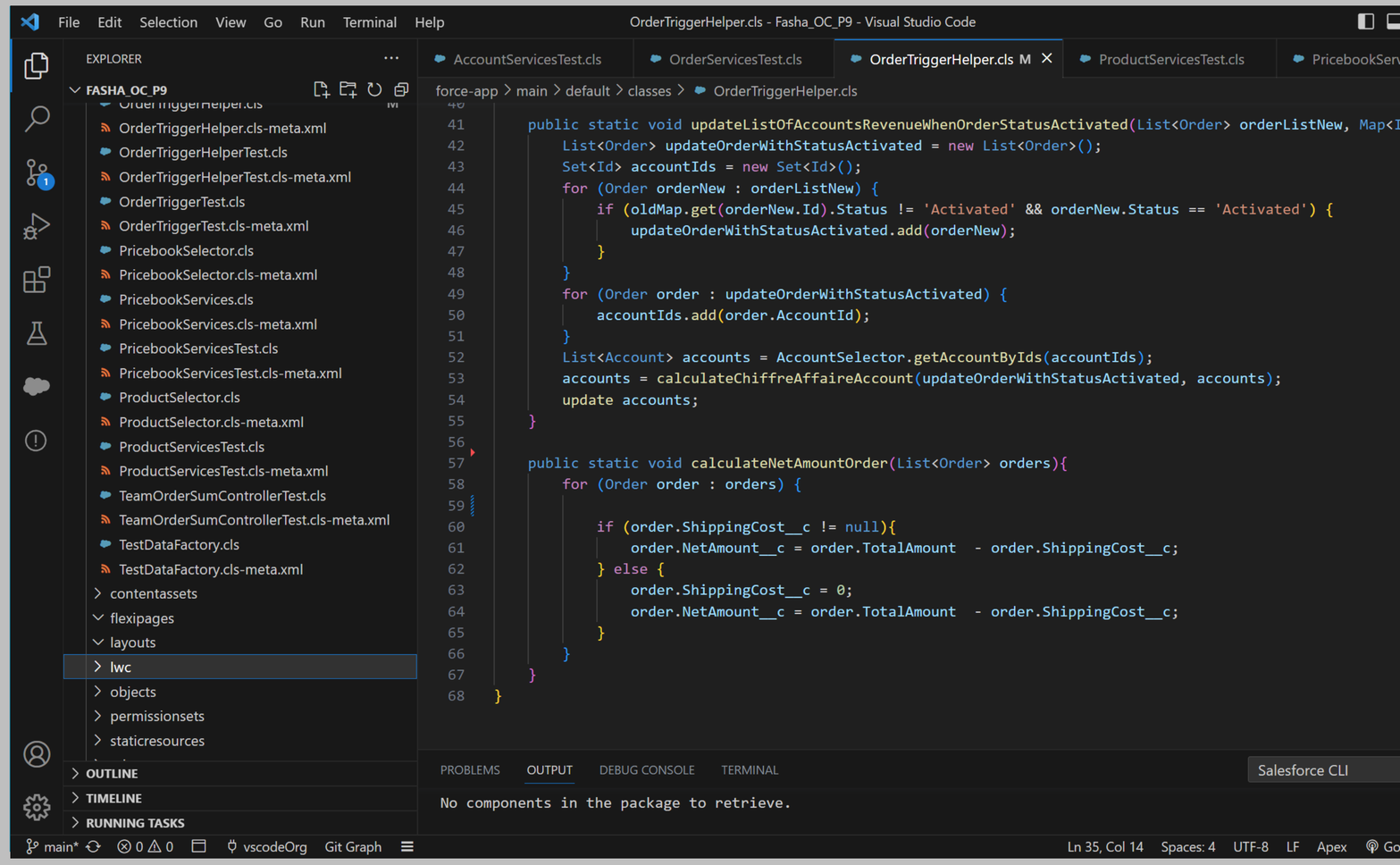
Correction du Trigger

Bonnes Pratiques

1 trigger par objet



TriggerHelper



Projet mené pour la Société Fasha

2

Le Batch

Projet mené pour la Société Fasha

Création de la mise à jour en Batch

Start method
Database.QueryLocator

```
global Database.QueryLocator start(Database.BatchableContext bc) {
    return Database.getQueryLocator(
        [  SELECT Id, Chiffre_affaire__c FROM Account WHERE Id IN (SELECT AccountId FROM Order WHERE Status =
        'Activated')]);
}
```

