

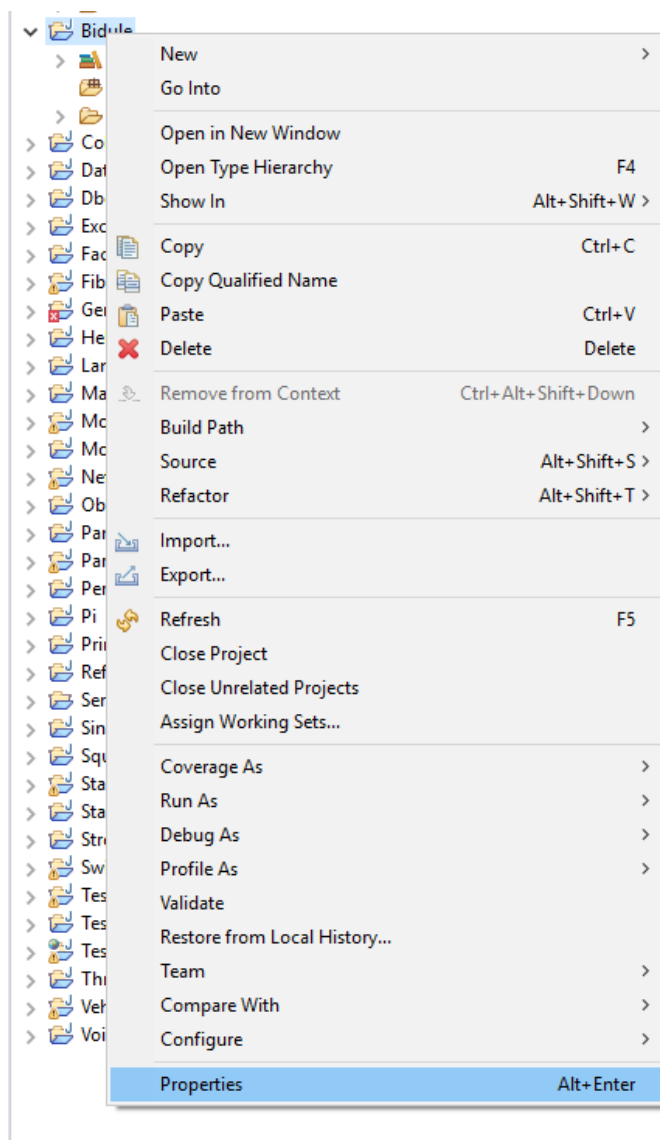
Tests Unitaires

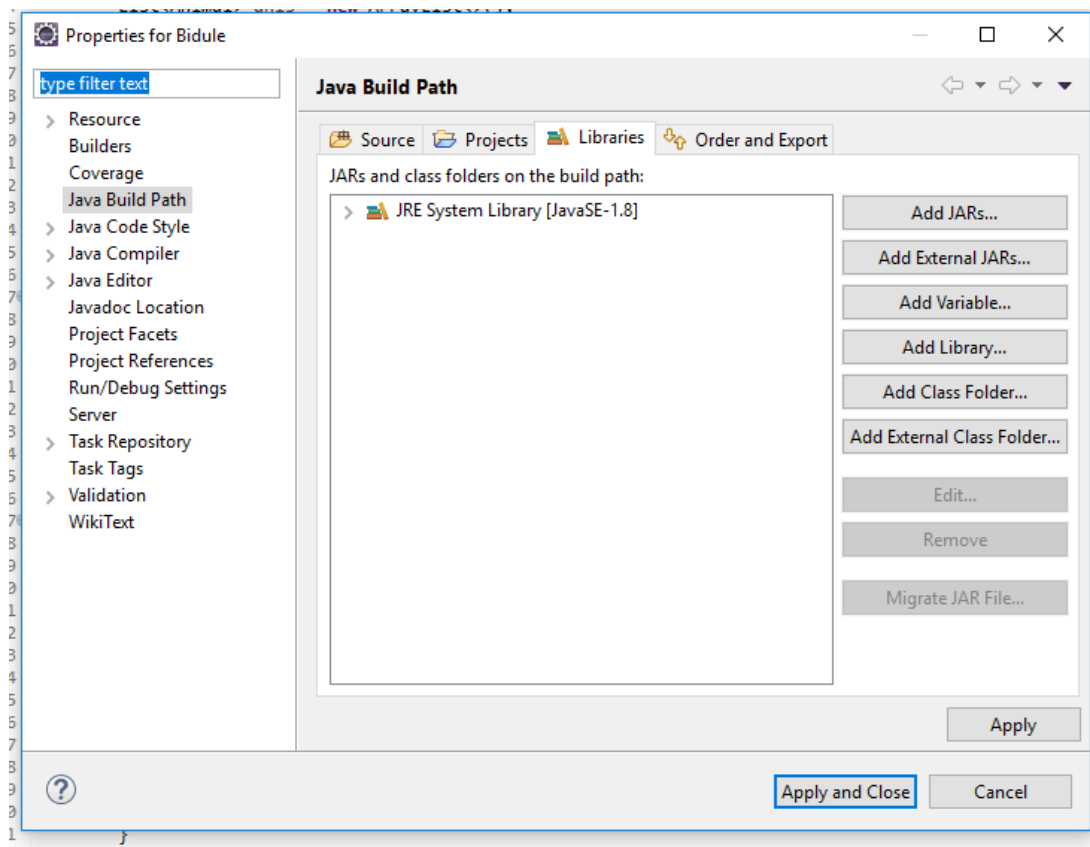
Intro

Les Tests unitaires servent à automatiser les tests d'objets. Par exemple, dans le cas des DAO, on vérifie que les méthodes des DAO font bien ce qu'on attend d'elles.

