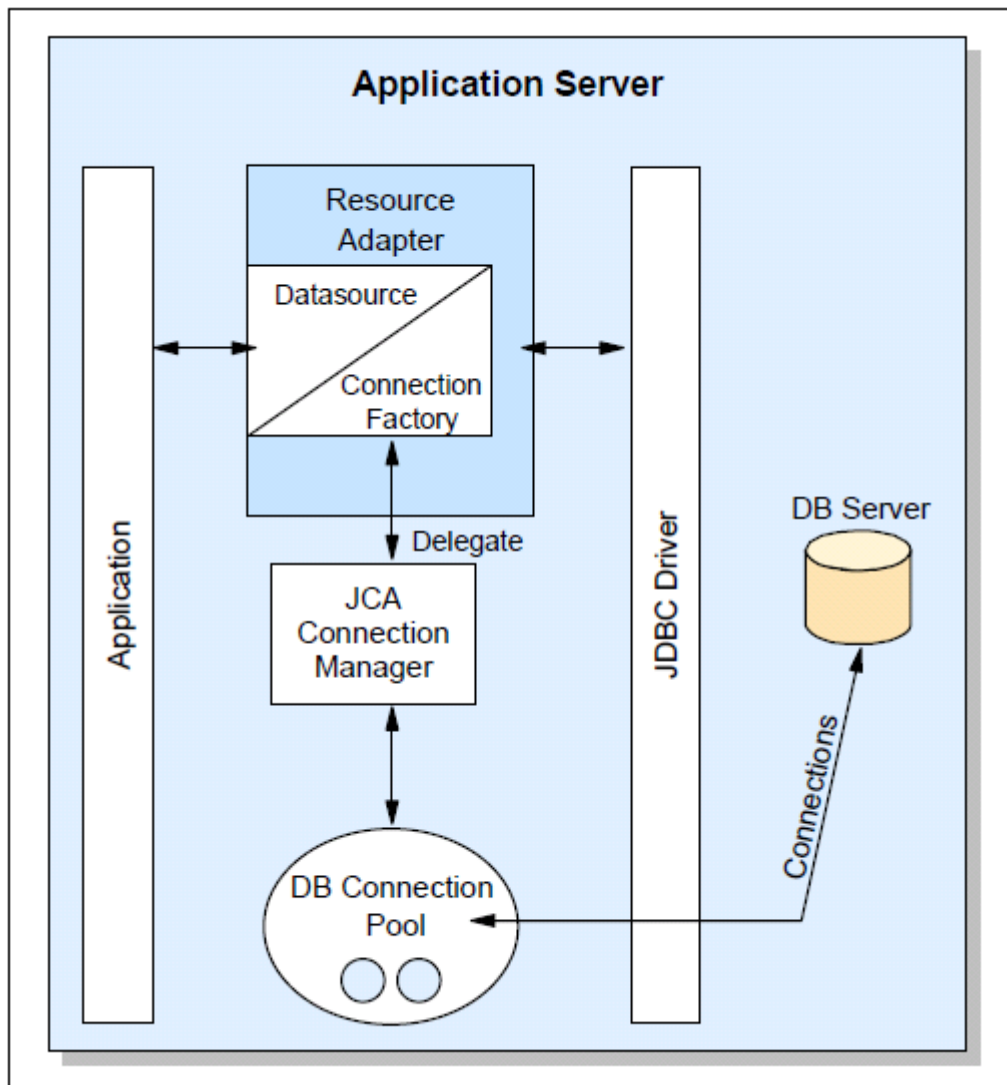


## 하나씩 쉽게 따라 해보는 IBM WebSphere Application Server(WAS) v7 – 6 – A

Original Editor : 이정운 ([juwlee@kr.ibm.com](mailto:juwlee@kr.ibm.com))

Updated by SunYeon Min ([symin@kr.ibm.com](mailto:symin@kr.ibm.com)) on Apr, 2010

하나씩 쉽게 따라 해보는 IBM WAS v7 시리즈 그 여섯번째 이야기 DB 호출입니다. 본 강좌에서 말씀드리려는 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. DB2 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.

IBM DB2의 명령행 도구 > 명령창을 이용해서 해당 DB에 접속합니다.( ex: >db2 connect to sample)

```
DB2 CLP - DB2COPY1
C:\Program Files\IBM\SQLLIB\BIN>db2 connect to sample

데이터베이스 연결 정보

데이터베이스 서버          = DB2/NT 9.5.0
SQL 권한 부여 ID          = ADMINIST...
로컬 데이터베이스 별명    = SAMPLE

C:\Program Files\IBM\SQLLIB\BIN>
```

Data 를 넣어둘 샘플 테이블을 생성합니다.

