

# WEBT 설정방법 및 실행결과

---

## [Jeus에서 Tmax Service 호출]

일자	작성자
2013.12.04	김요셉

## Update History

[illegible]

# CONTENTS

## 1. WEBT

### 1) 개념

A. webt.properties를 이용

B. JEUSMain.xml를 이용

### 2) 실습

A. webt.properties

B. JEUSMain.xml

### 3) Source

A. tpcall\_form.jsp

B. tpcall\_not\_datasource.jsp와 tpcall\_datasource.jsp

### 4) 실행결과

A. tpcall\_not\_datasource.jsp

B. tpcall\_datasource.jsp

## 1. WEBT

### 1) 기본개념

- 원리 : JEUS의 웹페이지로 Tmax의 Service를 호출. 즉, JEUS & Tmax 연동
- 방법 : webt-xxxx.jar 파일이 필요하고, 아래 2가지 방법을 사용

#### A. webt.properties를 이용

- webt.properties 파일을 이용해서 수행.

#### B. JEUSMain.xml를 이용

- JEUSMain.xml의 <resource>절에 설정하여 수행

### 2) 설정

## A. webt.properties

- JEUSMain.xml의 해당 컨테이너 command-option에 webt.properties 경로를 적어줌

```
<command-option> ↵  
-Xms512m -Xmx512m -XX:PermSize=128m -XX:MaxPermSize=256m ↵  
-XX:+DisableExplicitGC -verbose:gc -Xloggc:C:/TmaxSoft/JEUS6.0/logs/gclog/con1.log  
-XX:+PrintGCDetails -XX:+PrintGCTimeStamps -XX:+PrintHeapAtGC ↵  
-XX:+HeapDumpOnOutOfMemoryError ↵  
-XX:HeapDumpPath=C:/TmaxSoft/JEUS6.0/logs/dump ↵  
-Dwebt.properties=C:/TmaxSoft/JEUS6.0/webt/webt.properties ↵  
↵  
</command-option> ↵
```

[그림] JEUSMain.xml – webt.properties 경로 지정

- 빨간부분을 환경에 맞게 Setting

```
# printed to standard out ↵
log.dir=C:/TmaxSoft/JEUS6.0/webt/webtlog1 ↵
↵
# set the name of the log file. ↵
default is webt.log ↵
log.file=webt.log ↵
↵
# set log buffering size. default is 0 ↵
log.bufsize=512 ↵
↵
# set log valid day -> seperated by date-string. default is -1
log.valid.days=1 ↵
↵
# # FDL related parameters ##### ↵
# set the fdl file. ↵
fdl.file=C:/TmaxSoft/JEUS6.0/webt/tmax.fdl ↵
```

[그림] webt.properties – log, fdl 경로지정

- 빨간부분을 환경에 맞게 Setting

```
# WebtConnectionGroup name list ↵
connectionPool.groups=tmax1 ↵
↵
# set connection group type valid values are shared,
connectionPool.tmax1.type=shared ↵
↵
# set Tmax Server Address. ↵
connectionPool.tmax1.hostAddr=192.168.190.128 ↵
↵
# set Tmax Server Port. ↵
connectionPool.tmax1.hostPort=8888 ↵
```

[그림] webt.properties – IP, Port, Group 명 지정

- tpcall\_form.jsp을 실행하여 not\_datasource방식으로 Tmax 서비스를 실행

## B. JEUSMain.xml

- name-prefix가 export-name(JNDI)이 됨

- 환경에 맞게 빨간부분을 수정

```
.
<property> ↵
  <key>log-file-name</key> ↵
  <value>C:/TmaxSoft/JEUS6.0/webt/webtlog1/webt_jndi.log</value>
</property> ↵
↵
<property> ↵
  <key>log-level</key> ↵
  <value>info</value> ↵
</property> ↵
↵
<property> ↵
  <key>fdl-file</key> ↵
  <value>C:/TmaxSoft/JEUS6.0/webt/tmax.fdl</value> ↵
</property> ↵
.
```

