

Projet 9

«*Testez vos développements Java*»

Travaux effectués

Version 2.0

1- Versions	3
2- Introduction	3
2.1- Objet du document	3
2.2- Références	3
3- Implémentations	4
3.1- TODO : Fonctionnalités demandées	4
3.2- Implémentations requises au projet	4
3.3- Correction d’erreur	4
4- Test unitaire	5
4.1- Couche Business	5
4.2- Couche Model	6
5- Test d’intégration	6
5.1- Couche Business	6
5.2- Couche Consumer	7

Versions

Auteur	Date	Description	Version
Dubois Geoffrey	17/10/2020	Création du document	1.0
Dubois Geoffrey	19/10/2020	première insertion de tâche	1.1
Dubois Geoffrey	23/10/2020	fin insertion de tâche	1.2
Dubois Geoffrey	09/11/2020	Mise en page, et structuration des idées	2.0

Introduction

Objet du document

Le présent document constitue l'ensemble des tâches effectuées depuis la prise du projet : 14 août 2020

Objectif du document:

- Répertorier l'ensemble des erreurs présents dans le projet.
- Lister les nouvelles fonctionnalités apportées.
- Cataloguer les tests amenés au projet.

Les éléments du présents dossiers découlent :

- d'une étude du code sources
- d'une mises en place de tests unitaires et fonctionnelles

Références

Pour de plus amples informations, se référer également aux éléments suivants :

README.md - information d'utilisation

doc : Dossier de diagramme

docker : Base de donnée de l'application

Implémentation

TODO : Fonctionnalités demandées

Couche Business, Classe ComptabiliteManagerImpl, AddReference() : *Implementation*

- 1- Recherche de la séquence d'écriture comptable du journal par rapport à l'année en cours et le journalCode associée.
- 2- Utiliser la dernière valeur de la séquence et lui ajouter 1. Si la séquence n'existe pas, création d'une nouvelle séquence avec la dernière valeur égale à 1.
- 3- Mise à jour de la référence en respectant la RG_compta_5.
- 4- Enregistrer / mettre à jour la valeur de la séquence en persistance.

Couche Business, Classe ComptabiliteManagerImpl, checkEcritureComptableUnit() : *Implementation*

- Ajout de la RG_Compta_5 :
 - 1- Vérifie que la référence respecte bien le bon format : XX-AAAA/#####.
 - 2- Vérifie que l'année de la référence correspond à la date de l'écriture.
 - 3- Vérifie que le journal code de la référence correspond au journal code de l'écriture.

Implémentations requises au projet

Couche Consumer, Classe ComptabiliteDao, getSequenceEcritureComptable() :
insertSequenceEcritureComptable() :
updateSequenceEcritureComptable() :
deleteSequenceEcritureComptable() :

- 1- Création des signatures des méthodes dans l'interface ComptabiliteDao.

Couche Consumer, Classe ComptabiliteDaoImpl, getSequenceEcritureComptable() :
insertSequenceEcritureComptable() :
updateSequenceEcritureComptable() :
deleteSequenceEcritureComptable() :

- 1- Création des méthodes dans ComptabiliteDaoImpl en @Override.
- 2- Implémentation des méthodes.
- 3- Création des méthode setSQL associée et de l'attribut statique SQL associé.

Couche Business, package testbusiness.business, Ajout de fichier bootstrapContext.xml :

- 1- Erreur survenu : impossible d'ouvrir le fichier bootStrapContext.xml
- 2- Configuration : Importation des fichiers de configurations.

Couche Model, Classe SequenceEcritureComptable, SequenceEcritureComptable : *Création classe*

- 1- Création des attributs : annee, derniereValeur, journalCode.
- 2- Création du constructeur : SequenceEcritureComptable() et SequenceEcritureComptable(String pJournalCode, Integer pAnnee, Integer pDerniereValeur).
- 3- Création des Setters et Getters.
- 4- Création de la méthode toString().

