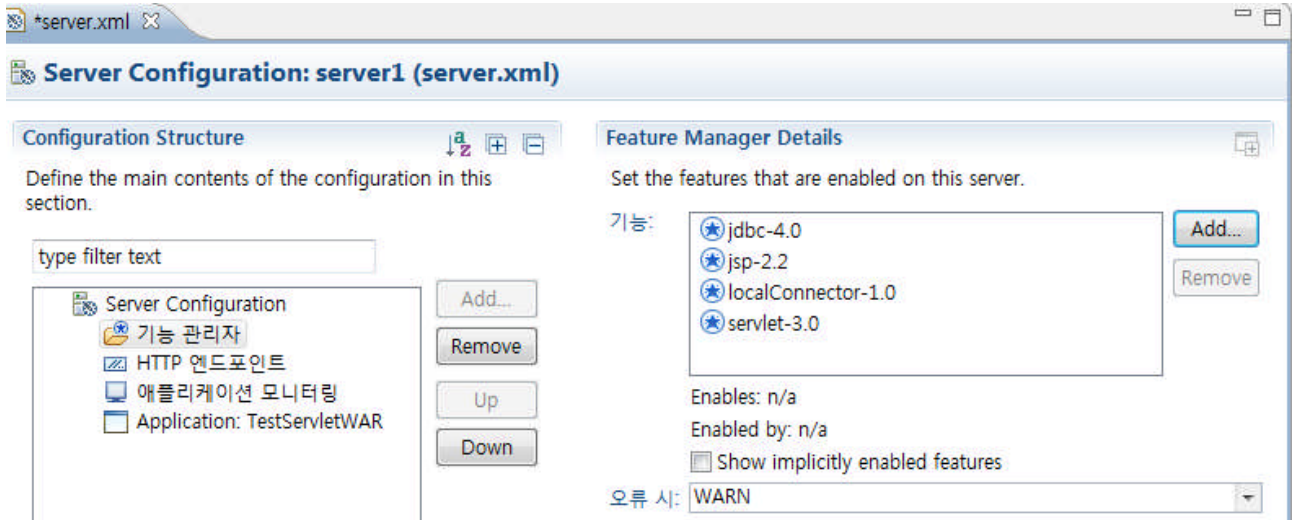


WebSphere Application Server V8.5
Liberty Profile – JDBC Test

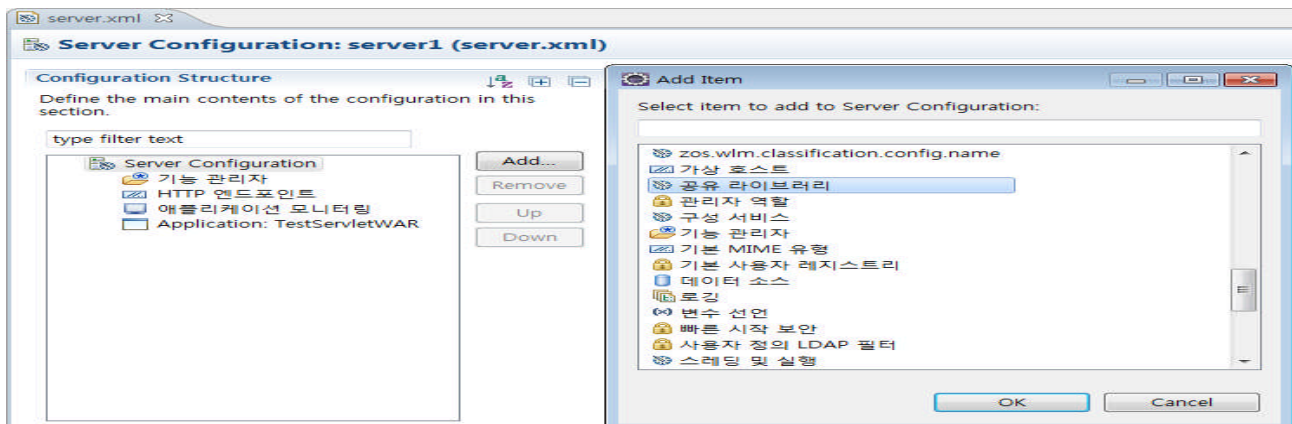


1) IBM WAS V8.5 Liberty Profile - JDBC 설정

1. 이번 강좌에서는 IBM WAS V8.5 Liberty Profile을 이용해서 JDBC를 테스트 해보도록 하겠습니다. 해당 테스트를 위해서 먼저 server.xml > 기능 관리자에서 Add 를 클릭하여 jdbc-4.0 기능을 추가합니다.