1

SELECT

Création de la mise à jour en Batch

Execute method
Database.BatchableContext

```
global void execute(Database.BatchableContext bc, List<Account> scope) {
    Set<Id> sAccountTurnoverToUpdate = (new Map<Id, SObject>(scope)).keySet();
    NetAmountCalculator.calculateAmount(sAccountTurnoverToUpdate);
}
```

2

Database.BatchableContext

➔ List<Account> scope

Création de la mise à jour en Batch

Finish method

finish()

```
global void finish(Database.BatchableContext bc) {
}
```

3

Finish

(Database.BatchableContext bc)

Magdalena Larmet | DEVELOPPEUR **SALESFORCE**

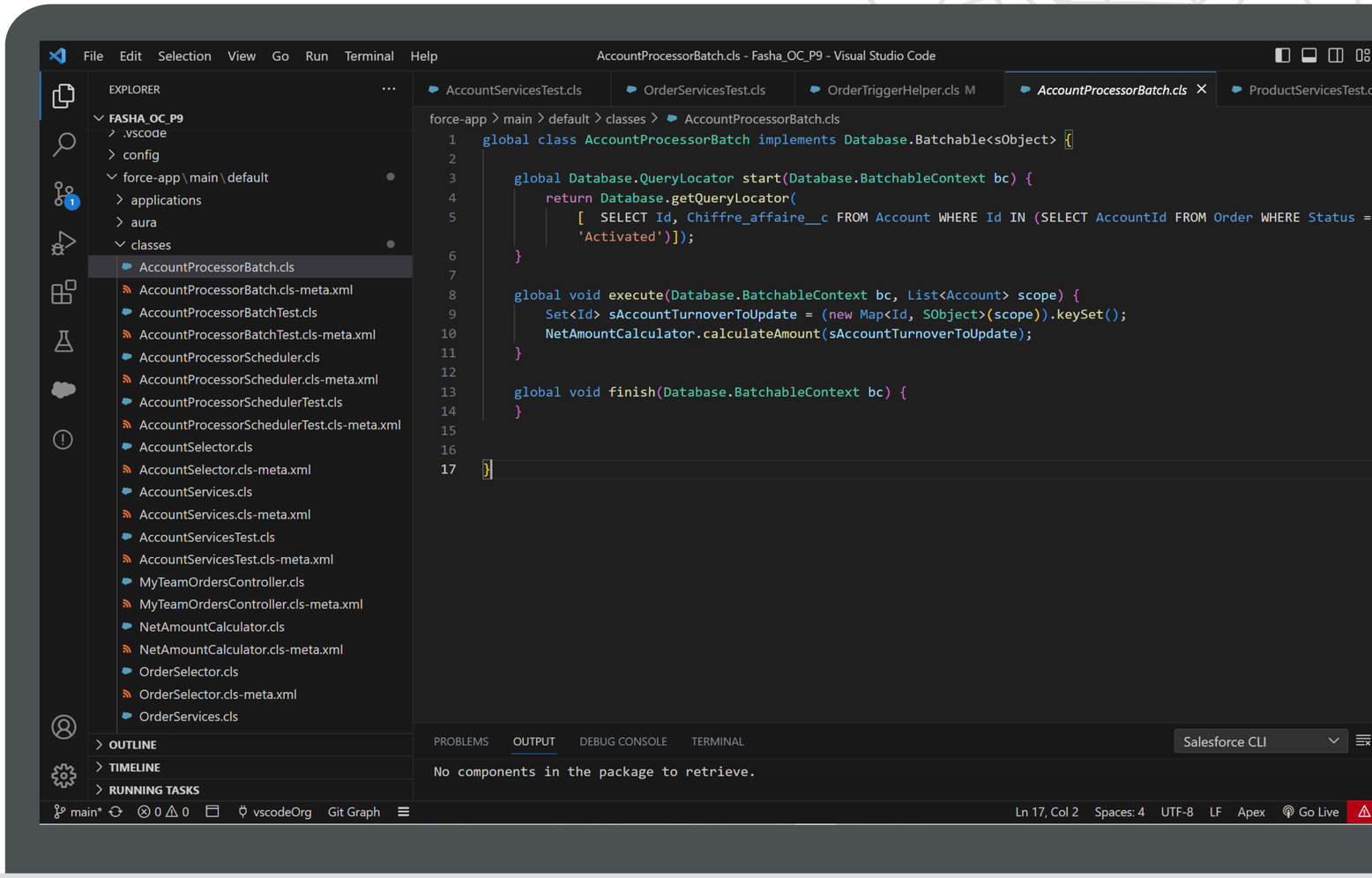
Création de la mise à jour en Batch

AccountProcessorBatch

Traitement des données en masse

Finish

(Database.BatchableContext bc)



