

JAVA.IO.DATAINPUTSTREAM.READ METHOD

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

Copyright © tutorialspoint.com

Description

The **java.io.DataInputStream.readbyte[]b, intoff, intlen** method reads **len** bytes from the contained input stream and allocate them in the buffer **b** starting at **b[off]**. The method is blocked until input data is available, an exception is thrown or end of file is detected.

Declaration

Following is the declaration for **java.io.DataInputStream.readbyte[]b, intoff, intlen** method:

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

Parameters

- **b** - the byte[] into which the data is read from the input stream.
- **off** - the start offset in the b[].
- **len** - the maximum number of bytes read.

Return Value

Total number of bytes read, else -1 if the stream has reached the end.

Exception

- **IOException** -- if an I/O error occurs, the first byte cannot be read, or close is invoked before this method.
- **NullPointerException** -- if b is null.
- **IndexOutOfBoundsException** -- if len is greater than b.length - off, off is negative, or len is negative

Example

The following example shows the usage of **java.io.DataInputStream.readbyte[]b, intoff, intlen** method.

```
package com.tutorialspoint;

import java.io.DataInputStream;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;

public class DataInputStreamDemo {
    public static void main(String[] args) throws IOException {

        InputStream is = null;
        DataInputStream dis = null;

        try{
            // create input stream from file input stream
            is = new FileInputStream("c:\\test.txt");

            // create data input stream
            dis = new DataInputStream(is);

            // count the available bytes form the input stream
            int count = is.available();
        }
    }
}
```

```

// create buffer
byte[] bs = new byte[count];

// read len data into buffer starting at off
dis.read(bs, 4, 3);

// for each byte in the buffer
for (byte b:bs)
{
    // convert byte into character
    char c = (char)b;

    // empty byte as char '0'
    if(b ==0)
        c='0';

    // print the character
    System.out.print(c);
}
}catch(Exception e){
    // if any I/O error occurs
    e.printStackTrace();
}finally{

    // releases any associated system files with this stream
    if(is!=null)
        is.close();
    if(dis!=null)
        dis.close();
}
}
}

```

Assuming we have a text file **c:/test.txt**, which has the following content. This will file will be used as an input for our example program:

ABCDEFGH

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

00000ABC0

Loading [MathJax]/jax/output/HTML-CSS/jax.js