

How to write to a file in java using FileOutputStream

BY CHAITANYA SINGH | FILED UNDER: [JAVA I/O](#)

Earlier we saw how to [create a file in Java](#). In this tutorial we will see how to write to a file in java using `FileOutputStream`. We would be using [write\(\) method](#) of `FileOutputStream` to write the content to the specified file. Here is the signature of `write()` method.

```
public void write(byte[] b) throws IOException
```

It writes `b.length` bytes from the specified byte array to this file output stream. As you can see this method needs array of bytes in order to write them into a file. Hence we would need to convert our content into array of bytes before writing it into the file.

Complete Code: Writing to a File

In the below example we are writing a `String` to a file. To convert the `String` into an array of bytes, we are using [getBytes\(\) method](#) of [String class](#).

```
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;

public class WriteFileDemo {
    public static void main(String[] args) {
        FileOutputStream fos = null;
        File file;
        String mycontent = "This is my Data which needs" +
            " to be written into the file";
        try {
            //Specify the file path here
            file = new File("C:/myfile.txt");
            fos = new FileOutputStream(file);

            /* This logic will check whether the file
             * exists or not. If the file is not found
             * at the specified location it would create
             * a new file*/
            if (!file.exists()) {
                file.createNewFile();
            }

            /*String content cannot be directly written into
             * a file. It needs to be converted into bytes
             */
            byte[] byteArray = mycontent.getBytes();

            fos.write(byteArray);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

```

        fos.flush();
        System.out.println("File Written Successfully");
    }
    catch (IOException ioe) {
        ioe.printStackTrace();
    }
    finally {
        try {
            if (fos != null)
            {
                fos.close();
            }
        }
        catch (IOException ioe) {
            System.out.println("Error in closing the Stream");
        }
    }
}
}
}

```

Output:

```
File Written Successfully
```