Projet mené pour la Société Fasha

Magdalena Larmet | DEVELOPPEUR **SALESFORCE**

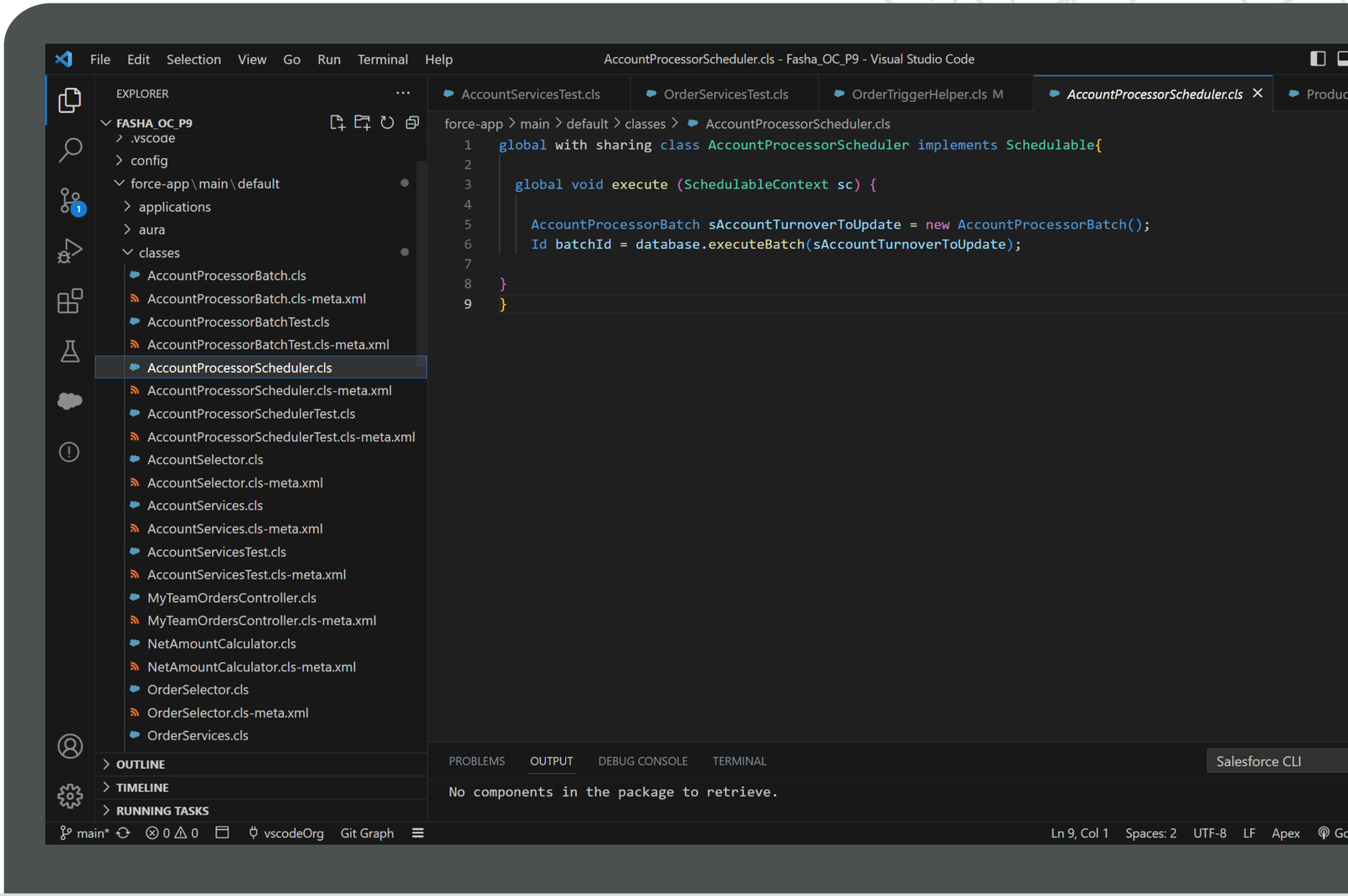
Création du Scheduler

Account Processor Scheduler

Execute

database.executeBatch

(sAccountTurnoverToUpdate)