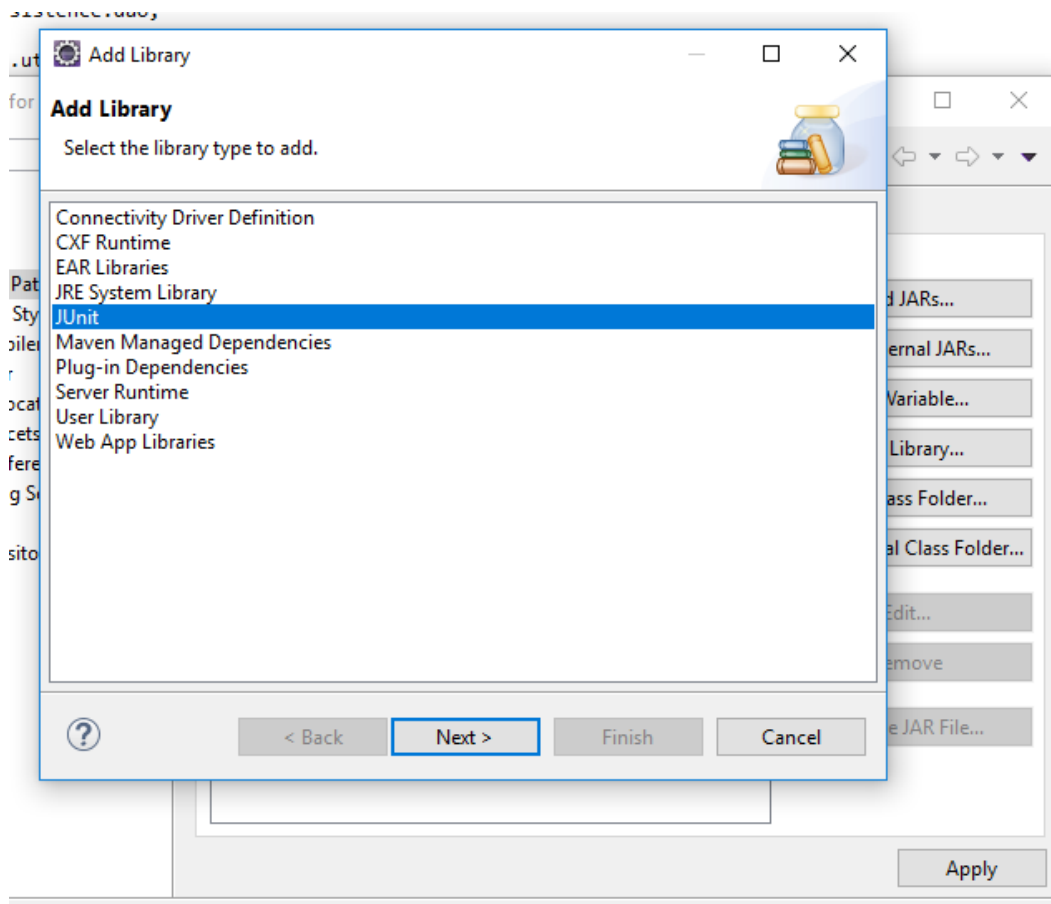
Mise en place

Click droit sur le **projet/properties/java build path/libraries**





Click sur **Add Library** et choix de **JUnit**



Puis choisir **JUnit 4**.