[그림] JEUSMain.xml – log, fdl 경로설정

- 아래 그림은 tmax1이 JNDI명이 됨

```
<property-group> ↵
  <name-prefix>tmax1</name-prefix>.
```

[그림] JEUSMain.xml – export-name 설정

```

<property> ↵
  <key>host-name</key> ↵
  <value>192.168.190.128</value>
</property> ↵
↵
<property> ↵
  <key>host-port</key> ↵
  <value>8888</value> ↵
</property> ↵

```

[그림] JEUSMain.xml – IP, Port 설정

```

<property> ↵
  <key>min</key> ↵
  <value>5</value> ↵
</property> ↵
↵
<property> ↵
  <key>max</key> ↵
  <value>20</value> ↵
</property> ↵
↵
<property> ↵
  <key>step</key> ↵
  <value>2</value> ↵
</property> ↵
↵
<property> ↵
  <key>period</key> ↵
  <value>60000</value> ↵
</property> ↵

```

[그림] JEUSMain.xml – min, max, step, period 설정

```

<name>webt</name> ↵
<class-name>tmax.webt.external.WebtResource</class-name> ↵

```

[그림] JEUSMain.xml - webt연동 필요 Class 설정



### 3) Source

#### A. tpcall\_form.jsp

- tpcall\_not\_datasource.jsp과 tpcall\_datasource.jsp로 나누어주기 위한

JSP 파일

```
<form action="tpcall_not_datasource.jsp">
<p> .<p> .<p> .<p> .
<table border="1" align="center" cellpadding="0" cellspacing="0">
<tr>
<td width="400" colspan="2" align="center" valign="center"><h3>:: TPCall Test Page ::</h3></td>
</tr>
<tr>
<td width="200" align="center"><font color="blue">버퍼 타입</font></td><td width="200" align="center"><select name='iType'><option value='1'>STRING<option va
</tr>
<tr>
<td width="200" align="center"><font color="blue">XA/NonXA</font></td><td width="200" align="center"><select name='iXA'><option value='1'>XA<option value='2
</tr>
<tr>
<td width="200" align="center"><font color="blue">서비스명</font></td><td width="200" align="center"><input type="text" name='iService' size=10 value=''></td>
</tr>
<tr>
<td width="200" align="center"><font color="blue">입력 데이터</font></td><td width="200" align="center"><input type="text" name='iData' size=10 value=''></td>
</tr>
<tr>
<td width="400" colspan="2" align="center"><input type="submit" value='실행'></td>
```

[그림] tpcall\_form.jsp

**B. tpcall\_not\_datasource.jsp와 tpcall\_datasource.jsp**

- tpcall\_not\_datasource.jsp는 \* webt.properties 이용
- tpcall\_datasource.jsp는 \* Resource 이용

- 빨간부분은 환경에 맞게 수정

```

    /**
    * 커넥션 얻어오기 webt.properties
    */
    /**
    // conn = new WebtConnection("192.168.190.128", "8888")
    conn = WebtConnectionPool.getConnection("tmax1");
    */

    /**
    * 커넥션 얻어오기 WebtDataSource
    */
    /*
final String TMAXSOURCE = "tmax1";
try {
    InitialContext ic = new InitialContext();
    ds = (WebtDataSource)ic.lookup(TMAXSOURCE);
    conn = ds.getConnection();
} catch (Exception ex) {
    ex.printStackTrace();
}
//System.out.println("getConnection");
*/

```

[그림] tpcall\_not\_datasource.jsp와 tpcall\_datasource.jsp – 접속 방식 차이

- Field의 경우 demo.f에 등록한 Field명을 적어줌

```
case 3:
    /*****
     * 필드버퍼 테스트시
     *****/
    service = new WebtRemoteService(iService, conn);

    WebtFieldSet sndset = new WebtFieldSet(service.createFieldBuffer());

    if (iService.equals("FDLToupper") || iService.equals("FDLTolower")) {
        sndset.add("INPUT", iData);
    }
    else if (iService.equals("FDLSEL")) {
        String id = iData;
        sndset.add("FDL_ID", id);
    }

    sndbuf = sndset.getFieldBuffer();
    break;
```

