



## RAPPORT PROJET D'ATELIER DE GENIE LOGICIEL



Université Cheikh Anta Diop de DAKAR

Section Informatique / Master 1 Sir-Soir

Participants : Khalifa Ababacar Sy SAMB

Demba TOURE

Djibril NDIAYE

29 NOVEMBRE 2020

Encadreur : Mr SAMB

# **PLAN**

## **INDRODUCTION**

### **I. PROJET JAVA MAVEN**

### **II. TESTS UNITAIRES**

### **III. GITHUB & GIT**

### **IV. CREATION DU FICHIER DOCKERFILE**

### **V. GITHUB ACTIONS**

### **VI. DOCKER HUB AVEC GITHUB ACTIONS**

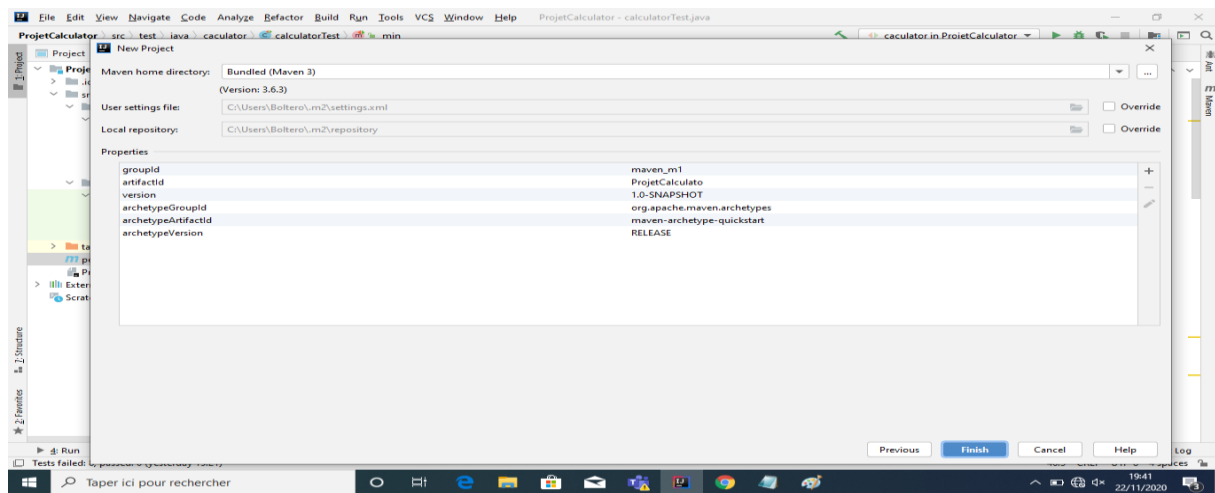
## **CONCLUSION**

## **INDRODUCTION**

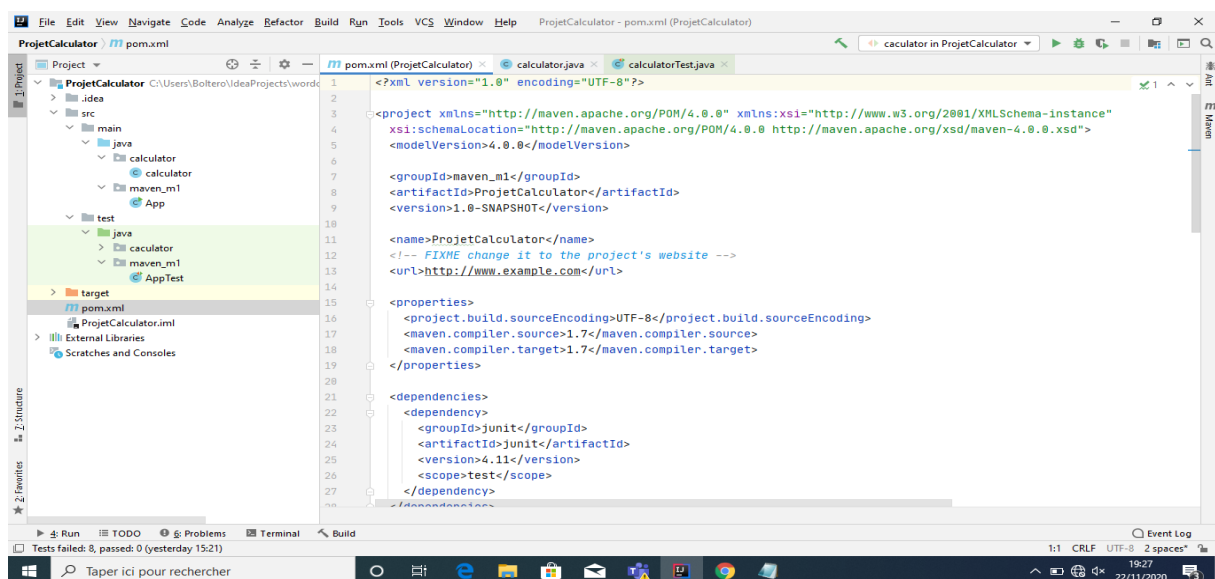
Le génie logiciel, l'ingénierie logicielle ou l'ingénierie du logiciel (en anglais: *software engineering*) est une science qui étudie les méthodes de travail et les bonnes pratiques des ingénieurs qui développent des logiciels.

Le génie logiciel s'intéresse en particulier aux procédures systématiques qui permettent d'arriver à ce que des logiciels de grande taille correspondent aux attentes du client, soient fiables, aient un coût d'entretien réduit et de bonnes performances tout en respectant les délais et les coûts de construction.

### **I. CREATION DU PROJET JAVA MAVEN**



➤ **Fichier pom :**



## ➤ Création de la classe Calculator

```
package calculator;

public class Calculator {
    public int sub(int a, int b) { return a + b ; }
    public int minus(int a, int b) { return a - b ; }
    public int multiply(int a, int b) { return a*b ; }
    public int divide(int a, int b){
        return a/b ;
    }
    public int min(int a, int b) { return Math.min(a,b); }
    public int max(int a, int b) { return Math.max(a,b); }

    public int minElement(int[] list) {
        int k = list[0];
        for (int i = 1; i < list.length; i++) {
            if ( list[i]<k) {
                k = list[i];
            }
        }
        return k;
    }

    public int maxElement(int[] list){
        int k =list[0];
        for(int i=1; i<list.length; i++){
            if(list[i] >k){
                k=list[i];
            }
        }
    }
}
```

```

    public int min(int a, int b){
        return Math.min(a,b);
    }
    public int max(int a, int b){
        return Math.max(a,b);
    }

    public int minElement(int[] list) {
        int k = list[0];
        for (int i = 1; i < list.length; i++) {
            if ( list[i]<k) {
                k = list[i];
            }
        }
        return k;
    }

    public int maxElement(int[] list){
        int k =list[0];
        for(int i=1; i<list.length; i++){
            if(list[i] >k){
                k=list[i];
            }
        }
        return k;
    }
}

```

## ➤ Création de la classe CalculatorTest

```

import calculator.Calculator;
import org.junit.Assert;
import org.junit.Before;
import org.junit.Test;

public class CalculatorTest {
    Calculator calculator;
    @Before
    public void setUp() throws Exception{
        calculator= new Calculator();
    }

    @Test
    public void testSub() {
        int resultat=calculator.sub( a: 3, b: 2);
        if(resultat !=5){
            Assert.fail();
        }
    }

    @Test
    public void testMinus() {
        Assert.assertEquals( expected: 0, calculator.minus( a: 7, b: 7));
    }
}

```

```

@Test
public void testMultiply() { Assert.assertEquals( expected: 18, calculator.multiply( a: 9, b: 2)); }

@Test
public void testDivide() {
    int result= calculator.divide( a: 15, b: 3);
    if (result !=5){
        Assert.fail();
    }
}

/* @Test(expected = ArithmeticException.class)
public void divideWiththrowExeption() {
    calculator.divide(6,0);
}
*/

@Test
public void testMin() {
    int min=calculator.min( a: 10, b: 20);
    if (min !=10) Assert.fail();
}

@Test
public void testMax() {
    int max=calculator.max( a: 2000, b: -2000);
    Assert.assertEquals( expected: 2000,max);
}

@Test
public void testMinElement() {
    int tab[]={
        1,3,5,9
    };
    int a=calculator.minElement(tab);
    Assert.assertEquals( expected: 1,a);
}

@Test
public void testMaxElement(){
    int tab[]={
        1,3,5,9
    };
    int a =calculator.maxElement(tab);
    Assert.assertEquals( expected: 9,a);
}

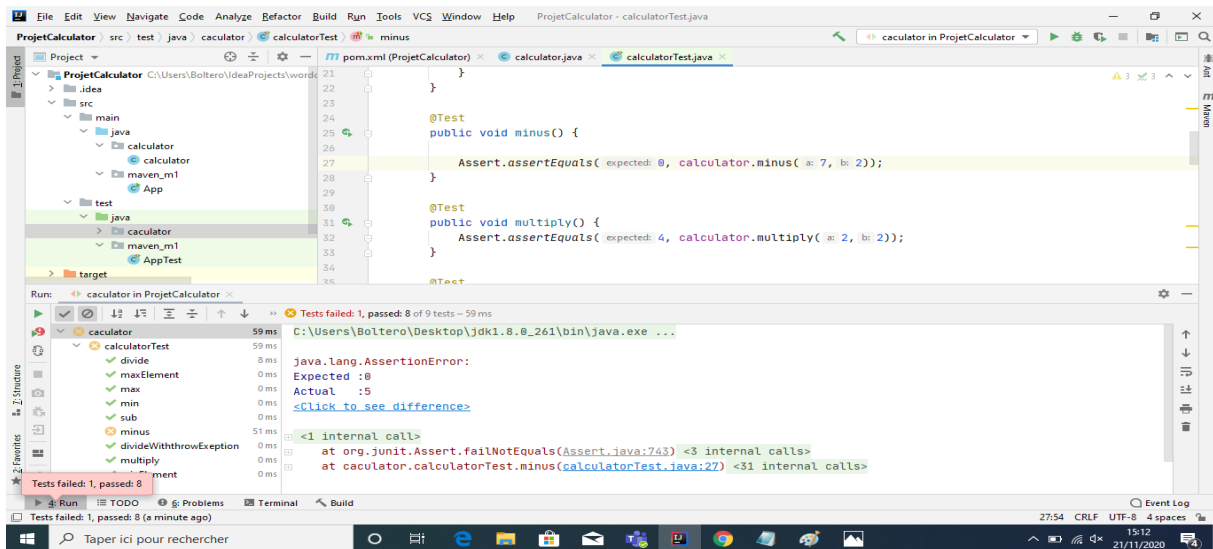
```

## II. REALISATION DES TESTES UNITAIRES

Une série de tests ont été réalisés avec ou sans bugs.