2. JDBC 기능을 추가했으면 다음으로는 JDBC Driver에서 사용할 jar파일을 공유 라이브러리로 만들기 위하여 Server Configuration에서 Add 버튼을 클릭하여 공유 라이브러리를 추가합니다.

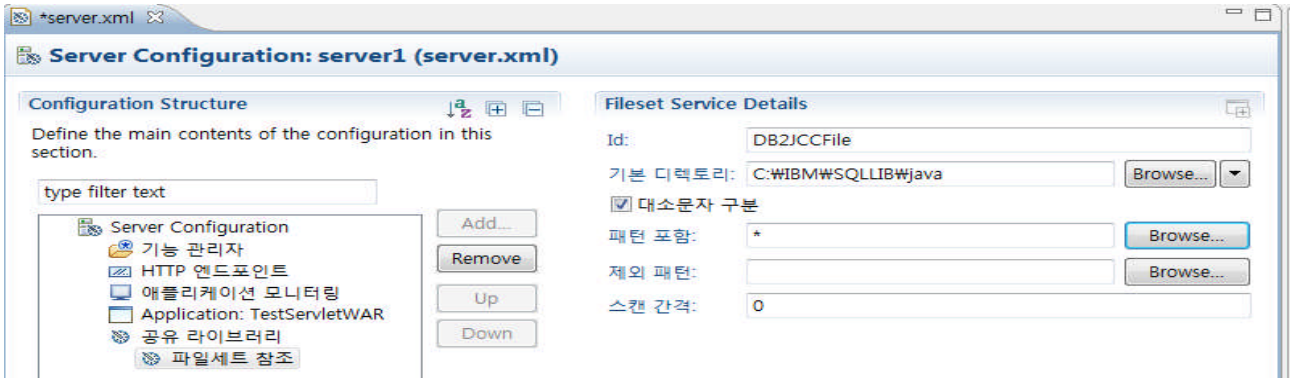


3. Id와 이름을 넣고 실제 파일 위치를 넣기 위해 파일 세트 참조 메뉴에서 Add를 클릭합니다.

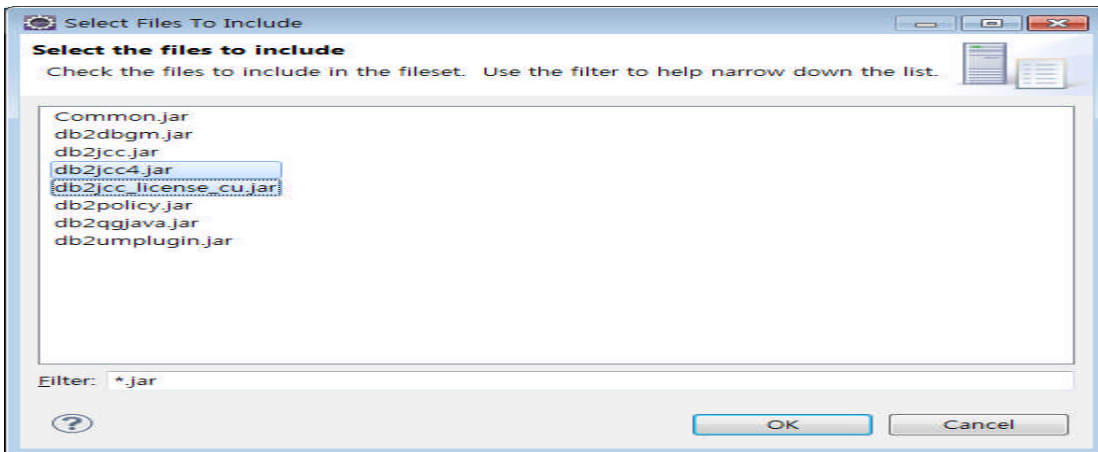


4. 파일세트 참조가 새로 생성된 것을 확인할 수 있으며 실제 jar파일이 위치한 디렉토리를 기본 디렉토리로 지정하고 jar파일을 연결하기 위하여 패턴 포함메뉴에 있는 Browse 를 클릭합니다.

