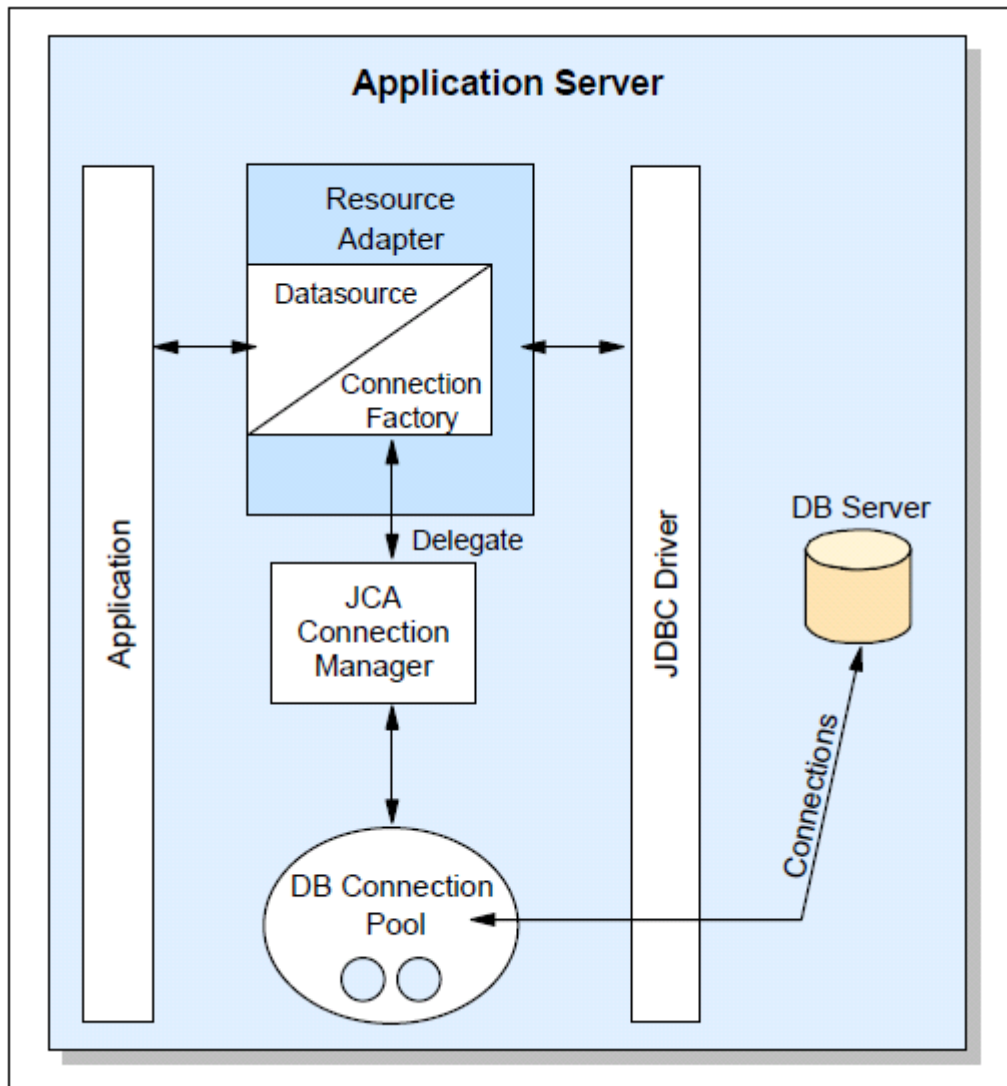


하나씩 쉽게 따라 해보는 IBM WebSphere Application Server(WAS) v7 – 6

이정운 (juwlee@kr.ibm.com)

하나씩 쉽게 따라 해보는 IBM WAS v7 시리즈 그 여섯번째 이야기 DB 호출입니다. 본 강좌에서 말씀드리는 DB 호출이란 이전 강좌에서 진행했던 DB 연결 설정을 이용하여(JDBC 프로바이더와 Data Source) 실제 DB에 접속하는 EJB 어플리케이션을 작성하고 이를 통해 DB 작업을 진행하는 것을 말하는 것입니다. 이전 강좌에서 준비된 연결을 이용해서 DB Data 를 받아오고, 다시 이것을 작업하여 DB 에 저장하는 등의 작업을 합니다. 이전에도 설명드렸지만 대부분의 기업에서 사용되는 어플리케이션은 DB 작업을 많이 하기 때문에 본 강좌가 많은 부분을 설명할 수는 없지만 기본적으로 어플리케이션과 WAS가 엮이고 어떤 식으로 동작하는지 이해하는데 많은 도움이 될 수 있을 것입니다. 그럼 강좌를 진행하도록 하겠습니다.