Couche Consumer, Classe SequenceEcritureComptableRM, Création de la classe SequenceEcritureComptableRM :

- 1- mapRow() : Permet de mapper les données du resultSet au bean

Correction d'erreur

Couche Model, Classe EcritureComptable, Correction d'erreur sur la méthode getTotalCredit() :

- 1- la méthode récupérait le débit de l'écriture comptable au lieu du crédit. getDebit() ==> getCredit()

Couche Model, Classe EcritureComptable, Correction d'erreur sur la méthode isEquilibree() :

- 1- Une erreur dans la comparaison était présente. equals() ==> compareTo()

Couche Model, Classe EcritureComptable, Correction d'erreur sur l'attribut reference :

- 1- Le format de l'attribut reference était incorrect. Le code journal prenait en compte des chiffres au lieu de l'ensemble des caractères. (regexp = «\\d{1,5}-\\d{4}/\\d{5}») ==> (regexp = «\\w{1,5}-\\d{4}/\\d{5}»)

Couche Model, Classe SequenceEcritureComptable, Correction d'erreur sur l'attribut journalCode :

- 1- L'attribut était absent. Ajout de l'attribut «private String journalCode;», de son getter , de son setter et de son remplissage dans le constructeur

Couche Business, Classe ComptabiliteManagerImpl, Correction d'erreur sur la méthodes updateEcritureComptable() :

- 1- la vérification de l'écriture comptable n'était pas effectué avant la mise à jour de l'écriture comptable. this.checkEcritureComptable(pEcritureComptable); avant getDaoProxy().getComptabiliteDao().updateEcritureComptable(pEcritureComptable);

Couche Consumer, fichier sqlContext.xml, Correction d’erreur sur la requête **SQLinsertListLigneEcritureComptable** :

1- une virgule était absente dans la requête d’insertion.

<pre><value> INSERT INTO myerp.ligne_ecriture_comptable (ecriture_id, ligne_id, compte_ comptable_numero, libelle, debit credit) VALUES (:ecriture_id, :ligne_id, :compte_ comptable_numero, :libelle, :debit, :credit) </value></pre>	==>	<pre><value> INSERT INTO myerp.ligne_ecriture_comptable (ecriture_id, ligne_id, compte_ comptable_numero, libelle, debit, credit) VALUES (:ecriture_id, :ligne_id, :compte_ comptable_numero, :libelle, :debit, :credit) </value></pre>
--	-----	---

Test unitaire

Couche Business

Couche Business, Classe ComptabiliteManagerImplTest, Test de la fonction **checkEcritureComptableUnit()** :

1- **checkEcritureComptableUnit_throwFunctionalException()** : Vérifie qu’une ecriture comptable vide déclenche une FunctionalException.

2- **checkEcritureComptableUnit_testPassed()** : Vérifie qu’une ecriture comptable réglementaire passe toute les règles unitaires.

3- **checkEcritureComptableUnitRG2_NoEquilibre_ThrowFunctionalException()** : Vérifie qu’une ecriture comptable non équilibré déclenche une FunctionalException car la RG_Compta_2 n’est pas respecté.

4- **checkEcritureComptableUnitRG2_Equilibre_DontThrowException()** : Vérifie qu’une ecriture comptable équilibré ne déclenche pas une FunctionalException.

5- **checkEcritureComptableUnitRG3_WithOneLine_throwFunctionalException()** : Vérifie qu’une ecriture comptable avec une ligne d’écriture déclenche une FunctionalException car la RG_Compta_3 n’est pas respecté.

6- **checkEcritureComptableUnitRG3_WithTwoDebit_throwFunctionalException()** : Vérifie qu’une ecriture comptable avec 2 débits et 0 crédit déclenche une FunctionalException car la RG_Compta_3 n’est pas respecté.

