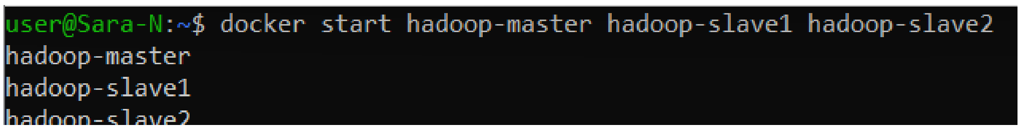
Résumé du TP2 - Programmation avec l'API HDFS

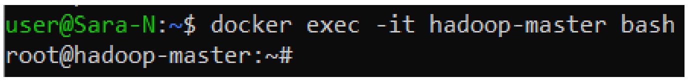
I. Démarrer le Cluster Hadoop

1. Démarrage des containers



```
user@Sara-N:~$ docker start hadoop-master hadoop-slave1 hadoop-slave2
hadoop-master
hadoop-slave1
hadoop-slave2
```

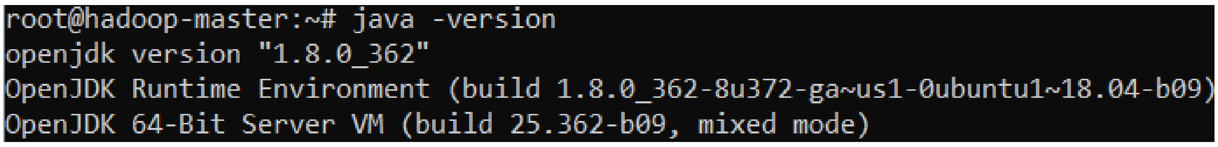
2. Accéder au master



```
user@Sara-N:~$ docker exec -it hadoop-master bash
root@hadoop-master:~#
```