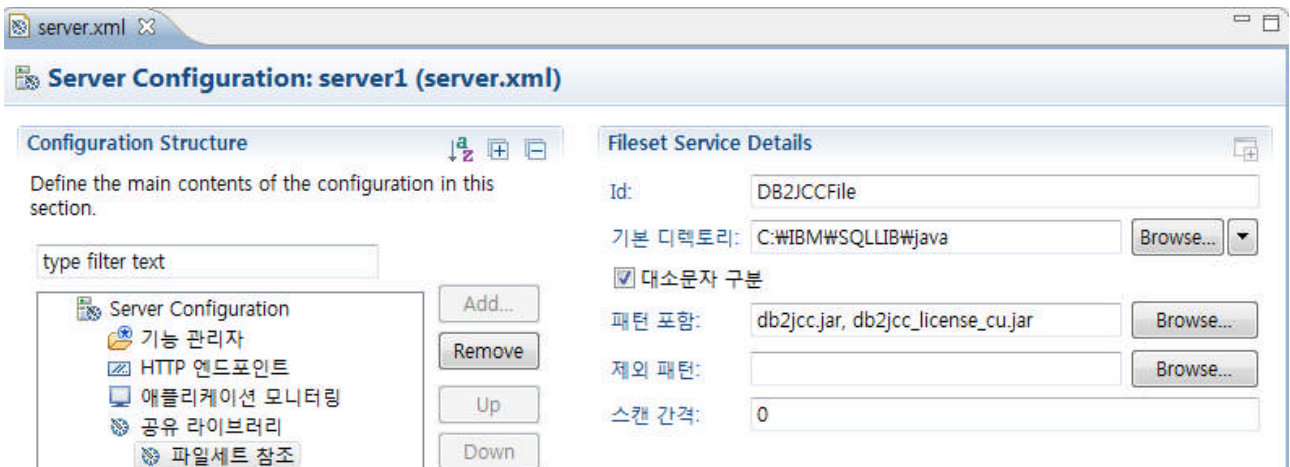
(참고로 해당 강좌는 DB2를 기반으로 작성되었습니다.)



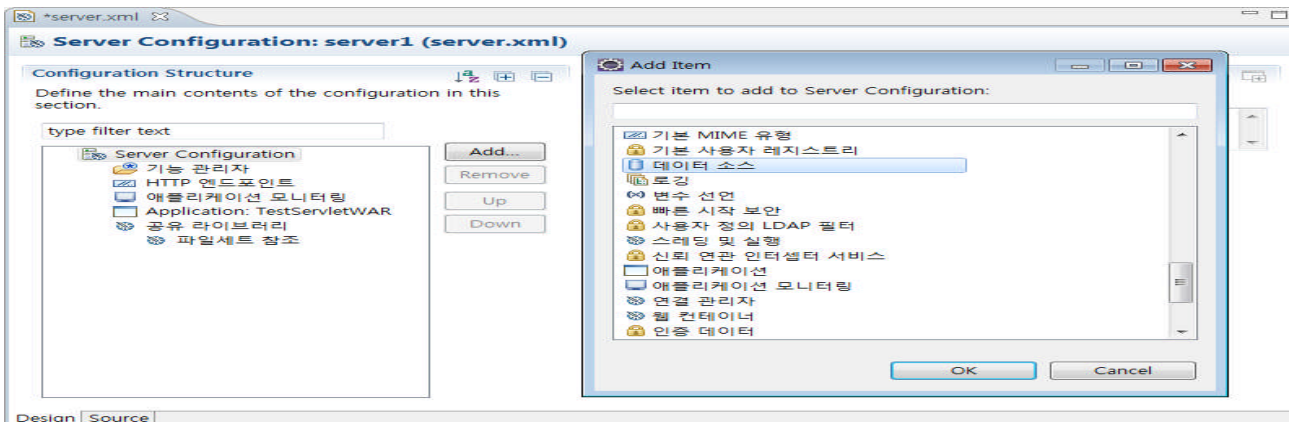
5. 팝업 창이 나오면 DB2의 JDBC Driver와 License파일을 선택한 후 OK를 클릭합니다.



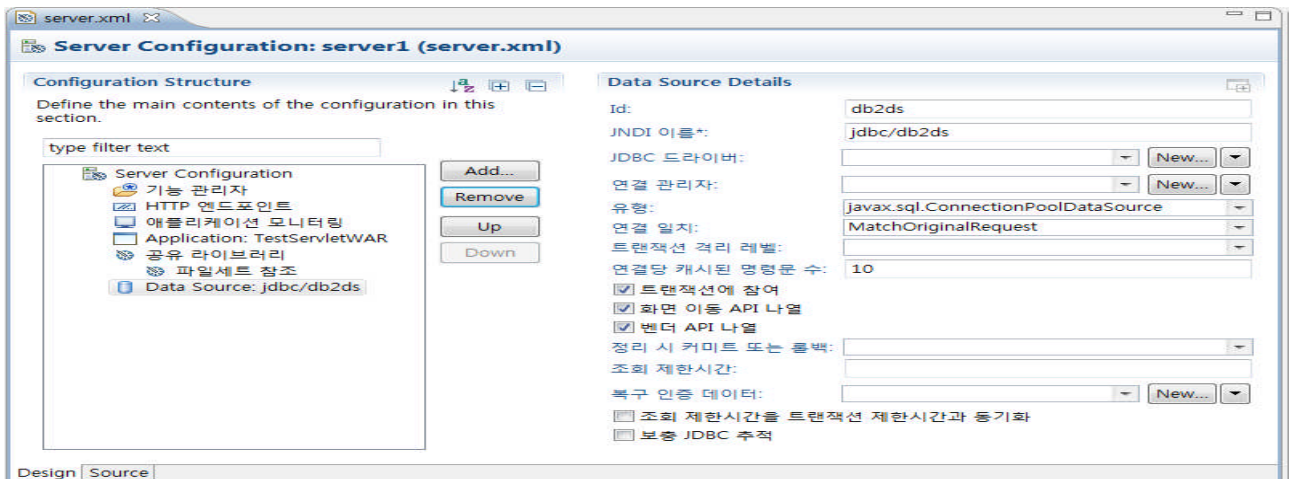
6. 하단과 같이 설정이 되었으면 기본적인 공유 라이브러리 설정은 완료되었습니다.



