

Junit : Framework pour les tests unitaires

```
setUp (@Before) : méthode appelée avant chaque test
tearDown (@After): méthode appelée après chaque test
@BeforeClass : méthode static appelée une seule fois avant tous les tests
@AfterClass : méthode static appelée une seule fois après tous les tests
testXXX (@Test) : méthode de test
```

Les Assertions : méthodes héritées de TestCase

```
assertTrue(String message, boolean b)
assertNull(String message, Object o)
assertNotNull(String message, Object o)
assertEquals(String message, Object test, Object atester) : test equals
assertSame(String message, Object test, Object atester) : test ==
assertArrayEquals( String message, Object test, Object atester) :
    pour les éléments de tableaux
...
fail (@Test(expected=XXXException.class)) :
    test si la méthode lève bien une exception
@Test(timeout=1000) : test que la méthode s'exécute en X ms.
```

Suite de tests :

```
@RunWith(Suite.class)
@SuiteClasses({TestClassXXX.class, TestClassYYY.class})
```

EasyMock

on définit l'implémentation à simuler et on lui donne les valeurs des tests pour la réalisation des tests unitaires avant implémentation :

```
ICalcul monMock = EasyMock.createMock(ICalcul.class);
EasyMock.expect(monMock.add(1, 5)).andReturn(6);
EasyMock.expect(monMock.pow(2.0, 10.0)).andReturn(1024.0);
EasyMock.expect(monMock.div(10.0, 5.0)).andReturn(2.0);
EasyMock.expect(monMock.div(10.0, 0.0)).andThrow(new ArithmeticException());
EasyMock.replay(monMock);
...
Et à la fin d'un test on vérifie que toutes les méthodes du Mock ont bien été
    invoquées :
EasyMock.verify(monMock);
```

MAVEN

Maven :
outil de build automatisé
Phases importantes de
construction du projet :

validate : validation du projet
compile : compilation du projet
test : compilation des tests et passage de ceux ci
package : mise en forme du livrable (jar, war, ear ...)
integration-test : tests qui doivent passer sur un environnement
verify : vérifier la qualité du livrable selon un série de critères
install : dépose le package livrable dans le repository local
deploy : publier le package sur un environnement partagé

Creation de projet

```
Projet java
C :>mvn archetype:create -DgroupId=fr.afcepf.qualimetrie -DartifactId=monProjet
Projet Ejb3
C :>mvn archetype:generate -DarchetypeGroupId=org.codehaus.mojo.archetypes -DarchetypeArtifactId=ejb-jee5
-DgroupId=fr.afcepf.qualimetrie -DartifactId=banque-facade
Projet Web
C:>mvn archetype:generate -DarchetypeArtifactId=maven-archetype-webapp -DgroupId=fr.afcepf.qualimetrie
-DartifactId=banque-web
```

Dépendances

scope :

compile (par défaut) : la dépendance est nécessaire à la compilation et l'exécution
provided : la dépendance est nécessaire mais ne fait pas partie du package final
runtime : utile que pour l'exécution, pas pour la compilation
test : uniquement pour les tests unitaires
system : comme runtime mais la dépendance doit être fournie manuellement

Recherche de dépendances

- <http://mvnrepository.com>
- <http://search.maven.org>

Configuration pour release

settings.xml

```
<server>
  <id>gizmo-repository</id>
  <username>admin</username>
  <password>*****</password>
</server>
```

pom.xml

```
<profiles>
  <profile>
    <id>gizmo-distribution</id>
    <distributionManagement>
      <snapshotRepository>
        <id>gizmo-repository</id>
        <name>Gizmo Repository</name>
        <url>http://localhost:9292/archiva/repository/snapshots/</url>
      </snapshotRepository>
    </distributionManagement>
  </profile>
  <profile>
    <id>gizmo-release</id>
    <distributionManagement>
      <repository>
        <id>gizmo-repository</id>
        <name>Gizmo Repository</name>
        <url>http://localhost:9292/archiva/repository/internal/</url>
      </repository>
    </distributionManagement>
  </profile>
</profiles>
<scm>
  <connection>scm:svn:http://repo-jenkins.googlecode.com/svn/trunk/resanet</connection>
  <developerConnection>
    scm:svn:https://repo-jenkins.googlecode.com/svn/trunk/resanet
  </developerConnection>
</scm>
```

Commandes pour release

```
C:>svn checkout http://repo-jenkins.googlecode.com/svn/trunk/resanet
C:>mvn release:prepare -DtagBase=https://repo-jenkins.googlecode.com/svn/trunk/resanet/tags
C:>mvn release:perform -Dgoals=install
```

Commandes pour verification

```
C:>svn log http://repo-jenkins.googlecode.com/svn/trunk/resanet/pom.xml
C:>svn log http://repo-jenkins.googlecode.com/svn/tags
```