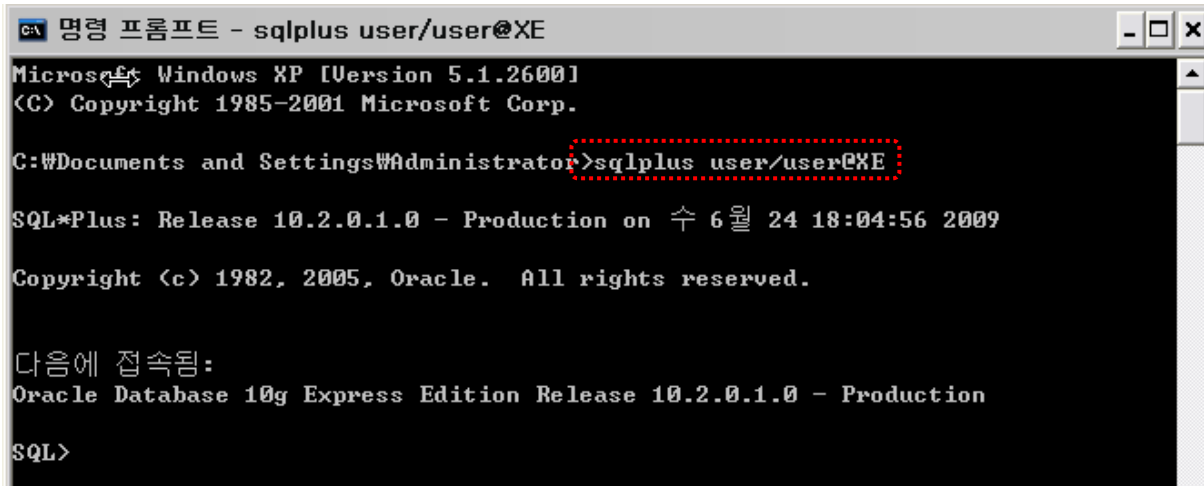


(참조: WebSphere Application Server V6.1: System Management and Configuration Redbook)

Part 1. Oracle DB 설정

DB 호출을 해보기 위해서는 먼저 DB 구성이 만들어져 있어야 합니다. 여기서 말하는 DB 구성이란 data가 들어갈 수 있는 Table 을 미리 만들어 두어야 한다는 것입니다. Table 이 마련되어 있어야 해당 Table 에 data 를 insert 하던가 select 하는 작업등을 진행할 수 있습니다. 그럼 Table 을 만들기 위해서 해당 DB로 접속합니다.

Windows 의 명령 프롬프트를 이용해서 sqlplus 사용자명/암호@서비스명 으로 해당 DB에 접속합니다. (ex: sqlplus user/user@XE)



```
C:\> sqlplus user/user@XE

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>sqlplus user/user@XE

SQL*Plus: Release 10.2.0.1.0 - Production on 수 6월 24 18:04:56 2009

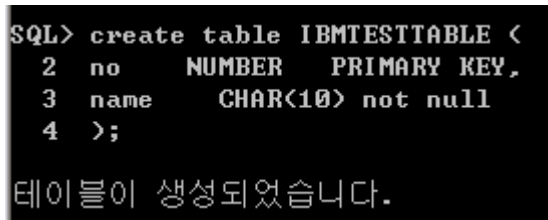
Copyright (c) 1982, 2005, Oracle. All rights reserved.

다음에 접속됨:
Oracle Database 10g Express Edition Release 10.2.0.1.0 - Production

SQL>
```

Data 를 넣어둘 샘플 테이블을 생성합니다.

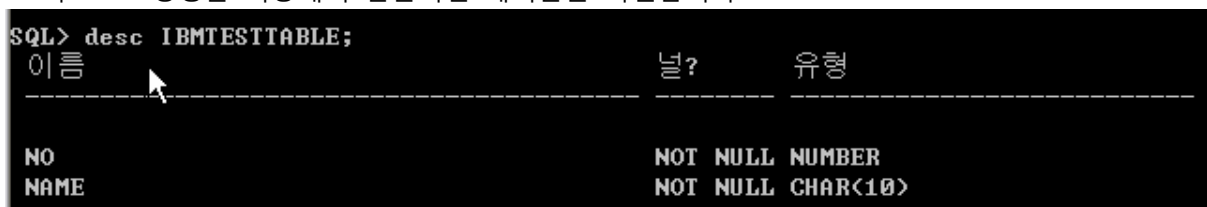
(말 그대로 샘플 테이블이기 때문에 본 형식이 아닌 원하는 형식으로 만드셔도 무방합니다.)



```
SQL> create table IBMTESSTABLE (
  2 no NUMBER PRIMARY KEY,
  3 name CHAR(10) not null
  4 );

테이블이 생성되었습니다.
```

DB의 Desc 명령을 이용해서 만들어진 테이블을 확인합니다.



```
SQL> desc IBMTESSTABLE;

이름          널?       유형
-----
NO            NOT NULL  NUMBER
NAME          NOT NULL  CHAR(10)
```

테이블을 만들었으면 실제로 data를 input 하여 정상적으로 동작하는지 확인해보도록 하겠습니다. 그전에 unique 한 PK 값을 쉽게 만들기 위해서 하단과 같이 sequence 를 만들도록 합니다. Sequence 는 Oracle 에서 제공하는 기능으로 일정 범위의 unique 한 숫자를 반환해주는 기능입니다. 반드시 필요한 기능은 아니지만 PK 값을 편하게 입력할 수 있기 때문에 만들고 사용하는 것이 편할 수도 있습니다.

```
SQL> CREATE sequence test_seq
2 increment BY 1
3 START with 1
4 MAXvalue 999999999
5 nocycle
6 nocache;
```

시퀀스가 생성되었습니다.

