

JAVA.UTIL.RESOURCEBUNDLE.GETBUNDLE METHOD

http://www.tutorialspoint.com/java/util/resourcebundle_getbundle_targetlocale.htm Copyright © tutorialspoint.com

Description

The **java.util.ResourceBundle.getBundle**

String **baseName**, *Locale* **targetLocale**, *ResourceBundle*.*Control* **control** method returns a resource bundle using the specified base name, target locale and control, and the caller's class loader.

Declaration

Following is the declaration for **java.util.ResourceBundle.getBundle** method

```
public static final ResourceBundle getBundle(String baseName, Locale
targetLocale, ResourceBundle.Control control)
```

Parameters

- **baseName** -- the base name of the resource bundle, a fully qualified class name
- **locale** -- the locale for which a resource bundle is desired
- **control** -- the control which gives information for the resource bundle loading process

Return Value

This method returns a resource bundle for the given base name and a Locale in locales

Exception

- **NullPointerException** -- if baseName, locales or control is **null**
- **MissingResourceException** -- if no resource bundle for the specified base name in any of the locales can be found.
- **IllegalArgumentException** -- if the given control doesn't perform properly (e.g., control.getCandidateLocales returns **null**.) Note that validation of control is performed as needed.

Example

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

```
package com.tutorialspoint;

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

public class ResourceBundleDemo {

    public static void main(String[] args) {

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

        // create a new ResourceBundle with default locale and a Control
        ResourceBundle bundle =
            ResourceBundle.getBundle("hello", Locale.US, rbc);

        // print the text assigned to key "hello"
        System.out.println(" " + bundle.getString("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:

```
Hello World!
```

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