Java dan AS/400

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME
1.1	2014-01		SF

Contents

1 Pengantar

Java adalah bahasa pemrograman yang sangat populer dan digunakan oleh banyak kalangan. Walaupun diciptakan oleh Sun Microsystem tetapi Java telah menjadi milik suatu komunitas yang amat luas. Saat ini hak cipta dan merek Java dimiliki oleh Oracle yang membeli Sun Microsystem walau demikian arah perkembangan teknologi Java tidak semata-mata ditentukan oleh Oracle. Komunitas pemakai Java yang banyak dan aktif ikut serta menentukan perkembangan Java. Lisensi teknologi Java saat ini adalah lisensi GPL versi 2. Lisensi ini adalah lisensi open source dan free software. Ketersediaan source Java menjamin kelangsungan hidup Java.

Sedangkan AS/400 (IBM i) adalah komputer bisnis yang dibuat IBM. Apa keunggulan AS/400?

1. Kompabilitas ke belakang yang panjang. Anda masih bisa menjalankan program yang anda buat 20-an tahun lalu di AS/400 model terbaru.
2. Kestabilan. AS/400 bisa dioperasikan selama bertahun-tahun tanpa mengalami gangguan. Jika terjadi kerusakan pada salah satu bagian mesin seperti hard disk maka AS/400 bisa diganti bagian yang rusak tanpa perlu melakukan shutdown.
3. Kemudahan pemakaian. Walaupun jumlah kemampuan yang disediakan oleh AS/400 luar biasa banyaknya akan tetapi semua kemampuan didokumentasikan dengan sangat baik. Pemakai juga dituntun oleh prompt sehingga tahu persis parameter mana yang dibutuhkan dan mana yang tidak. Tersedia juga sistem menu yang sangat lengkap.
4. Keamanan. Sangat sedikit eksploit keamanan yang bisa digunakan untuk menembus keamanan sistem secara teknis. Operating system dari AS/400 yaitu OS/400 dari sejak awal sudah dirancang mengutamakan keamanan. Tentu saja keamanan bukan sekedar masalah teknis tapi juga prosedur pemakaian sistem. Tapi secara teknis celah keamanan AS/400 sangatlah sedikit.
5. Skalabilitas. AS/400 tersedia dalam skala kecil dimana 1 mesin bisa digunakan puluhan user sampai skala besar dimana 1 mesin bisa digunakan ribuan user. Jika 1 mesin besar tidak cukup maka beberapa mesin besar bisa dikonfigurasi sehingga terlihat sebagai satu sistem saja dan bisa melayani puluhan ribu user.

Mengingat hal diatas maka pemakaian Java dan AS/400 untuk membangun platform software bisnis anda adalah suatu investasi bisnis yang baik untuk jangka panjang. Dalam tutorial berikut penulis akan membagikan beberapa tehnik pemakaian Java dan AS/400.

2 Perangkat Lunak

Dalam menuliskan tutorial ini penulis menggunakan perangkat lunak berikut:

1. Open JDK versi 7
2. Maven versi 3.1.1
3. NetBeans 7.4

Program Java yang dibuat di uji dengan dijalankan diatas Linux. Penulis menggunakan Manjaro Linux 64-bit. Program RPG yang dibuat duji dengan dijalankan diatas OS/400 v5r1.

3 AS/400 Hosting

Penulis menggunakan hosting AS/400 gratis dari situs http://rzkh.de. Daftarkan diri anda disana untuk mendapatkan akses gratis ke AS/400. Penulis mendaftarkan diri disana dengan nama user SAMUELF. Karena itu penulis mendapatkan 2 library yaitu SAMUELF1 dan SAMUELF2. Tutorial ini menggunakan dua library tersebut. Sesuaikan nama library dalam tutorial ini dengan nama library yang anda dapatkan.

4 Data Queue

RPG adalah bahasa pemrograman paling populer di AS400. Data queue adalah fasilitas messaging standar di AS/400. Data queue bisa digunakan untuk komunikasi antara program RPG dan Java. Dalam tutorial berikut penulis akan memberikan contoh program dimana program Java menuliskan data ke data queue. Data dalam data queue akan dibaca oleh program RPG lalu diproses. Hasil pemrosesan ditaruh dalam data queue. Program Java lalu membaca data queue dan menampilkan hasil proses dari RPG.