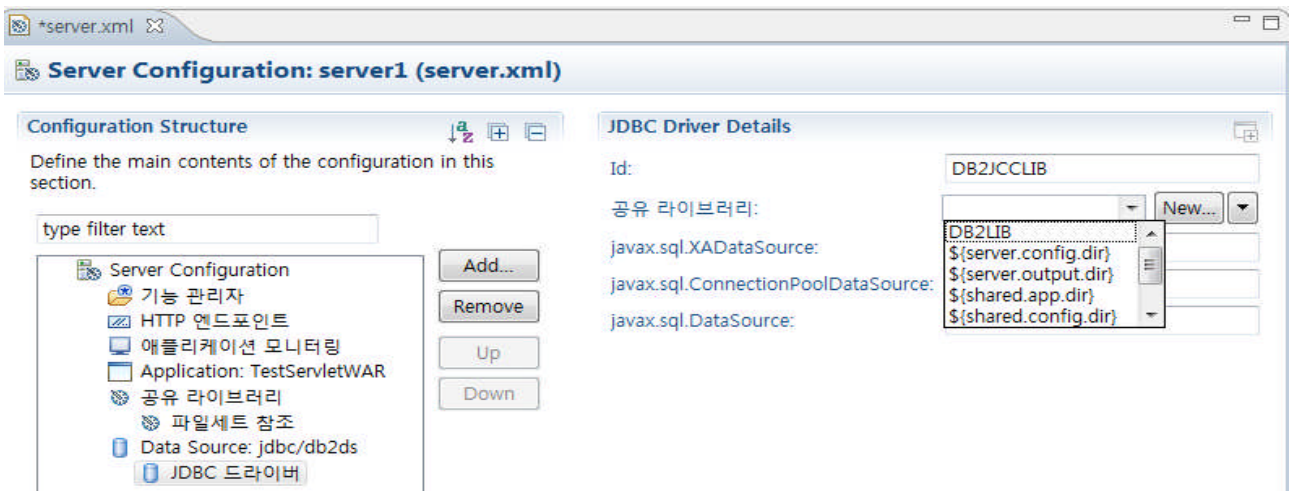
7. 이제 실제 필요한 데이터 소스를 설정하기 위하여 Server Configuration에서 Add 버튼을 클릭하여 데이터 소스를 추가합니다.



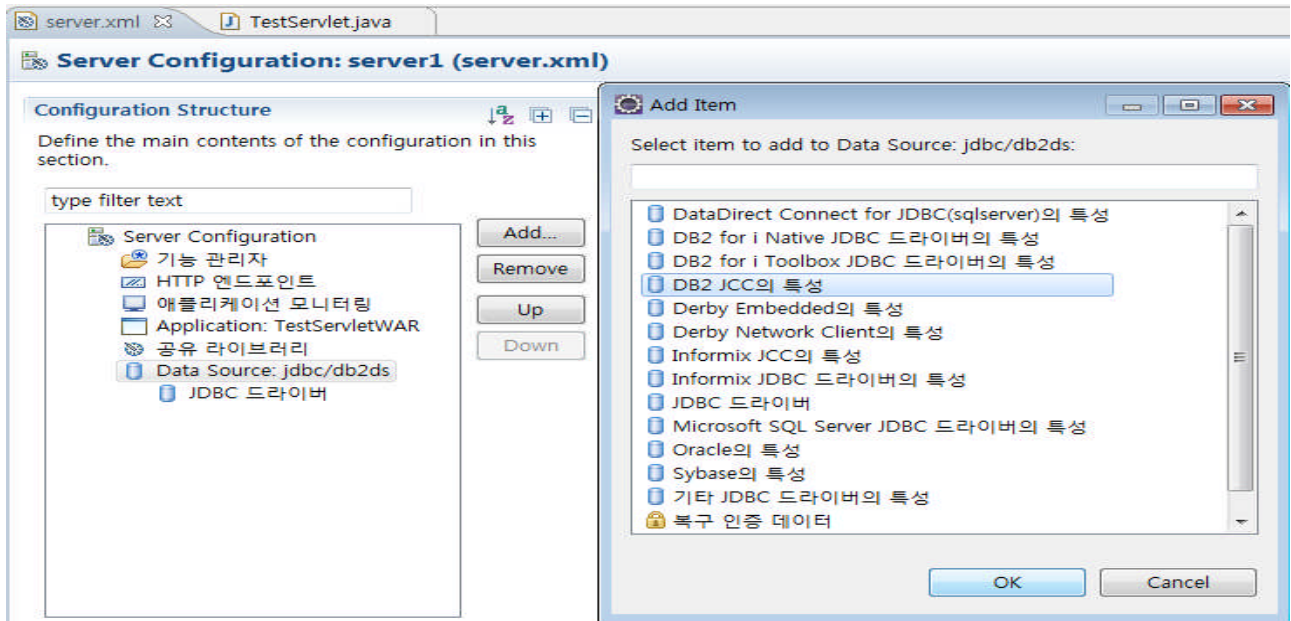
8. 데이터 소스에서 가장 중요한 JNDI 이름을 넣어주고 유형을 결정합니다. 이후 이전에 공유 라이브러리로 만들어둔 것과 JDBC 드라이버를 연결하기 위하여 JDBC 드라이버에서 New 버튼을 클릭합니다.