만들어진 sequence 를 이용해서 테이블에 data 를 하나 input 하고 select 를 이용하여 해당 Table을 조회 해 봅니다.

```
SQL> insert into IBMTESSTABLE(no, name) values(test_seq.NEXTVAL, 'IBMTTest');
```

1 개의 행이 만들어졌습니다.

```
SQL> select * from IBMTESSTABLE;
```

NO	NAME
1	IBMTTest

SQL>

정상적으로 Data 가 들어간 것을 확인하면 해당 내용에 대해 commit 합니다.

```
SQL> commit;
```

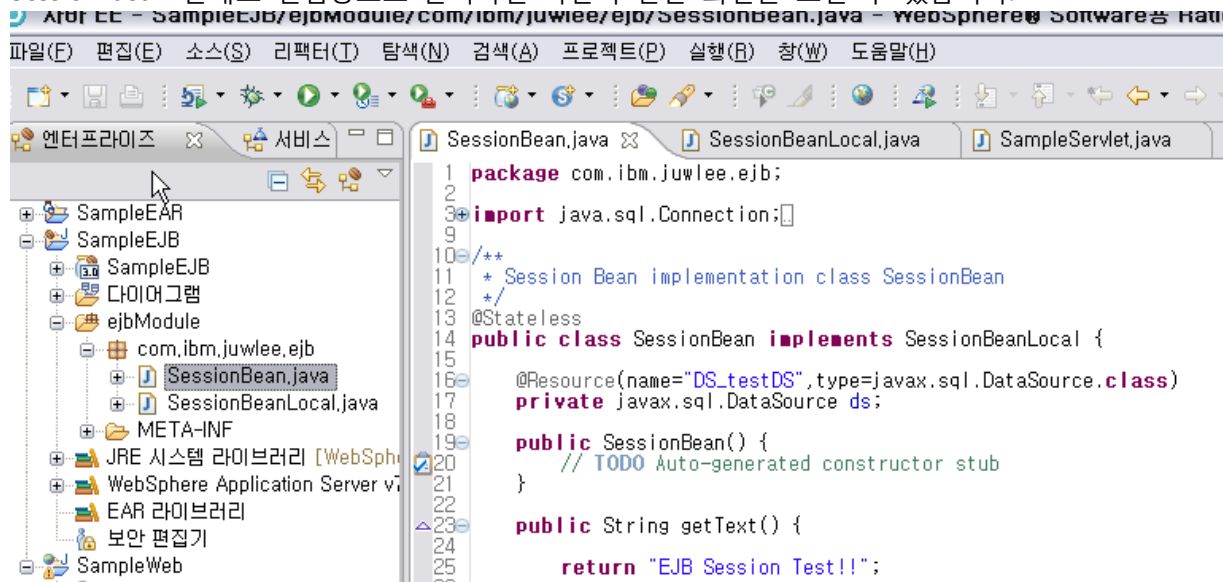
커밋이 완료되었습니다.

이로써, DB 호출전에 WAS 어플리케이션에서 호출해야할 DB 테이블 준비를 완료했습니다.

Part 2. DB 호출 어플리케이션 작성

DB 테이블이 준비되었으면 이제 실제 DB를 호출작업할 어플리케이션을 작성해보도록 하겠습니다. 전부 새로 만들 수도 있지만 재활용하는 차원에서 이전에 만들었던 SampleEAR 어플리케이션을 이용해서 추가하는 작업으로 어플리케이션을 작성하겠습니다.

SampleEAR 어플리케이션에 내용을 추가하기 위하여 RAD v7.5 를 열고 해당 어플리케이션의 Session bean 클래스 편집창으로 들어가면 하단과 같은 화면을 보실 수 있습니다.



지난 강좌에서 EJB Session bean 클래스로 만들었던 소스에 하단의 소스를 추가합니다. 하단의 소스는 EJB 3.0 규약에서 자원을 참조하고 look up 하는 소스로서 DB 와의 연결을 위한 Data source 의 객체를 받아올 수 있는 방법입니다. 사전에 설명드렸지만 WAS 에서는 JDBC provider 와의 연결을 위해 Data source 를 사용하며 어플리케이션은 그 Data source 객체를 받아와서 Connection 을 만들어 DB 호출작업을 할 수 있습니다.

```
@Stateless
public class SessionBean implements SessionBeanLocal {

    @Resource(name="DS_testDS", type=javax.sql.DataSource.class)
    private javax.sql.DataSource ds;
```