Mari kita membuat data queue. Panjang data queue harus sama dengan panjang data yang dikirim oleh program.

```
===>CRTDTAQ DTAQ(SAMUELF1/Q01) MAXLEN(36)
===>CRTDTAQ DTAQ(SAMUELF1/Q02) MAXLEN(33)
```

Buatlah member DTAQUE dalam file QRPGLESRC di library SAMUELF1 dengan tipe RPGLE.

```
===>STRSEU SRCFILE(SAMUELF1/QRPGLESRC) SRCMBR(DTAQUE) TYPE(RPGLE)
```

Lalu masukkan code berikut:

```
      * AUTHOR   : SAMUEL FRANKLYN <SFRANKLYN AT GMAIL.COM>
      * LOCATION : SAMUELF1/QRPGLESRC(DTAQUE)
      * DATE     : 2014-02-02
      *
     D Q01             S             10A   INZ('Q01')
     D Q02             S             10A   INZ('Q02')
     D LIB             S             10A   INZ('SAMUELF1')
     D Q01L            S              5  0 INZ(36)
     D Q02L            S              5  0 INZ(33)
     D WAIT            S              5  0 INZ(-1)
     D NOW             S               D
     D BDAY            S               D
      *
     D Q01DS           DS
     D  NM01                          30A
     D  BDAYS                          6S 0
      *
     D Q02DS           DS
     D  NM02                          30A
     D  AGE                            3S 0
      *
     C                   CALL      'QRCVDTAQ'
     C                   PARM                    Q01
     C                   PARM                    LIB
     C                   PARM                    Q01L
     C                   PARM                    Q01DS
     C                   PARM                    WAIT
      *
     C                   EVAL       NM02=NM01
     C                   EVAL       NOW=%DATE
     C                   EVAL       BDAY=%DATE(BDAYS:*YMD)
     C                   EVAL       AGE=%DIFF(NOW:BDAY:*YEARS)
      *
     C                   CALL      'QSNDDTAQ'
     C                   PARM                    Q02
     C                   PARM                    LIB
     C                   PARM                    Q02L
     C                   PARM                    Q02DS
      *
     C                   EVAL      *INLR=*ON
     C                   RETURN
```

Compile DTAQUE dengan command berikut:

```
===> CRTBNDRPG PGM(SAMUELF1/DTAQUE) SRCFILE(SAMUELF1/QRPGLESRC)
```

Saatnya membuat program Java anda dengan menggunakan Maven. Jalankan perintah berikut di command line

```
$ mvn archetype:generate -DinteractiveMode=false \
   -DarchetypeGroupId=org.apache.maven.archetypes \
   -DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.1 \
   -DgroupId=samuelf.web.id -DartifactId=dataqueue -Dpackage=dataqueue
```

Maven akan membuat file pom.xml, source file App.java dan AppTest.java. Hapuslah source file App.java dan AppTest.java. Sesuaikan isi file pom.xml anda hingga seperti dibawah ini:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
                      http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>samuelf.web.id</groupId>
  <artifactId>dataqueue</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>dataqueue</name>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>net.sf.jt400</groupId>
      <artifactId>jt400-full</artifactId>
      <version>4.7.0</version>
    </dependency>
  </dependencies>
</project>
```

Buat file DataQueueExample.java dengan isi seperti ini:

```
/*
 * Copyright 2013 Samuel Franklyn <sfranklyn@gmail.com>.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *      http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
```

```
 * limitations under the License.
 */
package dataqueue;

import com.ibm.as400.access.AS400;
import com.ibm.as400.access.AS400SecurityException;
import com.ibm.as400.access.AS400Text;
import com.ibm.as400.access.AS400ZonedDecimal;
import com.ibm.as400.access.CharacterFieldDescription;
import com.ibm.as400.access.DataQueue;
import com.ibm.as400.access.DataQueueEntry;
import com.ibm.as400.access.ErrorCompletingRequestException;
import com.ibm.as400.access.IllegalObjectTypeException;
import com.ibm.as400.access.ObjectDoesNotExistException;
import com.ibm.as400.access.Record;
import com.ibm.as400.access.RecordFormat;
import com.ibm.as400.access.ZonedDecimalFieldDescription;
import java.io.IOException;
import java.io.InputStream;
import java.math.BigDecimal;
import java.math.BigInteger;
import java.util.Properties;
import java.util.logging.FileHandler;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 * Data queue example.
 *
 * @author Samuel Franklyn <sfranklyn@gmail.com>
 */
public class DataQueueExample {

    private static final Logger logger = Logger.getLogger(
            DataQueueExample.class.getName());
    private final Properties configProperties = new Properties();

