

# Analisar CSV em Java

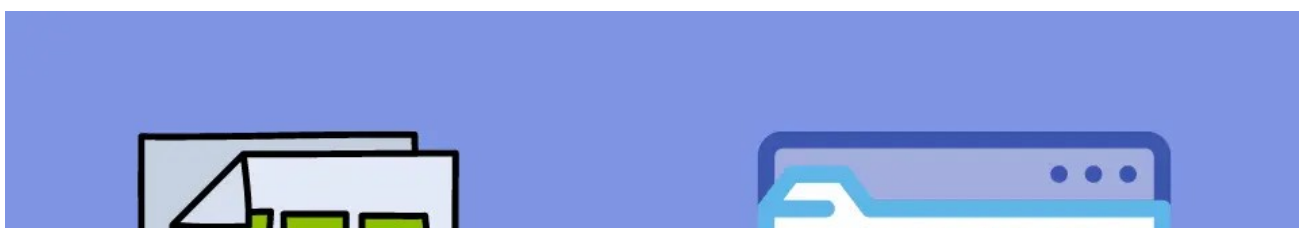
🏠 (/pt/) > HowTo (/pt/howto/) > Java Howtos (/pt/howto/java/) > Analisar CSV em Java (/pt/howto/java/parse-csv-in-java/)

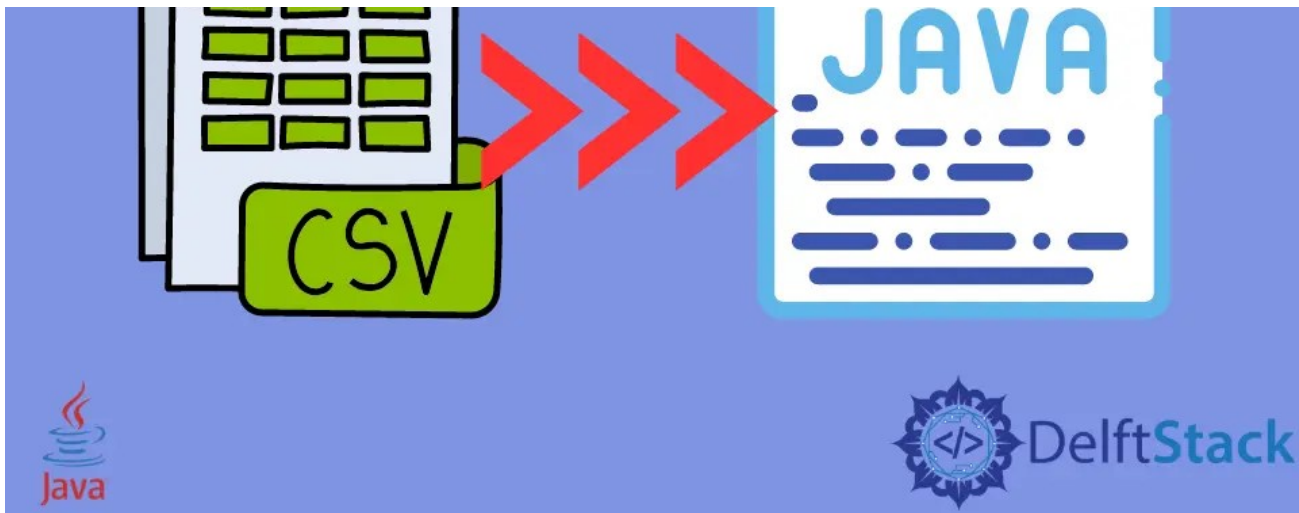
✍️ Rupam Yadav 🕒 12 outubro 2023

💎 Java (/pt/tags/java/) Java CSV (/pt/tags/java-csv/)  (http://www.facebook.com/sharer.php?src=bm&u=https%3a%2f%2fwww.delftstack.com%2fpt%2fhowto%2fjava%2fparse-csv-in-java%2f&t=Analisar%20CSV%20em%20Java)  (https://twitter.com/intent/tweet/?text=Java%20Howtos-Analisar%20CSV%20em%20Java&url=https%3a%2f%2fwww.delftstack.com%2fpt%2fhowto%2fjava%2fparse-csv-in-java%2f&hashtags=web,development)  (https://www.linkedin.com/shareArticle?mini=true&url=https%3a%2f%2fwww.delftstack.com%2fpt%2fhowto%2fjava%2fparse-csv-in-java%2f&title=Java%20Howtos-Analisar%20CSV%20em%20Java&source=https%3a%2f%2fwww.delftstack.com%2fpt%2fhowto%2fjava%2fparse-csv-in-java%2f&summary=Short%20summary)

## TABELA DE CONTEÚDO

1. Parse CSV utilizando o Scanner em Java
2. CSV Parse utilizando String.split() em Java
3. Analisar CSV usando openCSV em Java





Iremos abordar o tema da análise de um CSV em Java. Analisaremos os vários métodos para fazer a análise de Java CSV da amostra de CSV em Java.