## ■ Test sur la méthode minus ()

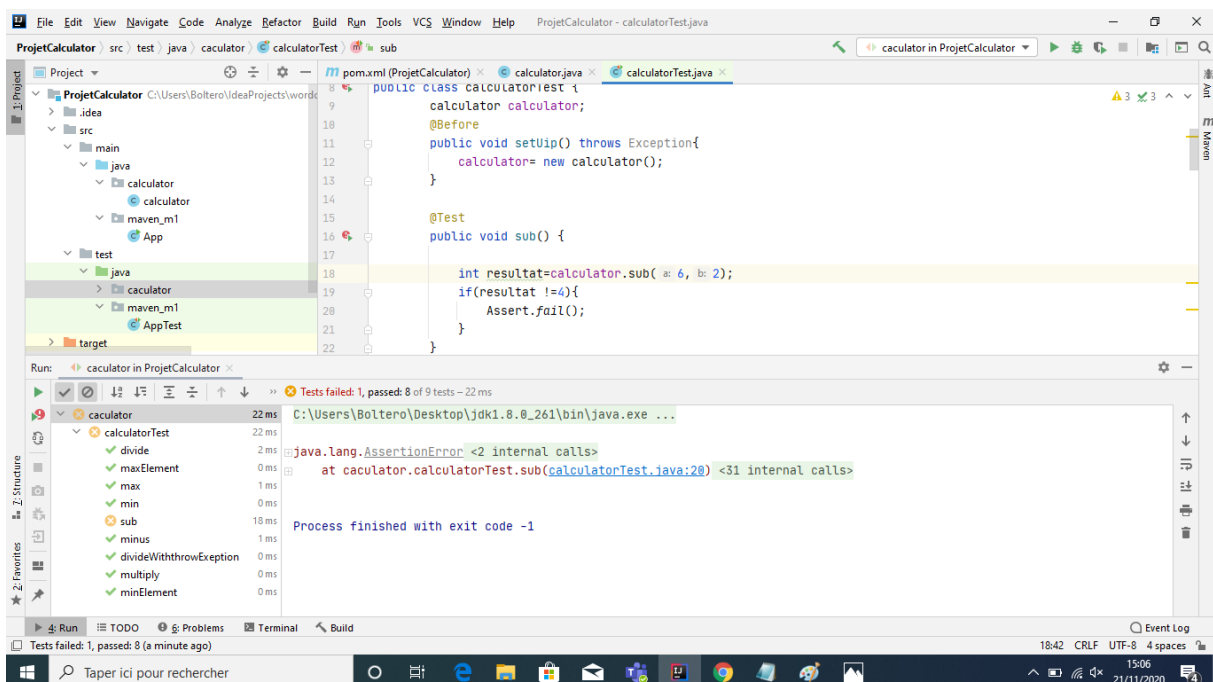
Comme **assertEquals** attend zéro 0 comme résultat, alors (7 - 2 = 5) entraîne une erreur après l'exécution du build



## ■ Test sur la méthode sub ()

Le résultat de l'addition est différent de 4 d'où l'erreur.

On garde toujours le test sur minus

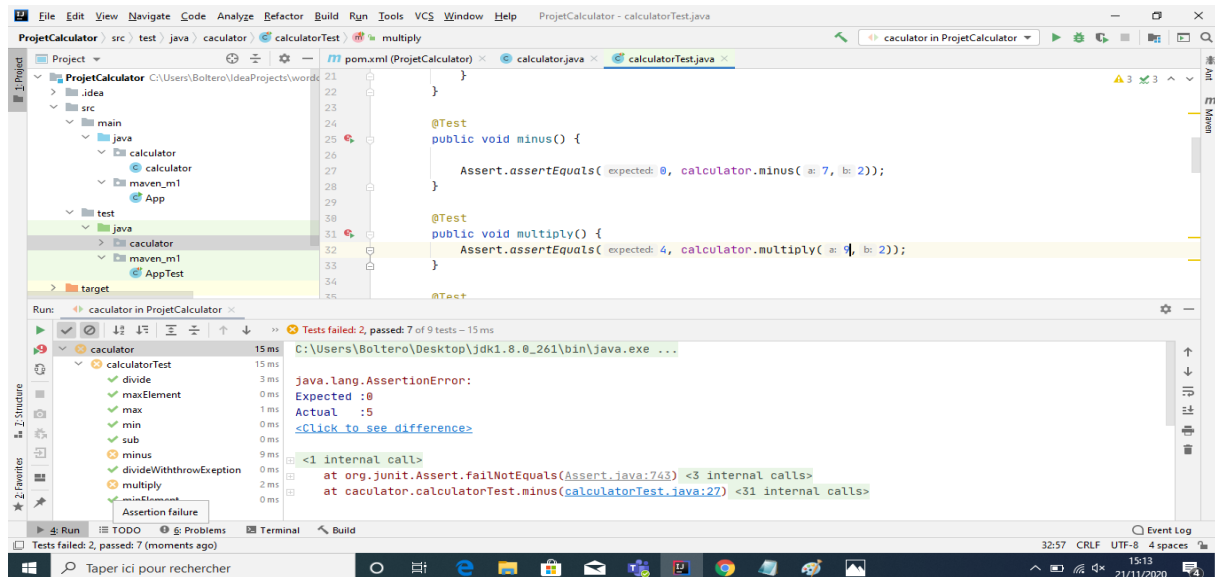




## ■ Test sur la méthode multiply ()

**assertEquals** attend zéro 4 comme résultat, alors ( $9 * 2 = 18$ ) entraine une erreur après l'exécution du build

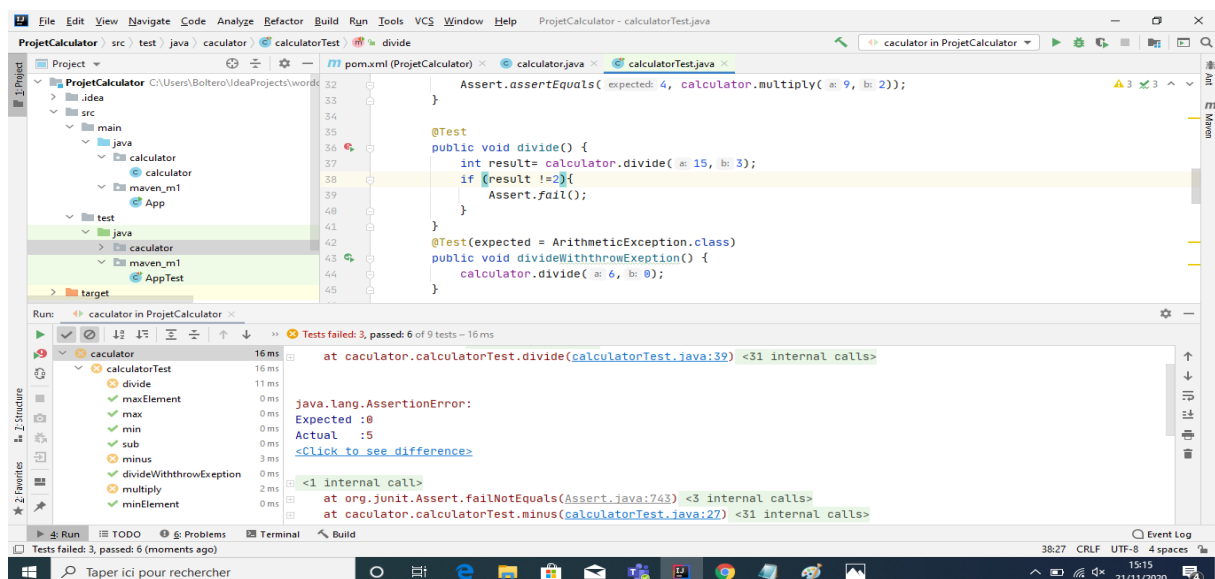
On garde toujours le test sur minus



## ■ Test sur la méthode divide ()

**Assert.fail ()** renvoie une erreur lorsque le résultat de la division est différent de 2, ce qui est le cas ici.

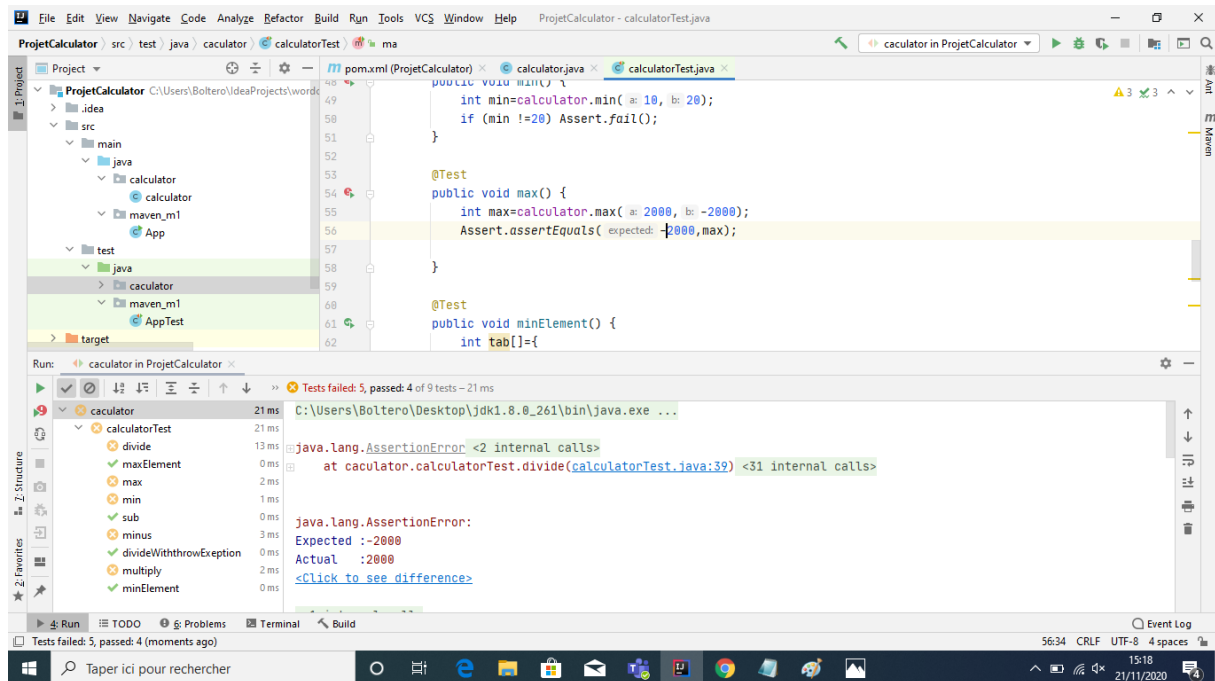
On garde toujours les tests sur minus et multiply