    /**
     * Load properties file from class path and write log into file.
     */
    public DataQueueExample() {
        try {
            InputStream inputStream = this.getClass().getClassLoader().
                    getResourceAsStream("config.properties");
            configProperties.load(inputStream);

            FileHandler fileHandler = new FileHandler("log.xml");
            logger.addHandler(fileHandler);
        } catch (IOException ex) {
            logger.log(Level.SEVERE, null, ex);
        }
    }

    /**
     * Write to data queue using Record object.
     *
     * @param name
     * @param birthday
     */
    public void writeDataQueue(String name, long birthday) {
        String systemName = configProperties.getProperty("as400.systemname");
        String userName = configProperties.getProperty("as400.username");
```

```
        String password = configProperties.getProperty("as400.password");
        String dataQueue = configProperties.getProperty("as400.dataqueue1");

        AS400 as400 = new AS400(systemName, userName, password);
        DataQueue dq = new DataQueue(as400, dataQueue);

        CharacterFieldDescription nameFD
                = new CharacterFieldDescription(new AS400Text(30, as400),
                        "NAME");
        ZonedDecimalFieldDescription birthdayFD
                = new ZonedDecimalFieldDescription(new AS400ZonedDecimal(6, 0),
                        "BIRTHDAY");

        RecordFormat recordFormat = new RecordFormat();
        recordFormat.addFieldDescription(nameFD);
        recordFormat.addFieldDescription(birthdayFD);

        Record record = new Record(recordFormat);
        record.setField("NAME", name);
        record.setField("BIRTHDAY", new BigDecimal(BigInteger.valueOf(birthday),
                0));

        try {
            byte[] byteData = record.getContents();
            dq.write(byteData);
        } catch (AS400SecurityException | ErrorCompletingRequestException |
                ObjectDoesNotExistException | IOException |
                InterruptedException | IllegalObjectTypeException ex) {
            logger.log(Level.SEVERE, null, ex);
        }

    }

    /**
     * Read from data queue using Record object.
     *
     * @return
     */
    public Record readDataQueue() {
        String systemName = configProperties.getProperty("as400.systemname");
        String userName = configProperties.getProperty("as400.username");
        String password = configProperties.getProperty("as400.password");
        String dataQueue = configProperties.getProperty("as400.dataqueue2");

        AS400 as400 = new AS400(systemName, userName, password);
        DataQueue dq = new DataQueue(as400, dataQueue);

        CharacterFieldDescription nameFD
                = new CharacterFieldDescription(new AS400Text(30, as400),
                        "NAME");
        ZonedDecimalFieldDescription ageFD
                = new ZonedDecimalFieldDescription(new AS400ZonedDecimal(3, 0),
                        "AGE");

        RecordFormat recordFormat = new RecordFormat();
        recordFormat.addFieldDescription(nameFD);
        recordFormat.addFieldDescription(ageFD);

        try {
            DataQueueEntry DQData = dq.read(-1);
            Record record = recordFormat.getNewRecord(DQData.getData());
            String name = (String) record.getField("NAME");
```

```
            BigDecimal age = (BigDecimal) record.getField("AGE");
            if (name != null) {
                return record;
            }
        } catch (AS400SecurityException | ErrorCompletingRequestException |
                IOException | IllegalObjectTypeException |
                InterruptedException | ObjectDoesNotExistException ex) {
            logger.log(Level.SEVERE, null, ex);
        }
        return null;
    }

}
```

Buat file DataQueueExampleTest.java dengan isi seperti ini:

```
/*
 * Copyright 2013 Samuel Franklyn <sfranklyn@gmail.com>.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *      http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
package dataqueue;

import com.ibm.as400.access.Record;
import java.io.UnsupportedEncodingException;
import java.math.BigDecimal;
import java.util.logging.Level;
import java.util.logging.Logger;
import static org.junit.Assert.assertTrue;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.junit.runners.JUnit4;

@RunWith(JUnit4.class)
public class DataQueueExampleTest {

    @Test
    public void dataQueueExampleTest() {
        Logger logger = Logger.getLogger(DataQueueExampleTest.class.getName());
        DataQueueExample dqe = new DataQueueExample();

        logger.info("Write Data Queue");
        dqe.writeDataQueue("SAMUEL FRANKLYN", 681125);

        logger.info("Read Data Queue");
        Record record = dqe.readDataQueue();
        String name;
        try {
            if (record != null) {
                name = (String) record.getField("NAME").toString().trim();
                BigDecimal age = (BigDecimal) record.getField("AGE");
                logger.log(Level.INFO, "NAME :{0} AGE:{1} years",
```