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

```
C:\Program Files\IBM\SQLLIB\BIN>db2 "create table test_table( col1 integer generated always as identity ( start with 1, increment by 1, no cache ), col2 varchar(15))"
DB20000I  SQL 명령이 완료되었습니다.

C:\Program Files\IBM\SQLLIB\BIN>
```

IBM DB2 > 일반관리 도구 > 제어 센터를 실행해서 Sample DB의 TEST\_TABLE을 확인합니다.

|                 |               |                 |  |  |  |
|-----------------|---------------|-----------------|--|--|--|
| TEST_TABLE      | ADMINISTRATOR | IBMDB2SAMPLE... |  |  |  |
| 표시된 148 항목의 148 |               |                 |  |  |  |

### 테이블 - TEST\_TABLE

스키마 : ADMINISTRATOR

작성자 : ADMINISTRATOR

컬럼 : 2

조치:

#### 컬럼

| 키 | 이름   | 데이터 유형  | 길이 | 널(NUL...) |
|---|------|---------|----|-----------|
|   | COL1 | INTEGER | 4  | 아니오       |
|   | COL2 | VARCHAR | 15 | 예         |

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 DB2 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.

```
C:\Program Files\IBM\SQLLIB\BIN>db2 "create sequence seq_1 start with 1 increment by 1 maxvalue 100 cycle"
DB20000I  SQL 명령이 완료되었습니다.
```

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

```
C:\Program Files\IBM\SQLLIB\BIN>db2 insert into test_table(col2) values ('Scott')
DB20000I  SQL 명령이 완료되었습니다.

C:\Program Files\IBM\SQLLIB\BIN>
C:\Program Files\IBM\SQLLIB\BIN>db2 select * from test_table

COL1          COL2
-----
          1 Scott

1 레코드가 선택되었습니다.

C:\Program Files\IBM\SQLLIB\BIN>
```

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

```
C:\Program Files\IBM\SQLLIB\BIN>db2 commit
DB20000I  SQL 명령이 완료되었습니다.

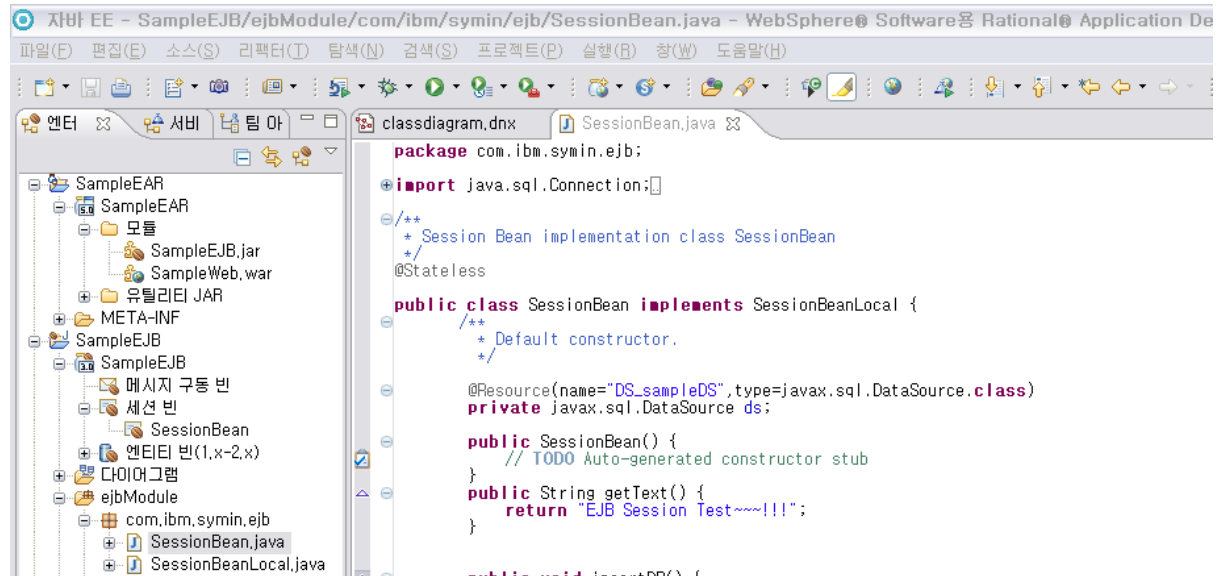
C:\Program Files\IBM\SQLLIB\BIN>
```

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.

```
public class SessionBean implements SessionBeanLocal {
    /**
     * Default constructor.
     */

    @Resource(name="DS_sampleDS", 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.