## ■ Test sur les méthodes min () et max ()

Pour `min ()` **Assert.fail ()** renvoie une erreur car le résultat est différent de 20, et pour `max ()` **assertEquals** attend -2000 comme résultat d'où l'erreur

On garde toujours les tests sur min, multiply et divide.

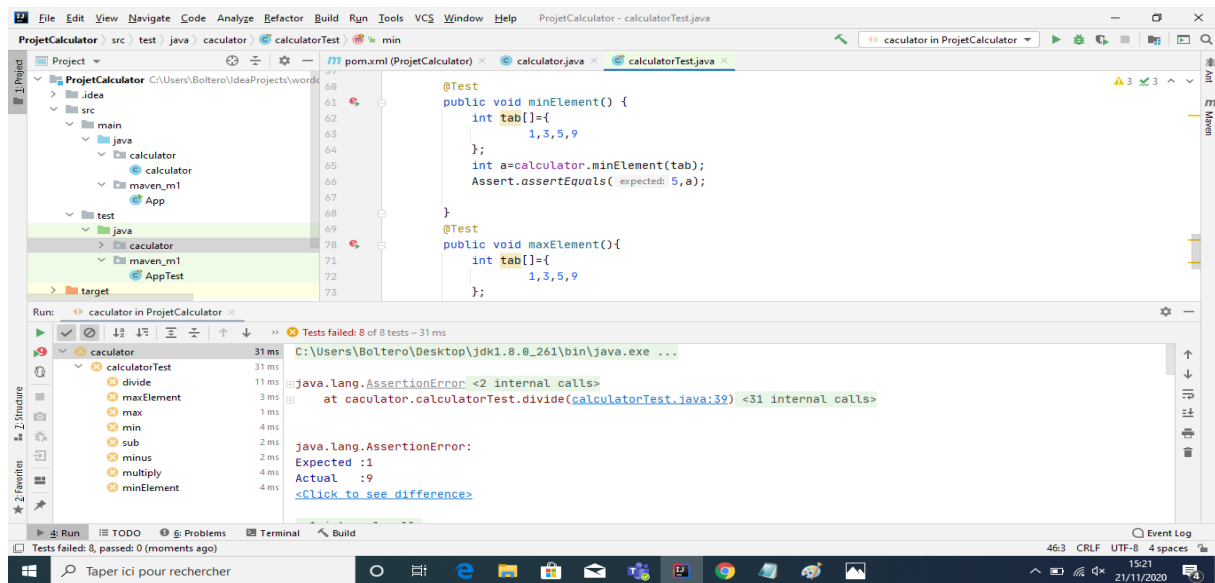


## ■ Test sur les méthodes `minElement ()` et `maxElement ()`

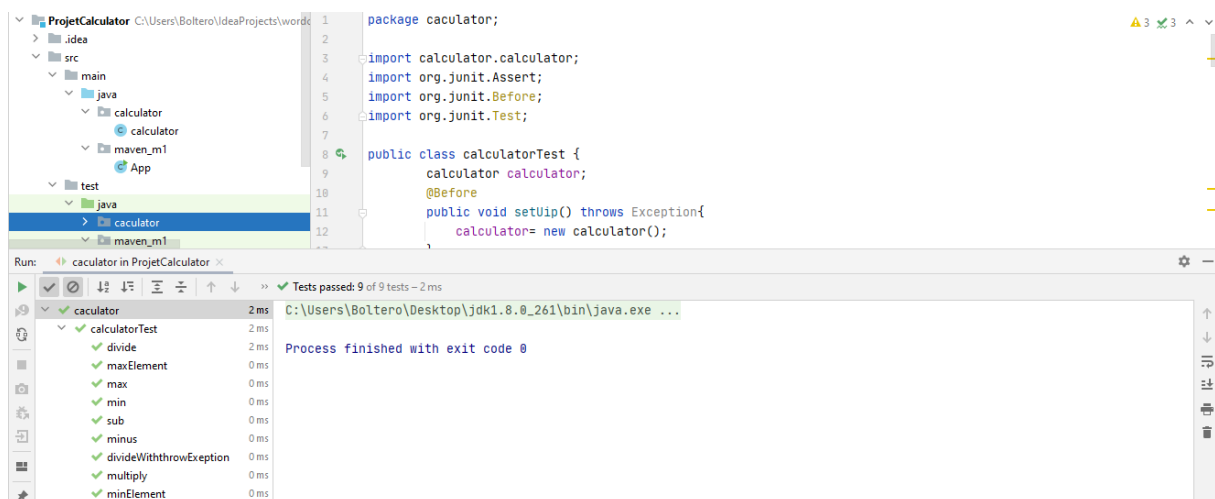
**minElement** : Le tableau d'élément renvoie 1 comme élément le plus petit alors que **assertEquals** attend 5 et donc erreur après l'exécution.

**maxElement** : Le tableau d'élément renvoie 9 comme plus grand élément alors que **assertEquals** attend 1 et donc erreur après l'exécution.

On garde toujours les tests sur min, multiply, divide min et max.



- A présent les tests réalisés donnent tous des valeurs attendues.



### III. GITHUB & GIT

#### ➤ GITHUB

Github est un outil qui va permettre de versionner votre code, c'est-à-dire que l'outil va gérer les différentes versions de votre code au fur et à mesure de son évolution. Il est très important de versionner son code aussi bien sur un projet solo et encore plus sur un projet d'équipe.

Grâce aux outils qu'elle fournit pour gérer les conflits éventuels résultant des changements apportés par plusieurs développeurs, il est possible de collaborer efficacement sur un même projet.

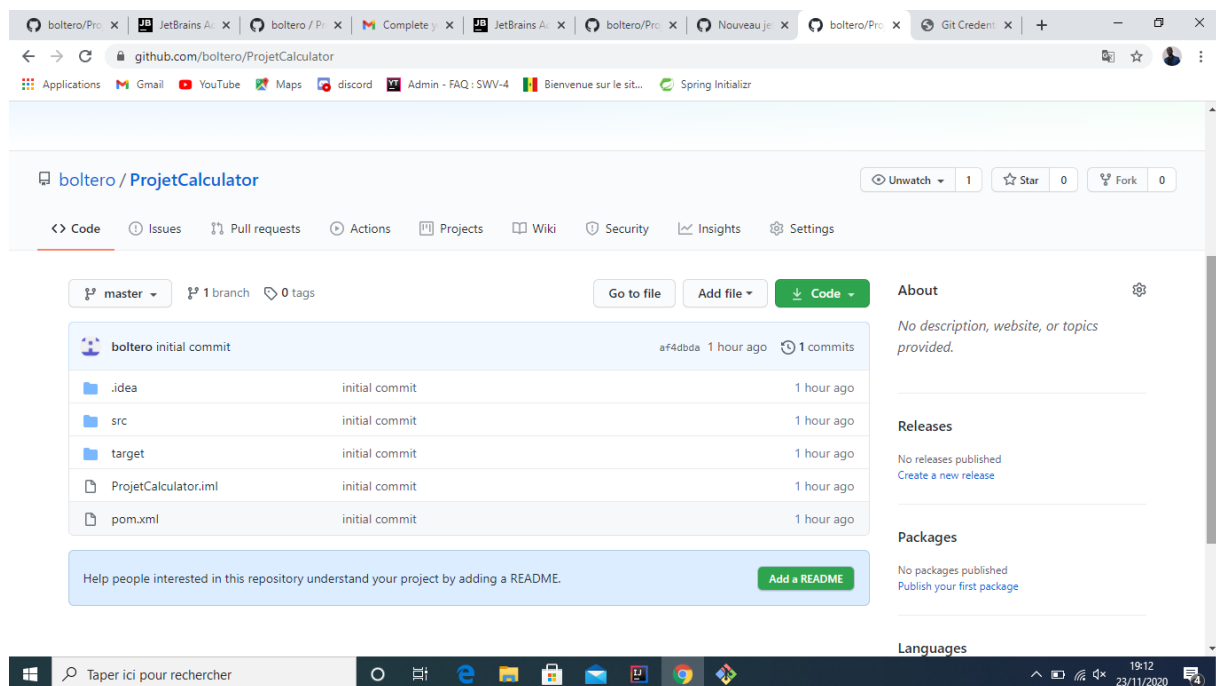
Il va permettre lorsqu'un bug survient après une pause sur le projet de pouvoir garder une trace de toutes les modifications faites.

GitHub facilite la programmation collaborative en mettant une interface Web à disposition du référentiel de code de Git, ainsi que des outils d'administration favorisant la collaboration.

GitHub permet aux développeurs de modifier, d'adapter et d'améliorer le logiciel gratuitement à partir de référentiels publics, mais facture les interventions pour les référentiels privés, selon trois formules de paiement.

Chaque référentiel public ou privé contient tous les fichiers d'un projet, ainsi que l'historique des révisions de chaque fichier. Plusieurs collaborateurs sont susceptibles de travailler dans les référentiels, qu'ils soient publics ou privés.

