

JAVA.LANG.SECURITYMANAGER.CHECKREAD METHOD

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

Copyright © tutorialspoint.com

Description

The **java.lang.SecurityManager.checkReadStringfile, Objectcontext** method throws a `SecurityException` if the specified security context is not allowed to read the file specified by the string argument. The context must be a security context returned by a previous call to `getSecurityContext`.

If context is an instance of `AccessControlContext` then the `AccessControlContext.checkPermission` method will be invoked with the `FilePermissionfile, " read "` permission. If context is not an instance of `AccessControlContext` then a `SecurityException` is thrown. If you override this method, then you should make a call to `super.checkRead` at the point the overridden method would normally throw an exception.

Declaration

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

```
public void checkRead(String file, Object context)
```

Parameters

- **file** -- the system-dependent file name.
- **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 does not have permission to read the specified file.
- **NullPointerException** -- if the file 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.checkRead` method.

```
package com.tutorialspoint;  
  
import java.security.AccessControlContext;  
import java.security.AccessController;  
  
public class SecurityManagerDemo extends SecurityManager {  
  
    public static void main(String[] args) {  
  
        // get current 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.checkRead("test.txt", 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)
Loading [MathJax]/jax/output/HTML-CSS/jax.js
```