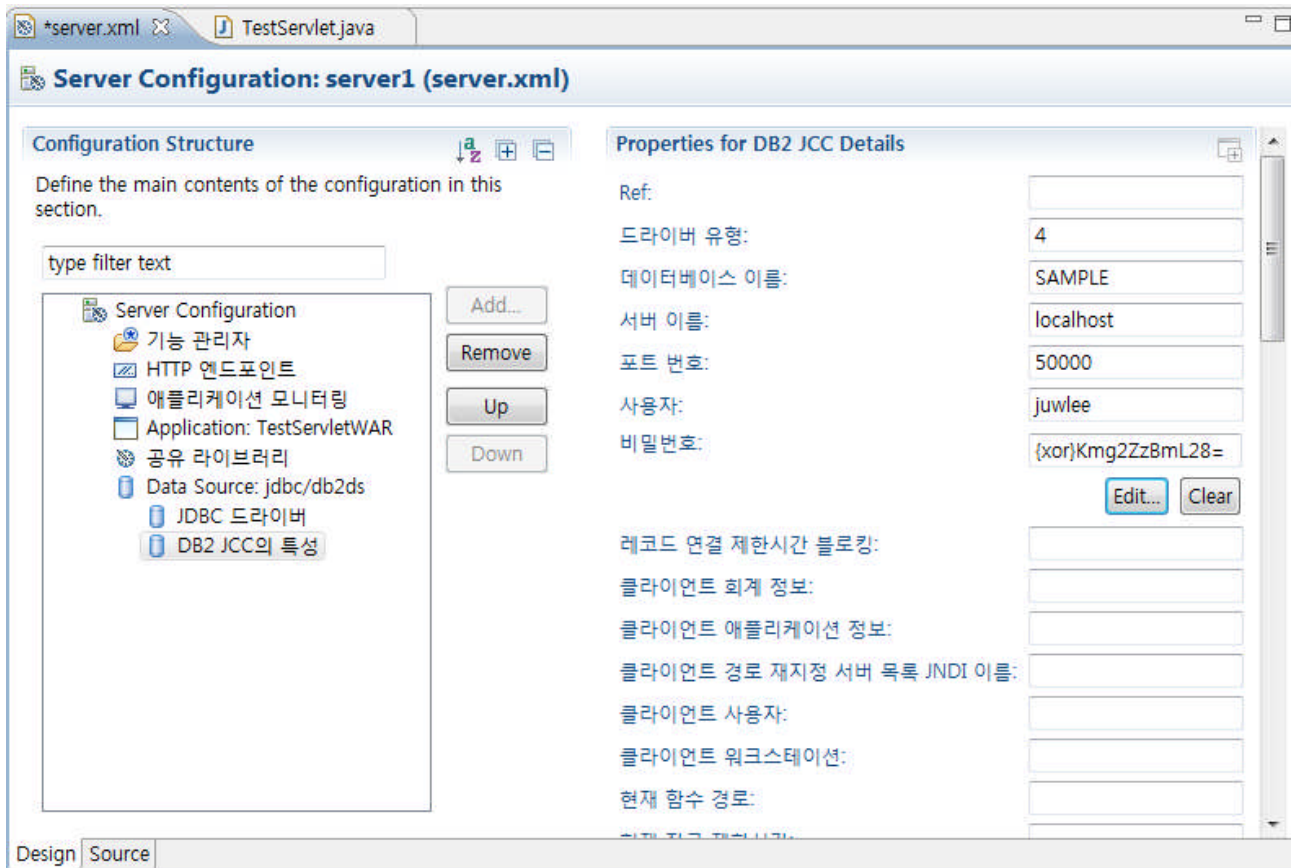
9. 새롭게 생성되는 JDBC 드라이버를 확인하고 공유 라이브러리 부분에서 이전에 만들어둔 DB2LIB 라이브러리를 선택하여 줍니다.



