DataSource Ciphered Password

Example datasource-ciphered-password can be browsed at https://github.com/apache/tomee/tree/master/examples/datasource-ciphered-password

Datasource Ciphered Password example

This example shows how to use a ciphered password with an OpenEJB datasource.

It shows how to implement its own encryption too.

Configuration

The configuration is simply a datasource configuration with an additionnal parameter "PasswordCipher" to specify the encryption to use.

Example using Static3DES encryption:

```
properties.setProperty("ProtectedDatasource", "new://Resource?type=DataSource");
properties.setProperty("ProtectedDatasource.JdbcDriver", "org.hsqldb.jdbcDriver");
properties.setProperty("ProtectedDatasource.JdbcUrl", "jdbc:hsqldb:mem:protected");
properties.setProperty("ProtectedDatasource.UserName", "user");
// the plain text password is "YouLLN3v3rFindM3"
properties.setProperty("ProtectedDatasource.Password",
"fEroTNXjaL5SOTyRQ92x3DNVS/ksbtgs");
properties.setProperty("ProtectedDatasource.PasswordCipher", "Static3DES");
properties.setProperty("ProtectedDatasource.JtaManaged", "true");
```

Using its own implementation

The example implement a reverse encryption which simply reverse the password to encrypt/decrypt.

The implementation is done with commons-lang library:

```java
public static class ReverseEncryption implements PasswordCipher {
    @Override public char[] encrypt(String plainPassword) {
        return StringUtils.reverse(plainPassword).toCharArray();
    }

    @Override public String decrypt(char[] encryptedPassword) {
        return new String(encrypt(new String(encryptedPassword)));
    }
}
```

To be functional it needs the file META-INF/org.apache.openejb.resource.jdbc.PasswordCipher/reverse.

The file name (reverse) define the encryption name to use for the PasswordCipher parameter.

This file simply contains the implementation class of the encryption.

Then you simply declare this encryption for your datasource:

```
properties.setProperty("ProtectedDatasource", "new://Resource?type=DataSource");
properties.setProperty("ProtectedDatasource.JdbcDriver", "org.hsqldb.jdbcDriver");
properties.setProperty("ProtectedDatasource.JdbcUrl", "jdbc:hsqldb:mem:protected");
properties.setProperty("ProtectedDatasource.UserName", USER);
properties.setProperty("ProtectedDatasource.Password", "3MdniFr3v3NLLuoY");
properties.setProperty("ProtectedDatasource.PasswordCipher", "reverse");
properties.setProperty("ProtectedDatasource.JtaManaged", "true");
```

Documentation

For more information please see the documentation

Full Test Source

```java
package org.superbiz;

import org.apache.commons.lang.StringUtils;
import org.apache.openejb.resource.jdbc.PasswordCipher;
import org.junit.BeforeClass;
import org.junit.Test;

import javax.annotation.Resource;
import javax.ejb.embeddable.EJBContainer;
import javax.naming.Context;
import javax.sql.DataSource;
import java.sql.Connection;
```

```java
import java.sql.DriverManager;
import java.sql.Statement;
import java.util.Properties;

import static junit.framework.Assert.assertNotNull;

public class DataSourceCipheredExampleTest {
    private static final String USER = DataSourceCipheredExampleTest.class
.getSimpleName().toUpperCase();
    private static final String PASSWORD = "YouLLN3v3rFindM3";
    private static final String DATASOURCE_URL = "jdbc:hsqldb:mem:protected";

    @Resource
    private DataSource dataSource;

    @BeforeClass
    public static void addDatabaseUserWithPassword() throws Exception {
        Class.forName("org.hsqldb.jdbcDriver");
        Connection conn = DriverManager.getConnection(DATASOURCE_URL, "sa", "");
        conn.setAutoCommit(true);
        Statement st = conn.createStatement();
        st.executeUpdate("CREATE USER " + USER + " PASSWORD '" + PASSWORD + "';");
        st.close();
        conn.commit();
        conn.close();
    }

    @Test
    public void accessDatasource() throws Exception {
        // define the datasource
        Properties properties = new Properties();
        properties.setProperty("ProtectedDatasource", "new://Resource?type=DataSource
");
        properties.setProperty("ProtectedDatasource.JdbcDriver",
"org.hsqldb.jdbcDriver");
        properties.setProperty("ProtectedDatasource.JdbcUrl", DATASOURCE_URL);
        properties.setProperty("ProtectedDatasource.UserName", USER);
        properties.setProperty("ProtectedDatasource.Password",
"fEroTNXjaL5SOTyRQ92x3DNVS/ksbtgs");
        properties.setProperty("ProtectedDatasource.PasswordCipher", "Static3DES");
        properties.setProperty("ProtectedDatasource.JtaManaged", "true");

        // start the context and makes junit test injections
        EJBContainer container = EJBContainer.createEJBContainer(properties);
        Context context = container.getContext();
        context.bind("inject", this);

        // test the datasource
        assertNotNull(dataSource);
        assertNotNull(dataSource.getConnection());
```

```java
        // closing the context
        container.close();
    }

    @Test
    public void accessDatasourceWithMyImplementation() throws Exception {
        // define the datasource
        Properties properties = new Properties();
        properties.setProperty("ProtectedDatasource", "new://Resource?type=DataSource
");
        properties.setProperty("ProtectedDatasource.JdbcDriver",
"org.hsqldb.jdbcDriver");
        properties.setProperty("ProtectedDatasource.JdbcUrl",
"jdbc:hsqldb:mem:protected");
        properties.setProperty("ProtectedDatasource.UserName", USER);
        properties.setProperty("ProtectedDatasource.Password", "3MdniFr3v3NLLuoY");
        properties.setProperty("ProtectedDatasource.PasswordCipher", "reverse");
        properties.setProperty("ProtectedDatasource.JtaManaged", "true");

        // start the context and makes junit test injections
        EJBContainer container = EJBContainer.createEJBContainer(properties);
        Context context = container.getContext();
        context.bind("inject", this);

        // test the datasource
        assertNotNull(dataSource);
        assertNotNull(dataSource.getConnection());

        // closing the context
        container.close();
    }

    public static class ReverseEncryption implements PasswordCipher {
        @Override
        public char[] encrypt(String plainPassword) {
            return StringUtils.reverse(plainPassword).toCharArray();
        }

        @Override
        public String decrypt(char[] encryptedPassword) {
            return new String(encrypt(new String(encryptedPassword)));
        }
    }
}
```