7- **checkEcritureComptableUnitRG3_WithTwoCredit_throwFunctionalException()** : Vérifie qu’une ecriture comptable avec 0 débits et 2 crédit déclenche une FunctionalException car la RG_Compta_3 n’est pas respecté.

8- **checkEcritureComptableUnitRG3_WithThreeCredit_throwFunctionalException()** : Vérifie qu’une ecriture comptable avec 0 débits et 3 crédit déclenche une FunctionalException car la RG_Compta_3 n’est pas respecté.

9- **checkEcritureComptableUnitRG3AndRG2_WithThreeLine_NoThrowFunctionalException()** : Vérifie qu’une ecriture comptable avec 3 lignes ne déclenche pas une FunctionalException.

10- **checkEcritureComptableUnitRG5_BadFormat_ThrowFunctionalException(String ref)** : Vérifie que l’ensemble des ecritures comptables avec un mauvais format déclenche une FunctionalException car la RG_Compta_5 n’est pas respecté.

11- **checkEcritureComptableUnitRG5_BadYear_ThrowFunctionalException()** : Vérifie qu’une ecriture comptable avec une erreur dans l’année de la référence déclenche une FunctionalException car la RG_Compta_5 n’est pas respecté.

12- **checkEcritureComptableUnitRG5_BadCode_ThrowFunctionalException()** : Vérifie qu’une ecriture comptable avec une erreur dans le journal code de la référence déclenche une FunctionalException car la RG_Compta_5 n’est pas respecté.

Couche Business, Classe ComptabiliteManagerImplTest, Test de la fonction **checkEcritureComptableContext()** :

1- **checkEcritureComptableContext_NoReturn_testPassed()** : Vérifie qu’une ecriture comptable ne déclenche pas une FunctionalException.

2- **checkEcritureComptableContextRG6_double_throwFunctionalException()** : Vérifie qu’une ecriture comptable qui existe déjà déclenche une FunctionalException car la RG_Compta_6 n’est pas respecté.

3- **checkEcritureComptableContextRG6_NewEcritureComptableFound_throwFunctionalException()** : Vérifie qu’une ecriture comptable qui possède une valeur plus récente déclenche une FunctionalException car la RG_Compta_6 n’est pas respecté.

4- **checkEcritureComptableContextRG6_EcritureComptableNotFound_throwFunctionalException()** : Vérifie qu’une ecriture comptable qui n’existe pas déclenche une FunctionalException car la RG_Compta_6 n’est pas respecté.

Couche Business, Classe ComptabiliteManagerImplTest, Test de la fonction **checkEcritureComptable()** :

1- **checkEcritureComptable_NoReturn_NotThrowException()** : Vérifie qu’une ecriture comptable ne déclenche pas une FunctionalException.

Couche Business, Classe ComptabiliteManagerImplTest, Test de la fonction **addReference()** :

1- **addReference_NoReturn_EcritureComptable()** : Vérifie que l’ajout d’une référence d’une ecriture comptable ne déclenche pas une Exception.

2- **addReference_NoReturn_EcritureComptable_ThrowNotFoundException()** : Vérifie que l’ajout d’une référence d’une ecriture comptable déclenche une NotFoundException.

Couche Model

Couche Model, Classe EcritureComptableTest, Test de la classe **EcritureComptableTest** :

- 1- **validateSettersAndGetters()** : Vérifie que les getters and setters sont fonctionnels
- 2- **isEquilibree()** et **isNotEquilibree()** : Séparation des tests : test unitaire
- 3- **getTotalDebit_returnBigDecimal_EcritureComptable()** : Vérifie que la fonction renvoie la bonne somme de débit d’une écriture comptable
- 3- **getTotalCredit_returnBigDecimal_EcritureComptable()** : Vérifie que la fonction renvoie la bonne somme de crédit d’une écriture comptable

Couche Model, Classe LigneEcritureComptableTest, Test de la classe **LigneEcritureComptableTest** :