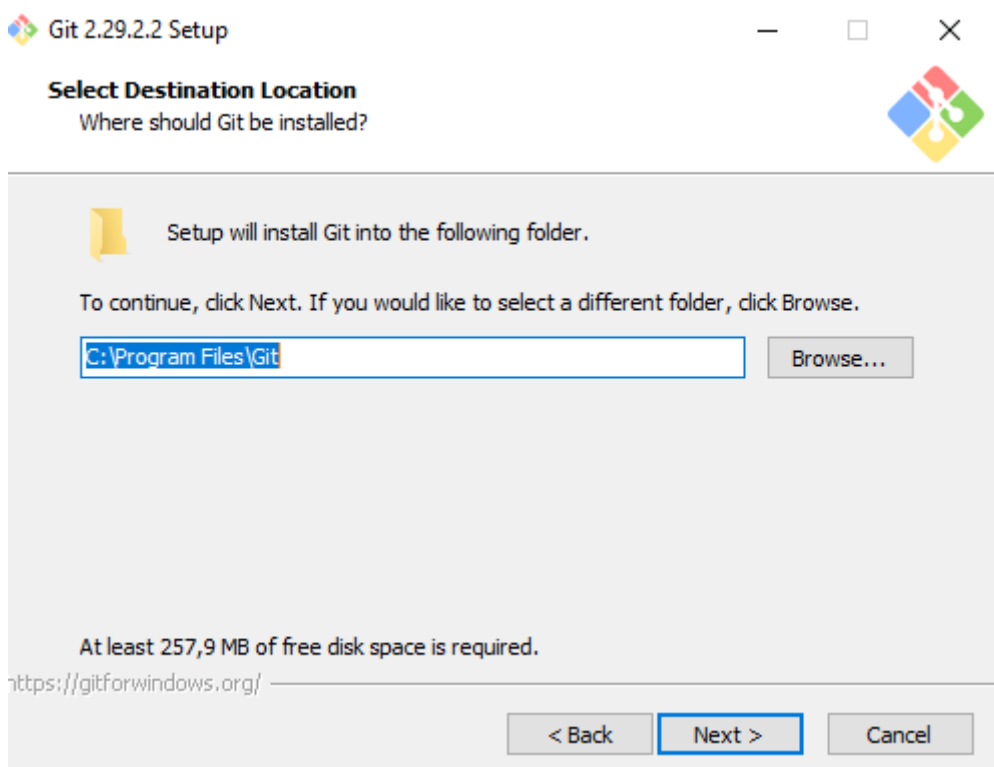
**De ce fait après la connexion on a eu à créer un nouveau repository et effectuer un push de notre projet**



## ➤ GIT

Git est un logiciel de gestion de versions décentralisé et libre, créé par Linus Torvalds, créateur du noyau Linux. Il dispose d'une grande communauté très active ce qui facilite la collaboration et les échanges.

## Installation et configuration sous Windows



**Select Components**

Which components should be installed?



Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- ☐ Additional icons
  - ☐ On the Desktop
- ☒ Windows Explorer integration
  - ☒ Git Bash Here
  - ☒ Git GUI Here
- ☒ Git LFS (Large File Support)
- ☒ Associate .git\* configuration files with the default text editor
- ☒ Associate .sh files to be run with Bash
- ☐ Use a TrueType font in all console windows
- ☐ Check daily for Git for Windows updates

Current selection requires at least 257,9 MB of disk space.

<https://gitforwindows.org/>

&lt; Back

Next &gt;

Cancel

**Select Start Menu Folder**

Where should Setup place the program's shortcuts?



Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

Git

Browse...

☐ Don't create a Start Menu folder

<https://gitforwindows.org/>

&lt; Back

Next &gt;

Cancel

**Choosing the default editor used by Git**

Which editor would you like Git to use?



Use Vim (the ubiquitous text editor) as Git's default editor

The [Vim editor](#), while powerful, [can be hard to use](#). Its user interface is unintuitive and its key bindings are awkward.

**Note:** Vim is the default editor of Git for Windows only for historical reasons, and it is highly recommended to switch to a modern GUI editor instead.

**Note:** This will leave the 'core.editor' option unset, which will make Git fall back to the 'EDITOR' environment variable. The default editor is Vim - but you may set it to some other editor of your choice.

<https://gitforwindows.org/>

&lt; Back

Next &gt;

Cancel

**Adjusting the name of the initial branch in new repositories**

What would you like Git to name the initial branch after "git init"?

☒ **Let Git decide**

Let Git use its default branch name (currently: "master") for the initial branch in newly created repositories. The Git project [intends](#) to change this default to a more inclusive name in the near future.

☐ **Override the default branch name for new repositories**

**NEW!** Many teams already renamed their default branches; common choices are "main", "trunk" and "development". Specify the name "git init" should use for the initial branch:

This setting does not affect existing repositories.

<https://gitforwindows.org/>

&lt; Back

Next &gt;

Cancel

**Adjusting your PATH environment**

How would you like to use Git from the command line?

☐ **Use Git from Git Bash only**

This is the most cautious choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.

☒ **Git from the command line and also from 3rd-party software**

**(Recommended)** This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from Git Bash, the Command Prompt and the Windows PowerShell as well as any third-party software looking for Git in PATH.

☐ **Use Git and optional Unix tools from the Command Prompt**

Both Git and the optional Unix tools will be added to your PATH.  
**Warning:** This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.

<https://gitforwindows.org/>

&lt; Back

Next &gt;

Cancel

**Choosing HTTPS transport backend**

Which SSL/TLS library would you like Git to use for HTTPS connections?

☒ **Use the OpenSSL library**

Server certificates will be validated using the ca-bundle.crt file.

☐ **Use the native Windows Secure Channel library**

Server certificates will be validated using Windows Certificate Stores. This option also allows you to use your company's internal Root CA certificates distributed e.g. via Active Directory Domain Services.

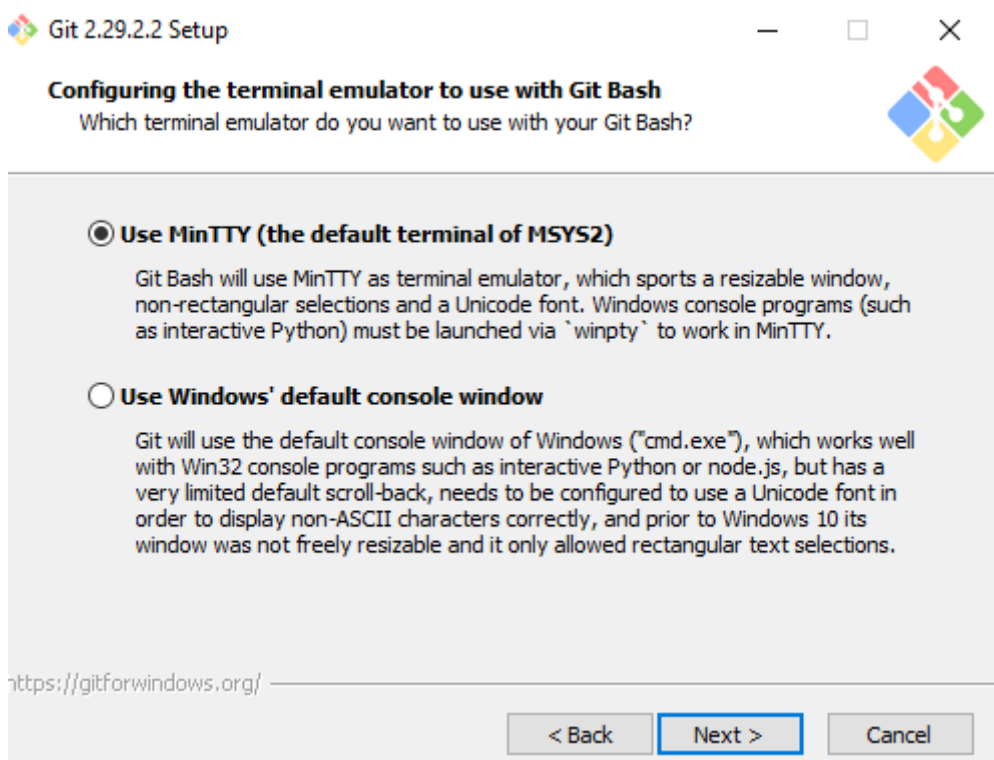
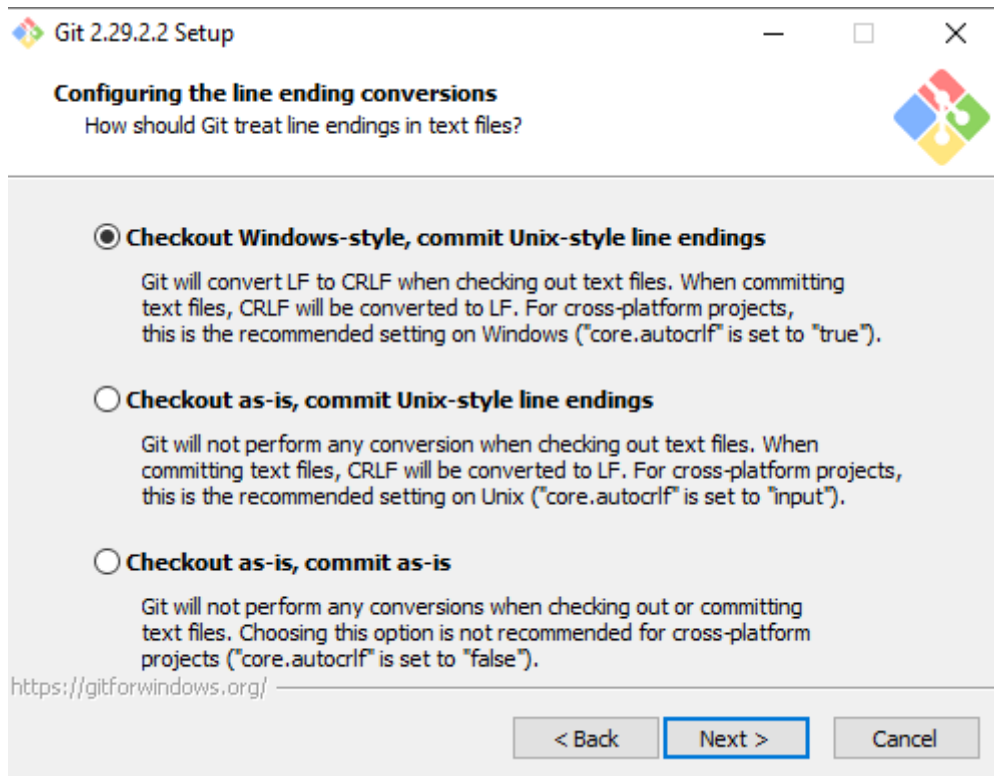
<https://gitforwindows.org/>