10. 이제 마지막으로 사용자/비밀번호와 같은 데이터 소스의 세부 속성을 넣기 위하여 만들어진 데이터 소스에서 Add 버튼을 클릭하여 DB2 JCC 의 특성을 선택합니다.



11. 기본 정보를 확인하고 데이터베이스 이름, 사용자, 비밀번호를 입력합니다.



2) JDBC 테스트

1. 이전까지 IBM WAS V8.5 Liberty Profile 을 위한 JDBC 설정을 모두 완료하였습니다. 이제 실제적으로 JDBC 를 테스트 해보기 위하여 실제 샘플로 쓸 수 있는 DB2 의 테이블과 값을 확인합니다.

(여기서는 SAMPLE DB의 EMPLOYEE 테이블의 FIRSTNAME만 조회하는 것을 해보겠습니다.)

The screenshot shows the DB2 console interface. On the left is a tree view of the database structure. The main area displays the 'EMPLOYEE' table details, including its schema (JUWLEE), creator (JUWLEE), and column definitions. A table below shows the first few rows of data:

EMPNO	FIRSTNAME
000010	CHRISTINE
000020	MICHAEL
000030	SALLY
000050	JOHN
000060	IRVING
000070	EVA
000090	EILEEN
000100	THEODORE
000110	VINCENZO
000120	SEAN

2. DB2 테이블과 값을 확인했으면 실제 이를 조회하기 위해 TestServlet 소스를 수정합니다. dbCall()이라는 메소드를 추가적으로 만들고 @Resource 어노테이션을 활용하여 'jdbc/db2ds' 자원을 주입하여 간단하게 executeQuery 메소드로 조회하는 내용입니다.

```

        out.println("<html><head><title>Hello World</title></head>");
        out.println("<body><h1>Hello World!!</h1><br>");
        out.println("<h1>DB Call</h1>" + dbCall() + "</body>");
        out.println("</html>");
    }

    @Resource(name="jdbc/db2ds")
    DataSource ds1;
    protected String dbCall() {
        String firstName = "test";
        Statement stmt = null;
        Connection con = null;

        try {
            con = ds1.getConnection();
            stmt = con.createStatement();
            ResultSet result = stmt.executeQuery("select FIRSTNAME from Employee where empno='000010'");
            result.next();
            firstName = result.getString(1);
        } catch (SQLException e) {
            e.printStackTrace();
        }

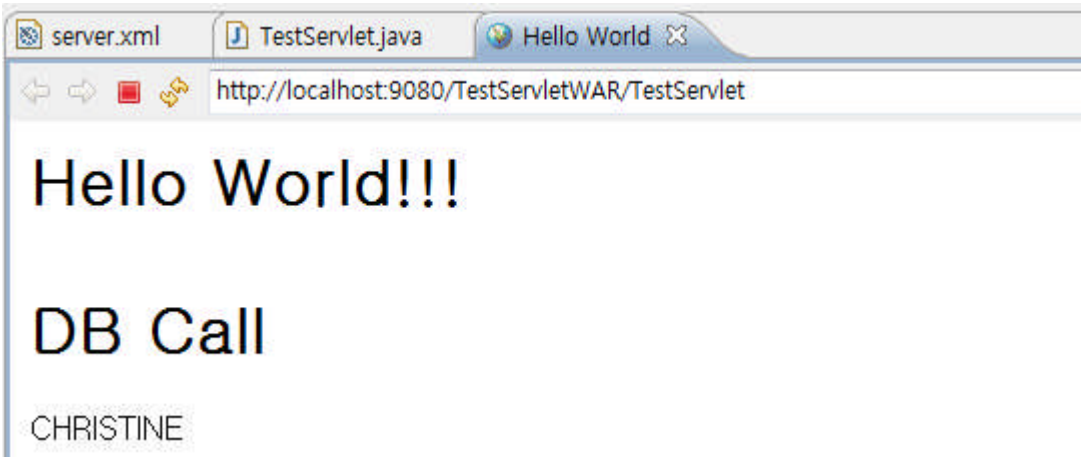
        return firstName;
    }
}

```

3. 애플리케이션 수정이 끝났으면 해당 서버를 시작합니다. 이전과 다르게 추가적으로 이번 강좌에서 설정한 데이터 소스와 JDBC Driver 관련한 내용이 추가적으로 나오는 것을 확인할 수 있습니다.