이제 insertDB() 라는 메소드를 하나 더 추가합니다. 해당 메소드는 Data source 를 이용하여 DB 에 접속가능한 connection 을 만들고 data 를 insert 하는 SQL 구문을 수행하는 샘플 소스입니다.

```

protected void insertDB() {

    String inst_sql = "insert into IBMTESSTABLE(no, name) values(test_seq.NEXTVAL, ?)";

    Connection conn = null;
    PreparedStatement pstmt = null;

    try{
        conn=ds.getConnection();
        pstmt=conn.prepareStatement(inst_sql);
        pstmt.setString(1, "IBMTesT");

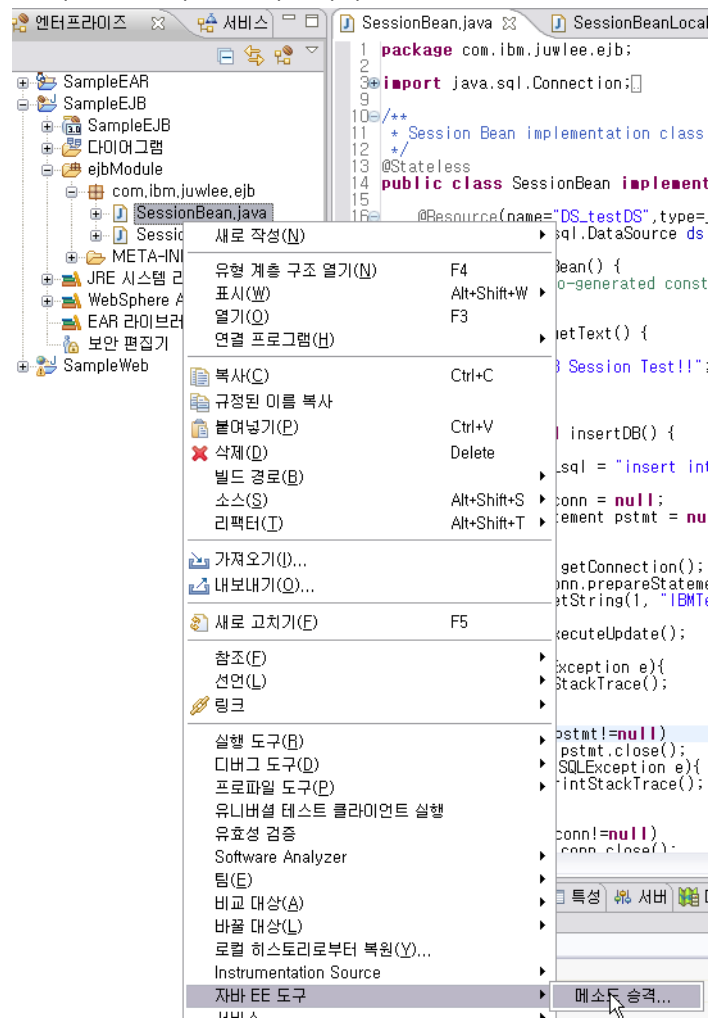
        pstmt.executeUpdate();

    }catch(SQLException e){
        e.printStackTrace();
    }finally {
        try{
            if(pstmt!=null)
                pstmt.close();
        }catch(SQLException e){
            e.printStackTrace();
        }
        try{
            if(conn!=null)
                conn.close();
        }catch(SQLException e){
            e.printStackTrace();
        }
    }

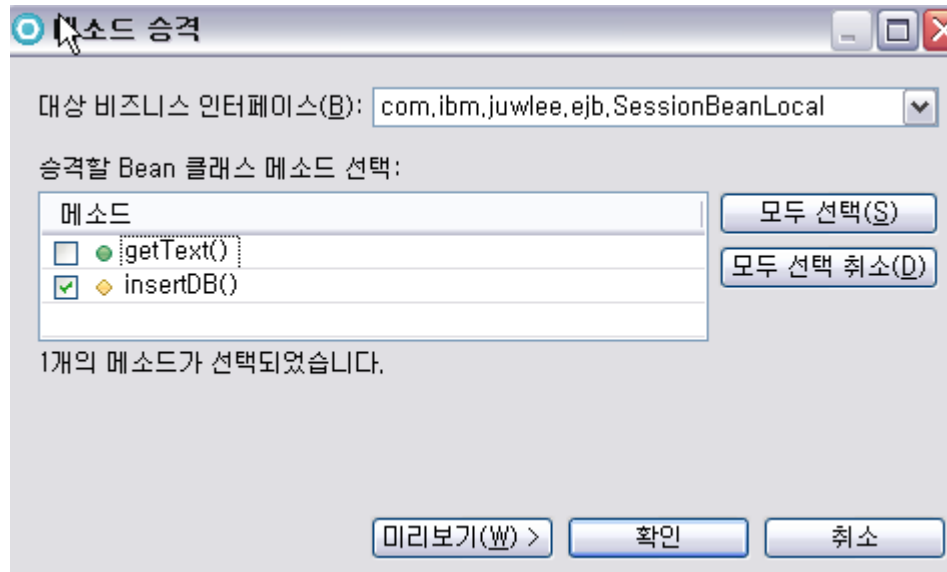
}
}

```

EJB 의 Session bean 클래스에 insertDB() 라는 메소드를 하나 추가하였고 이 메소드를 EJB 외부에서 호출할 예정이기 때문에 지난 강좌에서 설명한 것과 같이 마우스 우클릭 후에 자바 EE 도구 > 메소드 승격을 선택합니다.



메소드 승격 마법사에서 위에서 만들었던 insertDB() 를 선택하고 확인을 선택합니다.



메소드 승격이 완료되면 로컬 인터페이스에 insertDB() 메소드가 등록되므로 이제 동적 웹프로젝트에 가서 이전에 작업한 SampleServlet 페이지에 하단처럼 insertDB() 메소드를 호출하는 부분을 추가합니다.