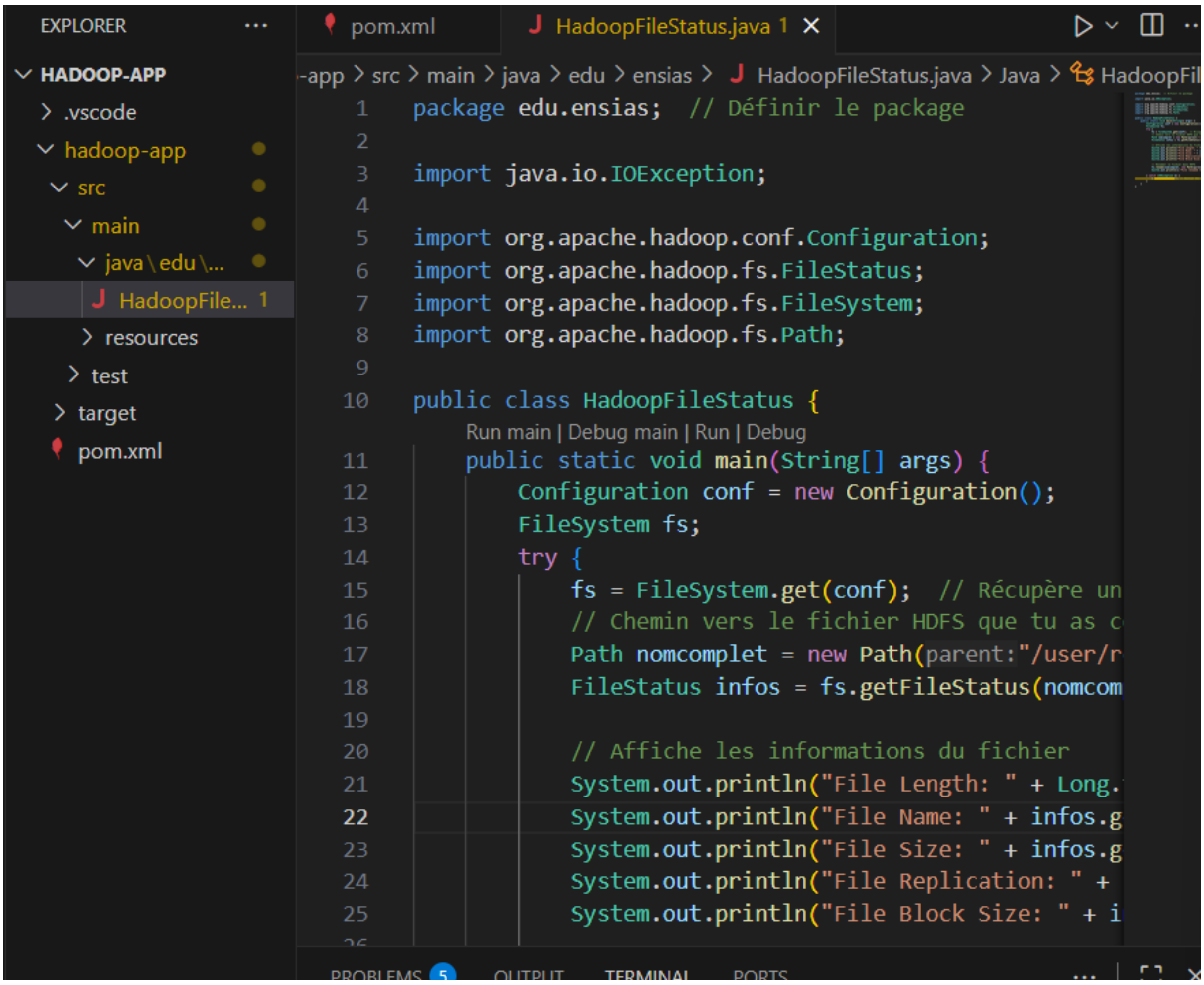
II.Programmation avec l'api HDFS

1. Installation de l'environnement de développement



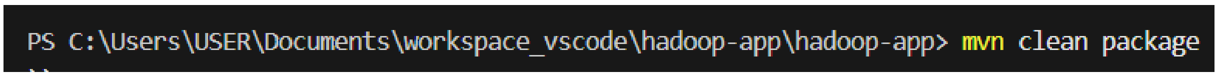
```
root@hadoop-master:~# java -version
openjdk version "1.8.0_362"
OpenJDK Runtime Environment (build 1.8.0_362-8u372-ga~us1-0ubuntu1~18.04-b09)
OpenJDK 64-Bit Server VM (build 25.362-b09, mixed mode)
```

2.Premier exemple



```java
package edu.ensias;  // Définir le package

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FileStatus;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;

public class HadoopFileStatus {
    public static void main(String[] args) {
        Configuration conf = new Configuration();
        FileSystem fs;
        try {
            fs = FileSystem.get(conf);  // Récupère un
            // Chemin vers le fichier HDFS que tu as c
            Path nomcomplet = new Path(parent:"/user/r
            FileStatus infos = fs.getFileStatus(nomcom

            // Affiche les informations du fichier
            System.out.println("File Length: " + Long.
            System.out.println("File Name: " + infos.g
            System.out.println("File Size: " + infos.g
            System.out.println("File Replication: " +
            System.out.println("File Block Size: " + i
```

- Créer un fichier jar qu’on va nommer HadoopFileStatus.jar

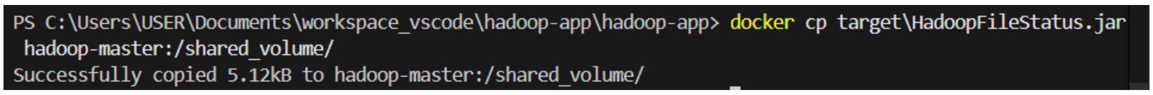


```
PS C:\Users\USER\Documents\workspace_vscode\hadoop-app\hadoop-app> mvn clean package
```

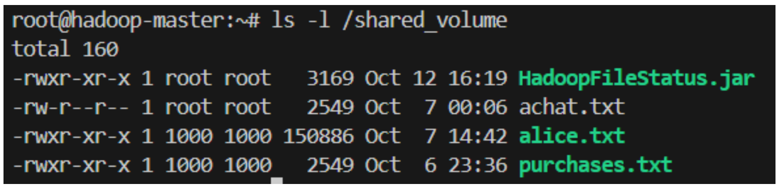
[INFO] Building jar: C:\Users\USER\Documents\workspace_vscode\hadoop-app\hadoop-app\target\hadoop-app-1.0-SNAPSHOT.jar

Puis j’ai renommé manuellement le jar ‘HadoopFileStatus.jar’ .

