



Synthèse Projet 9

« Testez vos développements Java »

Mathieu Le Cardinal
Tuteur : Jacky Revet

10 juillet 2019

Sommaire

1. Correction des 4 Bugs.....	p3
2. Etude des <<TODO>>.....	p4
3. Couverture de code.....	p5

1. Correction Bugs

Quatre bugs avaient été inséré dans le projet et étaient à corriger, nous les avons listé en précisant les corrections apportés :

- Classe [EcritureComptable](#) [module Model]: la méthode getTotalCredit() faisait appel à vEcritureComptable.getDebit() au lieu de vEcritureComptable.getCredit().
- Classe [EcritureComptable](#) [module Model]: j'ai remplacé la méthode equals() par compareTo() dans la méthode isEquilibree() car avec equals() 2 valeurs pouvaient être égale même si l'arrondi étaient différent.
- Classe [EcritureComptable](#) [module Model]: Correction d'une erreur dans l'expression régulière permettant de valider le format de la référence d'une écriture comptable.
- Fichier [sqlContext.xml](#) [module Consumer]: Correction d'une erreur de syntaxe dans la requête SQLinsertListLigneEcritureComptable.

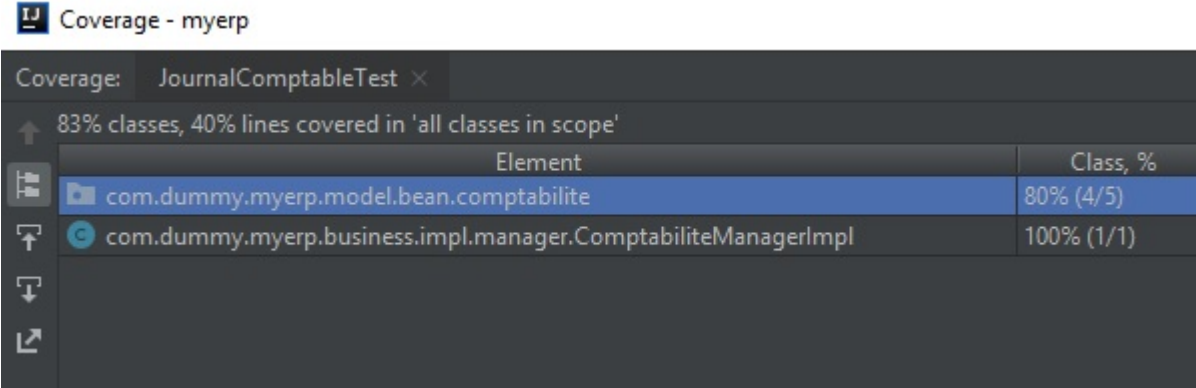
2. Etude des <<TODO>>

Rappel sur les <<TODO>> de Myerp et les solutions apportés :

- [module Model] [EcritureComptable](#) Méthode public BigDecimal getTotalDebit() à tester : Test réalisé.
- [module Model] Mettre en place des test unitaires pour les beans de Myerp : Tests Unitaires implémenté.
- [module Business] [ComptabiliteManagerImpl](#) Méthode addReference() à implémenter : Implémentation réalisée.
- [module Business] [ComptabiliteManagerImpl](#) Méthode addReference() à tester : Test réalisé.
- [module Business] [ComptabiliteManagerImpl](#) Méthode getValSequenceJournalComptable() à tester : Test réalisé.
- [module Business] [ComptabiliteManagerImpl](#) Méthode checkEcritureComptableUnit() à implémenter : Implémentation réalisée.
- [module Business] [ComptabiliteManagerImpl](#) Implémentation et test de la RG5 : Test et implémentation réalisés.
- [module Business] [ComptabiliteManagerImpl](#) Implémentation et test de la RG7 : Test et implémentation réalisés.
- [module Business] [ComptabiliteManagerImpTest](#) Mettre en place des test unitaires pour la partie business de Myerp : Tests Unitaires implémentés.
- [module Business] [ComptabiliteManagerImplTest](#) Mettre en place des test d'intégrations les modules de Myerp : Tests d'intégration implémentés.

3. Couverture de Code

Rappel sur la couverture de code avec un objectif de 75 % de l'application :



The screenshot shows a 'Coverage - myerp' window with a tab for 'JournalComptableTest'. It displays a summary of 83% classes and 40% lines covered. Below this is a table with two columns: 'Element' and 'Class, %'. The table lists two elements: a package 'com.dummy.myerp.model.bean.comptabilite' with 80% coverage (4/5) and a class 'com.dummy.myerp.business.impl.manager.ComptabiliteManagerImpl' with 100% coverage (1/1).

Coverage: JournalComptableTest	
83% classes, 40% lines covered in 'all classes in scope'	
Element	Class, %
com.dummy.myerp.model.bean.comptabilite	80% (4/5)
com.dummy.myerp.business.impl.manager.ComptabiliteManagerImpl	100% (1/1)

Code Coverage des test de Myerp : 90 %