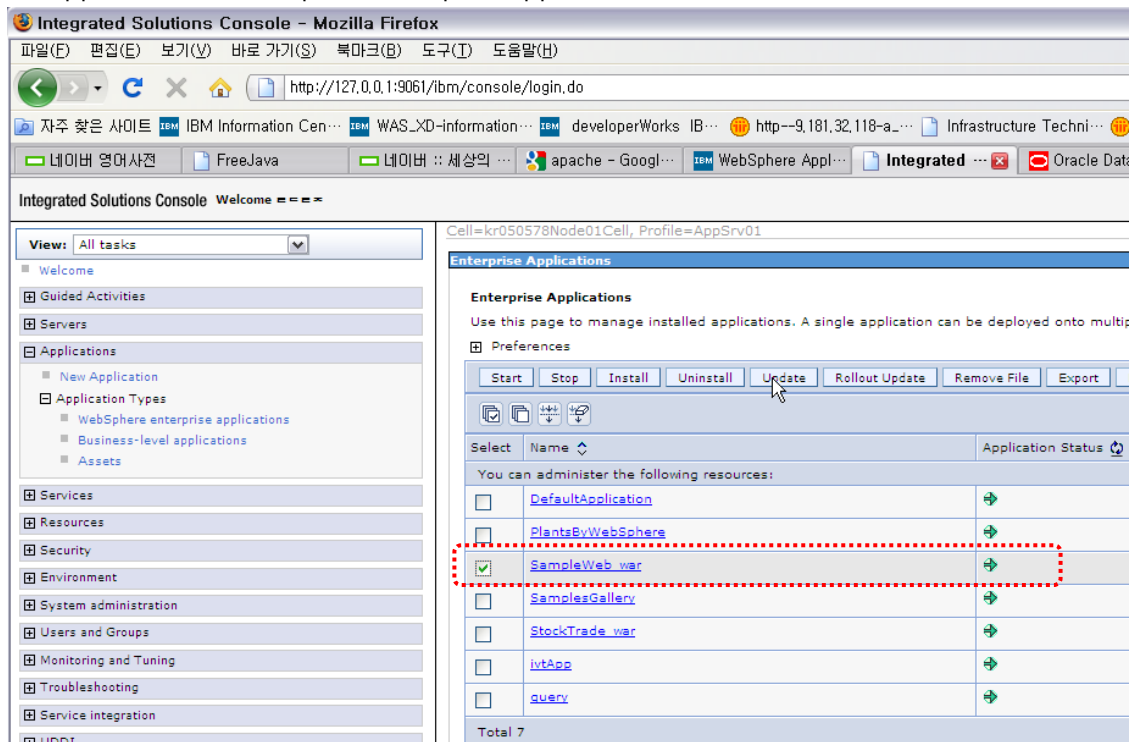
```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    PrintWriter out = response.getWriter();
    out.println("Hello Servlet World!!!");
    out.println("EJB call : " + ejb3Session.getText() );
    ejb3Session.insertDB();
}
```

여기까지 잘 작업하셨다면 DB 호출을 위한 EJB 소스 작업은 잘 완료된 것입니다. 그럼 이제 해당 EAR 을 반출하고 WAS 에 배포하는 단계로 넘어가도록 하겠습니다.

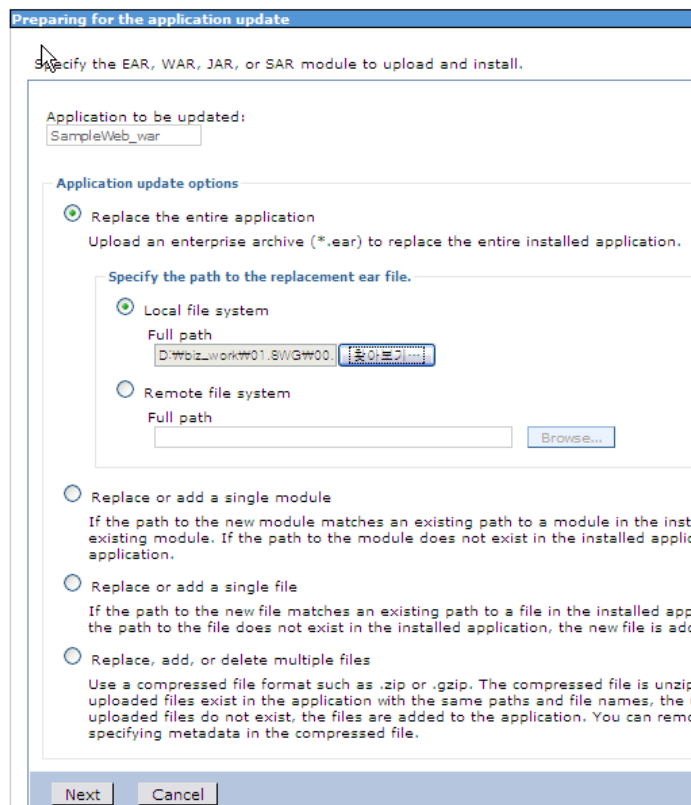
(EAR 반출하는 방법이 생각나지 않으신 분들은 이전 강좌를 참고하세요.)

Part 3. DB 호출 어플리케이션 배포

DB 호출 어플리케이션 작성이 완료되었으면 이를 배포하기 위하여 IBM WAS 관리콘솔로 접근하여 Applications > WebSphere enterprise application 메뉴로 갑니다.



이전에 작업했던 어플리케이션을 선택하고 업데이트 버튼을 클릭하면 하단과 같은 업데이트 마법사가 나오고 방금전에 작성하고 반출했던 EAR 파일을 선택한 후 다음을 클릭합니다.



이전 강좌에서 진행한 것 처럼 빠른 설치를 선택하고 다음을 클릭합니다.

Preparing for the application update

How do you want to install the application?

