

Documentation des Tests Unitaires

Résumé

Métrique	Valeur
Framework	JUnit 5 (Jupiter)
Total Tests	11
Réussis	11 oui
Échecs	0
Temps d'exécution	~0.1s

Structure des Tests

```
src/test/java/com/portfoliotracker/  
├─ model/  
│   └─ AssetTest.java          (5 tests)  
└─ service/  
    └─ EncryptionServiceTest.java (6 tests)
```

AssetTest.java

Classe testée: `com.portfoliotracker.model.Asset`

Objectif: Vérifier les calculs financiers sur les actifs (quantités, prix moyen, investissement).

Tests

Méthode	Description	Input	Expected
testGetTotalQuantity_withBuyAndSell	Calcul quantité avec achats et ventes	BUY 2 + BUY 1.5 - SELL 0.5	3.0
testGetTotalQuantity_withRewards	Prise en compte des rewards	REWARD 0.01 + BUY 1.0	1.01
testGetAverageBuyPrice	Prix moyen d'achat pondéré	BUY 1@40000 + BUY 1@50000	45000
testGetTotalInvested	Total investi (incluant frais)	BUY 1@40000+100 + BUY 0.5@50000+50	65150
testEmptyAsset	Asset sans transactions	(vide)	0 pour tous

Code Exemple

```
@Test
void testGetAverageBuyPrice() {
    asset.addTransaction(new Transaction(TransactionType.BUY, 1.0, 40000, ...));
    asset.addTransaction(new Transaction(TransactionType.BUY, 1.0, 50000, ...));

    assertEquals(45000, asset.getAverageBuyPrice(), 0.001);
}
```

EncryptionServiceTest.java

Classe testée: com.portfoliotracker.service.EncryptionService

Objectif: Vérifier le chiffrement XOR symétrique pour la protection des données.

Tests

Méthode	Description	Résultat Attendu
testEncryptDecrypt_roundTrip	Chiffrer puis déchiffrer	Données originales récupérées
testEncryptDecrypt_withSpecialCharacters	Caractères spéciaux (€, accents)	Préservation des caractères
testEncryption_changesCiphertext	Le chiffrement modifie les données	Texte chiffré ≠ texte original
testSetPassphrase_enablesEncryption	Activation via passphrase	isEncryptionEnabled() = true
testEmptyPassphrase_disablesEncryption	Passphrase vide désactive	isEncryptionEnabled() = false
testEncrypt_withEmptyKey_throwsException	Clé vide lève exception	IllegalArgumentException

Code Exemple

```
@Test
void testEncryptDecrypt_roundTrip() {
    String originalText = "Hello Portfolio Tracker!";
    String passphrase = "secretKey123";

    byte[] encrypted = encryptionService.encrypt(originalText.getBytes(), passphrase);
    byte[] decrypted = encryptionService.decrypt(encrypted, passphrase);

    assertEquals(originalText, new String(decrypted));
}
```

Exécution des Tests

```
# Exécuter tous les tests
mvn test

# Exécuter avec détails
mvn test -Dsurefire.useFile=false

# Exécuter un test spécifique
mvn test -Dtest=AssetTest
```

Couverture

Composant	Couvert par Tests
Model.Asset	oui - Calculs financiers
Service.EncryptionService	oui - Encryption/Decryption
Service.EventService	Attention: Non (dépendance fichiers)
Service.PortfolioService	Attention: Non (dépendance persistance)
Controllers	non - Non (nécessite JavaFX Test)

Résultats (14 janvier 2026)

```
[INFO] -----  
[INFO]  T E S T S  
[INFO] -----  
[INFO] Running com.portfoliotracker.model.AssetTest  
[INFO] Tests run: 5, Failures: 0, Errors: 0, Skipped: 0  
[INFO] Running com.portfoliotracker.service.EncryptionServiceTest  
[INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0  
[INFO]  
[INFO] Results:  
[INFO] Tests run: 11, Failures: 0, Errors: 0, Skipped: 0  
[INFO] BUILD SUCCESS
```