```

public void insertDB() {
    String inst_sql = "insert into administrator.test_table(col2) values ('IBMTTest')";

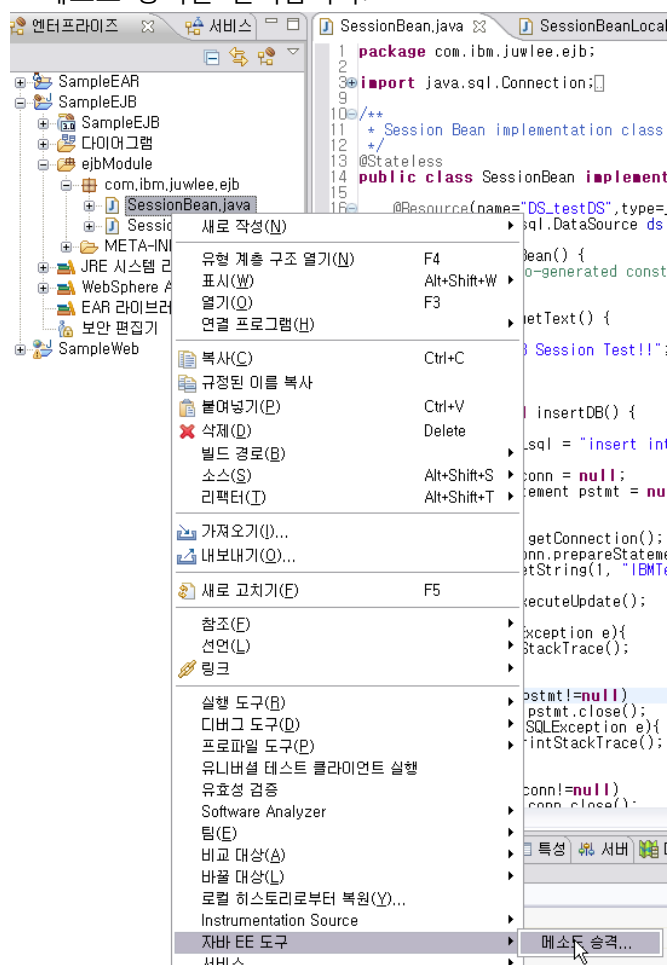
    Connection conn = null;
    PreparedStatement pstmt = null;

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

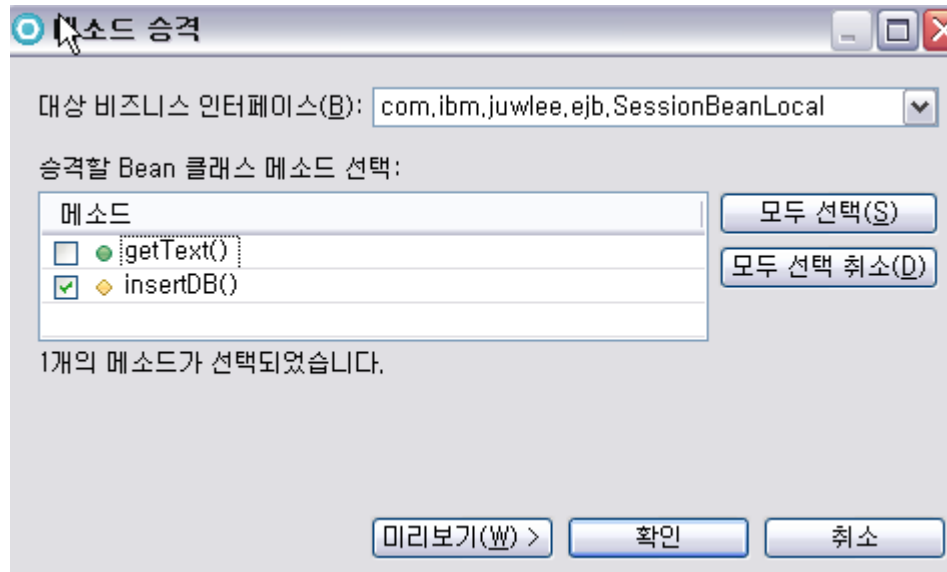
        pstmt.executeUpdate();
    }catch(SQLException e){
        e.printStackTrace();
    }finally {
        try{
            if(pstmt!=null)
                pstmt.close();
        }catch(SQLException e){
            e.printStackTrace();
        }
        try{
            if(conn!=null)
                conn.close();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