```
                        new Object[]{name, age});
                assertTrue(name.equals("SAMUEL FRANKLYN"));
                assertTrue(age.longValue() == 45);
            }
        } catch (UnsupportedEncodingException ex) {
            logger.log(Level.SEVERE, null, ex);
        }
    }
}
```

Untuk bisa menjalankan program java ini anda harus mencopy dari file config-example.properties di direktori src/main/java/resource ke file config.properties lalu menyesuaikan isinya. Isi dari config-example.properties

```
# Copyright 2013 Samuel Franklyn <sfranklyn@gmail.com>.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#      http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
as400.systemname=pub1.rzkh.de
as400.username=USER
as400.password=PASSWORD
as400.dataqueue1=/QSYS.LIB/SAMUELF1.LIB/Q01.DTAQ
as400.dataqueue2=/QSYS.LIB/SAMUELF1.LIB/Q02.DTAQ
```

Untuk menjalankan tutorial ini masuklah kedalam AS/400 lalu panggil PGDQ.

```
===>CALL DTAQUE
```

Lalu panggil Maven dengan goal test:

```
$ mvn test
```

5 JDBC

Kita bisa mengakses database di AS/400 dengan menggunakan API standard dari Java yaitu JDBC (Java Database Connectivity). Untuk bisa mencoba hal ini kita perlu menciptakan tabel di AS/400. Buat member PERSON di source file QSQLSRC dengan type SQL.

```
===>STRSEU SRCFILE(SAMUELF1/QSQLSRC) SRCMBR(PERSON) TYPE(SQL)
```

Lalu isi PERSON dengan statement SQL berikut:

```
     /* AUTHOR   : SAMUEL FRANKLYN <SFRANKLYN AT GMAIL.COM> */
     /* LOCATION : SAMUELF1/QSQLSRC(PERSON)                 */
     /* DATE     : 2014-02-02                               */
     /*                                                     */
CREATE TABLE SAMUELF1/PERSON
(
  PS_NAME  CHAR(30),
  PS_BDAY  NUMERIC( 6, 0),
  PRIMARY KEY (PS_NAME)
);
```

Ciptakan tabel dengan menjalankan perintah:

```
===>RUNSQLSTM SRCFILE(SAMUELF1/QSQLSRC) SRCMBR(PERSON)
```

Statement SQL dalam PERSON akan dijalankan dan tabel akan tercipta dalam library SAMUELF1.

Saatnya membuat program Java anda dengan menggunakan Maven. Jalankan perintah berikut di command line

```
$ mvn archetype:generate -DinteractiveMode=false \
   -DarchetypeGroupId=org.apache.maven.archetypes \
   -DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.1 \
   -DgroupId=samuelf.web.id -DartifactId=jdbc -Dpackage=jdbc
```

Maven akan membuat file pom.xml, source file App.java dan AppTest.java. Hapuslah source file App.java dan AppTest.java. Sesuaikan isi file pom.xml anda hingga seperti dibawah ini:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
                      http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>samuelf.web.id</groupId>
  <artifactId>jdbc</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>jdbc</name>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>net.sf.jt400</groupId>
      <artifactId>jt400-full</artifactId>
      <version>4.7.0</version>
    </dependency>
  </dependencies>
</project>
```

