

JAVA.LANG.CLASS.FORNAME METHOD

http://www.tutorialspoint.com/java/lang/class_forname_loader.htm

Copyright © tutorialspoint.com

Description

The **java.lang.Class.forName**(*String* name, *boolean* initialize, *ClassLoader* loader) method returns the Class object associated with the class or interface with the given string name, using the given class loader.

The specified class loader is used to load the class or interface. If the parameter **loader** is null, the class is loaded through the bootstrap class loader. The class is initialized only if the **initialize** parameter is true and if it has not been initialized earlier.

Declaration

Following is the declaration for **java.lang.Class.forName** method

```
public static Class<?> forName(String name, boolean initialize, ClassLoader loader)
throws ClassNotFoundException
```

Parameters

- **name** -- This is the fully qualified name of the desired class.
- **initialize** -- This shows whether the class must be initialized.
- **loader** -- This is the class loader from which the class must be loaded.

Return Value

This method returns the class object representing the desired class.

Exception

- **LinkageError** -- if the linkage fails.
- **ExceptionInInitializerError** -- if the initialization provoked by this method fails.
- **ClassNotFoundException** -- if the class cannot be located by the specified class loader.

Example

The following example shows the usage of java.lang.Class.forName method.

```
package com.tutorialspoint;

import java.lang.*;

public class ClassDemo {

    public static void main(String[] args) {

        try {
            Class cls = Class.forName("ClassDemo");

            // returns the ClassLoader object
            ClassLoader cLoader = cls.getClassLoader();

            /* returns the Class object associated with the class or interface
            with the given string name, using the given classloader. */
            Class cls2 = Class.forName("java.lang.Thread", true, cLoader);

            // returns the name of the class
            System.out.println("Class = " + cls.getName());
        }
    }
}
```

```
        System.out.println("Class = " + cls2.getName());
    }
    catch(ClassNotFoundException ex) {
        System.out.println(ex.toString());
    }
}
```

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

```
Class = ClassDemo
Class = java.lang.Thread
```

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