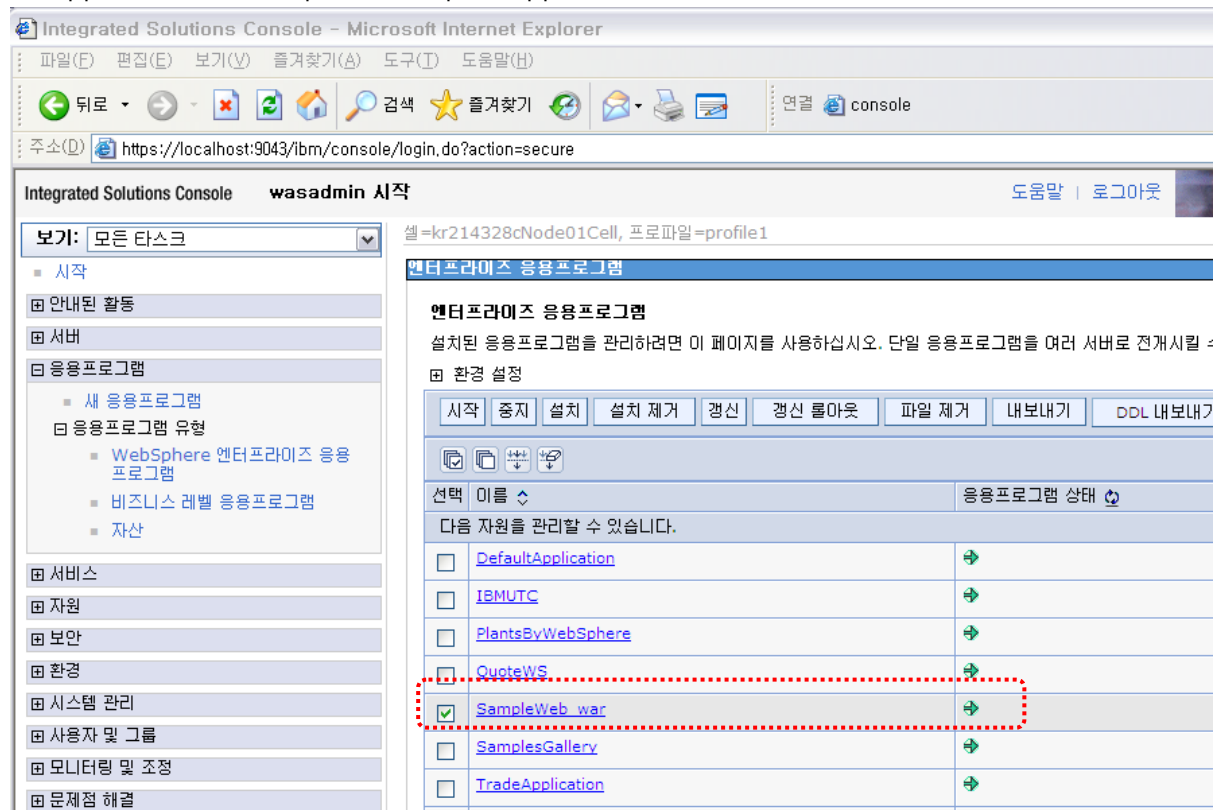
```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    PrintWriter out = response.getWriter();
    out.println("Hello Servlet World!!!");
    out.println("EJB call : " + ejb3Session.getText()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

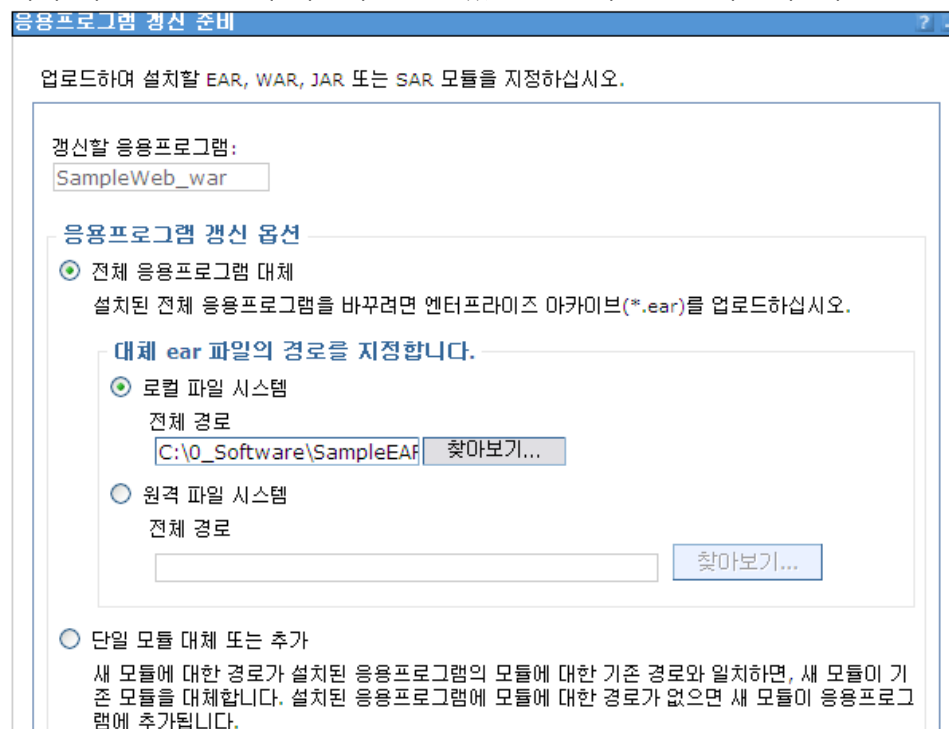
(EAR 반출하는 방법이 생각나지 않으신 분들은 이전 강좌를 참고하세요.)

### 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.

응용프로그램 경신 준비

응용프로그램 설치 방법

☒ 빠른 경로 - 추가 정보가 필요한 경우에만 프롬프트됩니다.

☐ 세부사항 - 모든 설치 옵션 및 매개변수를 표시합니다.

사용할 바인딩을 지정

새 바인딩과 기존 바인딩 병합

☒ 기본 바인딩 및 맵핑을 생성하도록 선택

이전

다음

취소

어플리케이션 설치 마법사의 옵션 설정이 나오면 확인하고 다음을 누릅니다.

→ 단계 1: 설치 옵션 선택

단계 2 서버에 모듈 맵핑

\* 단계 3 **자원으로 자원 참조 맵핑**

단계 4 요약

설치 옵션 선택

사용자 응용프로그램을 준비하고 설치하는 데 사용 가능한 다양한 옵션을 지정하십시오.

☐ JavaServer 페이지 파일 사전 컴파일

응용프로그램을 설치할 디렉토리

\$(APP\_INSTALL\_ROOT)/kr

☒ 응용프로그램 분배

☐ 2진 구성 사용

☐ 엔터프라이즈 Bean 전개

☒ 자원에 대한 MBeans 작성

☐ 웹 및 EJB 모듈의 설정을 다시 로드하여 클래스 대체

다시 로드 간격(초)

☐ 웹 서비스 전개

입력 유효성 검증 값(깨짐/경고/실패)

경고

☐ 프로세스 임베디드 구성

파일 권한

모든 파일 읽기 허용(쓰기 불가능)

실행 파일 실행 허용

모두에게 HTML 및 이미지 파일 읽기 허용

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

응용프로그램 빌드 ID

Unknown

☐ 원격 자원으로 포함(include) 디스패치 허용

☐ 원격 자원에서 포함(include) 서비스 허용

비동기 요청 디스패치 유형

사용 불가능

☐ EJB 참조 대상이 자동으로 해석하도록 허용

다음

취소



어플리케이션, 서버 매핑 화면이 나오면 확인 후에 다음으로 넘어갑니다.

엔터프라이즈 응용프로그램과 모듈에 대한 설치 옵션을 지정합니다.

**단계 1** 설치 옵션 선택

→ **단계 2: 서버에 모듈 매핑**

★ **단계 3: 자원으로 자원 참조 매핑**

단계 4 요약

**서버에 모듈 매핑**

응용프로그램에 포함된 모듈을 설치할 Application Server나 Application Server 클러스터와 같은 대상을 지정하십시오. 모듈은 동일한 Application Server에 설치되거나 여러 Application Server 간에 분산될 수 있습니다. 또한 이 응용프로그램에 대한 요청의 라우터로 서비스를 제공할 웹 서버를 대상으로 지정하십시오. 웹 서버 각각의 플러그인 구성 파일(plugin-cfg.xml)은 웹 서버가 라우트되는 응용프로그램을 기초로 생성됩니다.

클러스터 및 서버:

WebSphere:cell=kr214328cNode01Cell,node=kr214328cNode01,server=server1

| 선택                       | 모듈            | URI                                | 서버   |
|--------------------------|---------------|------------------------------------|--|
| <input type="checkbox"/> | SampleEJB.jar | SampleEJB.jar,META-INF/ejb-jar.xml | WebSphere:cell=kr214328cNode01Cell,node=kr214328cNode01,server=server1 |
| <input type="checkbox"/> | SampleWeb     | SampleWeb.war,WEB-INF/web.xml      | WebSphere:cell=kr214328cNode01Cell,node=kr214328cNode01,server=server1 |

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

엔터프라이즈 응용프로그램과 모듈에 대한 설치 옵션을 지정합니다.

**단계 1** 설치 옵션 선택

단계 2 서버에 모듈 매핑

→ **단계 3: 자원으로 자원 참조 매핑**

단계 4 요약

**자원으로 자원 참조 매핑**

응용프로그램에 정의된 각 자원 참조가 자원에 매핑되어야 합니다.

**javax.sql.DataSource**

| 선택                       | 모듈            | EJB         | URI                                | 자원 참조       | 대상 자원 JNDI 이름                        | 로그인 구성                 |
