



Testez vos développements Java

Méthodes



- Modification de l'environnement de test
- Écriture des tests
- Correction des bugs
- Implémentation des TODOs
- Refactoring et correction du code avec Sonar
- Intégration Continue Travis/Jenkins



Modification de l'environnement de test

Séparation des environnements

- Base PostgreSQL pour le profile de production
- Base H2 en mémoire pour profiles de test

```
<profile>
  <id>production</id>
  <properties>
    <myerp.database>MYERP</myerp.database>
  </properties>
  <dependencies>
    <!-- ===== POSTGRES -->
    <dependency>
      <groupId>org.postgresql</groupId>
      <artifactId>postgresql</artifactId>
      <scope>runtime</scope>
    </dependency>
  </dependencies>
</profile>

<!-- Profile utilisé pour lancer les tests d'intégration de la couche Business -->
<profile>
  <id>test-business</id>
  <properties>
    <spring.profiles.active>test-business</spring.profiles.active>
    <myerp.database>TEST</myerp.database>
  </properties>
  <dependencies>
    <!-- ===== H2 -->
    <dependency>
      <groupId>com.h2database</groupId>
      <artifactId>h2</artifactId>
      <scope>test</scope>
    </dependency>
  </dependencies>
</profile>
```

Passage à Junit5

- Utilisation de Junit5
- Ajout de AssertJ pour les assertions
- Mockito pour simuler les classes
- Mise à jour des différentes dépendances

```
<dependency>
  <groupId>org.junit.jupiter</groupId>
  <artifactId>junit-jupiter-engine</artifactId>
  <version>${junit-jupiter.version}</version>
  <scope>test</scope>
</dependency>

<dependency>
  <groupId>org.junit.vintage</groupId>
  <artifactId>junit-vintage-engine</artifactId>
  <version>5.1.0</version>
  <scope>test</scope>
</dependency>

<dependency>
  <groupId>org.assertj</groupId>
  <artifactId>assertj-core</artifactId>
  <version>${assertj-core.version}</version>
  <scope>test</scope>
</dependency>

<dependency>
  <groupId>org.mockito</groupId>
  <artifactId>mockito-junit-jupiter</artifactId>
  <version>${mockito-junit-jupiter.version}</version>
  <scope>test</scope>
</dependency>
```


Outils d'analyse de code

- maven-project-info-reports
- maven-surefire-report
- maven-checkstyle
- cobertura-maven
- jacoco-maven
- sonar-maven

```
<!-- ===== -->
<!-- SONARQUBE -->
<!-- ===== -->
<sonar.projectKey>pedsf1968_myerp</sonar.projectKey>
<sonar.moduleKey>${project.artifactId}</sonar.moduleKey>
<sonar.organization>pedsf1968</sonar.organization>
<sonar.host.url>https://sonarcloud.io</sonar.host.url>
<sonar.login>9e78c9485e384e93f7bf7a2f68c024a6c15c6576</sonar.login>
<sonar.language>java</sonar.language>
<sonar.java.coveragePlugin>jacoco</sonar.java.coveragePlugin>
<sonar.dynamicAnalysis>reuseReports</sonar.dynamicAnalysis>
<sonar-maven-plugin.version>3.7.0.1746</sonar-maven-plugin.version>
<sonar.coverage.jacoco.xmlReportPaths>target/site/jacoco/jacoco.xml,../target/site/jacoco
<sonar.coverage.exclusions>
  myerp-technical/src/main/java/com/dummy/myerp/technical/exception/*.java,
  myerp-technical/src/main/java/com/dummy/myerp/technical/util/spring/NullFactoryBean
  myerp-consumer/src/main/java/com/dummy/myerp/consumer/dao/impl/db/rowmapper/comptab
  myerp-consumer/src/main/java/com/dummy/myerp/consumer/dao/impl/db/DBN.java,
</sonar.coverage.exclusions>
```



Écriture des tests

Écriture des tests unitaires



- Pour les MODEL dans myerp-model
- Pour les méthodes de bases dans myerp-consumer
- Pour les méthodes de bases dans myerp-busines

Écriture des tests d'intégration



Myerp-consumer sur les
méthodes des classes

- CompteComptable
- JournalComptable
- EcritureComptable
- SequenceEcritureComptable

Myerp-business sur les
méthodes

- De vérification d'EcritureComptable
- CRUD d'EcritureComptable



Correction des bugs

EcritureComptable.getTotalCredit()



