

1. Accès distants aux ejb3.1 de Jboss 7.1.1

L'accès externe et distant au serveur Jboss 7 via le registre des noms logiques JNDI a été complètement chamboulé (comparé aux anciennes versions 3, 4.2 ,et 5.1 de Jboss). Le numéro de port (par défaut) de la partie JNDI est 4447 (et plus 1099) .

1.1. Liste des librairies nécessaires (coté client)

La librairie suivante doit être à recopiée dans le classpath de l'application cliente externe :

- **jboss-client.jar** (à récupérer dans **JBOSS7_HOME/bin/client**)

Eventuellement , junit-4.8.1.jar peut être ajouté (à des fins de tests unitaires).

1.2. Configuration et code d'accès distant à un ejb "@Remote"

```
java.util.Properties jndiProps = null;
jndiProps= new java.util.Properties();
jndiProps.setProperty(Context.URL_PKG_PREFIXES,
                        "org.jboss.ejb.client.naming");
try {
Context ic = new InitialContext(jndiProps);
serviceGestionComptes=(GestionComptes)
    ic.lookup("ejb:minibank-jee6/minibank-jee6EJB//GestionComptesImpl!
tp.myapp.minibank.impl.domain.ejb.itf.GestionComptesRemote");
Compte cpt = serviceGestionComptes.getCompteByNum(1L);
System.out.println("compte 1 = " + cpt.toString());
//Assert.assertTrue(cpt.getNumero()==1L);
}
catch (NamingException e){
e.printStackTrace();
}
```

NB1:Le simple fait d'avoir préciser la propriété **URL_PKG_PREFIXES** à la valeur **"org.jboss.ejb.client.naming"** couplé au chargement des bonnes librairies en mémoire provoque une *lecture et interprétation automatique* d'un fichier **jboss-ejb-client.properties** qui doit être présent à la racine du classpath du client.

Ce fichier **jboss-ejb-client.properties** doit ressembler au suivant :

```
remote.connectionprovider.create.options.org.xnio.Options.SSL_ENABLED=false
remote.connections=default
remote.connection.default.host=localhost
remote.connection.default.port=4447
remote.connection.default.connect.options.org.xnio.Options.SASL_POLICY_NOANONYMOUS=false
```

NB2 : depuis JEE6 (et EJB3.1) , les noms JNDI doivent suivre des formats normalisés :

// **jndi external names (from external EJB client app):**

//**ejb:**<app-name>/<module-name>/<distinct-name>/<bean-name>!<fully-qualified-classname-of-the-remote-interface>

Le "distinct-name" peut être vide

Exemple :

```
"ejb:minibank-jee6/minibank-jee6EJB//GestionComptesImpl!
tp.myapp.minibank.impl.domain.ejb.itf.GestionComptesRemote"
```

Attention : un chemin jndi de type

"java:global/minibank-jee6/minibank-jee6EJB/GestionComptesImpl!
tp.myapp.minibank.impl.domain.ejb.itf.GestionComptesRemote" utilisé depuis un lookup
déclenché par un programme externe à Jboss mène à l'exception suivante :
"javax.naming.NoInitialContextException: Need to specify class name in environment or
system property" .

==> il faut donc absolument que le chemin JNDI soit au format attendu
"ejb:...." mentionné précédemment !

1.3. Configuration d'un accès distant à une file JMS

```
public static void main(String[] args) {
    try {
        Properties jndiProps = new Properties();
        jndiProps.put(Context.INITIAL_CONTEXT_FACTORY,
            "org.jboss.naming.remote.client.InitialContextFactory");
        jndiProps.put(Context.PROVIDER_URL, "remote://localhost:4447");
        jndiProps.put(Context.SECURITY_PRINCIPAL, "admin");
        jndiProps.put(Context.SECURITY_CREDENTIALS, "pwd");
        //avec admin = utilisateur ajouté via la commande JBOSS7_HOME/bin/add-user
        //mot de passe=pwd et rôles associés admin,guest
        //et avec "guest" = rôle configuré sur la partie "messaging" de standalone(-full).xml
        Context ic = new InitialContext(jndiProps);

        QueueConnectionFactory factory = (QueueConnectionFactory)
            ic.lookup("jms/RemoteConnectionFactory");
        //avec <entry name="java:jboss/exported/jms/RemoteConnectionFactory"/>
        //dans standalone(-full).xml

        Queue queue = (Queue) ic.lookup("jms/queue/test");
        // avec queue/test doit etre exporté dans standalone(-full).xml
        //<entry name="java:jboss/exported/jms/queue/test"/>

        QueueConnection cnn = factory.createQueueConnection(
            jndiProps.getProperty(Context.SECURITY_PRINCIPAL),
            jndiProps.getProperty(Context.SECURITY_CREDENTIALS));
        QueueSession session = cnn.createQueueSession(false,
            QueueSession.AUTO_ACKNOWLEDGE);
```

```
    TextMessage msg = session.createTextMessage();
    msg.setText("<msg>message in the bottle</msg>");
    QueueSender sender = session.createSender(queue);
    sender.send(msg);
    System.out.println("Message sent successfully to remote queue.");
} catch (Exception e) {
    e.printStackTrace();
}
}
```

NB1: Les connexions JNDI et JMS sont sécurisées avec Jboss 7 et doivent être associées à une authentification d'un utilisateur configuré du serveur Jboss7 ayant le rôle "guest" (ou ...) lui même configuré dans standalone(-full).xml

La commande interactive **add-user** de JBOSS7_HOME/bin permet d'ajouter un utilisateur avec un mot de passe crypté dans standalone/configuration/**application-users.properties** et avec une liste de rôles en clair dans **application-roles.properties** .

NB2 : Les noms JNDI qui sont vus depuis les applications externes sont ceux qui sont préfixés par "java:jboss/exported" dans standalone(-full).xml