|--------------------------|---------------|-------------|------------------------------------|-------------|--------------------------------------|------------------------|
| <input type="checkbox"/> | SampleEJB.jar | SessionBean | SampleEJB.jar,META-INF/ejb-jar.xml | DS_sampleDS | <input type="text" value="찾아보기..."/> | 자원 권한: 컨테이너 인증 메소드: 없음 |

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

| 선택                               | 이름                        | JNDI 이름                                | 범위                                  | 설명  |
|----------------------------------|---------------------------|--|-------------------------------------|---|
| <input checked="" type="radio"/> | DS_sampleDS               | jdbc/sample                            | Cell=kr214328cNode01Cell            | DB2 Universal Driver DataSource                               |
| <input type="radio"/>            | Default DataSource        | DefaultDataSource                      | Node=kr214328cNode01,Server=server1 | Datasource for the WebSphere Default Application              |
| <input type="radio"/>            | DefaultEJBTimerDataSource | jdbc/DefaultEJBTimerDataSource         | Node=kr214328cNode01,Server=server1 | Default data source for WebSphere EJB Timer Service           |
| <input type="radio"/>            | PLANTSDB                  | jdbc/PlantsByWebSphereDataSource       | Node=kr214328cNode01,Server=server1 | Data source for the Plants by WebSphere entity beans          |
| <input type="radio"/>            | PLANTSDBNonJTA            | jdbc/PlantsByWebSphereDataSourceNONJTA | Node=kr214328cNode01,Server=server1 | Data source for the Plants by WebSphere entity beans, non-JTA |

총계 5

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

엔터프라이즈 응용프로그램과 모듈에 대한 설치 옵션을 지정합니다.

단계 1  
설치 옵션  
선택

단계 2  
서버에 모  
듈 맵핑

→ 단계 3: **자원으로  
자원 참조 맵핑**

단계 4  
요약

자원으로 자원 참조 맵핑

응용프로그램에 정의된 각 자원 참조가 자원에 맵핑되어야 합니다.

javax.sql.DataSource

복수 JNDI 이름 설정 ▾

자원 인증 메소드 수정...

확장 특성...

| 선택                       | 모듈            | EJB         | URI                                | 자원 참조       | 대상 자원 JNDI 이름                             | 로그인 구성                          |
|--------------------------|---------------|-------------|------------------------------------|-------------|---|---------------------------------|
| <input type="checkbox"/> | SampleEJB.jar | SessionBean | SampleEJB.jar,META-INF/ejb-jar.xml | DS_sampleDS | <div>jdbc/sample</div> <div>찾아보기...</div> | 자원 권한:<br>컨테이너<br>인증 메소드:<br>없음 |