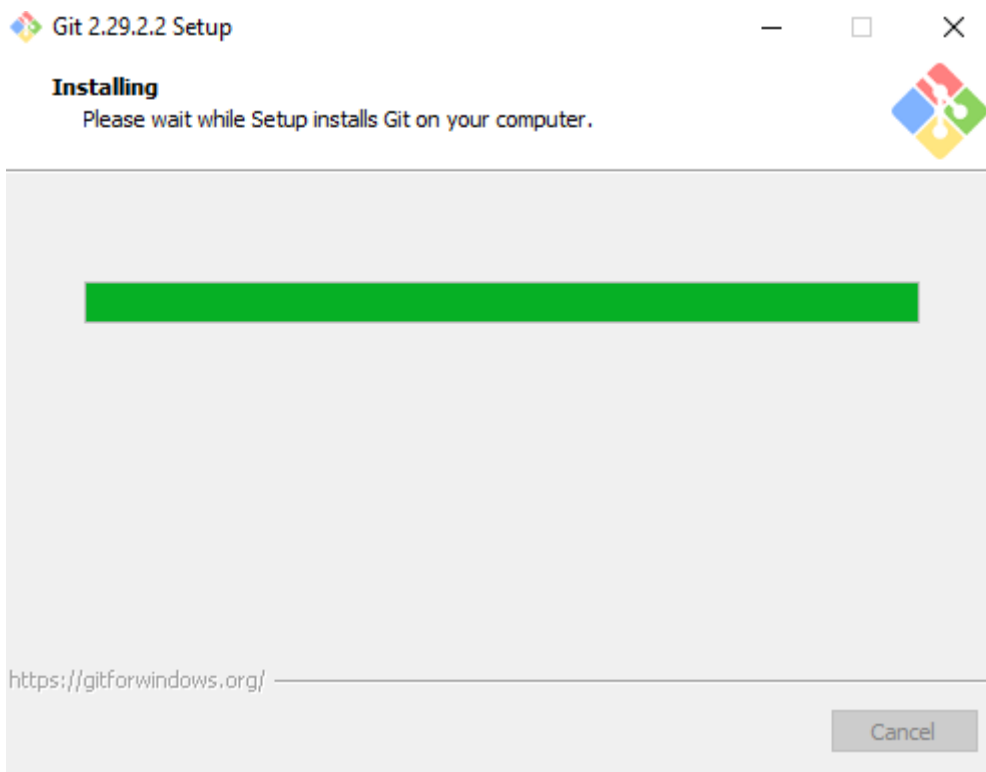
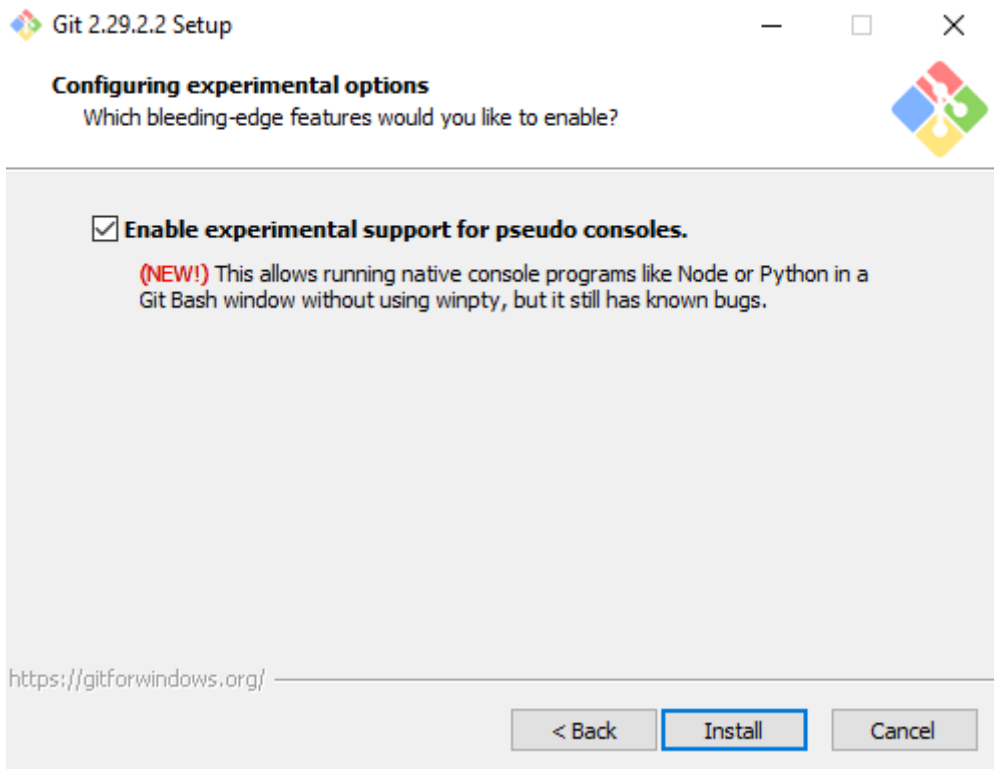
&lt; Back

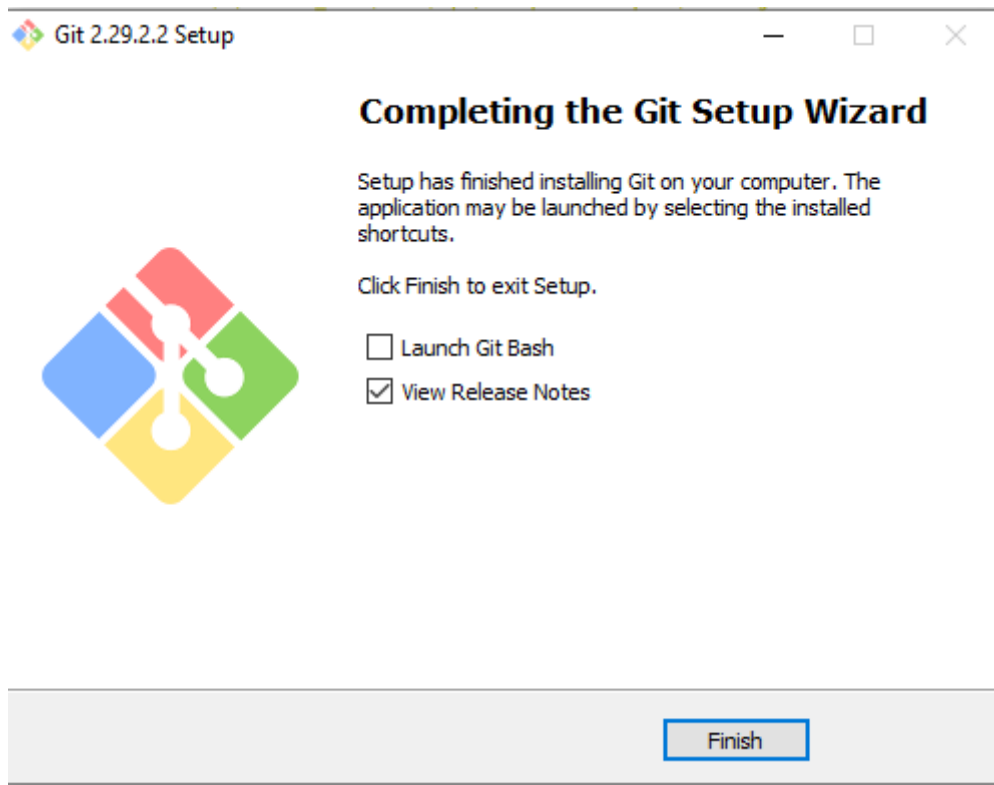
Next &gt;

