Comment lire un gros fichier texte ligne par ligne en Java

Table des matières

https://www.delftstack.com/fr/howto/java/how-to-read-a-large-text-file-line-by-line-in-java/

Ce tutoriel traite des méthodes permettant de lire efficacement, ligne par ligne, un gros fichier texte en Java.

Il existe de nombreuses méthodes pour lire un fichier texte en Java. Cependant, ce tutoriel est spécifique à la lecture de gros fichiers texte, et nous allons discuter des trois méthodes les plus efficaces pour lire rapidement des gros fichiers texte.

L’outil `BufferedReader` pour lire un fichier ligne par ligne en Java

La classe `BufferedReader` en Java lit le texte à partir d’un flux d’entrée de caractères donné, en mettant en mémoire tampon les caractères pour permettre une lecture efficace des caractères, des tableaux et des lignes. Cette méthode permet une lecture efficace ligne par ligne pour les fichiers d’entrée dont la taille est considérablement importante.

L’exemple ci-dessous illustre l’utilisation de `BufferedReader` pour lire un fichier `txt` et sortir son contenu ligne par ligne.

```
import java.io.*;

public class Main {
    public static void main(String[] args) {
        String file = "my-file.txt";
        try(BufferedReader br = new BufferedReader(new FileReader(file)))
        {
            String line;
            while ((line = br.readLine()) != null) {
            System.out.println(line);
            }
        }
        catch (IOException e) {
            System.out.println("An error occurred.");
            e.printStackTrace();
        }
```

```
    }
}
```

Stream pour lire un fichier ligne par ligne en Java

Les utilisateurs de Java 8 et plus peuvent également utiliser `Stream` pour lire des fichiers volumineux ligne par ligne. L'exemple ci-dessous illustre l'utilisation de `Stream` pour lire un fichier `txt` et afficher son contenu ligne par ligne.

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

public class Main {
    public static void main(String[] args) {
        String file = "my-file.txt";
        try (Stream<String> stream = Files.lines(Paths.get(fileName))) {
            stream.forEach(System.out::println);
        }
        catch (IOException e) {
            System.out.println("An error occurred.");
            e.printStackTrace();
        }
    }
}
```

Les deux méthodes discutées ci-dessus permettent de lire le fichier d'entrée ligne par ligne au lieu de lire le fichier entier en mémoire. Ces deux méthodes sont donc très efficaces si nous avons un énorme fichier qui ne peut pas être entièrement lu en mémoire.

Cependant, si notre mémoire est suffisamment grande pour lire le fichier d'entrée entièrement, nous pouvons également essayer la méthode ci-dessous.

Le `Scanner` pour lire un fichier ligne par ligne en Java

La classe `Scanner` en Java est un simple scanner de texte qui peut analyser des types primitifs et des chaînes de caractères à l'aide d'expressions régulières. Le `Scanner(File source)` lit le fichier complet en mémoire et le traite ensuite ligne par ligne.

L'exemple ci-dessous illustre l'utilisation de `Scanner` pour lire un fichier `txt` et sortir son contenu ligne par ligne.

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

public class Main {
    public static void main(String [] args) throws IOException {
        String fileName = "my-file.txt";
        Scanner scan = new Scanner(new File(fileName));
        while(scan.hasNextLine()){
            String line = scan.nextLine();
```

```
            System.out.println(line);
        }
    }
}
```

Nous avons discuté de trois méthodes pour lire un gros fichier texte en Java et le traiter ligne par ligne. Chaque méthode présente des contraintes et des avantages que nous devons prendre en compte lorsque nous décidons de la méthode à utiliser dans un scénario particulier.