Projet mené pour la Société Fasha

3

Les Classes

Projet mené pour la Société Fasha

Nomenclature des Classes

Des Classes Spécialisées
Les Classes Selector

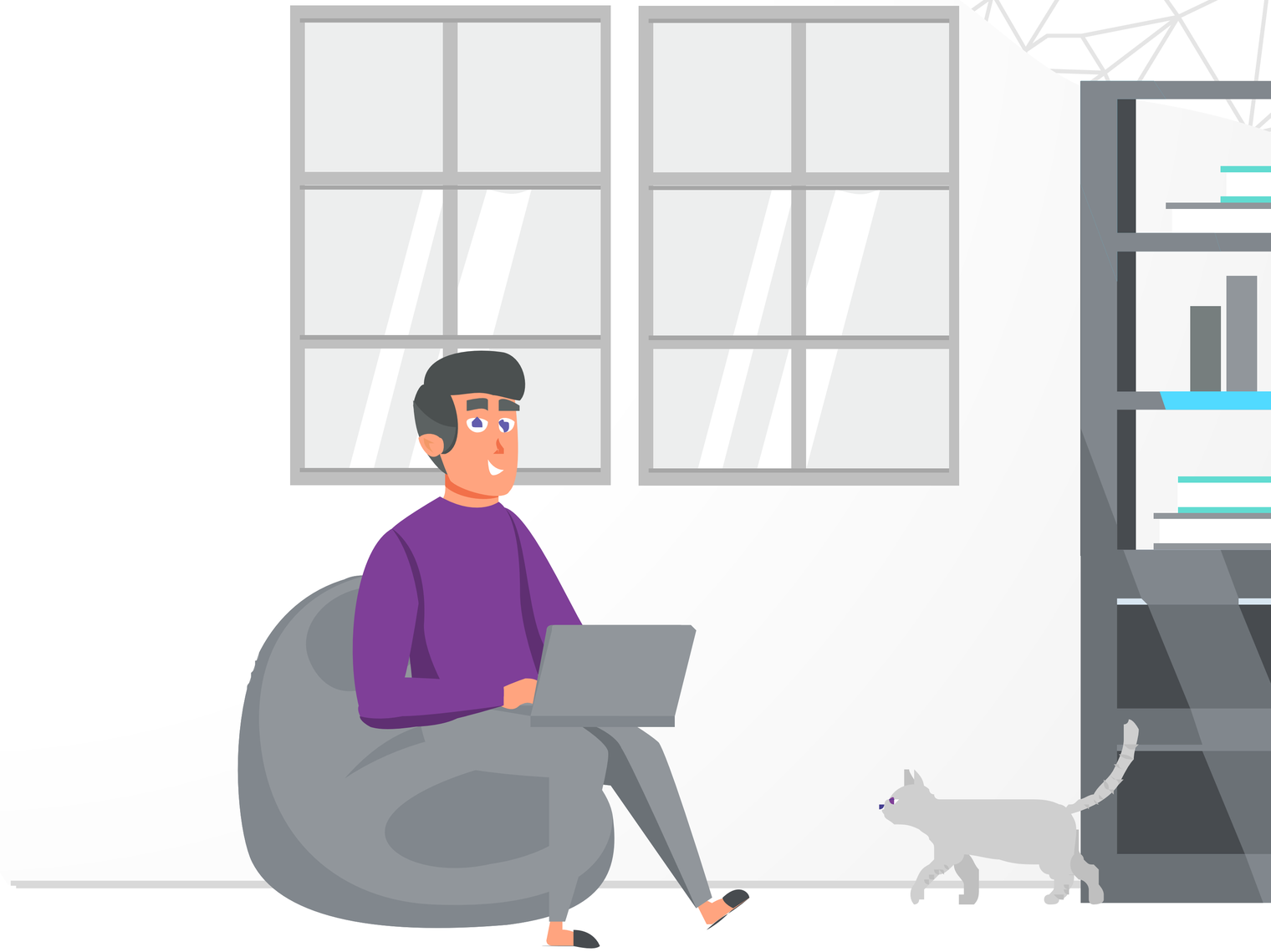
AccountSelector

OrderSelector

PricebookSelector

Sélection de données (GET)