4. TestServlet에 테스트를 수행하여 봅니다. 하단과 같이 DB2 의 SAMPLE DB에 저장되어 있는 값 ('CHRISTINE')이 정상적으로 나오면 테스트를 무사히 완료하신 것 입니다.



Appendix #1 – 다른 DB 연결 샘플

For DB2®

```
<dataSource id="db2" jndiName="jdbc/db2">
  <jdbcDriver libraryRef="DB2JCC4Lib"/>
  <properties.db2.jcc databaseName="SAMPLEDB" serverName="localhost" portNumber="50000"/>
</dataSource>

<library id="DB2JCC4Lib">
  <fileset dir="C:/DB2/java" includes="db2jcc4.jar db2jcc_license_cisuz.jar"/>
</library>
```

For DB2 on iSeries® (Native)

```
<dataSource id="db2iNative" jndiName="jdbc/db2iNative">
  <jdbcDriver libraryRef="DB2iNativeLib"/>
  <properties.db2.i.native databaseName="*LOCAL"/>
</dataSource>

<library id="DB2iNativeLib">
  <fileset dir="/QIBM/ProdData/java400/jdk6/lib/ext" includes="db2_classes16.jar"/>
</library>
```

For DB2 on iSeries (Toolbox)

```
<dataSource id="db2iToolbox" jndiName="jdbc/db2iToolbox">
  <jdbcDriver libraryRef="DB2iToolboxLib"/>
  <properties.db2.i.toolbox databaseName="SAMPLEDB" serverName="localhost"/>
</dataSource>

<library id="DB2iToolboxLib">
  <fileset dir="/QIBM/ProdData/Http/Public/jt400/lib" includes="jt400.jar"/>
</library>
```

For Derby Embedded

```
<dataSource id="derbyEmbedded" jndiName="jdbc/derbyEmbedded">
  <jdbcDriver libraryRef="DerbyLib"/>
  <properties.derby.embedded databaseName="C:/databases/SAMPLEDB" createDatabase="create"/>
</dataSource>

<library id="DerbyLib">
  <fileset dir="C:/db-derby-10.8.1.2-bin/lib"/>
</library>
```

For Derby Network Client

```
<dataSource id="derbyClient" jndiName="jdbc/derbyClient">
  <jdbcDriver libraryRef="DerbyLib"/>
  <properties.derby.client databaseName="C:/databases/SAMPLEDB" createDatabase="create"
serverName="localhost" portNumber="1527"/>
</dataSource>

<library id="DerbyLib">
  <fileset dir="C:/db-derby-10.8.1.2-bin/lib"/>
</library>
```

For Informix® JCC

```
<dataSource id="informixjcc" jndiName="jdbc/informixjcc">
  <jdbcDriver libraryRef="DB2JCC4Lib"/>
  <properties.informix.jcc databaseName="SAMPLEDB" serverName="localhost" portNumber="1526"/>
</dataSource>

<library id="DB2JCC4Lib">
  <fileset dir="C:/Drivers/jcc/4.8" includes="db2jcc4.jar db2jcc_license_cisuz.jar"/>
</library>
```


For Informix JDBC

```
<dataSource id="informix" jndiName="jdbc/informix">
  <jdbcDriver libraryRef="InformixLib"/>
  <properties.informix databaseName="SAMPLEDB" ifxIFXHOST="localhost" serverName="ol_machinename"
portNumber="1526"/>
</dataSource>

<library id="InformixLib">
  <fileset dir="C:/Drivers/informix" includes="ifxjdbc.jar ifxjdbcx.jar"/>
</library>
```

For Microsoft SQL Server (Microsoft JDBC driver)

```
<dataSource id="mssqlserver" jndiName="jdbc/mssqlserver">
  <jdbcDriver libraryRef="MSJDBClib"/>
  <properties.microsoft.sqlserver databaseName="SAMPLEDB" serverName="localhost" portNumber="1433"/>
</dataSource>

<library id="MSJDBClib">
  <fileset dir="C:/sqljdbc_4.0/enu" includes="sqljdbc4.jar"/>
</library>
```

For Microsoft SQL Server (DataDirect Connect for JDBC driver)

```
<dataSource id="ddsqlserver" jndiName="jdbc/ddsqlserver">
  <jdbcDriver libraryRef="DataDirectLib"/>
  <properties.datadirect.sqlserver databaseName="SAMPLEDB" serverName="localhost" portNumber="1433"/>
</dataSource>

<library id="DataDirectLib">
  <fileset dir="C:/DataDirect/Connect-4.2/lib" includes="sqlserver.jar"/>
</library>
```