Cancel









## Ouverture du terminal de git

Aller sur le projet, clic droit puis sur **git bash here**

Après authentification on lance les commandes git pour faire des commits, ajouter et pusher le projet sur git. On obtient ainsi :

```
MINGW64:/c/ProjetCalculator
Boltero@DESKTOP-ET2UJ07 MINGW64 ~
$ cd C:\ProjetCalculator

Boltero@DESKTOP-ET2UJ07 MINGW64 /c/ProjetCalculator (master)
$ git init
Reinitialized existing Git repository in C:/ProjetCalculator/.git/

Boltero@DESKTOP-ET2UJ07 MINGW64 /c/ProjetCalculator (master)
$ git status
On branch master
nothing to commit, working tree clean

Boltero@DESKTOP-ET2UJ07 MINGW64 /c/ProjetCalculator (master)
$ git add .

Boltero@DESKTOP-ET2UJ07 MINGW64 /c/ProjetCalculator (master)
$ git commit -m "premiere commit"
On branch master
nothing to commit, working tree clean

Boltero@DESKTOP-ET2UJ07 MINGW64 /c/ProjetCalculator (master)
$ AC

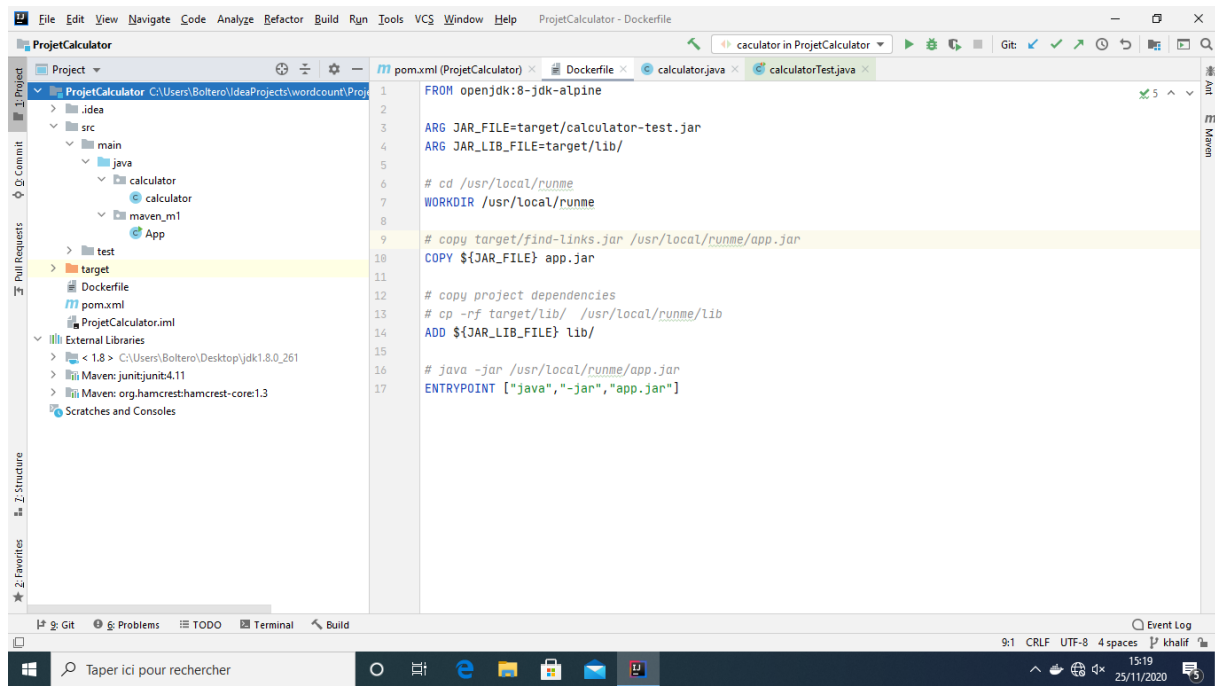
Boltero@DESKTOP-ET2UJ07 MINGW64 /c/ProjetCalculator (master)
$ git remote add origin https://github.com/boltero/ProjetCalculator.git
fatal: remote origin already exists.

Boltero@DESKTOP-ET2UJ07 MINGW64 /c/ProjetCalculator (master)
$ git push origin master
Enumerating objects: 35, done.
Counting objects: 100% (35/35), done.
Delta compression using up to 8 threads
Compressing objects: 100% (24/24), done.
Writing objects: 100% (35/35), 6.56 KiB | 231.00 KiB/s, done.
Total 35 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/boltero/ProjetCalculator.git
 * [new branch]      master -> master

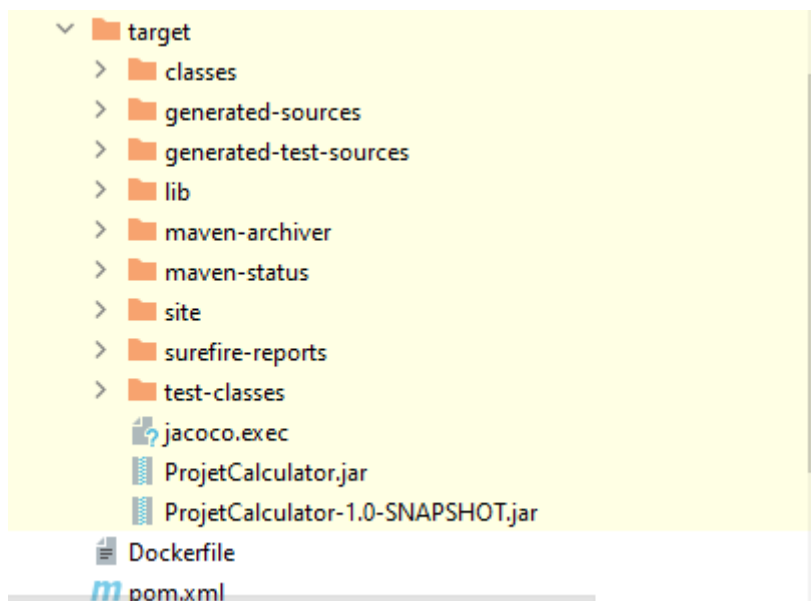
Boltero@DESKTOP-ET2UJ07 MINGW64 /c/ProjetCalculator (master)
```

On a ensuite créé une branche avec la commande **git checkout –b khalif**, puis y faire des modifications et faire les merges sur la branche master.

## IV. CREATION ET BUILD DU FICHIER DOCKERFILE



- Génération des fichiers jar et du répertoire lib sous le Target après compilation



- Lancement du build avec création de l'image : **docker build -t projetimage :1.0** sur la branche **khalif**.

```
MINGW64/C:/Users/Boltero/IdeaProjects/wordcount/ProjetCalculator/ProjetCalculator
$ docker build -t projetimage:1.0 .
#1 [internal] load .dockerignore
#1 transferring context: 28.0s done
#1 DONE 1.0s

#2 [internal] load build definition from Dockerfile
#2 transferring dockerfile: 451B 0.2s done
#2 DONE 1.3s

#3 [internal] load metadata for docker.io/library/openjdk:8-jdk-alpine
#3 DONE 4.8s

#4 [1/4] FROM docker.io/library/openjdk:8-jdk-alpine@sha256:94792824df2df33...
#4 resolve docker.io/library/openjdk:8-jdk-alpine@sha256:94792824df2df33402f201713f932b58cb9de94a0cd524164a0f2283343547b3 done
#4 sha256:c2274a1a0e2786ee9101b08f76111f9ab8019e368dce1e325d3c284a0ca33397 0B / 70.73MB 0.5s
#4 sha256:94792824df2df33402f201713f932b58cb9de94a0cd524164a0f2283343547b3 1.64kB / 1.64kB done
#4 sha256:44b3ca369e9492527a266273ce58c71a81f4f4c5076f6ebdb1a3f39015dce71 947B / 947B done
#4 sha256:c332a0991a80cf83172847c8beed6f6e4540b793c2b787f8b8be4534217 3.40kB / 3.40kB done
#4 sha256:e7c96db7181be991f19a9fb6975cddb73c65f4a2681348e63a141a2192a5f10 916.46kB / 2.76MB 2.4s
#4 sha256:f910a506b6cb1dbec766725d7035f6f95ae2bf2bea6224dbe8c7c6ad4f3664a2 0B / 238B 0.5s
#4 ...

#6 [internal] load build context
#6 transferring context: 294.18kB 1.0s done
#6 DONE 1.6s

#4 [1/4] FROM docker.io/library/openjdk:8-jdk-alpine@sha256:94792824df2df33...
#4 sha256:f910a506b6cb1dbec766725d7035f6f95ae2bf2bea6224dbe8c7c6ad4f3664a2 238B / 238B 1.7s
#4 sha256:f910a506b6cb1dbec766725d7035f6f95ae2bf2bea6224dbe8c7c6ad4f3664a2 238B / 238B 1.8s done
#4 sha256:e7c96db7181be991f19a9fb6975cddb73c65f4a2681348e63a141a2192a5f10 269.30kB / 2.76MB 2.1s
#4 sha256:e7c96db7181be991f19a9fb6975cddb73c65f4a2681348e63a141a2192a5f10 728.05kB / 2.76MB 2.4s
#4 sha256:e7c96db7181be991f19a9fb6975cddb73c65f4a2681348e63a141a2192a5f10 916.46kB / 2.76MB 2.5s
#4 sha256:e7c96db7181be991f19a9fb6975cddb73c65f4a2681348e63a141a2192a5f10 1.31MB / 2.76MB 2.8s
#4 sha256:e7c96db7181be991f19a9fb6975cddb73c65f4a2681348e63a141a2192a5f10 1.49MB / 2.76MB 2.9s
#4 sha256:e7c96db7181be991f19a9fb6975cddb73c65f4a2681348e63a141a2192a5f10 1.77MB / 2.76MB 3.1s
#4 sha256:e7c96db7181be991f19a9fb6975cddb73c65f4a2681348e63a141a2192a5f10 2.10MB / 2.76MB 3.3s
#4 sha256:e7c96db7181be991f19a9fb6975cddb73c65f4a2681348e63a141a2192a5f10 2.39MB / 2.76MB 3.5s
#4 sha256:e7c96db7181be991f19a9fb6975cddb73c65f4a2681348e63a141a2192a5f10 2.55MB / 2.76MB 3.6s
#4 sha256:e7c96db7181be991f19a9fb6975cddb73c65f4a2681348e63a141a2192a5f10 2.76MB / 2.76MB 3.8s
#4 sha256:c2274a1a0e2786ee9101b08f76111f9ab8019e368dce1e325d3c284a0ca33397 3.72MB / 70.73MB 4.0s
#4 sha256:e7c96db7181be991f19a9fb6975cddb73c65f4a2681348e63a141a2192a5f10 2.76MB / 2.76MB 4.0s done
#4 extracting sha256:e7c96db7181be991f19a9fb6975cddb73c65f4a2681348e63a141a2192a5f10
#4 extracting sha256:e7c96db7181be991f19a9fb6975cddb73c65f4a2681348e63a141a2192a5f10 0.2s done
#4 extracting sha256:f910a506b6cb1dbec766725d7035f6f95ae2bf2bea6224dbe8c7c6ad4f3664a2 done
#4 sha256:c2274a1a0e2786ee9101b08f76111f9ab8019e368dce1e325d3c284a0ca33397 7.36MB / 70.73MB 5.6s
#4 sha256:c2274a1a0e2786ee9101b08f76111f9ab8019e368dce1e325d3c284a0ca33397 11.17MB / 70.73MB 6.9s
#4 sha256:c2274a1a0e2786ee9101b08f76111f9ab8019e368dce1e325d3c284a0ca33397 14.97MB / 70.73MB 7.9s
#4 sha256:c2274a1a0e2786ee9101b08f76111f9ab8019e368dce1e325d3c284a0ca33397 18.90MB / 70.73MB 8.9s
#4 sha256:c2274a1a0e2786ee9101b08f76111f9ab8019e368dce1e325d3c284a0ca33397 22.69MB / 70.73MB 10.0s
#4 sha256:c2274a1a0e2786ee9101b08f76111f9ab8019e368dce1e325d3c284a0ca33397 26.45MB / 70.73MB 11.0s
#4 sha256:c2274a1a0e2786ee9101b08f76111f9ab8019e368dce1e325d3c284a0ca33397 30.06MB / 70.73MB 11.9s
#4 sha256:c2274a1a0e2786ee9101b08f76111f9ab8019e368dce1e325d3c284a0ca33397 33.91MB / 70.73MB 13.0s
#4 sha256:c2274a1a0e2786ee9101b08f76111f9ab8019e368dce1e325d3c284a0ca33397 37.84MB / 70.73MB 14.2s
#4 sha256:c2274a1a0e2786ee9101b08f76111f9ab8019e368dce1e325d3c284a0ca33397 41.58MB / 70.73MB 15.1s
#4 sha256:c2274a1a0e2786ee9101b08f76111f9ab8019e368dce1e325d3c284a0ca33397 45.22MB / 70.73MB 16.4s
#4 sha256:c2274a1a0e2786ee9101b08f76111f9ab8019e368dce1e325d3c284a0ca33397 48.82MB / 70.73MB 17.4s
#4 sha256:c2274a1a0e2786ee9101b08f76111f9ab8019e368dce1e325d3c284a0ca33397 52.47MB / 70.73MB 18.5s
#4 sha256:c2274a1a0e2786ee9101b08f76111f9ab8019e368dce1e325d3c284a0ca33397 56.50MB / 70.73MB 19.7s
#4 sha256:c2274a1a0e2786ee9101b08f76111f9ab8019e368dce1e325d3c284a0ca33397 60.10MB / 70.73MB 21.3s
#4 sha256:c2274a1a0e2786ee9101b08f76111f9ab8019e368dce1e325d3c284a0ca33397 63.88MB / 70.73MB 22.6s
#4 sha256:c2274a1a0e2786ee9101b08f76111f9ab8019e368dce1e325d3c284a0ca33397 67.95MB / 70.73MB 24.1s
#4 sha256:c2274a1a0e2786ee9101b08f76111f9ab8019e368dce1e325d3c284a0ca33397 70.73MB / 70.73MB 28.2s done
#4 extracting sha256:c2274a1a0e2786ee9101b08f76111f9ab8019e368dce1e325d3c284a0ca33397
#4 extracting sha256:c2274a1a0e2786ee9101b08f76111f9ab8019e368dce1e325d3c284a0ca33397 0.8s done
#4 DONE 30.9s

#5 [2/4] WORKDIR /usr/local/runme
#5 DONE 6.4s

#7 [3/4] COPY target/ProjetCalculator.jar app.jar
#7 DONE 0.9s

#8 [4/4] ADD target/lib/ lib/
#8 DONE 0.9s

#9 exporting to image
#9 exporting layers
#9 exporting layers 1.7s done
#9 writing image sha256:59461e8a7f828f8dce6122f4dc331cbb4dd4f048f7f8de73df09644326f36f
#9 writing image sha256:59461e8a7f828f8dce6122f4dc331cbb4dd4f048f7f8de73df09644326f36f 0.1s done
#9 naming to docker.io/library/projetimage:1.0 0.1s done
#9 DONE 1.9s

Boltero@DESKTOP-ET2U307 MINGW64 ~/IdeaProjects/wordcount/ProjetCalculator/ProjetCalculator (khalif)
$
```