☒ Fast Path - Prompt only when additional information is required.

☐ Detailed - Show all installation options and parameters.

Specify bindings to use

merge new and existing bindings ▼

☒ Choose to generate default bindings and mappings

Previous Next Cancel

어플리케이션 설치 마법사의 옵션 설정이 나오면 확인하고 다음을 누릅니다.

Specify options for installing enterprise applications and modules.

→ Step 1: Select installation options

Step 2: Map modules to servers

★ Step 3: Map resource references to resources

Step 4: Summary

Select installation options

Specify the various options that are available to prepare and in

☐ Precompile JavaServer Pages files

Directory to install application
\$(APP_INSTALL_ROOT)/\

☒ Distribute application

☐ Use Binary Configuration

☐ Deploy enterprise beans

☒ Create MBeans for resources

☐ Override class reloading settings for Web and EJB modules

Reload interval in seconds
[]

☐ Deploy Web services

Validate Input off/warn/fail
warn ▼

☐ Process embedded configuration

File Permission

Allow all files to be read but not written to
Allow executables to execute
Allow HTML and image files to be read by everyone

.*\.*dll=755#.*\.*so=755#.*\.*a=755#.*\.*sl=755

Application Build ID
Unknown

☐ Allow dispatching includes to remote resources

☐ Allow servicing includes from remote resources

Asynchronous Request Dispatch Type
Disabled ▼

☐ Allow EJB reference targets to resolve automatically

Next Cancel

어플리케이션, 서버 매핑 화면이 나오면 확인 후에 다음으로 넘어갑니다.

Specify options for installing enterprise applications and modules.

Step 1 Select installation options

→ **Step 2: Map modules to servers**

★ **Step 3** Map resource references to resources

Step 4 Summary

Map modules to servers

Specify targets such as application servers or clusters of application servers where you want to install the modules that are contained in your application. Modules can be installed on the same application server or dispersed among several application servers. Also, specify the Web servers as targets that serve as routers for requests to this application. The plug-in configuration file (plugin-cfg.xml) for each Web server is generated, based on the applications that are routed through.

Clusters and servers:
WebSphere:cell=kr050578Node01Cell,node=kr050578Node01,server=server1

Select	Module	URI	Server
<input type="checkbox"/>	SampleEJB.jar	SampleEJB.jar,META-INF/ejb-jar.xml	WebSphere:cell=kr050578Node01Cell,node=kr050578Node01,server=server1
<input type="checkbox"/>	SampleWeb	SampleWeb.war,WEB-INF/web.xml	WebSphere:cell=kr050578Node01Cell,node=kr050578Node01,server=server1

이전 강좌의 배포와 다른 화면을 볼 수 있으실 텐데 해당 화면은 EJB 어플리케이션에서 참조한 자원을 실제적으로 WAS 에서 설정된 Data source 와 연결하기 위한 화면입니다. 타겟의 빈 칸 밑의 Browse 를 클릭합니다.

Specify options for installing enterprise applications and modules.

Step 1 Select installation options

Step 2 Map modules to servers

→ **Step 3: Map resource references to resources**

Step 4 Summary

Map resource references to resources

Each resource reference that is defined in your application must be mapped to a resource.

javax.sql.DataSource

Select	Module	EJB	URI	Resource Reference	Target Resource JNDI Name	Login configuration
<input type="checkbox"/>	SampleEJB.jar	SessionBean	SampleEJB.jar,META-INF/ejb-jar.xml	DS_testDS	<input type="text" value="Browse..."/>	Resource authorization: Container Authentication method: None

WAS 에 등록된 자원중 이전 강좌에서 만들었던 XA용 데이터 소스인 DS_testXA Data source 를 선택하고 확인을 누릅니다.

Resources that can be used to bind to the resource-reference of a bean. Resources shown here are only those available to that module carrying the bean. This is determined by the targets to which that module is mapped. Resources available to a module can come from a hierarchical scope of a bean. If resources at different scopes have the same JNDI name, the one at the lower scope will override the parent. The overridden resources are not shown here.

Select	Name	JNDI NAME	Scope	Description
<input checked="" type="radio"/>	DS_testXA	jdbc/testXA	Cell=kr050578Node01Cell	New JDBC Datasource
<input type="radio"/>	Default Datasource	DefaultDatasource	Node=kr050578Node01,Server=server1	Datasource for the WebSphere Default Application
<input type="radio"/>	DefaultEJBTimerDataSource	jdbc/DefaultEJBTimerDataSource	Node=kr050578Node01,Server=server1	Default data source for WebSphere EJB Timer Service
<input type="radio"/>	PLANTSDB	jdbc/PlantsByWebSphereDataSource	Node=kr050578Node01,Server=server1	Data source for the Plants by WebSphere entity beans
<input type="radio"/>	PLANTSDBNonJTA	jdbc/PlantsByWebSphereDataSourceNONJTA	Node=kr050578Node01,Server=server1	Data source for the Plants by WebSphere entity beans, non-JTA

Total 5

자원 맵핑 메뉴의 타겟 부분의 빈칸이 적절하게 JNDI 이름으로 채워진 것을 확인하고 다음을 클릭합니다.

Specify options for installing enterprise applications and modules.