- Copier le jar créé vers le dossier de partage /hadoop_project

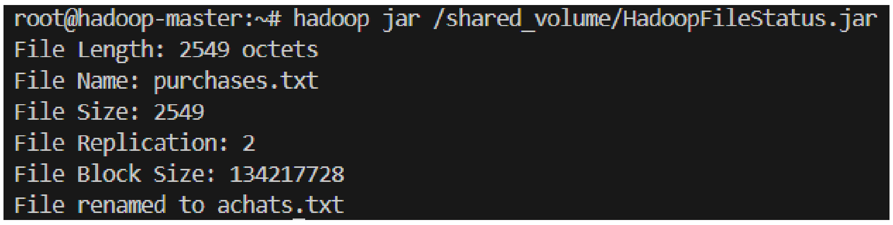


```
PS C:\Users\USER\Documents\workspace_vscode\hadoop-app\hadoop-app> docker cp target\HadoopFileStatus.jar
 hadoop-master:/shared_volume/
Successfully copied 5.12kB to hadoop-master:/shared_volume/
```



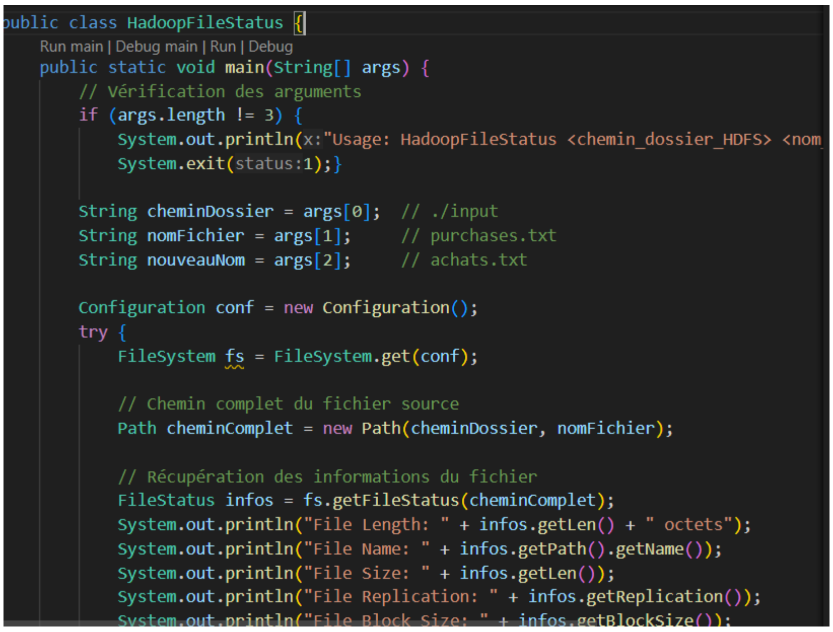
```
root@hadoop-master:~# ls -l /shared_volume
total 160
-rwxr-xr-x 1 root root   3169 Oct 12 16:19 HadoopFileStatus.jar
-rw-r--r-- 1 root root   2549 Oct  7 00:06 achat.txt
-rwxr-xr-x 1 1000 1000 150886 Oct  7 14:42 alice.txt
-rwxr-xr-x 1 1000 1000   2549 Oct  6 23:36 purchases.txt
```

- Lancer la commande



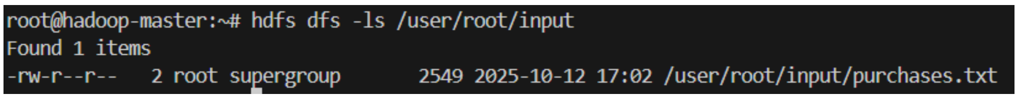
```
root@hadoop-master:~# hadoop jar /shared_volume/HadoopFileStatus.jar
File Length: 2549 octets
File Name: purchases.txt
File Size: 2549
File Replication: 2
File Block Size: 134217728
File renamed to achats.txt
```

- Modifier la classe HadoopFileStatus pour qu’elle puisse lire les paramètres chemin_fichier nom_fichier nouveau_nom_fichier lors de l’exécution



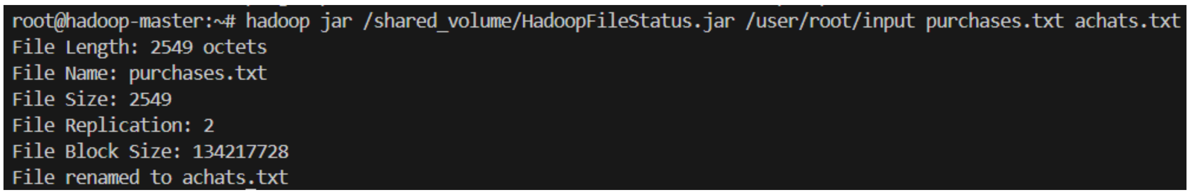
```
ublic class HadoopFileStatus {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        // Vérification des arguments
        if (args.length != 3) {
            System.out.println(x:"Usage: HadoopFileStatus <chemin_dossier_HDFS> <nom
            System.exit(status:1);}

        String cheminDossier = args[0];  // ./input
        String nomFichier = args[1];     // purchases.txt
        String nouveauNom = args[2];     // achats.txt

        Configuration conf = new Configuration();
        try {
            FileSystem fs = FileSystem.get(conf);

            // Chemin complet du fichier source
            Path cheminComplet = new Path(cheminDossier, nomFichier);

            // Récupération des informations du fichier
            FileStatus infos = fs.getFileStatus(cheminComplet);
            System.out.println("File Length: " + infos.getLen() + " octets");
            System.out.println("File Name: " + infos.getPath().getName());
            System.out.println("File Size: " + infos.getLen());
            System.out.println("File Replication: " + infos.getReplication());
            System.out.println("File Block Size: " + infos.getBlockSize());
```

