3.25 JMX

Généralités

Le support de JMX dans Spring offre des fonctionnalités simples et transparentes pour intégrer votre application Spring dans une infrastructure JMX. Spring JMX vous permet entre autres d'enregistrer automatiquement n'importe quel bean Spring en tant que **MBean** JMX.

La classe principale de Spring JMX est **MBeanExporter**. Cette classe est responsable de prendre en charge et d'enregistrer vos beans Spring dans un **MBeanServer**.

Pour exposer les propriétés et les méthodes d'un bean en tant qu'attributs et opérations d'un MBean, il faut simplement configurer une instance de la classe MBeanExporter dans le fichier de configuration.

La propriété "beans" de la classe MBeanExporter permet de savoir exactement quels beans doivent être exportés au MBeanServer.

Spring JMX offre aussi la possibilité de créer un connecteur serveur pour permettre l'accès au MBeanServer à distance.

Exemple: surveillance d'un service CMIS par une application "cliente"

Une application de gestion de fichiers s'appuie sur un serveur CMIS pour stocker ses fichiers. Si ce serveur CMIS n'est pas disponible (pour maintenance, par exemple), l'application doit continuer à fonctionner en "mode dégradé" .

Le service **cmisFileStorageService** doit savoir en temps réel si le service CMIS est actif ou non. Le bean **jmxTestCmis,** grâce à sa méthode **getCmisIsReady()**, donne cette information.

Les attributs et méthodes de ce bean sont exposés comme MBean grâce à la classe Spring **MBeanExporter.**

Les définition des beans par Spring :

```
<bean id="cmisFileStorageService" class="org.esupportail.application.service.fileStorage.
CmisFileStorageServiceImpl" lazy-init="true">
        <description>A bean to manage files upload and download to the CMIS server.</description>
         ...
        <property name="jmxTestCmis" ref="jmxTestCmis"/>
    </bean>

    <bean id="jmxTestCmis"
        class="org.esupportail.application.service.fileStorage.JmxTestCmisImpl">
        ...
    </bean>

    <!-  JMX to control CMIS ->
    <bean id="exporter" class="org.springframework.jmx.export.MBeanExporter">
        <property name="beans">
            <map>
                <entry key="bean:name=testCmis" value-ref="jmxTestCmis"/>
            </map>
        </property>
    </bean>
```

La classe métier utilisant le bean JMX

Avant toute opération sur les fichiers elle contrôle la disponibilité du serveur CMIS et affiche un message d'erreur s'il n'est pas actif.

```
public class CmisFileStorageServiceImpl  {
     ....

    /**
     * Bean to test if cmis is ready.
     */
    private JmxTestCmis jmxTestCmis;

    public void afterPropertiesSet() throws Exception{
          ....
          Assert.notNull(this.jmxTestCmis,
             "property jmxTestCmis of class " + this.getClass().getName() + " can not be null");
    }
    ...
    /**
     * @param jmxTestCmis the jmxTestCmis to set
     */
    public void setJmxTestCmis(final JmxTestCmis jmxTestCmis){
          this.jmxTestCmis = jmxTestCmis;
    }

    public FileStorage getFile(...) throws IOException {
        FileStorage file = null;
        if (jmxTestCmis.getCmisIsReady()) {
            file= ...;
        }
        else {
             System.err.println("--------le service CMIS est indisponible");
        }

        return file;
    }
}
```

Le bean JMX (MBean)

```
package org.esupportail.application.services.fileStorage;
public interface JmxTestCmis {
    /**
     * Enable the use of CMIS.
     */
    void enabled();

    /**
     * Disable the use of CMIS.
     */
    void disabled();

    /**
     * @return the cmisIsReady
     */
    Boolean getCmisIsReady();
}
```

```
package org.esupportail.application.services.fileStorage;
import java.io.Serializable;
public class JmxTestCmisImpl implements JmxTestCmis, Serializable {
    /**
     * True if cmis is ready.
     * Default value = true.
     */
    private Boolean cmisIsReady;
    public JmxTestCmisImpl() {
        super();
        cmisIsReady = true;
    }

    @Override
    public void disabled() {
        cmisIsReady = false;
    }

    @Override
    public void enabled() {
        cmisIsReady = true;
    }

    public Boolean getCmisIsReady() {
        return cmisIsReady;
    }
}
```

Activation de l'agent JMX

Si l'application écoute le port jmx 9656, par exemple, au lancement de l'application, on aura les options suivantes:

```
-Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.port=9656
-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false
```

