Acceso a Datos

UT7 – BASES DE DATOS XML.

ACCESO A BASEX DESDE JAVA

1. Preparación

Para poder acceder a BaseX desde Java es necesario realizar las siguientes operaciones:

1. Instalar la API XQJ de BaseX, desde su página web http://xqj.net/basex/ . En nuestro caso nos hemos bajado la versión 1.4.0, en concreto el archivo basex-xqj-1.4.0.zip, lo hemos descomprimido y hemos añadido los .jar al Classpath de nuestro proyecto en java.

2. Arrancar el Servidor de BaseX:

2. Configurar conexión

- **XQDataSource**. Identifica el origen de datos a partir del cual se va a crear la conexión. Cada implementación definirá las propiedades necesarias para efectuar la conexión. Por ejemplo, este código realiza la conexión con BaseX, hay que poner el nombre del servidor (Localhost), el puerto de la BD (1984), la base de datos (ACADT), el usuario (admin) y su password (admin).

```
XQDataSource xqd = new BaseXXQDataSource();
xqd.setProperty("serverName", "localhost");
xqd.setProperty("port", "1984");
xqd.setProperty("databaseName", "ACADT");
xqd.setProperty("user", "admin");
xqd.setProperty("password", "admin");
```

2. Configurar conexión

- **XQConnection**. Representa una sesión con la base de datos, manteniendo información de estado, transacciones, expresiones ejecutadas y resultados. Por ejemplo:

```
XQConnection xqc = xqd.getConnection();
```

También se puede indicar el usuario y paswoord que abre la sesión:

```
XQConnection conn = server.getConnection("admin","admin");
```

La conexión la cerramos escribiendo `xqc.close()`

3. Procesar los resultados de una consulta

- **XQExpression:** Objeto creado a partir de una conexión para la ejecución de una expresión. Devuelve un *XQResultsetSequence* con los datos obtenidos. Igual que en *XQExpression*, la ejecución se produce llamando al método *executeQuery*.
- **XQResultSecuence**: Resultado de la ejecución de una sentencia. Es un objeto recorrible. Este ejemplo obtiene los empleados del departamento 10, utilizamos getItemAsString para que devuelva los elementos como cadenas:

```
XQConnection xqc = xqd.getConnection();

XQExpression xqe = xqc.createExpression();
XQResultSequence xqr = xqe.executeQuery
                              ("/EMPLEADOS/EMPLEADO[DEPT_NO="+dep+"]");

while (xqr.next())
{
System.out.println(xqr.getItemAsString(null));
}
```

3. Procesar los resultados de una consulta

Los paquetes que tenemos que importar son:

```
import javax.xml.xquery.XQConnection;
import javax.xml.xquery.XQDataSource;
import javax.xml.xquery.XQException;
import javax.xml.xquery.XQExpression;
import javax.xml.xquery.XQResultSequence;
import net.xqj.basex.BaseXXQDataSource;
```

3. Procesar los resultados de una consulta

El código completo del programa sería el siguiente:

```
public static void GetEmpleadosByDep(int dep)
{
 try {
    XQDataSource xqd = new BaseXXQDataSource();
    xqd.setProperty("serverName", "localhost");
    xqd.setProperty("port", "1984");
    xqd.setProperty("databaseName", "ACADT");
    xqd.setProperty("user", "admin");
    xqd.setProperty("password", "admin");

    XQConnection xqc = xqd.getConnection();
    XQExpression xqe = xqc.createExpression();
    XQResultSequence xqr = xqe.executeQuery
                                ("/EMPLEADOS/EMPLEADO[DEPT_NO="+dep+"]");
          while (xqr.next())
          {
            System.out.println(xqr.getItemAsString(null));
          }

          xqc.close();

          } catch (XQException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
          }
  }
```

```
public static void GetEmpleadosByDep(int dep)
{
try {
XQDataSource xqd = new BaseXXQDataSource();
xqd.setProperty("serverName", "localhost");
xqd.setProperty("port", "1984");
xqd.setProperty("databaseName", "ACADT");
xqd.setProperty("user", "admin");
xqd.setProperty("password", "admin");

XQConnection xqc = xqd.getConnection();
XQExpression xqe = xqc.createExpression();
XQResultSequence xqr = xqe.executeQuery
("/EMPLEADOS/EMPLEADO[DEPT_NO="+dep+"]");
while (xqr.next())
{
System.out.println(xqr.getItemAsString(null));
}

xqc.close();

} catch (XQException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
}
```