**VDO.AI**  
([https://vdo.ai/contact?](https://vdo.ai/contact?utm_medium=video&utm_term=delftstack.com&utm_source=vdoai_logo)  
Powered By [vdo.ai/](https://vdo.ai/contact?utm_medium=video&utm_term=delftstack.com&utm_source=vdoai_logo)  
[contact?](https://vdo.ai/contact?utm_medium=video&utm_term=delftstack.com&utm_source=vdoai_logo)  
[utm\\_medium=video&utm\\_term=delftstack.com&utm\\_source=vdoai\\_logo](https://vdo.ai/contact?utm_medium=video&utm_term=delftstack.com&utm_source=vdoai_logo))

	A	B	C	D
1	Id	UserName	Age	Job
2	1	John Doe	24	Developer
3	2	Alex Johnson	43	Project Manager
4	3	Mike Stuart	26	Designer
5	4	Tom Sean	31	CEO
6				

## Parse CSV utilizando o Scanner em Java

A primeira e mais clássica forma de analisar um ficheiro CSV é utilizando a classe `Scanner` de Java. No exemplo, obtemos o ficheiro utilizando `File` e depois lêmo-lo por `Scanner`.

O mais importante a notar neste exemplo é que temos de conhecer o separador utilizado no CSV. Utilizamos a vírgula no CSV original como separador; é por isso que podemos especificar `,` como um delimitador no código.

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

public class Main {
    public static void main(String[] args) throws FileNotFoundException {
        File getCSVFiles = new File("/test/example.csv");
        Scanner sc = new Scanner(getCSVFiles);
        sc.useDelimiter(",");
        while (sc.hasNext()) {
            System.out.print(sc.next() + " | ");
        }
        sc.close();
    }
}
```

Resultado:

```
Id | UserName | Age | Job
1 | John Doe | 24 | Developer
2 | Alex Johnson | 43 | Project Manager
3 | Mike Stuart | 26 | Designer
4 | Tom Sean | 31 | CEO |
```

# CSV Parse utilizando `String.split()` em Java

No exemplo seguinte, utilizamos um método `split()` que funciona com a classe `String`. Podemos utilizar o `BufferedReader` para ler o ficheiro CSV e fazer loop através dele lendo cada linha até chegar a `null`.

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

public class Main {
    public static void main(String[] args) throws FileNotFoundException {
        String line = "";
        final String delimiter = ",";
        try {
            String filePath = "/test/example.csv";
            FileReader fileReader = new FileReader(filePath);

            BufferedReader reader = new BufferedReader(fileReader);
            while ((line = reader.readLine()) != null) // loops through every line until
            {
                String[] token = line.split(delimiter); // separate every token by comma
                System.out.println(token[0] + " | " + token[1] + " | " + token[2] + " | " + token[3]);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

## Resultado:

```
Id | UserName | Age | Job
1 | John Doe | 24 | Developer

2 | Alex Johnson | 43 | Project Manager
3 | Mike Stuart | 26 | Designer
4 | Tom Sean | 31 | CEO
```

# Analisar CSV usando OpenCSV em Java

Várias bibliotecas podem ajudar-nos a analisar o CSV em Java. Uma delas é OpenCSV. No exemplo abaixo, `CSVReader()` pega num `fileReader` com o ficheiro CSV e depois devolve um array de strings.

```
import com.opencsv.CSVReader;
import java.io.*;
import java.util.List;
import java.util.Scanner;
import java.util.Vector;

public class Main {
    public static void main(String[] args) throws Exception {
        try {
            String filePath = "/test/example.csv";
            FileReader fileReader = new FileReader(filePath);

            CSVReader openCSVReader = new CSVReader(fileReader);
            String[] record;

            while ((record = openCSVReader.readNext()) != null) {
                for (String token : record) {
                    System.out.print(token + "\t");
                }
                System.out.println();
            }

        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

## Resultado:

Id	UserName	Age	Job
1	John Doe	24	Developer
2	Alex Johnson	43	Project Manager
3	Mike Stuart	26	Designer
4	Tom Sean	31	CEO

**Autor: Rupam Yadav** (/pt/author/rupam-yadav/)

(/pt/author/rupam-yadav/)



Rupam Yadav  
avatar

(/pt/author/rupam-yadav/)

Rupam Saini is an android developer, who also works sometimes as a web developer., He likes to read books and write about various things.

 LinkedIn (<https://www.linkedin.com/in/rupam-saini-b0b752100/>)

## Artigo relacionado - Java CSV

(/pt/tags/java-csv/)

• Gravar Strings em Arquivo CSV em Java (/pt/howto/java/java-write-to-csv/)