Table of Contents

If you use database procedures and/or functions in your application, wouldn't it be great to call them with one simple call? Normally, a function or procedure call is very complex because you have to define a CallableStatement, set the output parameters, execute the statement, and read the result. And don't forget that the parameter index starts with 1 instead of 0!

Suppose we have a function (e.g., Oracle):

```
CREATE OR REPLACE FUNCTION execFunction(pNumber IN OUT NUMBER, pInText IN
VARCHAR2,
                                        pOutText OUT VARCHAR2) RETURN
VARCHAR2 IS
  res VARCHAR2(200);
  nr NUMBER := pNumber;
BEGIN
  pOutText := 'Out: '|| pOutText ||' In: '|| pInText;

  pNumber := pNumber + pNumber;

  RETURN 'IN-Param Nr: '|| nr;
END execFunction;
```

The function is simple but uses one in/out and one out parameter. A standard JDBC call looks like:

```
Connection con;

//create a DB connection

CallableStatement cstmt = con.prepareCall("{ ? = call EXECFUNCTION(?, ?, ?)
}");
cstmt.registerOutParameter(1, Types.VARCHAR);
cstmt.registerOutParameter(2, Types.DECIMAL);
cstmt.registerOutParameter(4, Types.VARCHAR);

cstmt.setObject(2, BigDecimal.valueOf(1), Types.DECIMAL);
cstmt.setObject(3, "ABC", Types.VARCHAR);
cstmt.execute();

Object oResult = cstmt.getObject(1);
```

The execution has more lines of code than the function in the database!

Now we call the same function through JVx:

```
Object oResult = dba.executeFunction("execFunction", Types.VARCHAR,
BigDecimal.valueOf(1),
                                     "ABC", null);
```

You are right, we have not support for in/out and out parameters! To support this kind of parameters, we have some special classes:

com.sibvisions.rad.persist.jdbc.param.InParam

com.sibvisions.rad.persist.jdbc.param.OutParam
com.sibvisions.rad.persist.jdbc.param.InOutParam

Use it as with the following code:

```
OutParam ouTextParam = new OutParam(InOutParam.SQLTYPE_VARCHAR);
InOutParam ioNumberParam = new InOutParam(InOutParam.SQLTYPE_DECIMAL,
                                          BigDecimal.valueOf(1));

Object oResult = dba.executeFunction("execFunction", Types.VARCHAR,
ioNumberParam,
                                     "ABC", ouTextParam);

Object oNumber = ioNumberParam.getValue();
Object oText = ouTextParam.getValue();
```

The `executeFunction` method supports standard Java objects and our special Param objects, as used in the previous example.

If you want to call a database procedure instead of a function, use the same classes and call `executeProcedure`.

From:
https://doc.sibvisions.com/ - **Documentation**

Permanent link:
https://doc.sibvisions.com/jvx/server/db/func_proc

Last update: **2020/06/29 13:06**