Step 1 Select installation options

Step 2 Map modules to servers

→ Step 3: Map resource references to resources

Step 4 Summary

Map resource references to resources

Each resource reference that is defined in your application must be mapped to a resource.

javax.sql.DataSource

Set Multiple JNDI Names ▾ Modify Resource Authentication Method... Extended Properties...

Select	Module	EJB	URI	Resource Reference	Target Resource JNDI Name	Login configuration
<input type="checkbox"/>	SampleEJB.jar	SessionBean	SampleEJB.jar,META-INF/ejb-jar.xml	DS_testDS	<input type="text" value="jdbc/testXA"/> Browse...	Resource authorization: Container Authentication method: None

Previous Next Cancel

마지막으로 요약정보를 확인하고 완료를 누릅니다.

Specify options for installing enterprise applications and modules.

Step 1 Select installation options

Step 2 Map modules to servers

Step 3 Map resource references to resources

→ Step 4: Summary

Summary

Summary of installation options

Options	Values
Precompile JavaServer Pages files	No
Directory to install application	\$(APP_INSTALL_ROOT)/kr050578Node01Cell
Distribute application	Yes
Use Binary Configuration	No
Deploy enterprise beans	No
Create MBeans for resources	Yes
Override class reloading settings for Web and EJB modules	No
Reload interval in seconds	
Deploy Web services	No
Validate Input off/warn/fail	warn
Process embedded configuration	No
File Permission	.*\,dll=755#.*\,so=755#.*\,a=755#.*\,sl=755
Application Build ID	Unknown
Allow dispatching includes to remote resources	No
Allow servicing includes from remote resources	No
Asynchronous Request Dispatch Type	Disabled
Allow EJB reference targets to resolve automatically	No
Application name	SampleWeb_war
Cell/Node/Server	Click here

Previous Finish Cancel

어플리케이션 배포 진행과정을 살펴보고 완료되면 저장(save)을 눌러 배포작업을 마무리 합니다.

Updating...

If there are enterprise beans in the application, the EJB deployment process can take several minutes. Do not save the configuration until the process completes.

Check the SystemOut.log on the deployment manager or server where the application is deployed for specific information about the EJB deployment process as it occurs.

ADMA5017: Uninstallation of SampleWeb_war started.

ADMA5104: The server index entry for WebSphere:cell=kr050578Node01Cell,node=kr050578Node01 is updated successfully.

ADMA5102: The configuration data for SampleWeb_war from the configuration repository is deleted successfully.

ADMA5011: The cleanup of the temp directory for application SampleWeb_war is complete.

ADMA5108: Application SampleWeb_war uninstalled successfully.

ADMA5016: Installation of SampleWeb_war started.

ADMA5067: Resource validation for application SampleWeb_war completed successfully.

ADMA5058: Application and module versions are validated with versions of deployment targets.

ADMA5005: The application SampleWeb_war is configured in the WebSphere Application Server repository.

ADMA5053: The library references for the installed optional package are created.

ADMA5005: The application SampleWeb_war is configured in the WebSphere Application Server repository.

ADMA5001: The application binaries are saved in D:\IBM\WebSphere7\AppServer\profiles\AppSrv01\wstemp\387907277\workspace\cells\kr050578Node01Cell\applications\SampleWeb_war.

ADMA5005: The application SampleWeb_war is configured in the WebSphere Application Server repository.

SECJ0400: Successfully updated the application SampleWeb_war with the appContextIDForSecurity information.

ADMA5005: The application SampleWeb_war is configured in the WebSphere Application Server repository.

ADMA5113: Activation plan created successfully.

ADMA5011: The cleanup of the temp directory for application SampleWeb_war is complete.

ADMA5013: Application SampleWeb_war installed successfully.

Application SampleWeb_war installed successfully.

To start the application, first save changes to the master configuration.

The application might not be immediately available while being started on all servers.

Changes have been made to your local configuration. You can:

- [Save](#) directly to the master configuration.
- [Review](#) changes before saving or discarding.

관리콘솔을 통해서 배포된 어플리케이션의 상태를 확인합니다.

Enterprise Applications

Use this page to manage installed applications. A single application can be deployed onto multiple servers.

Preferences

Select	Name	Application Status
<input type="checkbox"/>	DefaultApplication	➡
<input type="checkbox"/>	PlantsByWebSphere	➡
<input type="checkbox"/>	SampleWeb_war	➡
<input type="checkbox"/>	SamplesGallery	➡
<input type="checkbox"/>	StockTrade_war	➡
<input type="checkbox"/>	ivtApp	➡
<input type="checkbox"/>	query	➡

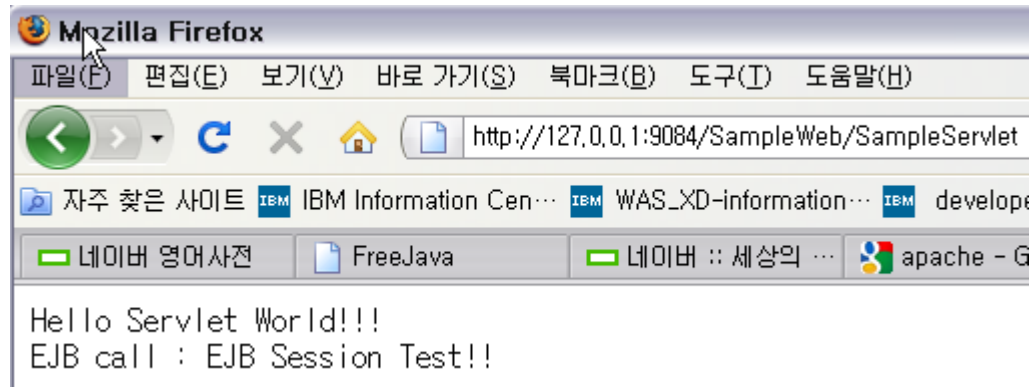
Total 7

여기까지가 DB 호출 어플리케이션 배포작업의 끝입니다. 그럼 이제 해당 어플리케이션에 대한 테스트를 진행해볼까요.

