

JAVA.UTIL.PROPERTIES.STORETOXML METHOD

http://www.tutorialspoint.com/java/util/properties_storetoxml.htm

Copyright © tutorialspoint.com

Description

The **java.util.Properties.storeToXML***OutputStream**String**comment* method Emits an XML document representing all of the properties contained in this table. An invocation of this method of the form `props.storeToXML(os, comment)` behaves in exactly the same way as the invocation `props.storeToXML(os, comment, " UTF - 8 " ;)`.

Declaration

Following is the declaration for **java.util.Properties.storeToXML** method

```
public void storeToXML(OutputStream os,String comment)
```

Parameters

- **out** -- the output stream on which to emit the XML document.
- **comments** -- a description of the property list, or null if no comment is desired.

Return Value

This method does not return a value

Exception

- **IOException** -- if writing this property list to the specified output stream throws an IOException.
- **ClassCastException** -- if this Properties object contains any keys or values that are not Strings.
- **NullPointerException** -- if out is null.

Example

The following example shows the usage of `java.util.Properties.storeToXML` method.

```
package com.tutorialspoint;

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.*;

public class PropertiesDemo {

    public static void main(String[] args) {
        Properties prop = new Properties();

        // add some properties
        prop.put("Height", "200");
        prop.put("Width", "15");

        try {

            // create a output and input as a xml file
            FileOutputStream fos = new FileOutputStream("properties.xml");
            FileInputStream fis = new FileInputStream("properties.xml");

            // store the properties in the specific xml
            prop.storeToXML(fos, "Properties Example");
```

```
// print the xml
while (fis.available() > 0) {
    System.out.print("'" + (char) fis.read());
}

} catch (IOException ex) {
    ex.printStackTrace();
}

}
```

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

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<comment>Properties Example</comment>
<entry key="Width">15</entry>
<entry key="Height">200</entry>
</properties>
```

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