- getAccount()
- getOrder()
- getPricebook()



Nomenclature des Classes

Des Classes Spécialisées
Les Classes Services

AccountServices

OrderServices

PricebookServices

AccountProcessorBatch

NetAmountCalculator

Traitement sur les données

- updateOrderStatus()
- updateAccountNetAmount()

4

Les Tests

Magdalena Larmet | DEVELOPPEUR **SALESFORCE**

Les tests & Best Practices

La création des données de Test

Test data Factory

- createAccount()
- createOrder()
- createPricebook()
- createProduct()
- ...

Les tests & Best Practices

La nomenclature des classes

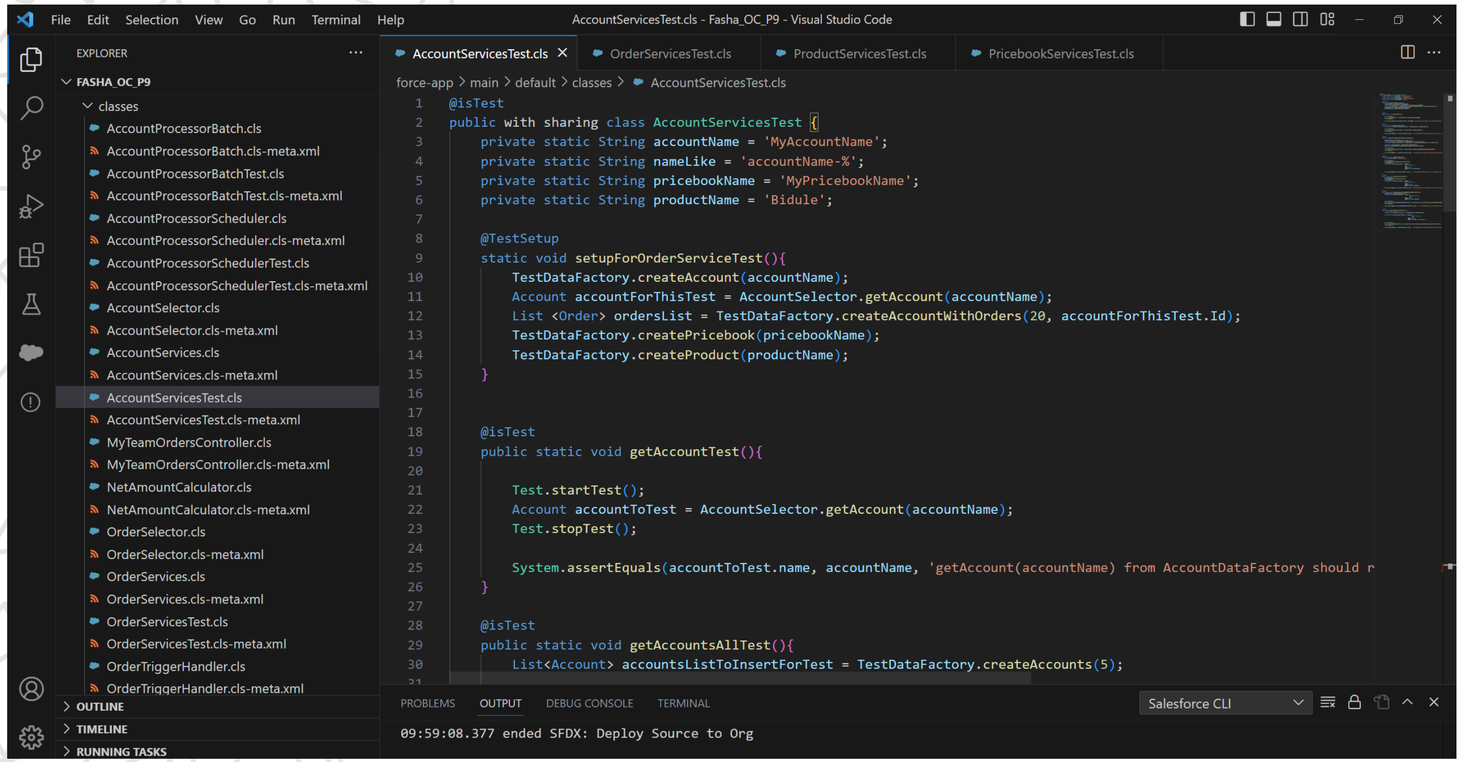
Une classe Test pour une classe Services