Original

```
public BigDecimal getTotalCredit() {  
    BigDecimal vRetour = BigDecimal.ZERO;  
    for (LigneEcritureComptable vLigneEcritureComptable : listLigneEcriture) {  
        // replace getDebit() call with getCredit  
        if (vLigneEcritureComptable.getCredit() != null) {  
            vRetour = vRetour.add(vLigneEcritureComptable.getDebit());  
        }  
    }  
    return vRetour;  
}
```

Correction

```
public BigDecimal getTotalCredit() {  
    BigDecimal vRetour = BigDecimal.ZERO;  
    for (LigneEcritureComptable vLigneEcritureComptable : listLigneEcriture) {  
        // replace getDebit() call with getCredit  
        if (vLigneEcritureComptable.getCredit() != null) {  
            vRetour = vRetour.add(vLigneEcritureComptable.getCredit());  
        }  
    }  
    return vRetour;  
}
```

EcritureComptable.isEquilibree()



Original

```
public boolean isEquilibree() {  
    boolean vRetour = this.getTotalDebit().equals(getTotalCredit());  
    return vRetour;  
}
```

Correction

```
public boolean isEquilibree() {  
    // equal compare string and not the value  
    // compareTo return 0 if it's the same value  
    return this.getTotalDebit().compareTo(getTotalCredit()) == 0;  
}
```

ResultSetHelper.getDate()



Original

```
public static Date getDate(ResultSet pRS, String pColName) throws SQLException {  
    Date vDate = pRS.getDate(pColName);  
    if (vDate != null) {  
        vDate = DateUtils.truncate(vDate, Calendar.DATE);  
    }  
    return vDate;  
}
```

Correction

```
public static Date getDate(ResultSet pRS, String pColName) throws SQLException {  
    if (pRS==null || pColName==null) {  
        throw new SQLException(ERROR_MESSAGE);  
    }  
  
    return pRS.getDate(pColName);  
}
```

sqlContext.xml

Original

```
<property name="sqlInsertListLigneEcritureComptable">
  <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>
</property>
```

Correction

```
<property name="sqlInsertListLigneEcritureComptable">
  <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>
</property>
```


ComptabiliteManagerImpl.updateEcritureComptable()



Original

```
public void updateEcritureComptable(EcritureComptable pEcritureComptable) {  
    TransactionStatus vTS = getTransactionManager().beginTransactionMyERP();  
    try {  
        getDaoProxy().getComptabiliteDao().updateEcritureComptable(pEcritureComptable);  
        getTransactionManager().commitMyERP(vTS);  
        vTS = null;  
    } finally {  
        getTransactionManager().rollbackMyERP(vTS);  
    }  
}
```

Correction

```
public void updateEcritureComptable(EcritureComptable pEcritureComptable) throws FunctionalException {  
    this.checkEcritureComptable(pEcritureComptable);  
    TransactionStatus vTS = getTransactionManager().beginTransactionMyERP();  
    try {  
        getDaoProxy().getComptabiliteDao().updateEcritureComptable(pEcritureComptable);  
        getTransactionManager().commitMyERP(vTS);  
        vTS = null;  
    } finally {  
        getTransactionManager().rollbackMyERP(vTS);  
    }  
}
```



Implémentation des TODOs

ComptabiliteManagerImpl.addReference()

```
public synchronized void addReference(EcritureComptable pEcritureComptable) throws FunctionalException {
    int numeroSequence = 1; // initialise with 1 to not increment
    SequenceEcritureComptable vSEC = new SequenceEcritureComptable();

    // 1. Remonter depuis la persistance la dernière valeur de la séquence du journal pour l'année de l'écriture
    Integer annee = Integer.parseInt(new SimpleDateFormat( pattern: "yyyy").format(pEcritureComptable.getDate()));
    SequenceEcritureComptable vLastSEC = getDaoProxy().getComptabiliteDao().getLastSeqOfTheYear(annee, pEcritureComptable.getJc

    // 2. S'il n'y a aucun enregistrement pour le journal pour l'année concernée :
    //     Utiliser le numéro 1. sinon dernière valeur + 1
    numeroSequence += (vLastSEC==null?0:vLastSEC.getDerniereValeur());
    vSEC.setAnnee(annee);
    vSEC.setDerniereValeur(numeroSequence);

    // 3. Mettre à jour la référence de l'écriture avec la référence calculée (RG_Compta_5)
    String reference = String.format("%s-%s/%05d",
        pEcritureComptable.getJournal().getCode(),
        annee,
        numeroSequence);
    pEcritureComptable.setReference(reference);

    // Enregistrer (insert/update) la valeur de la séquence en persistance
    getTransactionManager().beginTransactionMyERP();
    this.updateEcritureComptable(pEcritureComptable);

    if (numeroSequence == 1) {
        this.insertEcritureComptable(pEcritureComptable);
    } else {
        this.updateEcritureComptable(pEcritureComptable);
    }
}
```