Buat file JdbcExample.java dengan isi seperti ini:

```
/*
 * Copyright 2013 Samuel Franklyn <sfranklyn@gmail.com>.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *      http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
```

```
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
package jdbc;

import java.io.IOException;
import java.io.InputStream;
import java.math.BigDecimal;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Properties;
import java.util.logging.FileHandler;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 * Java Database Connectivity example
 *
 * @author Samuel Franklyn <sfranklyn@gmail.com>
 */
public class JdbcExample {

    private static final Logger logger = Logger.getLogger(
            JdbcExample.class.getName());
    private final Properties configProperties = new Properties();
    private Connection conn = null;

    /**
     * Load properties file from class path, write log into file and connect to
     * database.
     */
    public JdbcExample() {
        try {
            InputStream inputStream = this.getClass().getClassLoader().
                    getResourceAsStream("config.properties");
            configProperties.load(inputStream);

            FileHandler fileHandler = new FileHandler("log.xml");
            logger.addHandler(fileHandler);

            String driver = configProperties.getProperty("as400.driver");
            String jdbcUrl = configProperties.getProperty("as400.jdbcurl");
            String userName = configProperties.getProperty("as400.username");
            String password = configProperties.getProperty("as400.password");
            logger.info("Connect to AS/400");
            Class.forName(driver);
            conn = DriverManager.getConnection(jdbcUrl, userName, password);
        } catch (IOException | ClassNotFoundException | SQLException ex) {
            logger.log(Level.SEVERE, null, ex);
        }
    }

    /**
     * Insert record into person table
     *
     * @param psName
     * @param psBday
     * @return
```

```
     */
    public int insertRecord(String psName, BigDecimal psBday) {
        String insertRecord = "INSERT INTO PERSON ("
                + "PS_NAME,"
                + "PS_BDAY "
                + ") VALUES ("
                + "?,"
                + "?"
                + ")";
        try {
            PreparedStatement insertStmt = conn.prepareStatement(insertRecord);

            insertStmt.setString(1, psName);
            insertStmt.setBigDecimal(2, psBday);

            return insertStmt.executeUpdate();
        } catch (SQLException ex) {
            logger.log(Level.SEVERE, null, ex);
        }
        return 0;
    }

    /**
     * Select from person table using person name.
     *
     * @param psName
     * @return
     */
    public ResultSet selectRecord(String psName) {
        String selectRecord = "SELECT "
                + "PS_NAME,"
                + "PS_BDAY "
                + "FROM PERSON "
                + "WHERE "
                + "PS_NAME = ?";
        try {
            PreparedStatement selectStmt = conn.prepareStatement(selectRecord);
            selectStmt.setString(1, psName);

            ResultSet resultSet = selectStmt.executeQuery();
            if (resultSet.next()) {
                return resultSet;
            }
        } catch (SQLException ex) {
            logger.log(Level.SEVERE, null, ex);
        }
        return null;
    }

    /**
     * Update person birthday identified by person name
     *
     * @param psName
     * @param psBday
     * @return
     */
    public int updateRecord(String psName, BigDecimal psBday) {
        String updateRecord = "UPDATE PERSON "
                + "SET PS_BDAY = ? "
                + "WHERE "
                + "PS_NAME = ?";
        try {
```

```
            PreparedStatement updateStmt = conn.prepareStatement(updateRecord);

            updateStmt.setBigDecimal(1, psBday);
            updateStmt.setString(2, psName);

            return updateStmt.executeUpdate();
        } catch (SQLException ex) {
            logger.log(Level.SEVERE, null, ex);
        }
        return 0;
    }

    /**
     * Delete record identified by person name.
     *
     * @param psName
     * @return
     */
    public int deleteRecord(String psName) {
        String deleteRecord = "DELETE FROM PERSON "
                + "WHERE "
                + "PS_NAME = ?";
        try {
            PreparedStatement deleteStmt = conn.prepareStatement(deleteRecord);

            deleteStmt.setString(1, psName);

            return deleteStmt.executeUpdate();
        } catch (SQLException ex) {
            logger.log(Level.SEVERE, null, ex);
        }
        return 0;
    }
}
```

Buat file JdbcExampleTest.java dengan isi seperti ini:

```
/*
 * Copyright 2013 Samuel Franklyn <sfranklyn@gmail.com>.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *      http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
package jdbc;

import java.math.BigDecimal;
import java.math.BigInteger;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import static org.junit.Assert.assertTrue;
import org.junit.Test;
```

```
import org.junit.runner.RunWith;
import org.junit.runners.JUnit4;

@RunWith(JUnit4.class)
public class JdbcExampleTest {

    @Test
    public void jdbcExampleTest() {
        Logger logger = Logger.getLogger(JdbcExampleTest.class.getName());
        JdbcExample je = new JdbcExample();

        logger.info("Insert record");
        String psName = "SAMUEL FRANKLYN";
        BigDecimal psBday = new BigDecimal(BigInteger.valueOf(681125), 0);
        assertTrue(je.insertRecord(psName, psBday) > 0);

        logger.info("Select record");
        ResultSet resultSet = je.selectRecord(psName);
        try {
            if (resultSet != null) {
                String name = resultSet.getString(1).trim();
                BigDecimal bday = resultSet.getBigDecimal(2);

                logger.log(Level.INFO, "PS_NAME:{0}", name);
                logger.log(Level.INFO, "PS_BDAY:{0}", bday);

                assertTrue(psName.equals(name));
                assertTrue(bday.longValue() == 681125);
            }
        } catch (SQLException ex) {
            logger.log(Level.SEVERE, null, ex);
        }

        logger.info("Update record");
        psBday = new BigDecimal(BigInteger.valueOf(0), 0);
        assertTrue(je.updateRecord(psName, psBday) > 0);

        logger.info("Delete record");
        je.deleteRecord("SAMUEL FRANKLYN");
    }
}
```