Projet parent :

- banque
 - pom.xml

```
<modelVersion>4.0.0</modelVersion>
<groupId>fr.afcepf.qualimetrie</groupId>
<artifactId>banque</artifactId>
<version>1.0-SNAPSHOT</version>
<packaging>pom</packaging>
<name>banque</name>
<url>http://maven.apache.org</url>
<modules>
  <module>banque-web</module>
  <module>banque-ear</module>
  <module>banque-data</module>
  <module>banque-business</module>
  <module>banque-facade</module>
  <module>assurance-data</module>
  <module>assurance-business</module>
</modules>
```

Projet Ear :

- banque-ear
 - pom.xml

```
<modelVersion>4.0.0</modelVersion>
<parent>
  <groupId>fr.afcepf.qualimetrie</groupId>
  <artifactId>banque</artifactId>
  <version>1.0-SNAPSHOT</version>
</parent>
<artifactId>banque-ear</artifactId>
<packaging>ear</packaging>
<name>banque-ear</name>
<url>http://maven.apache.org</url>
<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
</properties>
<dependencies>
  <dependency>
    <groupId>fr.afcepf.qualimetrie</groupId>
    <artifactId>banque-web</artifactId>
    <version>1.0-SNAPSHOT</version>
    <type>war</type>
  </dependency>
  <dependency>
    <groupId>fr.afcepf.qualimetrie</groupId>
    <artifactId>banque-data</artifactId>
    <version>1.0-SNAPSHOT</version>
    <type>ejb</type>
  </dependency>
  ...
</dependencies>
```

plugins

javadoc

```
<reporting>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-javadoc-plugin</artifactId>
      <version>2.9.1</version>
      <configuration>
        <link>http://java.sun.com/j2se/1.5.0/docs/api</link>
      </configuration>
    </plugin>
  </plugins>
</reporting>
```

compiler

```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>2.0.2</version>
      <configuration>
        <source>1.5</source>
        <target>1.5</target>
      </configuration>
    </plugin>
  </plugins>
</build>
```

source plugin :

```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-source-plugin</artifactId>
      <version>2.2.1</version>
      <executions>
        <execution>
          <id>attach-sources</id>
          <phase>package</phase>
          <goals>
            <goal>jar-no-fork</goal>
          </goals>
        </execution>
        <execution>
          <id>attach-test-sources</id>
          <phase>package</phase>
          <goals>
            <goal>test-jar-no-fork</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
```

pour un jar executable
maven-shade-plugin pour inclure les dependances
maven-jar-plugin pour definir le classPath (par exemple)

```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-shade-plugin</artifactId>
      <version>2.1</version>
      <executions>
        <execution>
          <id>attach-sources</id>
          <phase>package</phase>
          <goals>
            <goal>shade</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-jar-plugin</artifactId>
      <version>2.4</version>
      <configuration>
        <archive>
          <manifest>
            <addClasspath>true</addClasspath>
            <mainClass>com.zenika.formation.App</mainClass>
          </manifest>
          <manifestEntries>
            <VersionId>${project.version}</VersionId>
            <ClassPathPrefix>${settings.localRepository}</ClassPathPrefix>
          </manifestEntries>
        </archive>
      </configuration>
    </plugin>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-source-plugin</artifactId>
      <version>2.2.1</version>
      <executions>
        <execution>
          <id>attach-sources</id>
          <phase>package</phase>
          <goals>
            <goal>jar-no-fork</goal>
          </goals>
        </execution>
        <execution>
          <id>attach-test-sources</id>
          <phase>package</phase>
          <goals>
            <goal>test-jar-no-fork</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
```

checkstyle-config (projet séparé):

- checkstyle-config
 - config
 - checkstyle.xml
 - src
 - pom.xml

```
<modelVersion>4.0.0</modelVersion>
<groupId>fr.afcepf.qualimetric</groupId>
<artifactId>checkstyle-config</artifactId>
<version>1.0</version>
<packaging>jar</packaging>
<build>
  <resources>
    <resource>
      <directory>config</directory>
    </resource>
  </resources>
</build>
```

projet avec checkstyle prédéfini :

- projet
 - src
 - main
 - test
 - pom.xml

```
<reporting>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-checkstyle-plugin</artifactId>
      <version>2.10</version>
      <configuration>
        <configLocation>checkstyle.xml</configLocation>
      </configuration>
    </plugin>
  </plugins>
</reporting>
<build>
  <extensions>
    <extension>
      <groupId>fr.afcepf.qualimetric</groupId>
      <artifactId>checkstyle-config</artifactId>
      <version>1.0</version>
    </extension>
  </extensions>
</build>
```