Building jar: C:\Users\USER\Documents\workspace_vscode\hadoop-app\hadoop-app\target\HadoopFileStatus.jar



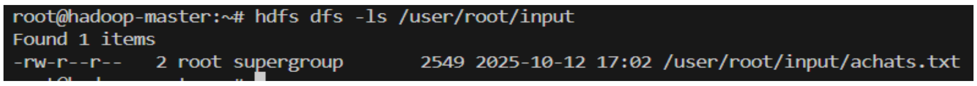
```
root@hadoop-master:~# hdfs dfs -ls /user/root/input
Found 1 items
-rw-r--r--   2 root supergroup       2549 2025-10-12 17:02 /user/root/input/purchases.txt
```

Exécuter le jar:



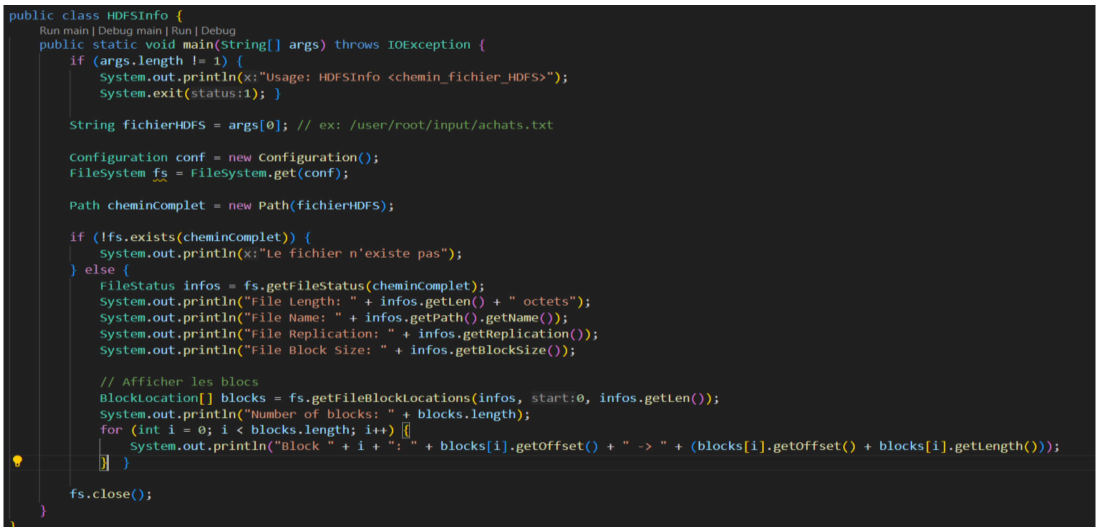
```
root@hadoop-master:~# hadoop jar /shared_volume/HadoopFileStatus.jar /user/root/input purchases.txt achats.txt
File Length: 2549 octets
File Name: purchases.txt
File Size: 2549
File Replication: 2
File Block Size: 134217728
File renamed to achats.txt
```

Input après:

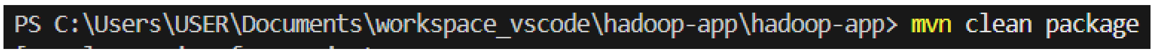


```
root@hadoop-master:~# hdfs dfs -ls /user/root/input
Found 1 items
-rw-r--r--   2 root supergroup       2549 2025-10-12 17:02 /user/root/input/achats.txt
```