- 1- **validateSettersAndGetters()** : Vérifie que les getters and setters sont fonctionnels

Couche Model, Classe JournalComptableTest, Test de la classe **JournalComptableTest** :

- 1- **validateSettersAndGetters()** : Vérifie que les getters and setters sont fonctionnels
- 2- **getByCode_returnJournalComptable_listJournalComptableAndCode()** : vérifie que la fonction récupère le bon journal dans une liste par rapport à son journalCode

Couche Model, Classe CompteComptableTest, Test de la classe **CompteComptableTest** :

- 1- **validateSettersAndGetters()** : Vérifie que les getters and setters sont fonctionnels
- 2- **getByNumero_returnCompteComptable_listCompteComptableAndNumero()** : Vérification de récupération d’un élè-ment d’une liste par rapport à son numéro

Couche Model, Classe SequenceEcritureComptableTest, Test de la classe **SequenceEcritureComptableTest** :

- 1- **validateSettersAndGetters()** : Vérifie que les getters and setters sont fonctionnels
- 2- **validateConstrutor()** : Test le constructeur de la classe SequenceEcritureComptable

Test d'intégration

Couche Business

Couche Business, Classe BusinessTest, Création de la classe **BusinessTest** :

- 1- **extends BusinessTestCase**

Couche Business, Classe BusinessTest, Test d’intégration lié à CompteComptable :

- 1- **getListCompteComptableTest_listCompteComptable_sizeListEqual7()** : Test de la récupération de la liste des comptes comptables.

Couche Business, Classe BusinessTest, Test d’intégration lié à JournalComptable :

- 1- **getListJournalComptableTest_listJournalComptable_sizeListEqual4()** : Test de la récupération de la liste des journaux comptables.

Couche Business, Classe BusinessTest, Test d’intégration lié à EcritureComptable :

- 1- **getListEcritureComptableTest_listEcritureComptable_sizeListEqual5()** : Test de la récupération de la liste des écritures comptables.
- 2- **insertEcritureComptableTest_NoReturn_checkEcritureComptable()** : Test d’insertion d’une écriture comp-table.
- 3- **insertEcritureComptable_NoReturn_throwFunctionalException()** : Test d’insertion d’une écriture ne respec-tant pas les règles de gestion.
- 4- **updateEcritureComptableTest_NoReturn_checkEcritureComptable()** : Test de mise à jour d’une écriture comp-table.
- 5- **updateEcritureComptableTest_NoReturn_incorrectList()** : Test de modification d’une écriture comptable avec une liste incorrect.
- 6- **updateEcritureComptableTest_NoReturn_IncorrectRef()** : Test de modification d’une écriture comptable avec une référence incorrecte.
- 7- **deleteEcritureComptableTest_NoReturn_checkEcritureComtpable()** : Test de suppression d’une écriture comp-table.

Couche Business, Classe BusinessTest, Test d’intégration lié à SequenceEcritureComptable :

- 1- **insertSequenceEcritureComptableTest_NoReturn_checkSequenceEcritureComptable()** : Test d’insertion de la séquence d’écriture comptable.
- 2- **updateSequenceEcritureComptableTest_NoReturn_checkSequenceEcritureComptable()** : Test de modification de la séquence d’écriture comptable.
- 3- **deleteSequenceEcritureComptableTest_NoReturn_checkSequenceEcritureComptable()** : Test de suppression d’une séquence d’écriture comptable.

Couche Business, Classe BusinessTest, Test d’intégration lié à addReference :

- 1- **addReferenceTest_NoReturn_checkAdd()** : Test d’ajout d’une référence à une écriture comptable avec addRe-fERENCE.