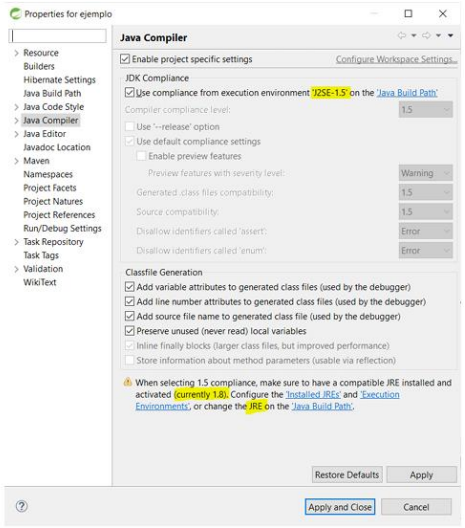
3. Procesar los resultados de una consulta

Si a la hora de ejecutarlo nos da el siguiente error:

```
Exception in thread "main" java.lang.ExceptionInInitializerError
at net.xqj.basex.bin.w.createExpression(Unknown Source)
at com.iesvjp.basex.App.GetEmpleadosByDep(App.java:30)
at com.iesvjp.basex.App.main(App.java:16)
Caused by: java.lang.RuntimeException: Fatal Error: XQJ implementation can not load
vital XML factories it needs to function. Can not instantiate the SAXParserFactory
'com.sun.org.apache.xerces.internal.jaxp.SAXParserFactoryImpl'. User does not have
correct access rights.
at net.xqj.basex.bin.u.<clinit>(Unknown Source)
... 3 more
```

3. Procesar los resultados de una consulta

Se debe a un error de permisos de acceso a las librerías de las últimas versiones del JDK, para que funcione tenemos que configurar una versión anterior del JDK, con la versión 1.5, funcionaría perfectamente:



(Otra solución sería crear un sandbox para poder utilizar los .jars del JDK desde código de terceros sin verificar. Como no es una tarea sencilla, vamos a optar por utilizar una versión anterior del JDK).

3. Procesar los resultados de una consulta

Una vez arreglado el problema, el resultado por consola sería:

```
<EMPLEADO>
  <EMP_NO>7782</EMP_NO>
  <APELLIDO>CEREZO</APELLIDO>
  <OFICIO>DIRECTOR</OFICIO>
  <DIR>7839</DIR>
  <FECHA_ALT>1991-06-09</FECHA_ALT>
  <SALARIO>2885</SALARIO>
  <DEPT_NO>10</DEPT_NO>
</EMPLEADO>
```

```
<EMPLEADO>
  <EMP_NO>7839</EMP_NO>
  <APELLIDO>REY</APELLIDO>
  <OFICIO>PRESIDENTE</OFICIO>
  <FECHA_ALT>1991-11-17</FECHA_ALT>
  <SALARIO>4100</SALARIO>
  <DEPT_NO>10</DEPT_NO>
</EMPLEADO>
<EMPLEADO>
  <EMP_NO>7934</EMP_NO>
  <APELLIDO>MUÑOZ</APELLIDO>
  <OFICIO>EMPLEADO</OFICIO>
  <DIR>7782</DIR>
  <FECHA_ALT>1992-01-23</FECHA_ALT>
  <SALARIO>1690</SALARIO>
  <DEPT_NO>10</DEPT_NO>
</EMPLEADO>
```

4. Serialización de los resultados de un consulta

El resultado de la consulta XQuery podemos guardarla en un fichero xml. Para ello utilizaremos el método *writeSequence* de la clase *XQResultSequence*. Que recibe dos parámetros:

- *OutputStream*: fichero donde guardaremos el resultado
- *Properties*: permite pasarle un valor nulo y en la serialización la API XQJ utilizará los valores por defecto en todos lo valores de serialización, obteniendo un documento sin codificar y sin identar.

```
XQExpression xqe = xqc.createExpression();
XQResultSequence xqr = xqe.executeQuery("for $book in /bib/book "
                  + "return $book/title");
     xqr.writeSequence(new FileOutputStream("resultado.xml"), new
     Properties());
```

4. Serialización de los resultados de un consulta

El resutado sería un archivo con el siguiente contenido:

```
<bib>
<book id="1">
<title>TCP/IP Illustrated</title>
<author>Stevens</author>
<publisher>Addison-Wesley</publisher>
<year>2002</year>
</book>
<bib>
```

No es un archivo XML válido, por lo tanto, necesitamos especificar las propiedades del XML (método, codificación, identación…)

```
Properties serializationProps = new java.util.Properties();
serializationProps.setProperty("method", "xml");
serializationProps.setProperty("indent", "yes");
serializationProps.setProperty("encoding", "UTF-8");
serializationProps.setProperty("omit-xml-declaration", "no");
```

4. Serialización de los resultados de un consulta

Le pasamos al *writeSequence* estas propiedades y el resultado sería:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<bib>
  <book id="1">
    <title>TCP/IP Illustrated</title>
    <author>Stevens</author>
    <publisher>Addison-Wesley</publisher>
    <year>2002</year>
  </book>
</bib>
```

4. Serialización de los resultados de un consulta

El código completo del programa sería:

```
public static void GetEmpleadosByDepToXML(int dep)
{
  //definimos las siguientes propiedades para el XML de salida
  Properties serializationProps = new java.util.Properties();
  serializationProps.setProperty("method", "xml");
  serializationProps.setProperty("indent", "yes");
  serializationProps.setProperty("encoding", "UTF-8");
  serializationProps.setProperty("omit-xml-declaration", "no");

  try {
      XQDataSource xqd = new BaseXXQDataSource();
     xqd.setProperty("serverName", "localhost");
     xqd.setProperty("port", "1984");
     xqd.setProperty("databaseName", "ACADT");
     xqd.setProperty("user", "admin");
     xqd.setProperty("password", "admin");
     XQConnection xqc = xqd.getConnection();
     XQExpression xqe = xqc.createExpression();
     XQResultSequence xqr = xqe.executeQuery("<EMPLEADOS>{/EMPLEADOS/
                                EMPLEADO[DEPT_NO="+dep+"]}</EMPLEADOS>");

     xqr.writeSequence(new FileOutputStream("resultado.xml"),
                                                 serializationProps);
     xqr.close();
     xqc.close();

} catch (XQException | FileNotFoundException e) {
   // TODO Auto-generated catch block
      e.printStackTrace();
}
}
```

```
public static void GetEmpleadosByDepToXML(int dep)
{
//definimos las siguientes propiedades para el XML de salida
Properties serializationProps = new java.util.Properties();
serializationProps.setProperty("method", "xml");
serializationProps.setProperty("indent", "yes");
serializationProps.setProperty("encoding", "UTF-8");
serializationProps.setProperty("omit-xml-declaration", "no");

try {
XQDataSource xqd = new BaseXXQDataSource();
xqd.setProperty("serverName", "localhost");
xqd.setProperty("port", "1984");
xqd.setProperty("databaseName", "ACADT");
xqd.setProperty("user", "admin");
xqd.setProperty("password", "admin");
XQConnection xqc = xqd.getConnection();
XQExpression xqe = xqc.createExpression();
XQResultSequence xqr = xqe.executeQuery("<EMPLEADOS>{/EMPLEADOS/
EMPLEADO[DEPT_NO="+dep+"]}</EMPLEADOS>);


xqr.writeSequence(new FileOutputStream("resultado.xml"),
serializationProps);
xqr.close();
xqc.close();

} catch (XQException | FileNotFoundException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
}
```

5. Actualizaciones en la BD

Si queremos realizar modificaciones en la Base de Datos y tenemos abierto la interfaz de usuario nos dará el siguiente error, será necesario cerrar la GUI:

```
javax.xml.xquery.XQException:
  bxerr:BXDB0007 - Database 'ACADT' is currently opened by another process.
  at AccesoBaseX.ConsultaBase(AccesoBaseX.java:97)
  at AccesoBaseX.main(AccesoBaseX.java:18)
```

El código de modificación del BD no varía con respecto al de las consultas. El siguiente ejemplo modifica la población de todas las personas a Plasencia:

5. Actualizaciones en la BD

```
public static void ModificarPoblacion()
  {
  try {
       XQDataSource xqd = new BaseXXQDataSource();
       xqd.setProperty("serverName", "localhost");
       xqd.setProperty("port", "1984");
       xqd.setProperty("databaseName", "ACADT");
       xqd.setProperty("user", "admin");
       xqd.setProperty("password", "admin");

       XQConnection xqc = xqd.getConnection();

       XQExpression xqe = xqc.createExpression();
       xqe.executeQuery("for $poblacion in /personas/persona/poblacion "+
                                                              "return "+
                                                              "replace
                          value of node $poblacion with 'Plasencia'");

       xqc.close();
  } catch (XQException e) {
            System.out.println("Error al actualizar la BD XML: "+
                                                    e.getMessage());
}}
```

```
public static void ModificarPoblacion()
{
try {
XQDataSource xqd = new BaseXXQDataSource();
xqd.setProperty("serverName", "localhost");
xqd.setProperty("port", "1984");
xqd.setProperty("databaseName", "ACADT");
xqd.setProperty("user", "admin");
xqd.setProperty("password", "admin");
XQConnection xqc = xqd.getConnection();
XQExpression xqe = xqc.createExpression();
xqe.executeQuery("for $poblacion in /personas/persona/poblacion "+
"return "+
"replace
value of node $poblacion with 'Plasencia'");
xqc.close();
} catch (XQException e) {
System.out.println("Error al actualizar la BD XML: "+
e.getMessage());
}}
```

Dudas y preguntas