3. Info d'un fichier sur HDFS

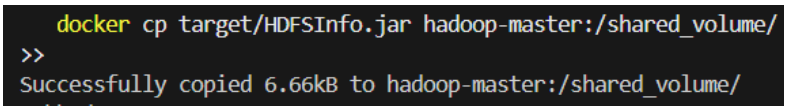


```java
public class HDFSInfo {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) throws IOException {
        if (args.length != 1) {
            System.out.println(x:"Usage: HDFSInfo <chemin_fichier_HDFS>");
            System.exit(status:1); }

        String fichierHDFS = args[0]; // ex: /user/root/input/achats.txt

        Configuration conf = new Configuration();
        FileSystem fs = FileSystem.get(conf);

        Path cheminComplet = new Path(fichierHDFS);

        if (!fs.exists(cheminComplet)) {
            System.out.println(x:"Le fichier n'existe pas");
        } else {
            FileStatus infos = fs.getFileStatus(cheminComplet);
            System.out.println("File Length: " + infos.getLen() + " octets");
            System.out.println("File Name: " + infos.getPath().getName());
            System.out.println("File Replication: " + infos.getReplication());
            System.out.println("File Block Size: " + infos.getBlockSize());

            // Afficher les blocs
            BlockLocation[] blocks = fs.getFileBlockLocations(infos, start:0, infos.getLen());
            System.out.println("Number of blocks: " + blocks.length);
            for (int i = 0; i < blocks.length; i++) {
                System.out.println("Block " + i + ": " + blocks[i].getOffset() + " -> " + (blocks[i].getOffset() + blocks[i].getLength()));
            } }

        fs.close();
    }
```

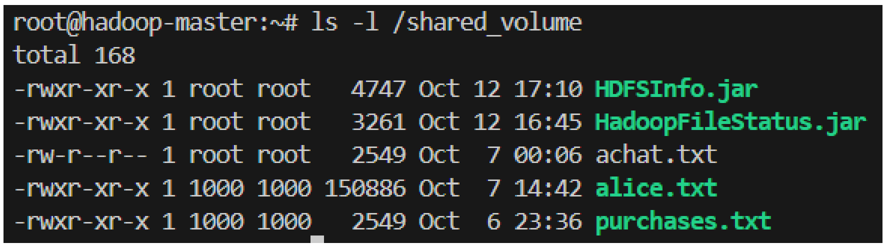


```
PS C:\Users\USER\Documents\workspace_vscode\hadoop-app\hadoop-app> mvn clean package
```

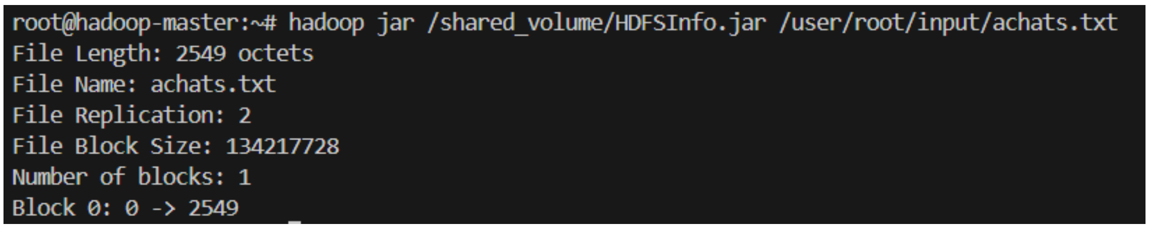
Building jar: C:\Users\USER\Documents\workspace_vscode\hadoop-app\hadoop-app\target\HDFSInfo.jar



```
   docker cp target/HDFSInfo.jar hadoop-master:/shared_volume/
>>
Successfully copied 6.66kB to hadoop-master:/shared_volume/
```