ComptabiliteManagerImpl.checkEcritureComptableRG5()



```
protected void checkEcritureComptableRG5(EcritureComptable pEcritureComptable) throws FunctionalException {
    // ===== RG_Compta_5 : Format et contenu de la référence
    // vérifier que l'année dans la référence correspond bien à la date de l'écriture, idem pour le code journal...
    if (pEcritureComptable.getJournal()!=null && pEcritureComptable.getReference()!=null) {
        String code = pEcritureComptable.getReference().split( regex: "-")[0];

        if (!code.equals(pEcritureComptable.getJournal().getCode())) {
            LOGGER.error("Pas le même Code de Journal Comptable : Code {} et Code dans Reference {}",
                pEcritureComptable.getJournal().getCode(),code);
            throw new FunctionalException(
                "La référence de l'écriture comptable doit correspondre au journal.");
        }

        String year = pEcritureComptable.getReference().split( regex: "[-/]")[1];
        java.sql.Date dateComptable = pEcritureComptable.getDate();
        String dateYear = String.valueOf(dateComptable.toLocalDate().getYear());

        if (!year.equals(dateYear)) {
            LOGGER.error("Pas le même Date de Journal Comptable : Date {} et Année dans Reference {}",dateYear,year);
            throw new FunctionalException(
                "La référence de l'écriture comptable doit correspondre au journal.");
        }
    }
}
```

ComptabiliteManagerImpl.checkEcritureComptableRG7()



```
protected void checkEcritureComptableRG7(EcritureComptable pEcritureComptable) throws FunctionalException {
    List<LigneEcritureComptable> ligneEcritureComptables = pEcritureComptable.getListLigneEcriture();

    for(LigneEcritureComptable ligneEcritureComptable : ligneEcritureComptables) {
        if(ligneEcritureComptable.getCredit() != null && getNumberOfDecimalPlaces(ligneEcritureComptable.getCredit()) > 2) {
            throw new FunctionalException("Un crédit ne peut pas avoir plus de deux chiffres après la virgule!");
        }
        if(ligneEcritureComptable.getDebit() != null && getNumberOfDecimalPlaces(ligneEcritureComptable.getDebit()) > 2) {
            throw new FunctionalException("Un débit ne peut pas avoir plus de deux chiffres après la virgule!");
        }
    }
}

int getNumberOfDecimalPlaces(BigDecimal bigDecimal) {
    String string = bigDecimal.stripTrailingZeros().toString();
    int index = string.indexOf('.');
    return index < 0 ? 0 : string.length() - index - 1;
}
```


SequenceEcritureComptableRM



```
package com.dummy.myerp.consumer.dao.impl.db.rowmapper.comptabilite;

import com.dummy.myerp.consumer.dao.impl.db.DBN;
import com.dummy.myerp.model.bean.comptabilite.SequenceEcritureComptable;
import org.springframework.jdbc.core.RowMapper;

import java.sql.ResultSet;
import java.sql.SQLException;

public class SequenceEcritureComptableRM implements RowMapper<SequenceEcritureComptable> {

    @Override
    public SequenceEcritureComptable mapRow(ResultSet resultSet, int i) throws SQLException {
        SequenceEcritureComptable vBean = new SequenceEcritureComptable();

        vBean.setAnnee(resultSet.getInt(DBN.ANNEE));
        vBean.setDerniereValeur(resultSet.getInt(DBN.DERNIERE_VALEUR));

        return vBean;
    }
}
```




