

JAVA.LANG.SECURITYMANAGER.CHECKLISTEN METHOD

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

Copyright © tutorialspoint.com

Description

The **java.lang.SecurityManager.checkListen** method throws a `SecurityException` if the calling thread is not allowed to wait for a connection request on the specified local port number.

If port is not 0, this method calls `checkPermission` with the `SocketPermission` "localhost:" + port, "listen". If port is zero, this method calls `checkPermission` with `SocketPermission` "localhost:1024-", "listen". If you override this method, then you should make a call to `super.checkListen` at the point the overridden method would normally throw an exception.

Declaration

Following is the declaration for **java.lang.SecurityManager.checkListen** method

```
public void checkListen(int port)
```

Parameters

- **port** -- the local port.

Return Value

This method does not return a value.

Exception

- **SecurityException** -- if the calling thread does not have permission to listen on the specified port.

Example

Our examples require that the permissions for each command is blocked. A new policy file was set that allows only the creating and setting of our Security Manager. The file is in C:/java.policy and contains the following text:

```
grant {  
    permission java.lang.RuntimePermission "setSecurityManager";  
    permission java.lang.RuntimePermission "createSecurityManager";  
    permission java.lang.RuntimePermission "usePolicy";  
};
```

The following example shows the usage of `lang.SecurityManager.checkListen` method.

```
package com.tutorialspoint;  
  
public class SecurityManagerDemo extends SecurityManager {  
  
    // checkListen needs to be overridden  
    @Override  
    public void checkListen(int port) {  
        throw new SecurityException();  
    }  
  
    public static void main(String[] args) {  
  
        // set the policy file as the system security policy  
        System.setProperty("java.security.policy", "file:C:/java.policy");  
  
        // create a security manager  
        SecurityManagerDemo sm = new SecurityManagerDemo();  
    }  
}
```

```
// set the system security manager
System.setSecurityManager(sm);

// perform the check
sm.checkListen(8080);

// print a message if we passed the check
System.out.println("Allowed!");
}
```

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

```
Exception in thread "main" java.lang.SecurityException
Loading [MathJax]/jax/output/HTML-CSS/jax.js
```