

Comment créer un fichier et y écrire des données en Java

<https://www.delftstack.com/fr/howto/java/how-to-create-a-file-and-write-data-to-it-in-java/>

Table des matières

PrintWriter pour créer un fichier en Java.....	1
java.nio.file pour créer un fichier en Java.....	2
Écriture en tampon pour créer un fichier en Java.....	2
FileWriter pour créer un fichier en Java.....	3

Ce tutoriel traite des méthodes permettant de créer un fichier texte et d'y écrire des données en Java.

Java dispose de plusieurs méthodes pour créer un fichier. La seule chose à retenir est que le bloc `try-catch` est nécessaire pour presque toutes les méthodes de création de fichiers afin que toute exception d'E/S soit traitée avec grâce. N'oubliez pas de l'utiliser.

PrintWriter pour créer un fichier en Java

Le `PrintWriter` en Java vous permet de créer un nouveau fichier avec le codage spécifié. Voici un exemple.

```
import java.io.*;

public class Main {
    public static void main(String[] args) {
        String fileName = "my-file.txt";
        String encoding = "UTF-8";
        try{
            PrintWriter writer = new PrintWriter(fileName, encoding);
            writer.println("The first line");
            writer.println("The second line");
            writer.close();
        }
        catch (IOException e){
            System.out.println("An error occurred.");
            e.printStackTrace();
        }
    }
}
```

java.nio.file pour créer un fichier en Java

Nous pouvons également utiliser `java.nio.file` pour créer un fichier et y écrire des données. L'exemple ci-dessous illustre cette approche.

```
import java.io.*;
import java.util.*;
import java.nio.file.*;
import java.nio.charset.*;

public class Main {
    public static void main(String[] args) {
        String fileName = "my-file.txt";
        try{
            List<String> lines = Arrays.asList("The first line", "The second line");
            Path file = Paths.get(fileName);
            Files.write(file, lines, StandardCharsets.UTF_8);
        }
        catch (IOException e){
            System.out.println("An error occurred.");
            e.printStackTrace();
        }
    }
}
```

Dans cet exemple, nous écrivons les données d'une `ArrayList` dans le fichier de telle sorte que chaque élément de la `ArrayList` soit écrit sur une nouvelle ligne du fichier de sortie. Le fichier de sortie ressemble à ceci:

```
> The first line
> The second line
```

Écriture en tampon pour créer un fichier en Java

L'exemple ci-dessous illustre la création et l'écriture dans un fichier à l'aide de `BufferedWriter`.

```
import java.io.*;
import java.nio.charset.*;
import java.util.*;

public class Main {
    public static void main(String[] args) {
        Charset utf8 = StandardCharsets.UTF_8;
        List<String> list = Arrays.asList("This is the first line", "This is the second line");
        String content = "This is my content";

        try
        {
```

```

//File name with path
String fileName = "my-file.txt";
File myFile = new File(fileName);
FileOutputStream fos = new FileOutputStream(myFile);
OutputStreamWriter osw = new OutputStreamWriter(fos);
Writer writer = new BufferedWriter(osw);
//Write data using a String variable
writer.write(content + "\n");

//Write data from an ArrayList
for (String s : list) {
    writer.write(s + "\n");
}

}
catch (IOException e){
    System.out.println("An error occurred.");
    e.printStackTrace();
}
}
}

```

Le fichier de sortie ressemble à ceci:

```

> This is my content
> This is the first line
> This is the second line

```

FileWriter pour créer un fichier en Java

Une autre méthode courante pour créer un nouveau fichier et y écrire en Java est d'utiliser `FileWriter`.

```

import java.io.*;

public class Main {
    public static void main(String[] args) {
        try {
            FileWriter myWriter = new FileWriter("my-file.txt");
            myWriter.write("We can also create a file and write to it using
FileWriter");
            myWriter.close();
        }
        catch (IOException e) {
            System.out.println("An error occurred.");
            e.printStackTrace();
        }
    }
}

```

Le fichier de sortie ressemble à ceci:

```

> We can also create a file and write to it using FileWriter

```