이전

다음

취소

마지막으로 요약정보를 확인하고 완료를 누릅니다.

엔터프라이즈 응용프로그램과 모듈에 대한 설치 옵션을 지정합니다.

단계 1  
설치 옵션  
선택

단계 2  
서버에 모  
듈 맵핑

단계 3  
자원으로  
자원 참조 맵핑

→ 단계 4: **요약**

요약

설치 옵션 요약

| 옵션                             | 값   |
|--------------------------------|---|
| JavaServer 페이지 파일 사전 컴파일       | 아니오   |
| 응용프로그램을 설치할 디렉토리               | \$(APP_INSTALL_ROOT)/kr214328cNode01Cell            |
| 응용프로그램 분배                      | 예   |
| 2진 구성 사용                       | 아니오   |
| 엔터프라이즈 Bean 전개                 | 아니오   |
| 자원에 대한 MBeans 작성               | 예   |
| 웹 및 EJB 모듈의 설정을 다시 로드하여 클래스 대체 | 아니오   |
| 다시 로드 간격(초)                    |   |
| 웹 서비스 전개                       | 아니오   |
| 입력 유효성 검증 값(퍼짐/경고/실패)          | 경고  |
| 프로세스 임베디드 구성                   | 아니오   |
| 파일 권한                          | .*\.*.dll=755#.*\.*.so=755#.*\.*.a=755#.*\.*.sl=755 |
| 응용프로그램 빌드 ID                   | Unknown   |
| 원격 자원으로 포함(include) 디스패치 허용    | 아니오   |
| 원격 자원에서 포함(include) 서비스 허용     | 아니오   |
| 비동기 요청 디스패치 유형                 | 사용 불가능  |
| EJB 참조 대상이 자동으로 해석하도록 허용       | 아니오   |
| 응용프로그램 이름                      | SampleWeb_war                                       |
| 셀/노드/서버                        | <a href="#">여기를 클릭하십시오.</a>                         |

이전

완료

취소

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

경신 중...

응용프로그램에 엔터프라이즈 Bean이 있는 경우, EJB 전개 프로세스에 몇 분이 소요될 수 있습니다. 프로세스가 완료될 때까지 구성을 저장하지 마십시오.

EJB 전개 프로세스 발생 시 이 프로세스에 대한 특정 정보는 응용프로그램이 전개되는 서버 또는 Deployment Manager에서 SystemOut.log를 확인하십시오.

ADMA5017: SampleWeb\_war 설치 제거가 시작되었습니다.

ADMA5104: WebSphere:cell=kr214328cNode01Cell,node=kr214328cNode01에 대한 서버 색인 항목이 갱신되었습니다.

ADMA5102: SampleWeb\_war에 대한 구성 데이터가 구성 저장소에서 삭제되었습니다.

ADMA5011: 응용프로그램 SampleWeb\_war의 temp 디렉토리 정리가 완료되지 않았습니다.

ADMA5106: SampleWeb\_war 응용프로그램이 설치 제거되었습니다.

ADMA5016: SampleWeb\_war 설치가 시작되었습니다.

ADMA5067: SampleWeb\_war 응용프로그램의 자원 유효성 검증이 완료되었습니다.

ADMA5058: 전개 대상 버전에 대해 응용프로그램 및 모듈 버전의 유효성을 검증했습니다.

ADMA5005: WebSphere Application Server 저장소에 SampleWeb\_war 응용프로그램이 구성되어 있습니다.

ADMA5053: 설치된 선택적 패키지에 대한 라이브러리 참조가 작성됩니다.

ADMA5005: WebSphere Application Server 저장소에 SampleWeb\_war 응용프로그램이 구성되어 있습니다.

ADMA5001: 응용프로그램 2진이 C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\wstemp\514564614\workspace\cells\kr214328cNode01Cell\applications\SampleWeb\_war에 저장되었습니다.

ADMA5005: WebSphere Application Server 저장소에 SampleWeb\_war 응용프로그램이 구성되어 있습니다.

SECJ0400: SampleWeb\_war 응용프로그램이 appContextIDForSecurity 정보로 갱신되었습니다.

ADMA5005: WebSphere Application Server 저장소에 SampleWeb\_war 응용프로그램이 구성되어 있습니다.

