JAVA.LANG.RUNTIME.LOAD METHOD

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

Description

The **java.lang.Runtime.load**$Stringfilename$ method loads the specified filename as a dynamic library. The filename argument must be a complete path name, $forexampleRuntime.\,getRuntime($.load $"\,/home/avh/lib/libX11.so\,"$;). First, if there is a security manager, its checkLink method is called with the filename as its argument. This may result in a security exception. This is similar to the method loadLibrary$String$, but it accepts a general file name as an argument rather than just a library name, allowing any file of native code to be loaded. The method System.load$String$ is the conventional and convenient means of invoking this method.

Declaration

Following is the declaration for **java.lang.Runtime.load** method

```
public void load(String filename)
```

Parameters

- **filename** -- the file to load.

Return Value

This method does not return a value.

Exception

- **SecurityException** -- if a security manager exists and its checkLink method doesn't allow loading of the specified dynamic library
- **UnsatisfiedLinkError** -- if the file does not exist
- **NullPointerException** -- if filename is null

Example

The following example shows the usage of lang.Runtime.load method.

```
package com.tutorialspoint;

public class RuntimeDemo {

   public static void main(String[] args) {

   // print when the program starts
   System.out.println("Program starting...");

   // load a library that is Windows/System32 folder
   System.out.println("Loading Library...");
   Runtime.getRuntime().load("C:/Windows/System32/crypt32.dll");
   System.out.println("Library Loaded.");

   }
}
```

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

```
Program starting...
Loading Library...
Library Loaded.
```

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