Couche Consumer

Couche Consumer, package *test-consumer*, Création de la partie **test** de **consumer** :

- 1- **SpringRegistry** : Ajout de la classe **SpringRegistry** : Registre des Beans Spring.
- 2- **ConsumerTestCase** : Ajout de la classe **ConsumerTestCase** : Classe mère des classes de test d'intégration de la couche Consumer.
- 3- **ConsumerTest**: Ajout de la classe **ConsumerTest** : Classe de test d'intégration de la couche Consumer.
- 4- **bootStrapContext.xml**: Ajout de la ressource **bootStrapContext** : fichier xml regroupant les ressources utilisés.
- 5- **datasourceContext.xml**: Ajout de la ressource **datasourceContext** : fichier xml regroupant les informations d'accès à la base de données.

Couche Consumer, *Classe ConsumerTest*, Implementation de méthode **createEcritureComptable()** :

- 1- **createEcritureComptable()** : création d'une écriture comptable suivant des argument et renvoie une *EcritureComptable*.

Couche Consumer, *Classe ConsumerTest*, Implementation de test d'intégration lié au **CompteComptable** :

- 1- **getListCompteComptableTest_listCompteComptable_checkFirstCompteComptableAndSizeListEqual7()** : Test de récupération de la liste de compte comptable

Couche Consumer, *Classe ConsumerTest*, Implementation de test d'intégration lié au **JournalComptable** :

- 1- **getListJournalComptableTest_listJournalComptable_checkFirstJournalComptableAndSizeListEqual4()** : Test de récupération de la liste de compte comptable

Couche Consumer, *Classe ConsumerTest*, Implementation de test d'intégration lié au **EcritureComptable** :

- 1- **getListEcritureComptableTest_listEcritureComptable_checkFirstEcritureComptableAndSizeListEqual5()** : Test de récupération de la liste des écritures comptables
- 2- **getEcritureComptableTest_EcritureComptable_checkEcritureComptable()** : Test de récupération d'une écriture comptable par son id
- 3- **getEcritureComptableTest_EcritureComptable_throwNotFoundException()** : Test de récupération d'une écriture comptable inexistant par son id
- 4- **Test de récupération d'une écriture comptable par sa référence** : Test de récupération d'une écriture comptable inexistant par son id
- 5- **getEcritureComptableByRefTest_EcritureComptable_throwNotFoundException()** : Test de récupération d'une écriture comptable inexistant par sa référence

Couche Consumer, *Classe ConsumerTest*, Implementation de test d'intégration lié au **EcritureComptable** :(suite)

- 6- **insertEcritureComptableTest_NoReturn_checkbyGetId()** : Test d'insertion d'une écriture comptable
- 7- **updateEcritureComptableTest_NoReturn_checkEcritureComptable()** : Test de mise à jour d'une écriture comptable
- 8- **deleteEcritureComptableTest_NoReturn_checkEcritureComptableNotFound()** : Test de suppression d'une écriture comptable

Couche Consumer, *Classe ConsumerTest*, Implementation de test d'intégration lié au **LigneEcritureComptable** :

- 1- **loadListLigneEcritureTest_NoReturn_checkListLigneEcritureComptableInEcritureComptable()** : Test de chargement de la liste des lignes d'écriture dans une écriture.

Couche Consumer, *Classe ConsumerTest*, Implementation de test d'intégration lié au **LigneEcritureComptable** :

- 1- **getSequenceEcritureComptableTest_SequenceEcritureComptable_withJournalCodeAndAnnee_checkDerniereValeur()** : Test de récupération d'une séquence d'écriture comptable.
- 2- **getSequenceEcritureComptableTest_SequenceEcritureComptable_throwNotFoundException()**: Test de récupération d'une séquence d'écriture comptable.
- 3- **deleteSequenceEcritureComptableTest_NoReturn_ThrowNotFoundException()**: Test de suppression d'une séquence d'écriture comptable
- 4- **insertSequenceEcritureComptableTest_NoReturn()**: Test d'insertion d'une séquence d'écriture comptable
- 5- **updateSequenceEcritureComptableTest_NoReturn()**: Test de mise à jour d'une séquence écriture comptable