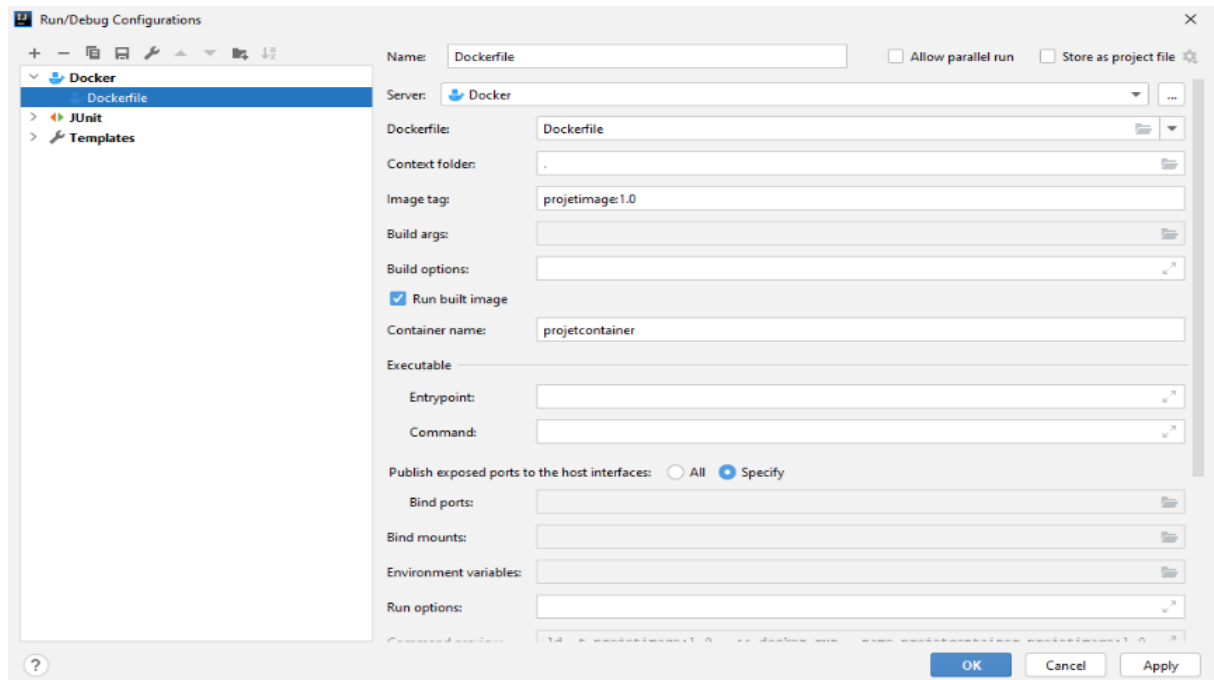
- Visualisation de l'image créer avec la commande : **docker images**

```

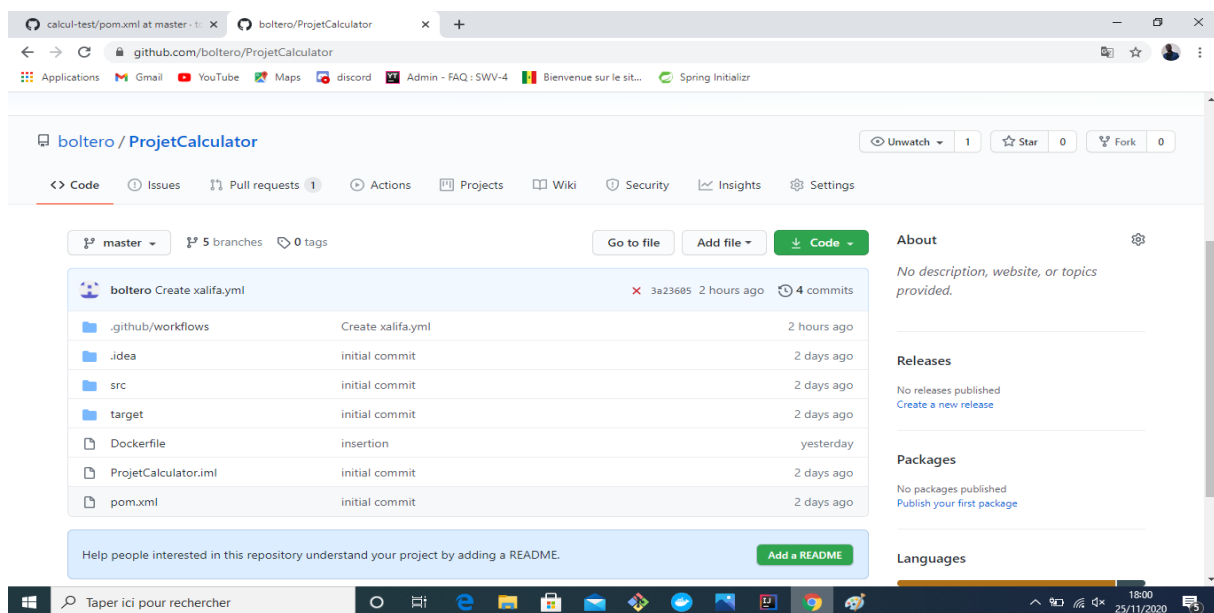
Boltero@DESKTOP-ET2UJ07 MINGW64 ~/IdeaProjects/wordcount/ProjetCalculator/ProjetCalculator (khalif)
$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
projetcalculator     1.0.0              59461e8a7f58       2 hours ago        105MB
projetimage          1.0                f48d9ca4cbf2       32 hours ago        99MB
docker/getting-started latest             67a3629d4d71       3 weeks ago        27.2MB
liliasfaxi/spark-hadoop hv-2.7.2           d64a47823a96       21 months ago      1.94GB

```

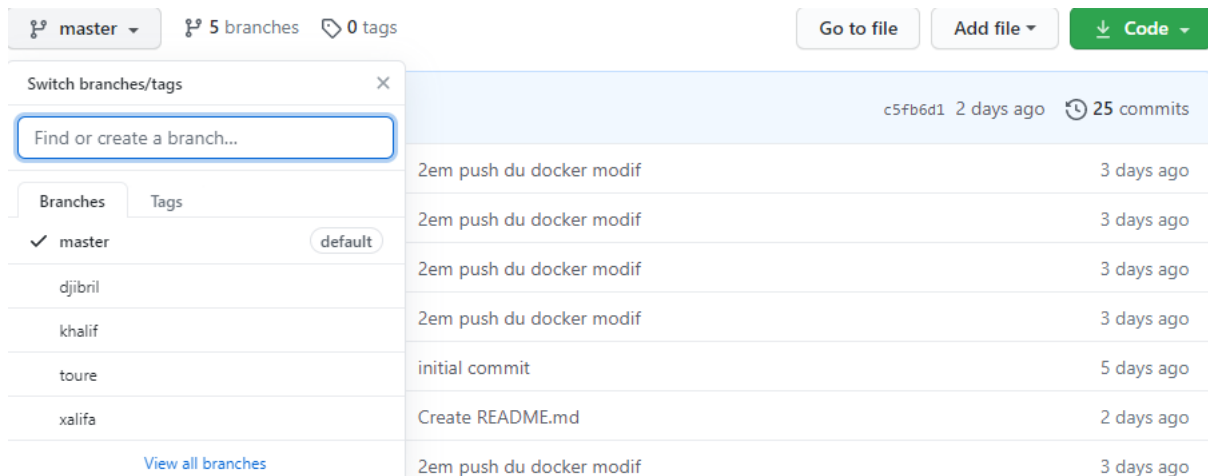
## ➤ Configuration du Dockerfile sur IntelliJ :