Untuk bisa menjalankan program java ini anda harus mencopy dari file config-example.properties di direktori src/main/java/resource ke file config.properties lalu menyesuaikan isinya. Isi dari config-example.properties

```
# Copyright 2013 Samuel Franklyn <sfranklyn@gmail.com>.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#      http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
as400.driver=com.ibm.as400.access.AS400JDBCDriver
as400.jdbcurl=jdbc:as400://pub1.rzkh.de/samuelf1
as400.username=USER
as400.password=PASSWORD
```

Untuk menjalankan panggil Maven dengan goal test:

```
$ mvn test
```

6 Command Call

Kita dapat memanggil command di AS/400 dengan menggunakan API CommandCall.

Saatnya membuat program Java anda dengan menggunakan Maven. Jalankan perintah berikut di command line

```
$ mvn archetype:generate -DinteractiveMode=false \
   -DarchetypeGroupId=org.apache.maven.archetypes \
   -DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.1 \
   -DgroupId=samuelf.web.id -DartifactId=commandcall -Dpackage=commandcall
```

Maven akan membuat file pom.xml, source file App.java dan AppTest.java. Hapuslah source file App.java dan AppTest.java. Sesuaikan isi file pom.xml anda hingga seperti dibawah ini:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
                      http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>samuelf.web.id</groupId>
  <artifactId>commandcall</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>commandcall</name>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>net.sf.jt400</groupId>
      <artifactId>jt400-full</artifactId>
      <version>4.7.0</version>
    </dependency>
  </dependencies>
</project>
```

Buat file CommandCallExample.java dengan isi seperti ini:

```
/*
 * Copyright 2013 Samuel Franklyn <sfranklyn@gmail.com>.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
```

```
 *
 *      http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
package commandcall;

import com.ibm.as400.access.AS400;
import com.ibm.as400.access.AS400SecurityException;
import com.ibm.as400.access.CommandCall;
import com.ibm.as400.access.ErrorCompletingRequestException;
import java.beans.PropertyVetoException;
import java.io.IOException;
import java.io.InputStream;
import java.util.Properties;
import java.util.logging.FileHandler;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 * Command call example.
 *
 * @author Samuel Franklyn <sfranklyn@gmail.com>
 */
public class CommandCallExample {

    private static final Logger logger = Logger.getLogger(
            CommandCallExample.class.getName());
    private final Properties configProperties = new Properties();

    /**
     * Load properties file from class path and write log into file.
     */
    public CommandCallExample() {
        try {
            InputStream inputStream = this.getClass().getClassLoader().
                    getResourceAsStream("config.properties");
            configProperties.load(inputStream);

            FileHandler fileHandler = new FileHandler("log.xml");
            logger.addHandler(fileHandler);
        } catch (IOException ex) {
            logger.log(Level.SEVERE, null, ex);
        }
    }

    public class CommandResult {

        private boolean result;
        private CommandCall commandCall;

        public boolean isResult() {
            return result;
        }

        public void setResult(boolean result) {
            this.result = result;
        }
```

```
        public CommandCall getCommandCall() {
            return commandCall;
        }

        public void setCommandCall(CommandCall commandCall) {
            this.commandCall = commandCall;
        }
    }

    /**
     * Call a command on AS/400.
     *
     * @param cmd
     * @return
     */
    public CommandResult commandCall(String cmd) {
        String systemName = configProperties.getProperty("as400.systemname");
        String userName = configProperties.getProperty("as400.username");
        String password = configProperties.getProperty("as400.password");

        AS400 as400 = new AS400(systemName, userName, password);
        CommandCall cc = new CommandCall(as400);

        try {
            CommandResult cr = new CommandResult();
            cr.result = cc.run(cmd);
            cr.commandCall = cc;
            return cr;
        } catch (AS400SecurityException | ErrorCompletingRequestException |
                IOException | InterruptedException |
                PropertyVetoException ex) {
            logger.log(Level.SEVERE, null, ex);
        }
        return null;
    }
}
```

Buat file CommandCallExampleTest.java dengan isi seperti ini:

```
/*
 * Copyright 2013 Samuel Franklyn <sfranklyn@gmail.com>.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *      http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
package commandcall;

import com.ibm.as400.access.AS400Message;
import commandcall.CommandCallExample.CommandResult;
import java.util.logging.Level;
import java.util.logging.Logger;
```