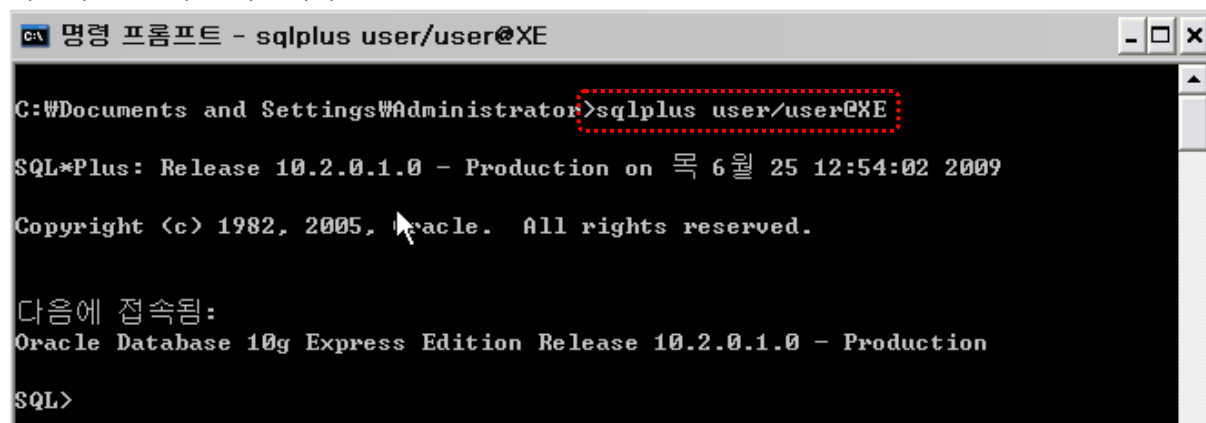
Part 4. DB 호출 어플리케이션 테스트

이전에 만들었던 Sample 어플리케이션을 재사용하여 소스를 추가하였기 때문에 SampleServlet 을 웹 브라우저를 활용하여 하단처럼 호출합니다.

([http://127.0.0.1:9084\(포트\)/SampleWeb\(ContextRoot\)/SampleServlet\(Servlet명\)](http://127.0.0.1:9084(포트)/SampleWeb(ContextRoot)/SampleServlet(Servlet명)))

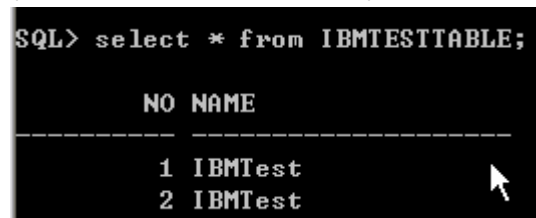


위와 같은 화면이 나오면 성공적으로 호출이 완료된 것입니다. 현재 어플리케이션의 메소드가 호출되면 insert 구문에 대한 SQL이 수행되는 것이므로 이를 확인하기 위하여 sqlplus 를 활용해서 하단처럼 DB에 접속합니다.



DB에 접속이 되었으면 insert 작업이 수행된 Table 을 select 문장으로 조회합니다.

(Select * from IBMTTESTTABLE;)



이전에 만들었던 Table 에 하나의 data 가 추가적으로 들어온 것을 확인하실 수 있습니다. 검증을 위하여 웹 브라우저에서 진행된 Servlet 페이지의 호출을 다시 한번 진행합니다.

Servlet 페이지 호출이 완료되면 다시 Sqlplus 에 가서 select 문의로 해당 Table 의 data 를 조회 합니다.

```
SQL> select * from IBMTTESTTABLE;
      NO NAME
-----
      1 IBMTTest
      2 IBMTTest
      3 IBMTTest
SQL>
```

위와 같이 이전에 비해 하나 더 추가된 table list 를 보실 수 있다면 이번 강좌도 성공적으로 완료하신 것 입니다. 여기까지 잘 따라오신 분들은 너무 고생하셨고 이번 강좌까지 나온 내용들에 대한 이해만 잘 하시면 기본적으로 어플리케이션을 작성해서 WAS에 배포하고 DB 연동하는 작업에 대한 기본은 이해하신 것입니다. 어때요... 참 쉽죠...^^& 그럼 휘리릭~~~

참고 1) IBM Information Center for WebSphere Application Server v7

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.multipatform.doc/info/welcome_nd.html

참고 2) InforCenter – Data source

(http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.base.doc/info/aes/aes/cdat_datasor.html?resultof=%22%64%61%74%61%22%20%22%73%6f%75%72%63%65%22%20%22%73%6f%75%72%63%22%20)

※이 자료의 저작권은 작성자에게 있으며 유포는 자유로이 허용되나 상업적으로 이용은 금합니다.