Intégration Continue avec Travis

Travis.yml

```
language: java
dist: trusty

# use Java 8
jdk: openjdk8

before_install:
  - chmod +x mvnw

after_success:
  - bash <(curl -s https://codecov.io/bash)

jobs:
  include:
    - stage: test
      script: ./mvnw clean test
    - stage: SonarCloud
      script: ./mvnw verify sonar:sonar
    - stage: Cobertura
      script: ./mvnw cobertura:cobertura

cache:
  directories:
    - $HOME/.m2
```

https://travis-ci.org

Search all repositories



My Repositories +

✓ pedsf1968/myerp # 76

⌚ Duration: 5 min 9 sec

📅 Finished: 5 days ago

✓ pedsf1968/escalade-rest # 3

⌚ Duration: 57 sec

📅 Finished: about a month ago

✓ pedsf1968/properties-server # 5

⌚ Duration: 1 min 10 sec

📅 Finished: 3 months ago

✓ pedsf1968/escalade-web # 2

⌚ Duration: 59 sec

📅 Finished: 4 months ago

○ pedsf1968/mastermind

⌚ Duration: -



pedsf1968 / myerp



build passing

Current Branches Build History Pull Requests

✓ develop starting README.md

↻ Commit e267871

🔗 Compare 34464a5...e267871

🔗 Branch develop

👤 DOS SANTOS FACAO

Build jobs

View config

✓ test

✓ # 76.1

AMD64

Trusty

</> JDK: openjdk8 Java

✓ Install

✓ # 76.2

AMD64

Trusty

</> JDK: openjdk8 Java

✓ SonarCloud

✓ # 76.3

AMD64

Trusty

</> JDK: openjdk8 Java

✓ Cobertura

✓ # 76.4

AMD64

Trusty

</> JDK: openjdk8 Java



Intégration Continue avec Jenkins

Docker-compose.yml



```
jenkins:
  image: jenkins/jenkins:lts
  container_name: jenkins
  privileged: true
  user: root
  restart: unless-stopped
  volumes:
    - ./docker/dev/jenkins:/var/jenkins_home
    - ./src:/var/jenkins_home/workspace/Myerp
  ports:
    - "5000:8080"
    - "443:8443"
    - "50000:50000"
  networks:
    net:
      ipv4_address: 10.5.0.4
```

Configuration – Maven Integration



Gestion des plugins

Ajouter, supprimer, activer ou désactiver des plugins qui peuvent étendre les fonctionnalités de Jenkins.

Maven

Installations Maven

Ajouter Maven

Liste des installations Maven sur ce système

Q maven

Mises à jour

Disponibles

Installés

Avancé

Installer Nom

Installer sans redémarrer

Télécharger maintenant et installer

Ajouter Maven

Maven

Nom Maven_3.6.6

☒ Install automatically

Install from Apache

Version 3.6.3

Configuration – JDK



Configuration globale des outils

Configurer les outils, leur localisation et les installateurs automatiques.



Enter Your Oracle Account

To access older JDK versions, you need to have an [Oracle Account](#).

Username

Password

OK

JDK

Installations JDK

Ajouter JDK

Liste des installations JDK sur ce système

Ajouter JDK

JDK

Nom

JDK_8.221

☒ Install automatically

☐ Install Oracle Java SE Development Kit from the website

Version

☒ J'approuve l'accord de licence Java SE Development Kit

 **Installing JDK requires Oracle account. Please enter your username/password**

Oracle Java SE 11+ is not available for business, commercial or production use without a commercial license. For more information, see [Oracle Java SE Licensing FAQ](#).
Public updates for Oracle Java SE 8 released after January 2019 will not be available for business, commercial or production use without a commercial license. For more information, see [Oracle Java SE Licensing FAQ](#).

Ajouter un installateur ▾

Configuration – JOB



Saisissez un nom

» Champ obligatoire



Construire un projet free-style

Ceci est la fonction principale de Jenkins qui sert à builder (construire) votre projet. Vous pouvez tout autre chose qu'un build logiciel.



Construire un projet maven

Construit un projet avec maven. Jenkins utilise directement vos fichiers POM et minimise les dépendances.



Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building complex workflows.



Construire un projet multi-configuration


Build


POM Racine


Goals et options


Configuration – Build





 Retour au tableau de bord


 **État**


 Modifications


 Répertoire de travail


 Lancer un build


 Supprimer Maven project


 Configurer

 Modules


 Rename


 **Historique des builds** [tendance](#)


 **#4** 1 juil. 2020 17:35


 **#3** 1 juil. 2020 17:35

Maven project Myerp

 [Espace de travail](#)

 [Changements récents](#)

 [Derniers résultats des tests](#) (aucune erreur)

 [Derniers résultats des tests](#) (aucune erreur)

Liens permanents

- [Dernier build \(#4\), il y a 21 mn](#)



SonarQube

https://sonarcloud.io/