```
import static org.junit.Assert.assertTrue;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.junit.runners.JUnit4;

@RunWith(JUnit4.class)
public class CommandCallExampleTest {

    @Test
    public void commandCallExampleTest() {
        Logger logger = Logger.getLogger(
                CommandCallExampleTest.class.getName());

        CommandCallExample cce = new CommandCallExample();
        CommandResult cr = cce.commandCall("CLRPFM FILE(SAMUELF1/PERSON)");

        if (cr.isResult()) {
            logger.info("Success");
        } else {
            logger.info("Fail");
        }

        AS400Message[] messagelist = cr.getCommandCall().getMessageList();
        for (AS400Message messagelist1 : messagelist) {
            logger.log(Level.INFO, "{0}: {1}",
                    new Object[]{messagelist1.getID(),
                        messagelist1.getText()});
        }

        assertTrue(cr.isResult());
    }
}
```

Untuk bisa menjalankan program java ini anda harus mencopy dari file config-example.properties di direktori src/main/java/resource ke file config.properties lalu menyesuaikan isinya. Isi dari config-example.properties

```
# Copyright 2013 Samuel Franklyn <sfranklyn@gmail.com>.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#      http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
as400.systemname=pub1.rzkh.de
as400.username=USER
as400.password=PASSWORD
```

Untuk menjalankan panggil Maven dengan goal test:

```
$ mvn test
```

7 Program Call

Selain memanggil command kita juga bisa memanggil program di AS/400. Program yang dipanggil menerima input paramater dan menghasilkan output parameter.

Buatlah member BRTDAY dalam file QRPGLESRC di library SAMUELF1 dengan tipe RPGLE.

```
===>STRSEU SRCFILE(SAMUELF1/QRPGLESRC) SRCMBR(BRTDAY) TYPE(RPGLE)
```

Lalu masukkan code berikut:

```
      * AUTHOR   : SAMUEL FRANKLYN <SFRANKLYN AT GMAIL.COM>
      * LOCATION : SAMUELF1/QRPGLESRC(BRTDAY)
      * DATE     : 2014-03-06
      *
     D NOW             S               D
     D BDAY            S               D
      *
     C     *ENTRY        PLIST
     C                   PARM                    BDAYS             6 0
     C                   PARM                    AGE               3 0
      *
     C                   EVAL       NOW=%DATE
     C                   EVAL       BDAY=%DATE(BDAYS:*YMD)
     C                   EVAL       AGE=%DIFF(NOW:BDAY:*YEARS)
      *
     C                   EVAL      *INLR=*ON
     C                   RETURN
```

Compile dengan perintah berikut:

```
===> CRTBNDRPG PGM(SAMUELF1/BRTDAY) SRCFILE(SAMUELF1/QRPGLESRC)
```

Saatnya membuat program Java anda dengan menggunakan Maven. Jalankan perintah berikut di command line

```
$ mvn archetype:generate -DinteractiveMode=false \
   -DarchetypeGroupId=org.apache.maven.archetypes \
   -DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.1 \
   -DgroupId=samuelf.web.id -DartifactId=pgmcall -Dpackage=pgmcall
```

Maven akan membuat file pom.xml, source file App.java dan AppTest.java. Hapuslah source file App.java dan AppTest.java. Sesuaikan isi file pom.xml anda hingga seperti dibawah ini:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
                      http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>samuelf.web.id</groupId>
  <artifactId>pgmcall</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>pgmcall</name>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
  </properties>
```

```
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>net.sf.jt400</groupId>
      <artifactId>jt400-full</artifactId>
      <version>4.7.0</version>
    </dependency>
  </dependencies>
</project>
```