For MySQL

```
<dataSource id="mysql" jndiName="jdbc/mysql">
  <jdbcDriver libraryRef="MySQLLib"/>
  <properties databaseName="SAMPLEDB" serverName="localhost" portNumber="3306"/>
</dataSource>

<library id="MySQLLib">
  <fileset dir="C:/mysql-connector-java-x.x.xx" includes="mysql-connector-java-x.x.xx.jar"/>
</library>
```

For Oracle

```
<dataSource id="oracle" jndiName="jdbc/oracle">
  <jdbcDriver libraryRef="OracleLib"/>
  <properties.oracle URL="jdbc:oracle:thin:@//localhost:1521/SAMPLEDB"/>
</dataSource>

<library id="OracleLib">
  <fileset dir="C:/Oracle/lib" includes="ojdbc6.jar"/>
</library>
```

For Sybase

```
<dataSource id="sybase" jndiName="jdbc/sybase">
  <jdbcDriver libraryRef="SybaseLib"/>
  <properties.sybase databaseName="SAMPLEDB" serverName="localhost" portNumber="5000"/>
</dataSource>

<library id="SybaseLib">
  <fileset dir="C:/Drivers/sybase" includes="jconn4.jar"/>
</library>
```

For solidDB®

```
<dataSource id="solidDB" jndiName="jdbc/solidDB">
  <jdbcDriver libraryRef="solidLib"/>
  <properties databaseName="SAMPLEDB" URL="jdbc:solid://localhost:2315"/>
</dataSource>

<library id="solidLib">
  <fileset dir="C:/Drivers/solidDB" includes="SolidDriver2.0.jar"/>
</library>
```

For a JDBC driver that is not known to the Liberty profile

```
<dataSource id="sample" jndiName="jdbc/sample" type="javax.sql.XADataSource">
  <jdbcDriver libraryRef="SampleJDBCLib" javax.sql.XADataSource="com.ibm.sample.SampleXADataSource"/>
  <properties databaseName="SAMPLEDB" hostName="localhost" port="12345"/>
</dataSource>

<library id="SampleJDBCLib">
  <fileset dir="C:/Drivers/SampleJDBC/" includes="sampleDriver.jar"/>
</library>
```

2) 결론

이번 강좌에서 IBM WAS V8.5 Liberty profile을 통한 JDBC 샘플 테스트를 수행하였습니다.

IBM WAS full profile과 설정 방식은 조금 다르지만 IBM WAS Liberty Profile을 활용하여 JDBC 조회 같은 작업을 문제없이 수행할 수 있습니다. 또한, IBM WAS Liberty Profile의 설정은 eclipse를 활용하여 진행하기 때문에 조금은 직관적인 설정이 가능하며 개발자들이 직접 설정하거나 server.xml 공유를 통하여 쉽게 개발 환경을 만들 수 있는 것을 확인할 수 있었습니다.

그럼 이번 강좌는 마무리하고 다음 강좌를 통해 또 다른 애플리케이션 테스트를 진행해 보도록 하겠습니다. Go Go !!!

9) 참고 자료

1. 이 가이드는 IBM WAS V8.5 최초 사용자를 위한 기본 가이드입니다.
2. IBM WAS 자체에 아직 익숙하지 않으신 분들은 가급적 기본강좌인 '하나씩 쉽게 따라 해보는 IBM WAS v7' 강좌를 먼저 읽고 이 강좌를 읽으시는 것이 훨씬 이해에 됩니다.
(http://www.websphere.pe.kr/xe/?mid=was_info_re&page=3&document_srl=800)
3. 가급적 IBM WAS V8.5 InfoCenter의 해당 카테고리를 한 번 읽어보고 난 후에 작업하시기 바랍니다.
4. InfoCenter – WebSphere Application Server V8.5
(<http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/index.jsp>)
5. InfoCenter – Configuring database connectivity in the Liberty profile
(http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/topic/com.ibm.websphere.wlp.nd.iseries.doc/topics/twlp_dep_configuring_ds.html?resultof=%22%43%6f%6e%66%69%67%75%72%69%6e%67%22%20%22%63%6f%6e%66%69%67%75%72%22%20%22%64%61%74%61%62%61%73%65%22%20%22%64%61%74%61%62%61%73%22%20%22%63%6f%6e%6e%65%63%74%69%76%69%74%79%22%20%22%63%6f%6e%6e%65%63%74%22%20%22%4c%69%62%65%72%74%79%22%20%22%6c%69%62%65%72%74%69%22%20%22%70%72%6f%66%69%6c%65%22%20%22%70%72%6f%66%69%6c%22%20)