

JAVA.UTIL.RESOURCEBUNDLE.CONTROL.GETFORMATS METHOD

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

Copyright © tutorialspoint.com

Description

The **java.util.ResourceBundle.Control.getFormats** method returns a List of Strings containing formats to be used to load resource bundles for the given baseName. The ResourceBundle.getBundle factory method tries to load resource bundles with formats in the order specified by the list. The list returned by this method must have at least one String.

The predefined formats are "java.class" for class-based resource bundles and "java.properties" for properties-based ones. Strings starting with "java." are reserved for future extensions and must not be used by application-defined formats.

Declaration

Following is the declaration for **java.util.Control.getFormats** method

```
public List<String> getFormats(String baseName)
```

Parameters

- **baseName** -- the base name of the resource bundle, a fully qualified class name

Return Value

This method returns a List of Strings containing formats for loading resource bundles.

Exception

- **NullPointerException** -- if baseName is **null**

Example

The following example shows the usage of java.util.ResourceBundle.Control.getFormats method.

```
package com.tutorialspoint;

import java.util.ResourceBundle;
import java.util.ResourceBundle.Control;

public class ResourceBundleControlDemo {

    public static void main(String[] args) {

        // create a new ResourceBundle.Control with default format
        ResourceBundle.Control rbc =
            ResourceBundle.Control.getControl(Control.FORMAT_DEFAULT);

        // print formats
        System.out.println(" " + rbc.getFormats("hello"));
    }
}
```

Assuming we have a resource file **hello_en_US.properties** available in your CLASSPATH, with the following content. This file will be used as an input for our example program:

```
hello=Hello World!
```

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

```
[java.class, java.properties]
```