[그림] tpccall\_not\_datasource.jsp와 tpccall\_datasource.jsp – Field

```

    /*****
     * 필드데이터 꺼내는 방법 1
     *****/
    WebtFieldSet rcvset = new WebtFieldSet(rcvbuf);

    int count = rcvset.count("FDL_ID");

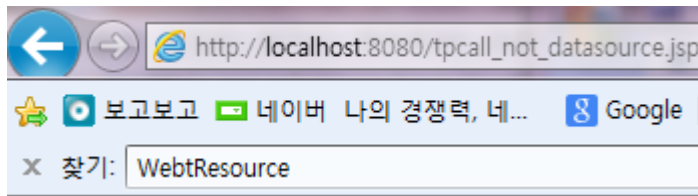
    out.println("<p>count=" + count);
    out.println("<p><table border=1><tr><td>FDL_ID</td><td>FDL_NAME</td><td>FDL_AGE</td></tr>");
    for (int i=1; i <= count; i++) {

        out.println("<tr>" +
            "<td>" + rcvset.getString("FDL_ID") + "</td>" +
            "<td>" + rcvset.getString("FDL_NAME") + "</td>" +
            "<td>" + rcvset.getString("FDL_AGE") + "</td>" +
            "</tr>");
    }
    out.println("</table>");
```

[그림] tpccall\_not\_datasource.jsp와 tpccall\_datasource.jsp – Field 응답

## 4) 실행결과

## A. tpcall\_not\_datasource.jsp

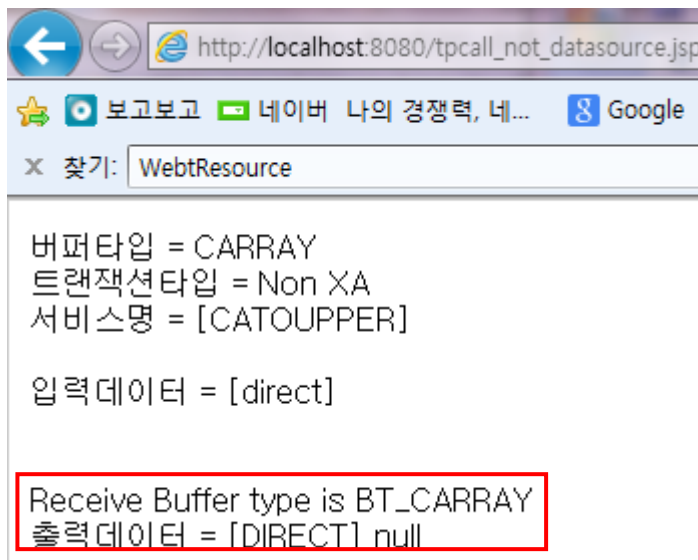


버퍼타입 = STRING  
트랜잭션타입 = Non XA  
서비스명 = [TOUPPER]

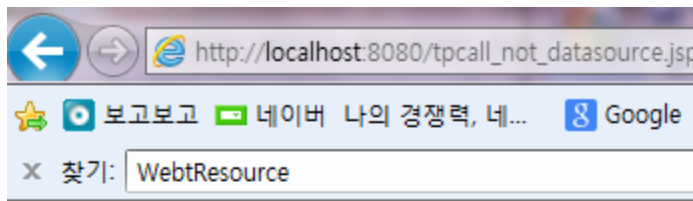
입력데이터 = [direct]

Receive Buffer type is BT\_STRING  
출력데이터 = [DIRECT] null

[그림] tpcall\_not\_datasource.jsp – String



[그림] tpcall\_not\_datasource.jsp - Carray



버퍼타입 = FIELD  
 트랜잭션타입 = Non XA  
 서비스명 = [FDLSEL]