Communication avec l'agent JMX

L'agent JMX est appelé par l'application délivrant le service CMIS

activationJMX.sh

```
#!/bin/csh
set cmdLineJMXJar=/opt/admin-java/jmx/cmdline-jmxclient.jar
set jmxHost=host.etablissement.fr
set jmXPort=9656
#No User and password so pass '-'
echo "active l'utilisation de CMIS pour les applications en ecoute sur le port 9656"
/opt/jdk1.6.0/bin/java -jar ${cmdLineJMXJar} - ${jmxHost}:${port} bean:name=testCmis enabled
```

desactivationJMX.sh

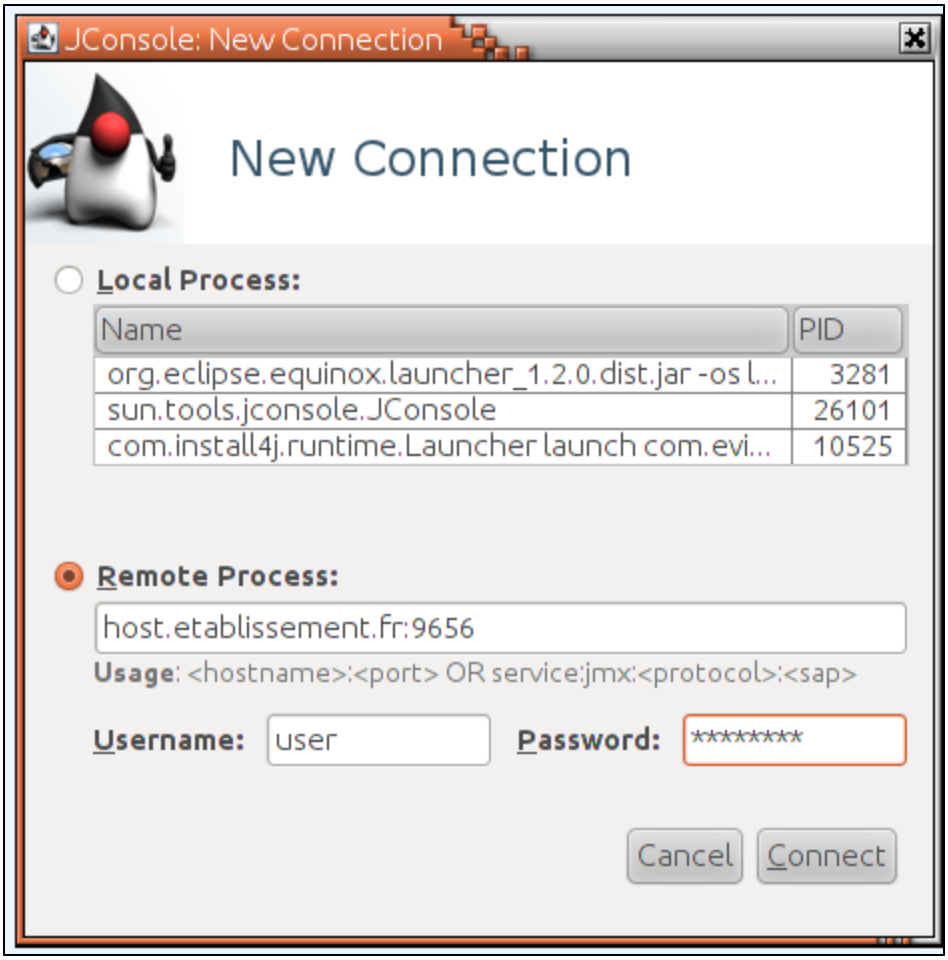
```
#!/bin/csh
set cmdLineJMXJar=/opt/admin-java/jmx/cmdline-jmxclient.jar
set port=9656
set jmxHost=host.etablissement.fr
#No User and password so pass '-'
echo "désactive l'utilisation de CMIS pour les applications en ecoute sur le port 9656"
/opt/jdk1.6.0/bin/java -jar ${cmdLineJMXJar} - ${jmxHost}:${port} bean:name=testCmis disabled
```