Comment ça marche ?

Pour qu'une classe soit considérée comme une classe JUnit, il faut qu'elle dérive de **TestCase**.

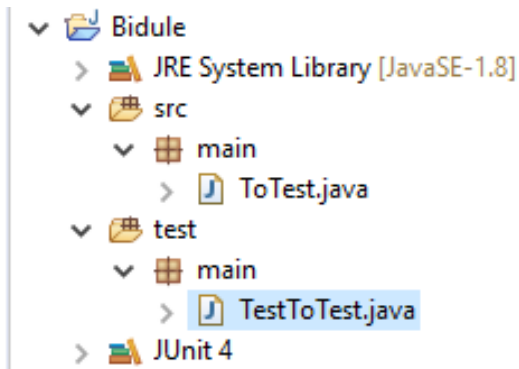
Pour qu'une méthode dans une classe JUnit soit exécutée, il faut que son nom soit de la forme **test...**, qu'elle soit sans paramètre et de type **public void**.

On a quelque chose comme ceci :

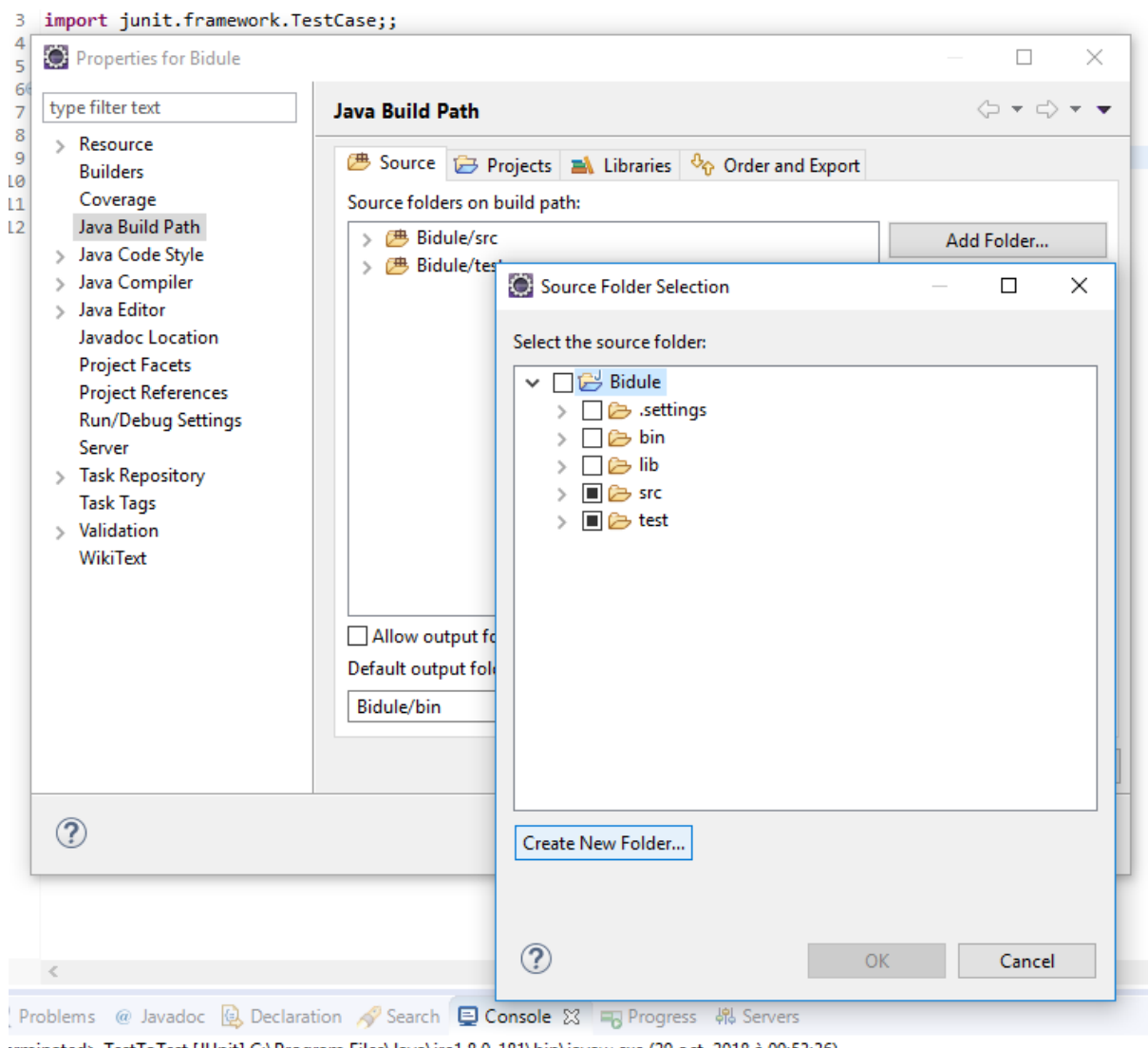
```
Class myTest extends TestCase {  
    public void testSomething() {  
        System.out.println("Je suis JUnit") ;  
    }  
}
```