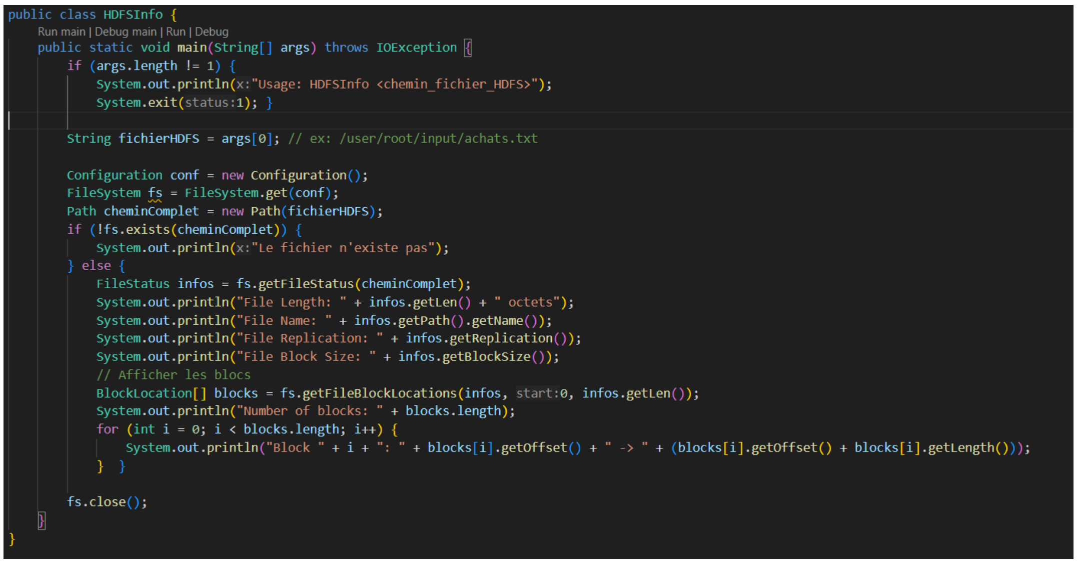


```
root@hadoop-master:~# ls -l /shared_volume
total 168
-rwxr-xr-x 1 root root   4747 Oct 12 17:10 HDFSInfo.jar
-rwxr-xr-x 1 root root   3261 Oct 12 16:45 HadoopFileStatus.jar
-rw-r--r-- 1 root root   2549 Oct  7 00:06 achat.txt
-rwxr-xr-x 1 1000 1000 150886 Oct  7 14:42 alice.txt
-rwxr-xr-x 1 1000 1000   2549 Oct  6 23:36 purchases.txt
```



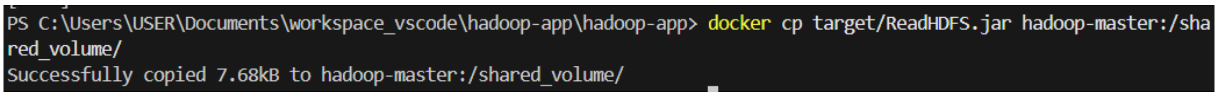
```
root@hadoop-master:~# hadoop jar /shared_volume/HDFSInfo.jar /user/root/input/achats.txt
File Length: 2549 octets
File Name: achats.txt
File Replication: 2
File Block Size: 134217728
Number of blocks: 1
Block 0: 0 -> 2549
```

4.Lire un fichier sur HDFS

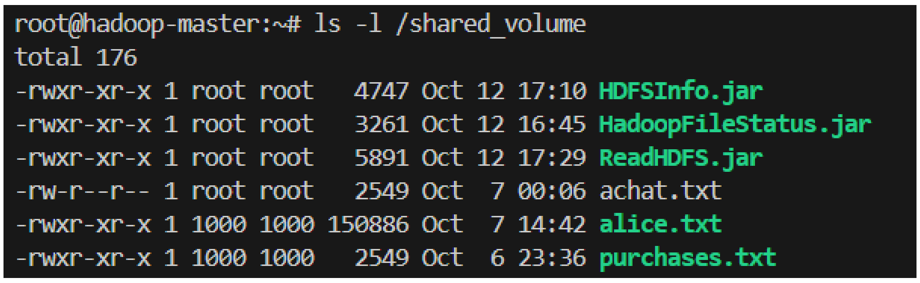


```java
public class HDFSInfo {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) throws IOException {
        if (args.length != 1) {
            System.out.println(x:"Usage: HDFSInfo <chemin_fichier_HDFS>");
            System.exit(status:1); }

        String fichierHDFS = args[0]; // ex: /user/root/input/achats.txt

        Configuration conf = new Configuration();
        FileSystem fs = FileSystem.get(conf);
        Path cheminComplet = new Path(fichierHDFS);
        if (!fs.exists(cheminComplet)) {
            System.out.println(x:"Le fichier n'existe pas");
        } else {
            FileStatus infos = fs.getFileStatus(cheminComplet);
            System.out.println("File Length: " + infos.getLen() + " octets");
            System.out.println("File Name: " + infos.getPath().getName());
            System.out.println("File Replication: " + infos.getReplication());
            System.out.println("File Block Size: " + infos.getBlockSize());
            // Afficher les blocs
            BlockLocation[] blocks = fs.getFileBlockLocations(infos, start:0, infos.getLen());
            System.out.println("Number of blocks: " + blocks.length);
            for (int i = 0; i < blocks.length; i++) {
                System.out.println("Block " + i + ": " + blocks[i].getOffset() + " -> " + (blocks[i].getOffset() + blocks[i].getLength()));
            }  }

        fs.close();
    }
}
```