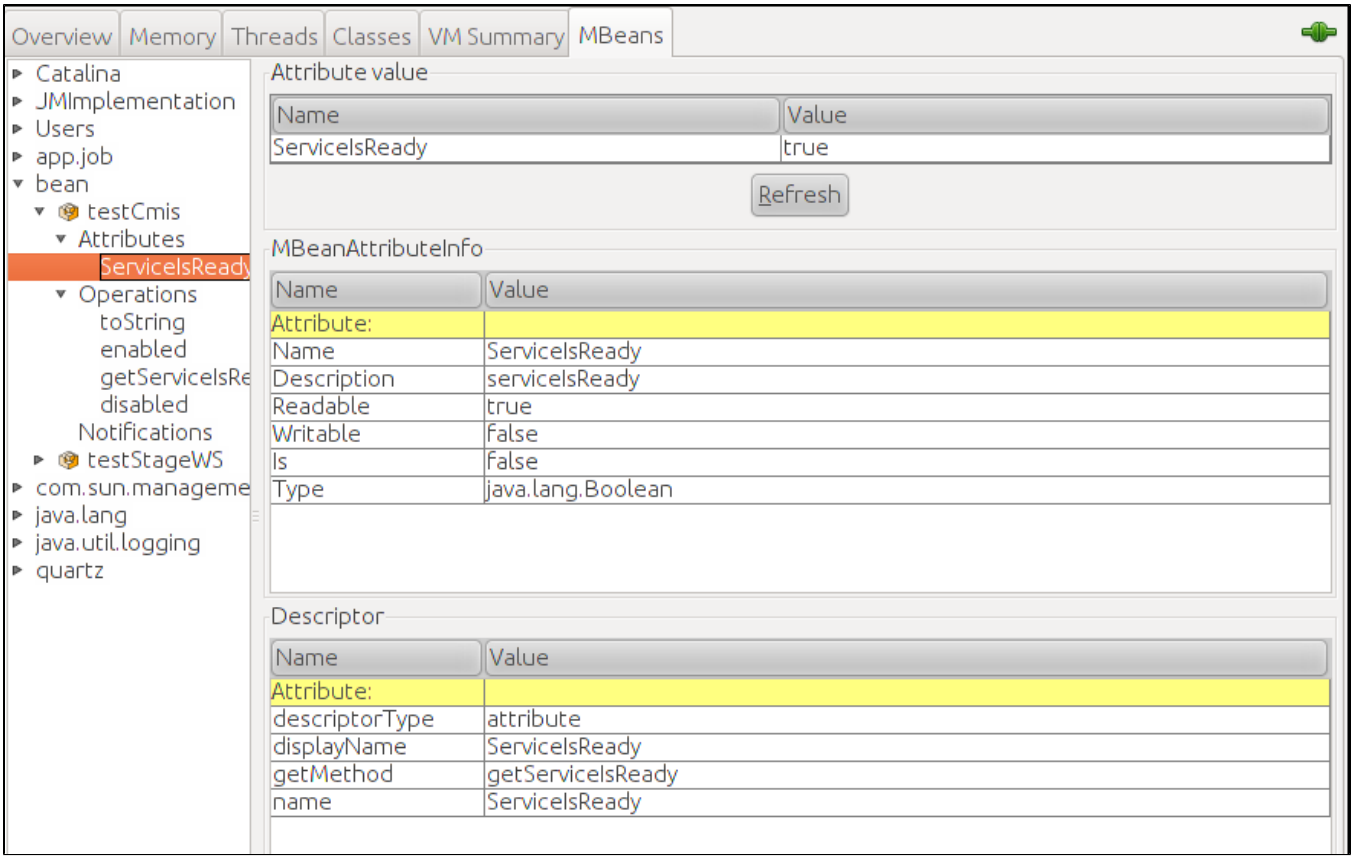
Remarques

1. L'application cliente JMX ne fait ici que lire l'état du serveur CMIS. On pourrait imaginer aller plus loin et vouloir l'administrer à distance. Le serveur de MBean (MbeanServer)
2. Le serveur de MBean (MbeanServer) utilisé ici est celui par défaut (celui de tomcat par exemple).

La console

La console java (jconsole) permet de visualiser et d'administrer les bean jmx instanciés sur un serveur d'application distant.




Overview
Memory
Threads
Classes
VM Summary
MBeans
Catalina
JMImplementation
Users
app.job
bean
testCmis
Attributes
ServiceIsReady
Operations
toString
enabled
getServiceIsRe
disabled
Notifications
testStageWS
com.sun.manageme
java.lang
java.util.logging
quartz
Attribute value
Name
Value
ServiceIsReady
true
Refresh
MBeanAttributeInfo
Name
Value
Attribute:
Name
ServiceIsReady
Description
serviceIsReady
Readable
true
Writable
false
Is
false
Type
java.lang.Boolean
Descriptor
Name
Value
Attribute:
descriptorType
attribute
displayName
ServiceIsReady
getMethod
getServiceIsReady
name
ServiceIsReady