ADMA5001: 응용프로그램 2진이 C:\Program Files\IBM\WebSphere\AppServer\profiles\profile1\wstemp\514564614\workspace\cells\kr214328cNode01Cell\applicat에 저장되었습니다.

ADMA5005: WebSphere Application Server 저장소에 SampleWeb\_war 응용프로그램이 구성되어 있습니다.

SECJ0400: SampleWeb\_war 응용프로그램이 appContextIDForSecurity 정보로 갱신되었습니다.

ADMA5005: WebSphere Application Server 저장소에 SampleWeb\_war 응용프로그램이 구성되어 있습니다.

CWSAD0040: Application Server 저장소에 SampleWeb\_war 응용프로그램이 구성되었습니다.

ADMA5113: 활성화 계획이 작성되었습니다.

ADMA5011: 응용프로그램 SampleWeb\_war의 temp 디렉토리 정리가 완료되지 않았습니다.

ADMA5013: SampleWeb\_war 응용프로그램이 설치되었습니다.

SampleWeb\_war 응용프로그램이 설치되었습니다.

응용프로그램을 시작하려면 우선 마스터 구성의 변경사항을 저장하십시오.

응용프로그램을 모든 서버에서 동시에 시작하는 경우, 즉시 사용하지 못할 수 있습니다.

로컬 구성에 변경사항이 작성되었습니다. 다음을 수행할 수 있습니다.

- 마스터 구성에 직접 저장
- 저장하거나 버리기 전에 변경사항 검토

관리콘솔을 통해서 배포된 어플리케이션의 상태를 확인합니다.

#### 엔터프라이즈 응용프로그램

설치된 응용프로그램을 관리하려면 이 페이지를 사용하십시오. 단일 응용프로그램을 여러 서버로 전개시킬 수 있습니다.

환경 설정

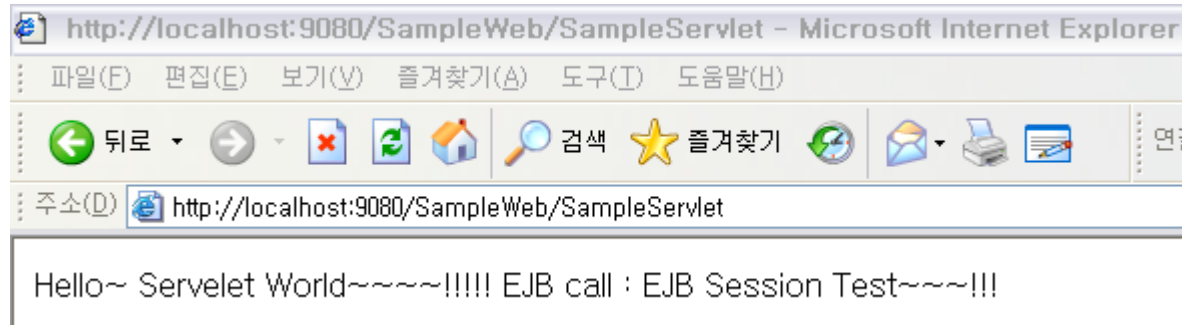
| 시작 중지 설치 설치 제거 갱신 갱신 롤아웃 파일 제거 내보내기 DDL 내보내기 파일 내보내기 |                                    |           |
|--|------------------------------------|-----------|
|  |                                    |           |
| 선택   | 이름                                 | 응용프로그램 상태 |
| 다음 자원을 관리할 수 있습니다.                                   |                                    |           |
| <input type="checkbox"/>                             | <a href="#">DefaultApplication</a> | ➡         |
| <input type="checkbox"/>                             | <a href="#">IBMUTC</a>             | ➡         |
| <input type="checkbox"/>                             | <a href="#">PlantsByWebSphere</a>  | ➡         |
| <input type="checkbox"/>                             | <a href="#">QuoteWS</a>            | ➡         |
| <input type="checkbox"/>                             | <a href="#">SampleWeb_war</a>      | ➡         |
| <input type="checkbox"/>                             | <a href="#">SamplesGallery</a>     | ➡         |
| <input type="checkbox"/>                             | <a href="#">TradeApplication</a>   | ➡         |
| <input type="checkbox"/>                             | <a href="#">iytApp</a>             | ➡         |
| <input type="checkbox"/>                             | <a href="#">query</a>              | ➡         |
| 총계 9   |                                    |           |

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

#### Part 4. DB 호출 어플리케이션 테스트

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

(<http://127.0.0.1:9080>(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 CLP를 활용해서 DB에 접속하고, insert 작업이 수행된 Table 을 select 문장으로 조회합니다.

(Select \* from test\_table)