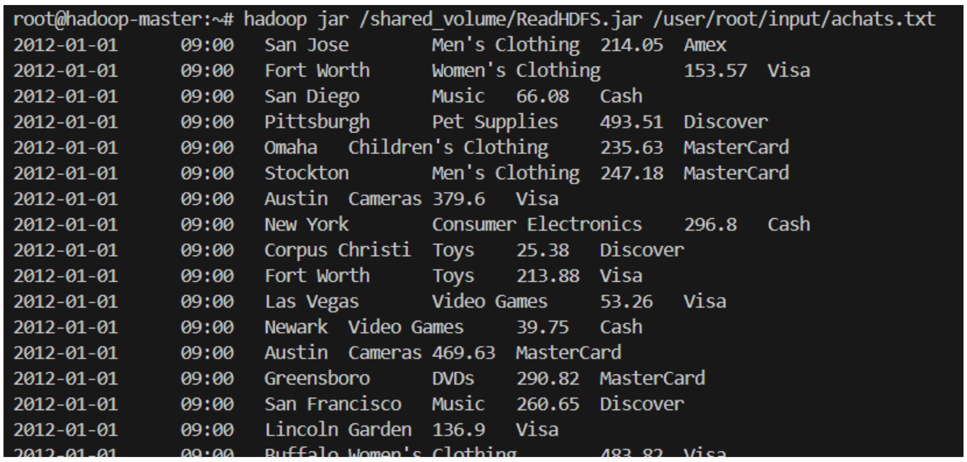
Building jar: C:\Users\USER\Documents\workspace_vscode\hadoop-app\hadoop-app\target\ReadHDFS.jar



```
PS C:\Users\USER\Documents\workspace_vscode\hadoop-app\hadoop-app> docker cp target/ReadHDFS.jar hadoop-master:/sha
red_volume/
Successfully copied 7.68kB to hadoop-master:/shared_volume/
```

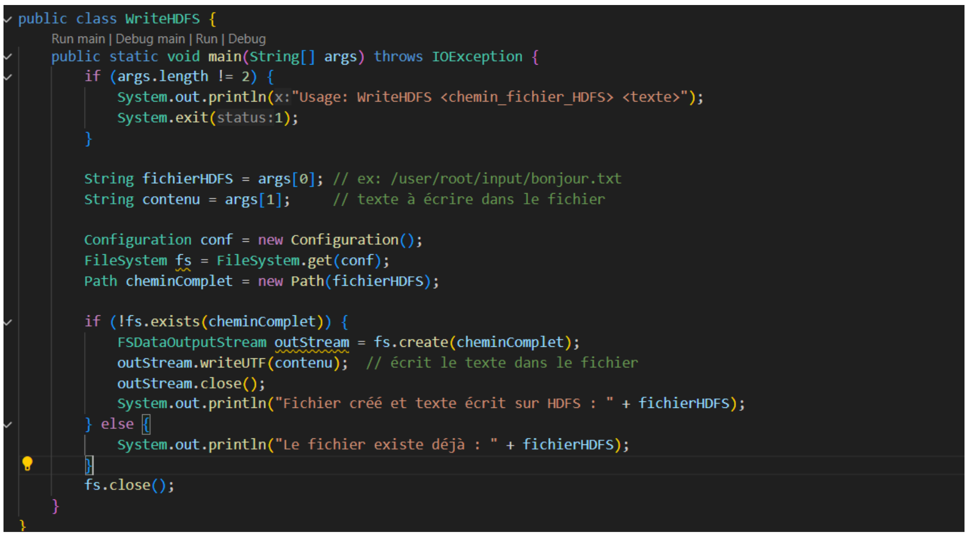


```
root@hadoop-master:~# ls -l /shared_volume
total 176
-rwxr-xr-x 1 root root   4747 Oct 12 17:10 HDFSInfo.jar
-rwxr-xr-x 1 root root   3261 Oct 12 16:45 HadoopFileStatus.jar
-rwxr-xr-x 1 root root   5891 Oct 12 17:29 ReadHDFS.jar
-rw-r--r-- 1 root root   2549 Oct  7 00:06 achat.txt
-rwxr-xr-x 1 1000 1000 150886 Oct  7 14:42 alice.txt
-rwxr-xr-x 1 1000 1000   2549 Oct  6 23:36 purchases.txt
```