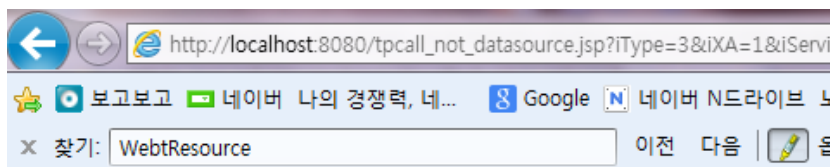
입력데이터 = [00]

Receive Buffer type is BT\_FIELD

count=1

FDL_ID	FDL_NAME	FDL_AGE
00	seb	11

[그림] tpcall\_not\_datasource.jsp – Field\_Nonxa



버퍼타입 = FIELD  
 트랜잭션타입 = XA  
 서비스명 = [FDLSEL]

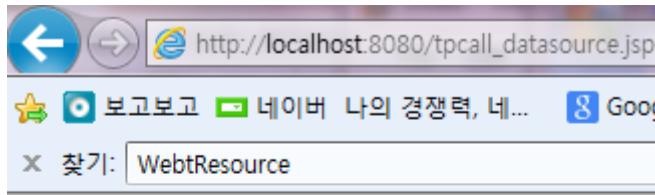
입력데이터 = [00]

Receive Buffer type is BT\_FIELD

count=1

FDL_ID	FDL_NAME	FDL_AGE
00	seb	11

[그림] tpcall\_not\_datasource.jsp – Field\_Xa

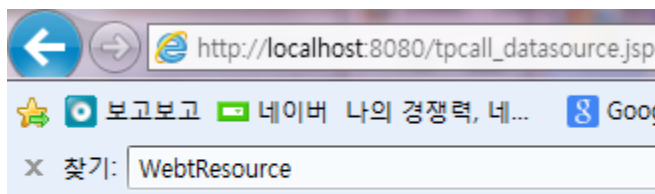
**B. tpcall\_datasource.jsp**

버퍼타입 = STRING  
트랜잭션타입 = Non XA  
서비스명 = [TOUPPER]

입력데이터 = [jndi]

Receive Buffer type is BT\_STRING  
출력데이터 = [JNDI] null

[그림] tpcall\_datasource.jsp - String

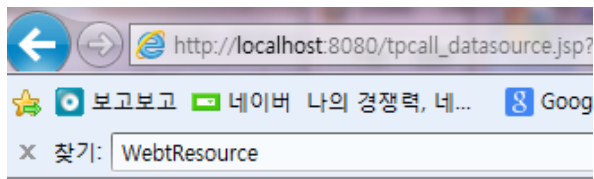


버퍼타입 = CARRAY  
트랜잭션타입 = Non XA  
서비스명 = [CATOUPPER]

입력데이터 = [jndi]

Receive Buffer type is BT\_CARRAY  
출력데이터 = [JNDI] null

[그림] tpcall\_datasource.jsp - Carray



버퍼타입 = FIELD

트랜잭션타입 = Non XA

서비스명 = [FDLSEL]

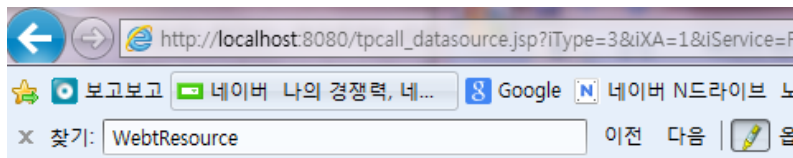
입력데이터 = [00]

Receive Buffer type is BT\_FIELD

count=1

FDL_ID	FDL_NAME	FDL_AGE
00	seb	11

[그림] tpcall\_datasource.jsp – Field\_Nonxa



버퍼타입 = FIELD

트랜잭션타입 = XA

서비스명 = [FDLSEL]

입력데이터 = [00]

Receive Buffer type is BT\_FIELD

count=1

FDL_ID	FDL_NAME	FDL_AGE
00	seb	11

[그림] tpcall\_datasource.jsp – Field\_Xa