

JAVA.IO.PUSHBACKINPUTSTREAM.READ METHOD

http://www.tutorialspoint.com/java/io/pushbackinputstream_read_byte_len.htm

Copyright © tutorialspoint.com

Description

The **java.io.PushbackInputStream.read** method reads up to len bytes of data from this input stream into an array of bytes. This method first reads any pushed-back bytes; after that, if fewer than len bytes have been read then it reads from the underlying input stream. If len is not zero, the method blocks until at least 1 byte of input is available; otherwise, no bytes are read and 0 is returned.

Declaration

Following is the declaration for **java.io.PushbackInputStream.read** method.

```
public int read(byte[] b, int off, int len)
```

Parameters

- **b** -- the buffer into which the data is read.
- **off** -- the start offset in the destination array b.
- **len** -- the maximum number of bytes read.

Return Value

This method returns the total number of bytes read into the buffer, or -1 if there is no more data because the end of the stream has been reached.

Exception

- **NullPointerException** -- If b is null.
- **IndexOutOfBoundsException** -- If off is negative, len is negative, or len is greater than b.length - off
- **IOException** -- if this input stream has been closed by invoking its close method, or an I/O error occurs.

Example

The following example shows the usage of **java.io.PushbackInputStream.read** method.

```
package com.tutorialspoint;

import java.io.*;

public class PushbackInputStreamDemo {

    public static void main(String[] args) {

        // declare a buffer and initialize its size:
        byte[] arrByte = new byte[1024];

        // create an array for our message
        byte[] byteArray = new byte[]{'H', 'e', 'l', 'l', 'o'};

        // create object of PushbackInputStream class for specified stream
        InputStream is = new ByteArrayInputStream(byteArray);
        PushbackInputStream pis = new PushbackInputStream(is);

        try {
```

```
// read a char into our array
pis.read(arrByte, 0, 3);

// print arrByte
for (int i = 0; i < 3; i++) {
    System.out.print((char) arrByte[i]);
}

} catch (Exception ex) {
    ex.printStackTrace();
}
}
```

Let us compile and run the above program, this will produce the following result:

```
He1
```

```
Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js
```