5 ways to convert InputStream to String in Java - Example Tutorial

InputStream to String Conversion Example

Converting `InputStream` to `String` in Java has become very easy after the introduction of `Scanner` class in Java 5 and due to the development of several open-source libraries like Apache commons `IOUtils` and Google Open source guava-libraries which provides excellent support to convert `InputStream` to `String` in Java program. we often need to convert `InputStream` to `String` while working in Java for example if you are reading XML files from InputStream and later performing XSLT transformation on it or if `InputStream` is reading data from text file or text input Source and we either want to log `String`s in a log file or want to operate on whole `String`.

Before Java 5 you would have to write lots of boilerplate code to read `String` line by line or byte by byte depending upon whether you are using either `BufferedReader` or not but as I said since JDK 5 added `Scanner` for reading input, its fairly easy to convert `InputStream` into `String`.

Before Converting any `InputStream` or `ByteStream` into `String` don't forget to provide character encoding or charset which tells Java Which characters to expect from those streams of bytes. in the absence of correct character encoding, you might alter the output because the same bytes can be used to represent different characters in a different encoding.

Another thing to keep in mind is that if you don't provide character encoding, Default character encoding in Java will be used which can be specified from System property "`file.encoding`" or "UTF-8" if file.encoding is not specified. In this Java tutorial, we will see 5 different examples of converting **`InputStream` to String** in Java both by using standard JDK libraries and using open source libraries.

How to convert InputStream to String in Java – 5 Examples

here are different ways to convert `InputStream` to `String` in Java, first we will see the most simple way of reading InputStream as String.

1. InputStream to String - Using Java 5 `Scanner`

java.util.`Scanner` has constructor which accept an `InputStream`, a character encoding and a delimiter to read `String` from `InputStream`. Here we have used delimiter as `"\A"` which is a boundary match for the beginning of the input as declared in `java.util.regex.Pattern` and that's why `Scanner` is returning whole `String` form `InputStream`.

I frequently use this technique to read input from users in Java using `System.in` which is most common example of `InputStream` in Java, but as demonstrated here this can also be used to read text file in Java.

```
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.util.Scanner;

/**
 * Java program example to demonstrate How to convert InputStream
into String by using JDK
 * Scanner utility. This program will work Java 5 onwards as Scanner
was added in Java 5.
 */
public class InputStreamTest {
```

```
  public static void main(String args[]) throws FileNotFoundException {
        FileInputStream fis = new FileInputStream("c:/sample.txt");
        String inputStreamString = new Scanner(fis,"UTF-
8").useDelimiter("\\A").next();
        System.out.println(inputStreamString);
    }
}

Output:
This String is read from InputStream by
changing InputStream to String in Java.
```