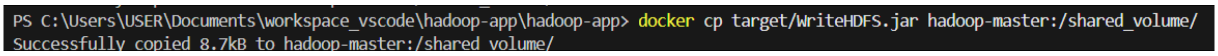
```
root@hadoop-master:~# hadoop jar /shared_volume/ReadHDFS.jar /user/root/input/achats.txt
2012-01-01      09:00   San Jose        Men's Clothing  214.05  Amex
2012-01-01      09:00   Fort Worth      Women's Clothing        153.57  Visa
2012-01-01      09:00   San Diego       Music   66.08   Cash
2012-01-01      09:00   Pittsburgh      Pet Supplies    493.51  Discover
2012-01-01      09:00   Omaha   Children's Clothing     235.63  MasterCard
2012-01-01      09:00   Stockton        Men's Clothing  247.18  MasterCard
2012-01-01      09:00   Austin  Cameras 379.6   Visa
2012-01-01      09:00   New York        Consumer Electronics    296.8   Cash
2012-01-01      09:00   Corpus Christi  Toys    25.38   Discover
2012-01-01      09:00   Fort Worth      Toys    213.88  Visa
2012-01-01      09:00   Las Vegas       Video Games     53.26   Visa
2012-01-01      09:00   Newark  Video Games     39.75   Cash
2012-01-01      09:00   Austin  Cameras 469.63  MasterCard
2012-01-01      09:00   Greensboro      DVDs    290.82  MasterCard
2012-01-01      09:00   San Francisco   Music   260.65  Discover
2012-01-01      09:00   Lincoln Garden  136.9   Visa
2012-01-01      09:00   Buffalo Women's Clothing        483.82  Visa
```

5.Ecrire un fichier sur HDFS

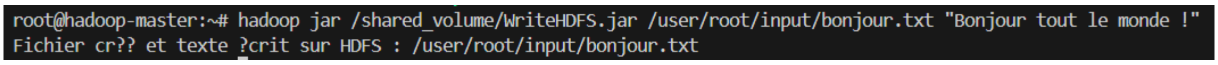


```java
public class WriteHDFS {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) throws IOException {
        if (args.length != 2) {
            System.out.println(x:"Usage: WriteHDFS <chemin_fichier_HDFS> <texte>");
            System.exit(status:1);
        }

        String fichierHDFS = args[0]; // ex: /user/root/input/bonjour.txt
        String contenu = args[1];     // texte à écrire dans le fichier

        Configuration conf = new Configuration();
        FileSystem fs = FileSystem.get(conf);
        Path cheminComplet = new Path(fichierHDFS);

        if (!fs.exists(cheminComplet)) {
            FSDataOutputStream outStream = fs.create(cheminComplet);
            outStream.writeUTF(contenu);  // écrit le texte dans le fichier
            outStream.close();
            System.out.println("Fichier créé et texte écrit sur HDFS : " + fichierHDFS);
        } else {
            System.out.println("Le fichier existe déjà : " + fichierHDFS);
        }
        fs.close();
    }
}
```

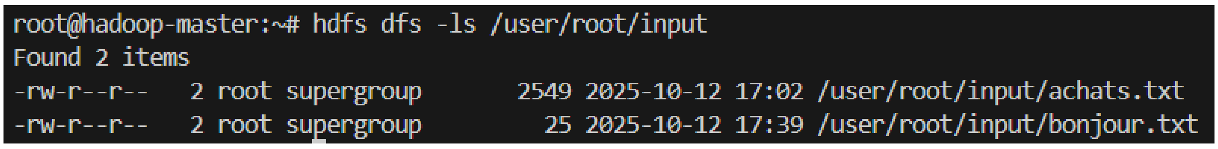
Building jar: C:\Users\USER\Documents\workspace_vscode\hadoop-app\hadoop-app\target\WriteHDFS.jar



```
PS C:\Users\USER\Documents\workspace_vscode\hadoop-app\hadoop-app> docker cp target/WriteHDFS.jar hadoop-master:/shared_volume/
Successfully copied 8.7kB to hadoop-master:/shared_volume/
```



```
root@hadoop-master:~# hadoop jar /shared_volume/WriteHDFS.jar /user/root/input/bonjour.txt "Bonjour tout le monde !"
Fichier cr?? et texte ?crit sur HDFS : /user/root/input/bonjour.txt
```



```
root@hadoop-master:~# hdfs dfs -ls /user/root/input
Found 2 items
-rw-r--r--   2 root supergroup       2549 2025-10-12 17:02 /user/root/input/achats.txt
-rw-r--r--   2 root supergroup         25 2025-10-12 17:39 /user/root/input/bonjour.txt
```