- OrderServicesTest
- PricebookServicesTest
- AccountServicesTest
- AccountProcessorBatchTest
- AccountProcessorSchedulerTest
- ...

Les tests & Best Practices

Les Best Practices des Tests Apex

La syntaxe



1. Utiliser @testSetup pour générer les données nécessaires pour chaque classe de test

3. Démarrer chaque test par Test.startTest()

4. Terminer chaque test par Test.stopTest()

5. Utiliser les instructions System.assert() après TestStopTest pour vérifier les données attendues

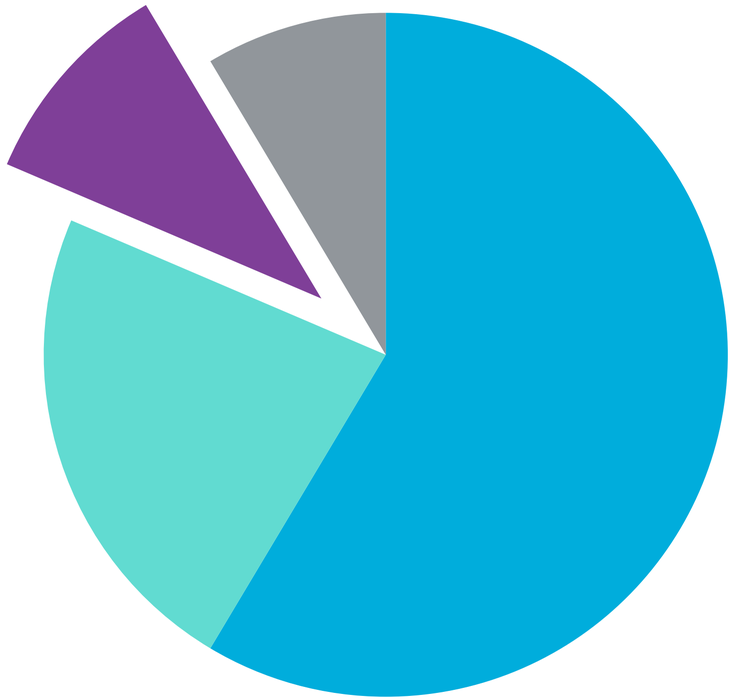
Les tests & Best Practices

Les Tests Fonctionnels

Couverture du code au moins à 75%

Overall Code Coverage

Class	Percent	Lines
Overall	**87%**	
AccountProcessorBatch	100%	7/7
AccountProcessorScheduler	100%	3/3
AccountSelector	100%	12/12
AccountServices	100%	10/10
MyTeamOrdersController	100%	5/5
NetAmountCalculator	100%	9/9
OrderSelector	61%	13/21
OrderServices	100%	15/15
OrderTrigger	100%	1/1
OrderTriggerHandler	100%	7/7
OrderTriggerHelper	75%	34/45
PricebookSelector	100%	11/11
PricebookServices	100%	6/6
ProductSelector	100%	4/4



Les tests & Best Practices

Les Bests Practices des Classes et des Tests

Ecriture de code optimisé pour le traitement de données de masse

Tous les tests et le code doivent marcher pour des données en bulk (avec un nb infini d'enregistrement) : pour se faire on crée un liste on stocke l'objet dans la liste et on récupère la liste d'objets)

```
trigger OpportunityTrigger on Opportunity(before insert) {
    Opportunity opp = Trigger.new[0];

    if(opp.Amount > 10000){
        assignSalesLeaderOwner(opp);
    }
}
```

```
trigger OpportunityTrigger on Opportunity(before insert) {
    for (Opportunity opp : Trigger.new) {
        if (opp.Amount > 10000) {
            assignSalesLeaderOwner(opp);
        }
    }
}
```

Magdalena Larmet

Salesforce Developer