Buat file PgmCallExample.java dengan isi seperti ini:

```
/*
 * Copyright 2013 Samuel Franklyn <sfranklyn@gmail.com>.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *      http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
package pgmcall;

import com.ibm.as400.access.AS400;
import com.ibm.as400.access.AS400PackedDecimal;
import com.ibm.as400.access.AS400SecurityException;
import com.ibm.as400.access.ErrorCompletingRequestException;
import com.ibm.as400.access.ObjectDoesNotExistException;
import com.ibm.as400.access.ProgramCall;
import com.ibm.as400.access.ProgramParameter;
import java.io.IOException;
import java.io.InputStream;
import java.util.Properties;
import java.util.logging.FileHandler;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 * Program call example.
 *
 * @author Samuel Franklyn <sfranklyn@gmail.com>
 */
public class PgmCallExample {

    private static final Logger logger = Logger.getLogger(
            PgmCallExample.class.getName());
    private final Properties configProperties = new Properties();

    /**
     * Load properties file from class path and write log into file.
     */
```

```
    public PgmCallExample() {
        try {
            InputStream inputStream = this.getClass().getClassLoader().
                    getResourceAsStream("config.properties");
            configProperties.load(inputStream);

            FileHandler fileHandler = new FileHandler("log.xml");
            logger.addHandler(fileHandler);
        } catch (IOException ex) {
            logger.log(Level.SEVERE, null, ex);
        }
    }

    public class ProgramResult {

        private boolean result;
        private ProgramCall pgmCall;
        private ProgramParameter[] parmList;

        public boolean isResult() {
            return result;
        }

        public void setResult(boolean result) {
            this.result = result;
        }

        public ProgramCall getPgmCall() {
            return pgmCall;
        }

        public void setPgmCall(ProgramCall pgmCall) {
            this.pgmCall = pgmCall;
        }

        public ProgramParameter[] getParmList() {
            return parmList;
        }

        public void setParmList(ProgramParameter[] parmList) {
            this.parmList = parmList;
        }
    }

    /**
     * Call a program on AS/400.
     *
     * @param pgm
     * @return
     */
    public ProgramResult pgmCall(String pgm) {
        String systemName = configProperties.getProperty("as400.systemname");
        String userName = configProperties.getProperty("as400.username");
        String password = configProperties.getProperty("as400.password");

        AS400 as400 = new AS400(systemName, userName, password);

        try {
            AS400PackedDecimal aspd= new AS400PackedDecimal(6, 0);
            ProgramParameter[] parmList = new ProgramParameter[2];
            parmList[0] = new ProgramParameter(aspd.toBytes(681125));
            parmList[1] = new ProgramParameter(3);
```

```
            ProgramCall pc = new ProgramCall(as400, pgm, parmList);
            ProgramResult pr = new ProgramResult();
            pr.result = pc.run();
            pr.pgmCall = pc;
            pr.parmList = parmList;
            return pr;
        } catch (AS400SecurityException | ErrorCompletingRequestException |
                IOException | InterruptedException |
                ObjectDoesNotExistException ex) {
            logger.log(Level.SEVERE, null, ex);
        }
        return null;
    }
}
```

Buat file PgmCallExampleTest.java dengan isi seperti ini:

```
/*
 * Copyright 2013 Samuel Franklyn <sfranklyn@gmail.com>.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *      http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
package pgmcall;

import com.ibm.as400.access.AS400Message;
import com.ibm.as400.access.AS400PackedDecimal;
import java.util.logging.Level;
import java.util.logging.Logger;
import static org.junit.Assert.assertTrue;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.junit.runners.JUnit4;
import pgmcall.PgmCallExample.ProgramResult;

@RunWith(JUnit4.class)
public class PgmCallExampleTest {

    @Test
    public void commandCallExampleTest() {
        Logger logger = Logger.getLogger(
                PgmCallExampleTest.class.getName());

        PgmCallExample pce = new PgmCallExample();
        ProgramResult pr = pce.pgmCall("/QSYS.LIB/SAMUELF1.LIB/BRTDAY.PGM");

        if (pr.isResult()) {
            logger.info("Success");
        } else {
            logger.info("Fail");
        }
```

```
        AS400PackedDecimal aspd= new AS400PackedDecimal(3, 0);
        byte[] ageBytes = pr.getParmList()[1].getOutputData();
        Double age = aspd.toDouble(ageBytes);
        logger.log(Level.INFO, "AGE: {0} years", age);

        AS400Message[] messagelist = pr.getPgmCall().getMessageList();
        for (AS400Message messagelist1 : messagelist) {
            logger.log(Level.INFO, "{0}: {1}",
                    new Object[]{messagelist1.getID(),
                        messagelist1.getText()});
        }

        assertTrue(pr.isResult());
    }
}
```

Untuk bisa menjalankan program java ini anda harus mencopy dari file config-example.properties di direktori src/main/java/resource ke file config.properties lalu menyesuaikan isinya. Isi dari config-example.properties

```
# Copyright 2013 Samuel Franklyn <sfranklyn@gmail.com>.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#      http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
as400.systemname=pub1.rzkh.de
as400.username=USER
as400.password=PASSWORD
```

Untuk menjalankan panggil Maven dengan goal test:

```
$ mvn test
```