```
C:\Program Files\IBM\SQLLIB\BIN>db2 select * from test_table

COL1          COL2
-----
          1  Scott
          2  IBMTTest

2 레코드가 선택되었습니다.

C:\Program Files\IBM\SQLLIB\BIN>
```

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 CLP창에 가서 select 문으로 해당 Table 의 data 를 조회합니다.

```
C:\Program Files\IBM\SQLLIB\BIN>db2 select * from test_table

COL1          COL2
-----
          1  Scott
          2  IBMTTest
          3  IBMTTest

3 레코드가 선택되었습니다.
```

위와 같이 이전과 비교하여 하나 더 추가된 table list 를 보실 수 있다면 이번 강좌도 성공적으로 완료하신 것입니다. 여기까지 잘 따라오신 분들은 너무 고생하셨고 이번 강좌까지 나온 내용들

에 대한 이해만 잘 하시면 기본적으로 어플리케이션을 작성해서 WAS에 배포하고 DB 연동하는  
작업에 대한 기본은 이해하신 것입니다. 어때요... 참 쉽죠...^^& 그럼 휘리릭~~~

[http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.multiprocessor.doc/info/welcome\\_nd.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.multiprocessor.doc/info/welcome_nd.html)

([http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.base.doc/info/ae/s/ae/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](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.base.doc/info/ae/s/ae/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))