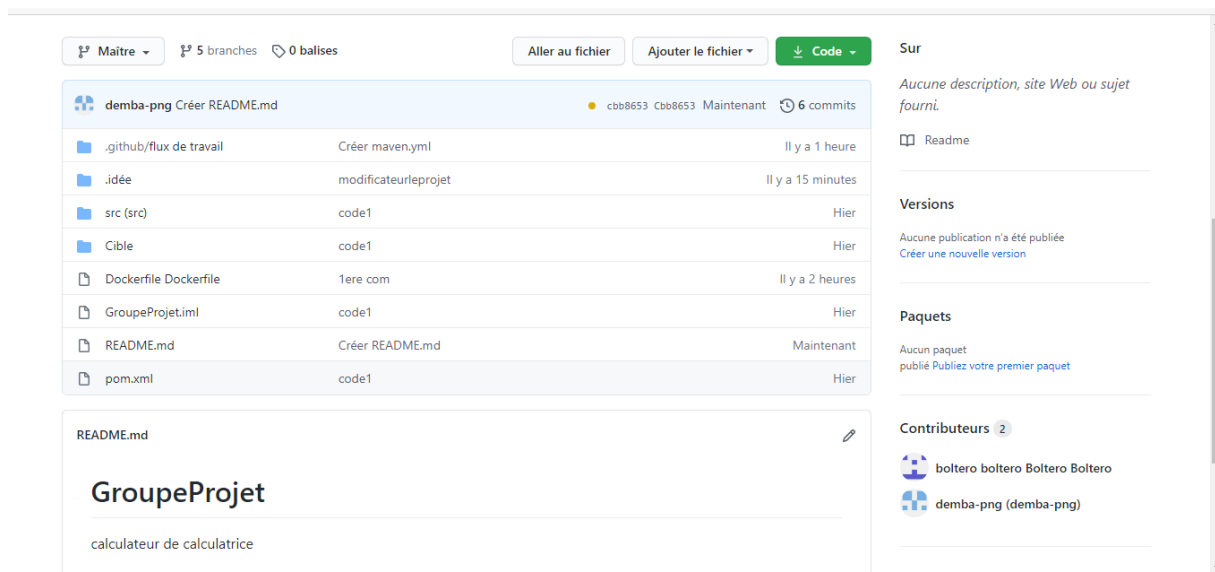
## ➤ Commit et push à nouveau du projet après build :



- Création d'autres branches avec la commande : **git checkout -b nombranche**




Commit sur la branche touré:




## V. GITHUB ACTION

Automatisez, personnalisez et exécutez vos flux de travail de développement logiciel directement dans votre référentiel avec les actions GitHub. Vous pouvez découvrir, créer et partager des actions pour effectuer n'importe quel travail de votre choix, y compris CI / CD, et combiner des actions dans un flux de travail entièrement personnalisé.




- Création et build du workflow

 boltero Create maven.yml ✓


Latest commit 0568f5b yesterday  History

 1 contributor


21 lines (15 sloc) | 399 Bytes

Raw Blame   

```
1 # This workflow will build a Java project with Maven
2 # For more information see: https://help.github.com/actions/language-and-framework-guides/building-and-testing-java-with-maven
3
4 name: CI
5
6 on:
7   push:
8     branches: [ master ]
9   pull_request:
10     branches: [ master ]
11
12 jobs:
13   build:
14
15     runs-on: ubuntu-latest
16
17     steps:
18     - uses: actions/checkout@v2
19
20     - name: Run test
21       run: mvn test
```

 Create maven.yml


master  0568f5b

 Re-run jobs ▾






▼ CI  1  
on: push

**build**

succeeded now in 17s

 Search logs

...

- >  Set up job 3s
- >  Run actions/checkout@v2 1s
- >  Run test 13s
- >  Post Run actions/checkout@v2 0s
- >  Complete job 0s



build

succeeded 3 minutes ago in 17s

Search logs

...

Run test13s

3319 downloaded from central: <https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-junit4/4.4.1/surefire-junit4-4.4.1.jar>  
(85 kB at 1.8 MB/s)

3320 [INFO]

3321 [INFO] -----

3322 [INFO] T E S T S

3323 [INFO] -----

3324 [INFO] Running org.example.CalculatorTest

3325 [INFO] Tests run: 8, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.066 s - in org.example.CalculatorTest

3326 [INFO]

3327 [INFO] Results:

3328 [INFO]

3329 [INFO] Tests run: 8, Failures: 0, Errors: 0, Skipped: 0

3330 [INFO]

3331 [INFO] -----

3332 [INFO] BUILD SUCCESS

3333 [INFO] -----

3334 [INFO] Total time: 11.213 s

3335 [INFO] Finished at: 2020-11-27T16:59:42Z

3336 [INFO] -----

Post Run actions/checkout@v20s

Complete job0s

## VI. DOCKER HUB AVEC GITHUB ACTIONS

```
1  name: Maven Docker CI/CD
2
3  on:
4    push:
5      branches: [ master ]
6    pull_request:
7      branches: [ master ]
8
9  jobs:
10    build_and_test:
11      runs-on: ubuntu-latest
12
13      services:
14        redis:
15          image: redis
16          options: >-
17            --health-cmd "redis-cli ping"
18            --health-interval 10s
19            --health-timeout 5s
20            --health-retries 5
21          ports:
22            - 6379:6379
23
24      steps:
25        - uses: actions/checkout@v2
26        - name: Set up JDK 14
27          uses: actions/setup-java@v1
28          with:
29            java-version: 14
```

---

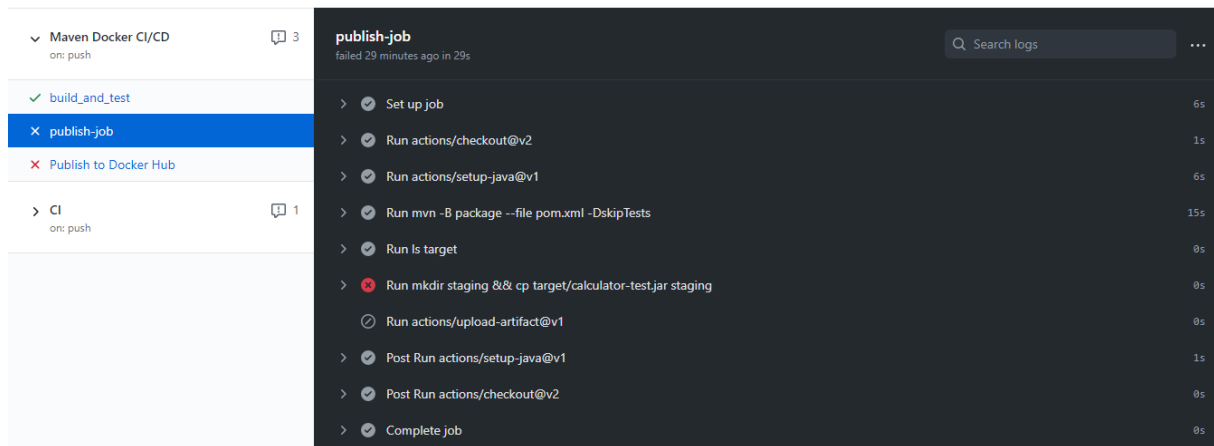
- name: Cache the Maven packages to speed up build  
uses: actions/cache@v1  
with:  
  path: ~/.m2  
  key: \${{ runner.os }}-m2-\${{ hashFiles('\*\*/pom.xml') }}  
  restore-keys: \${{ runner.os }}-m2
- name: Build and test project with Maven  
run: mvn -B package --file pom.xml

publish-job:

- runs-on: ubuntu-latest
- needs: [build\_and\_test]
- steps:
  - uses: actions/checkout@v2
  - uses: actions/setup-java@v1  
with:  
  java-version: 14
  - run: mvn -B package --file pom.xml -DskipTests
  - run: ls target
  - run: mkdir staging && cp target/calculator-test.jar staging
  - uses: actions/upload-artifact@v1  
with:  
  name: Package  
  path: staging

build-docker-image:

- name: Publish to Docker Hub
  - runs-on: ubuntu-latest
  - needs: [build\_and\_test]
  - steps:
    - uses: actions/checkout@v2
    - name: Login to Docker Hub  
run: docker login -u \${{ secrets.DOCKER\_USERNAME }} -p \${{ secrets.DOCKER\_PASSWORD }}
    - name: Build Container image  
run: docker build -t \${{ secrets.DOCKER\_REPO }}:latest .
    - name: Publish Docker image  
run: docker push \${{ secrets.DOCKER\_REPO }}
-



## CONCLUSION :

L'intégration continue repose souvent sur la mise en place d'une brique logicielle permettant l'automatisation de tâches : compilation, tests unitaires et fonctionnels, validation produit, tests de performances... À chaque changement du code, cette brique logicielle va exécuter un ensemble de tâches et produire un ensemble de résultats, que le développeur peut par la suite consulter. Cette intégration permet ainsi de ne pas oublier d'éléments lors de la mise en production et donc ainsi améliorer la qualité du produit.

Ainsi dans ce projet on a parcouru les principes de base relatifs à l'intégration continue et nous avons essayé de montrer son importance.

Ce projet nous permis de découvrir et maîtriser les outils Git, Github et docker, d'approfondir nos connaissances sur le plan de communications, de sauvegarde et de teste.