

Manejo de conectores Práctica 4. JDBC - Conectar con Sqlite

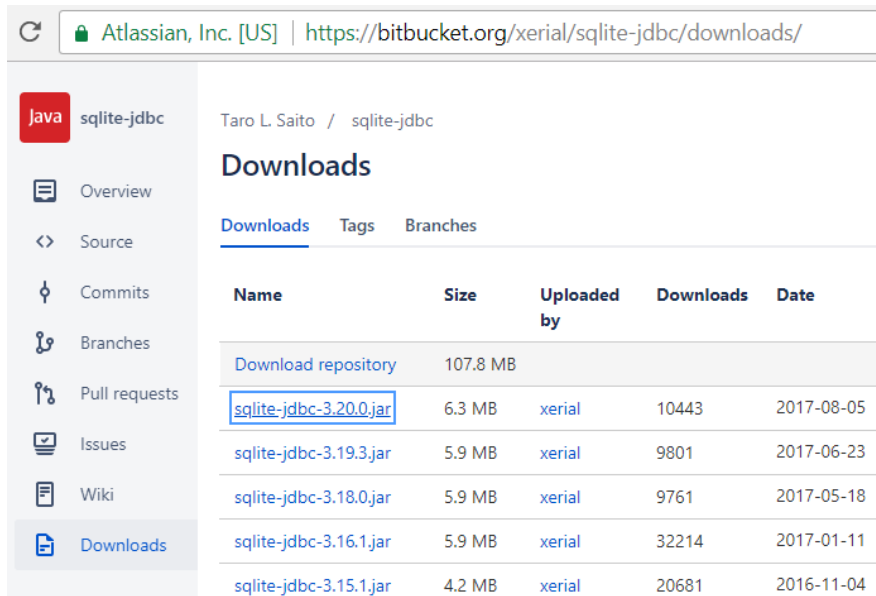
Ahora lo que vamos a hacer es conectar con la base de datos sqlite que creamos en una práctica anterior pero ahora conectaremos desde un programa Java con entorno Netbeans.

Preparación del entorno.

Para empezar tenemos que obtener el JAR que contiene el driver para sqlite.

Se recomienda obtenerlo de la distribución de la base de datos que haya instalada para evitar problemas de compatibilidad. Suele estar una de las carpetas del archivo descargado.

Si no lo tienes puedes conseguirlo aquí: <https://bitbucket.org/xerial/sqlite-jdbc/downloads/>



Name	Size	Uploaded by	Downloads	Date
Download repository	107.8 MB			
sqlite-jdbc-3.20.0.jar	6.3 MB	xerial	10443	2017-08-05
sqlite-jdbc-3.19.3.jar	5.9 MB	xerial	9801	2017-06-23
sqlite-jdbc-3.18.0.jar	5.9 MB	xerial	9761	2017-05-18
sqlite-jdbc-3.16.1.jar	5.9 MB	xerial	32214	2017-01-11
sqlite-jdbc-3.15.1.jar	4.2 MB	xerial	20681	2016-11-04

Nos descargamos el jar correspondiente sqlite-jdbc-X.X.X.jar.

En todos los proyectos donde queramos acceder a una base de datos sqlite tendremos que añadir el driver a las librerías del proyecto. Si quieres ayuda de cómo hacerlo puedes visitar: <http://www.sqlitetutorial.net/sqlite-java/sqlite-jdbc-driver/>

Conectar desde programa

Ahora podemos escribir nuestro primer programa, llámalo U2_P4_ConectarSqlite (nota. se ha incluido únicamente las instrucciones para establecer la conexión):

```
import java.sql.*;
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        try {
```

```
            // Empezamos por cargar el driver
```

```
            Class.forName("org.sqlite.JDBC");
```

```
            // Ahora podemos establecer la conexión mediante el DriverManager
```

```
            Connection micon =
```

```
            DriverManager.getConnection("jdbc:sqlite:C:/clase/bases/sqlite/ejemplo.db");
```

```
            // getConnection(url [, nombre de usuario, contraseña])
```

```
        // una vez la conexión establecida podemos ejecutar las instrucciones que
        // queramos contra la base de datos a través de la conexión.

        // Cuando ya no necesitamos acceder a la base de datos cerramos la conexión:
        micon.close();
    }
    catch (ClassNotFoundException cn) {cn.printStackTrace();}
    catch(SQLException e) { e.printStackTrace();}
} // fin de main
} // fin de la clase
```

Este programa no tiene mucho sentido, conectar para no hacer nada, pero nos hemos querido centrar en el punto que hay que practicar que es preparar el entorno y establecer la conexión.

En la siguiente parte de la unidad retomaremos este programa para añadirle el manejo de los datos una vez establecida la conexión.