

SECURITYMANAGER CHECKPERMISSION METHOD

http://www.tutorialspoint.com/java/lang/securitymanager_checkpermission_object.htm Copyright © tutorialspoint.com

Description

The **java.lang.SecurityManager.checkPermission***Permissionperm, Objectcontext* method throws a **SecurityException** if the specified security context is denied access to the resource specified by the given permission. The context must be a security context returned by a previous call to **getSecurityContext** and the access control decision is based upon the configured security policy for that security context.

If context is an instance of **AccessControlContext** then the **AccessControlContext.checkPermission** method is invoked with the specified permission. If context is not an instance of **AccessControlContext** then a **SecurityException** is thrown.

Declaration

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

```
public void checkPermission(Permission perm, Object context)
```

Parameters

- **perm** -- the requested permission.
- **context** -- a system-dependent security context.

Return Value

This method does not return a value.

Exception

- **SecurityException** -- if the specified security context is not an instance of **AccessControlContext** *e. g.* , *isnull*, or is denied access to the resource specified by the given permission.
- **NullPointerException** -- if the permission argument is null.

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.checkPermission** method.

```
package com.tutorialspoint;  
  
import java.io.FilePermission;  
import java.security.AccessControlContext;  
import java.security.AccessController;  
  
public class SecurityManagerDemo extends SecurityManager {  
    public static void main(String[] args) {
```

```
// get current security context
AccessControlContext con = AccessController.getContext();

// 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.checkPermission(new FilePermission("test.txt", "read,write"), con);

// 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.security.AccessControlException: access denied
(java.io.FilePermission test.txt read.write)
Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js
```