Règles d'écriture

On écrit toutes les classes de tests dans une autre source que les autres classes (celles qui ne sont pas des classes de test). On appelle cette source **test**.

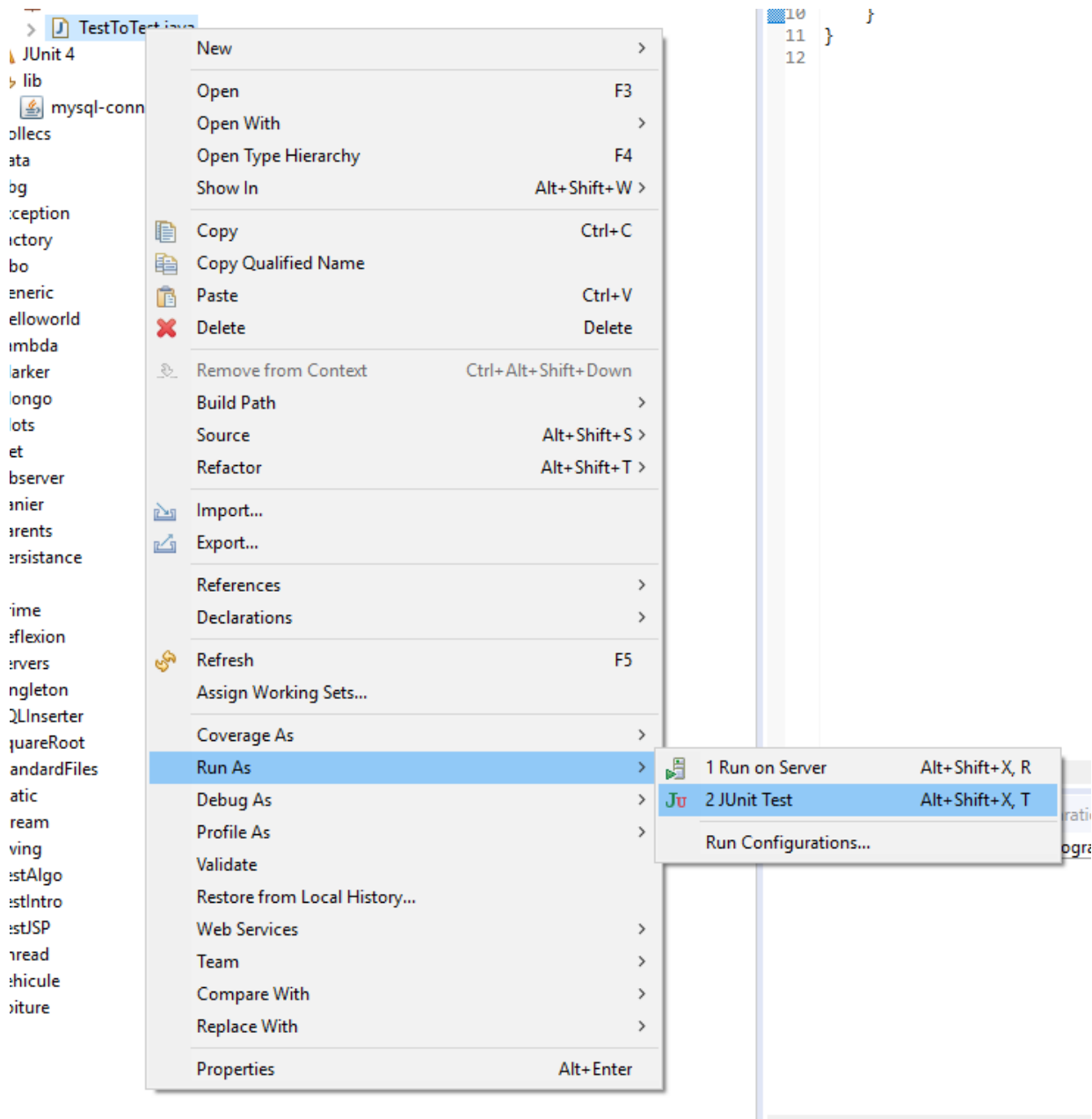


Pour créer la nouvelle source : Click droit sur le **projet/properties/java build path/source** et on crée un nouveau répertoire (**test**, donc).

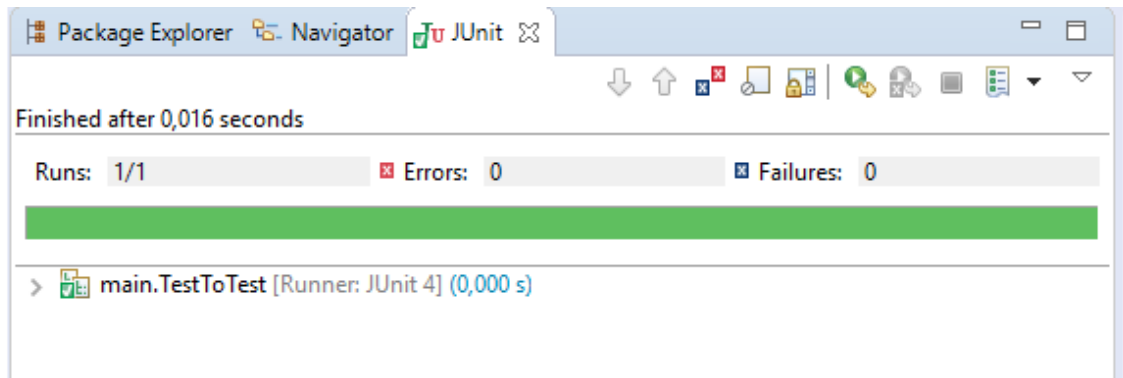


Comment lance-t'on une classe JUnit ?

On se positionne sur la classe de test et on fait : **click droit+Run as/JUnit test**



On arrive alors sur le lanceur JUnit qui lance toutes les méthodes de la classe de test.



Les asserts

Le but des classes et des méthodes de test est de vérifier que les méthodes qu'on teste se comportent bien comme elles ont censées se comporter.

On teste en général les retours des méthodes et on vérifie qu'on a les valeurs attendues. Pour cela, on utilise les **assert...** et en particulier les **assertEquals** qui testent que deux valeurs sont égales.

Ce n'est pas clair ?

Prenons un exemple

Soit la classe **ToTest**.

```
public class ToTest {  
    public ToTest() {  
        // TODO Auto-generated constructor stub  
    }  
    public int inc(int i) {  
        return i+1;  
    }  
}
```

On voit qu'elle n'a qu'une méthode qui incrémente la valeur qu'on lui passe en paramètre.

On va donc tester cette méthode et vérifier qu'elle incrémente bien la valeur.

On a donc la classe de test suivante :

```
import junit.framework.TestCase;;  
  
public class TestToTest extends TestCase {  
    public void testInc() {  
        int i = 1;  
        int j = new ToTest().inc(i);  
        assertEquals(j,i+1);  
    }  
}
```

On voit ici que le **assertEquals** teste bien que **j** (retour de **inc()**) est égal à **i+1**.

En plus

Dans une classe dérivant de **TestCase**, Il existe 2 fonctions qui sont appelées avant et après toute fonction de test (de type **test...()**).

Il s'agit respectivement de :

- `setUp()`
- `tearDown()`