

JAVA.UTIL.RESOURCEBUNDLE.GETBUNDLE METHOD

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

Copyright © tutorialspoint.com

Description

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

String *baseName*, *Locale* *targetLocale*, *ClassLoader* *loader*, *ResourceBundle.Control* *control* method returns a resource bundle using the specified base name, target locale, class loader and control. Unlike the `getBundle` factory methods with no control argument, the given control specifies how to locate and instantiate resource bundles.

Declaration

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

```
public static ResourceBundle getBundle(String baseName, Locale targetLocale, ClassLoader loader, 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
- **loader** -- the class loader from which to load the resource bundle
- **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 locale

Exception

- **NullPointerException** -- if `baseName`, `locale`, `loader` or `control` is **null**
- **MissingResourceException** -- if no resource bundle for the specified base name 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) {

        ClassLoader cl = ClassLoader.getSystemClassLoader();

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

        // create a new ResourceBundle with specified locale
        // and SystemClassLoader and a Control
```

```
ResourceBundle bundle =  
ResourceBundle.getBundle("hello", Locale.US, cl, 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/jax.js
```