Running

```
-----------------------------------------------------
```

```
  T E S T S
-------------------------------------------------------
Running org.superbiz.DataSourceCipheredExampleTest
Apache OpenEJB 4.0.0-beta-1    build: 20111002-04:06
http://tomee.apache.org/
INFO - openejb.home = /Users/dblevins/examples/datasource-ciphered-password
INFO - openejb.base = /Users/dblevins/examples/datasource-ciphered-password
INFO - Using 'javax.ejb.embeddable.EJBContainer=true'
INFO - Configuring Service(id=Default Security Service, type=SecurityService,
provider-id=Default Security Service)
INFO - Configuring Service(id=Default Transaction Manager, type=TransactionManager,
provider-id=Default Transaction Manager)
INFO - Configuring Service(id=ProtectedDatasource, type=Resource, provider-id=Default
JDBC Database)
INFO - Found EjbModule in classpath: /Users/dblevins/examples/datasource-ciphered-
password/target/test-classes
INFO - Beginning load: /Users/dblevins/examples/datasource-ciphered-
password/target/test-classes
INFO - Configuring enterprise application: /Users/dblevins/examples/datasource-
ciphered-password
WARN - Method 'lookup' is not available for 'javax.annotation.Resource'. Probably
using an older Runtime.
INFO - Configuring Service(id=Default Managed Container, type=Container, provider-
id=Default Managed Container)
INFO - Auto-creating a container for bean datasource-ciphered-password.Comp:
Container(type=MANAGED, id=Default Managed Container)
INFO - Auto-linking resource-ref
'java:comp/env/org.superbiz.DataSourceCipheredExampleTest/dataSource' in bean
datasource-ciphered-password.Comp to Resource(id=ProtectedDatasource)
INFO - Auto-linking resource-ref
'java:comp/env/org.superbiz.DataSourceCipheredExampleTest/dataSource' in bean
org.superbiz.DataSourceCipheredExampleTest to Resource(id=ProtectedDatasource)
INFO - Enterprise application "/Users/dblevins/examples/datasource-ciphered-password"
loaded.
INFO - Assembling app: /Users/dblevins/examples/datasource-ciphered-password
INFO - Jndi(name="java:global/datasource-ciphered-password/datasource-ciphered-
password.Comp!org.apache.openejb.BeanContext$Comp")
INFO - Jndi(name="java:global/datasource-ciphered-password/datasource-ciphered-
password.Comp")
INFO -
Jndi(name="java:global/EjbModule86823325/org.superbiz.DataSourceCipheredExampleTest!or
g.superbiz.DataSourceCipheredExampleTest")
INFO -
Jndi(name="java:global/EjbModule86823325/org.superbiz.DataSourceCipheredExampleTest")
INFO - Created Ejb(deployment-id=datasource-ciphered-password.Comp, ejb-
name=datasource-ciphered-password.Comp, container=Default Managed Container)
INFO - Created Ejb(deployment-id=org.superbiz.DataSourceCipheredExampleTest, ejb-
name=org.superbiz.DataSourceCipheredExampleTest, container=Default Managed Container)
INFO - Started Ejb(deployment-id=datasource-ciphered-password.Comp, ejb-
name=datasource-ciphered-password.Comp, container=Default Managed Container)
INFO - Started Ejb(deployment-id=org.superbiz.DataSourceCipheredExampleTest, ejb-
```

```
name=org.superbiz.DataSourceCipheredExampleTest, container=Default Managed Container)
INFO - Deployed Application(path=/Users/dblevins/examples/datasource-ciphered-
password)
INFO - EJBContainer already initialized.  Call ejbContainer.close() to allow
reinitialization
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.331 sec

Results :

Tests run: 2, Failures: 0, Errors: 0, Skipped: 0
```