Table of Contents

If you use a SecurityManager, you can restrict the access to life cycle objects. But sometimes you need restrictions for custom objects or methods of a life cycle object.

We are not big fans of “configuration til death” because config files are nothing more than config files. Nobody should fill config files with application logic. Keep your logic in your source code.

If you implement `com.sibvisions.rad.server.security.IObjectAccessController`, you'll have full control over object and method calls. The interface offers following methods:

```
public boolean isObjectAccessAllowed(AbstractObjectProvider pProvider,
ISession pSession,
                                     Map pLifeCycleObject, String
pObjectName,
                                     Object pObject);

public boolean isMethodInvocationAllowed(AbstractObjectProvider pProvider,
                                         ISession pSession, String
pObjectName,
                                         Object pObject, String pMethodName,
                                         Object... pParams);
```

The first method checks if it's allowed to call an object from a life cycle object. You have access to all relevant objects like life cycle object, called object, and the object name.

The second method checks if it's allowed to call a method from an object. You have access to all relevant information like object to call, method name, parameter, and the object instance itself.

Example

This implementation only allows calls for an object with the name “address”.

SimpleAddressAccessController.java

```
public class SimpleAddressAccessController implements
IObjectAccessController
{
        public boolean isObjectAccessAllowed(AbstractObjectProvider
pProvider,
                                             ISession pSession, Map
pLifeCycleObject,
                                             String pObjectName, Object
pObject)
        {
                if ("address".equals(pObjectName))
                {
                        return true;
                }

                return false;
        }
```

```
        public boolean isMethodInvocationAllowed(AbstractObjectProvider
pProvider,
                                                 ISession pSession,
String pObjectName,
                                                 Object pObject, String
pMethodName,
                                                 Object... pParams)
        {
                if ("address".equals(pObjectName))
                {
                        return true;
                }

                return false;
        }

}
```

We use the object name for our checks. It is also possible to check class names or instances and is not a problem to read the configuration from an XML file if you prefer.

From:
https://doc.sibvisions.com/ - **Documentation**

Permanent link:
https://doc.sibvisions.com/jvx/server/security/object_method_access

Last update: **2020/06/25 10:52**