



Java Spreadsheet API

Alexander Pinnow

ALEXANDER . PINNOW @ GOOGLEMAIL . COM

01.12.2012

FIX Agora is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

FIX Agora is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

You should have received a copy of the GNU Library General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Table of Contents

Objectives.....	4
About FIX Agora	4
Prerequisites	4
Business Scenario.....	5
Java Implementation.....	5

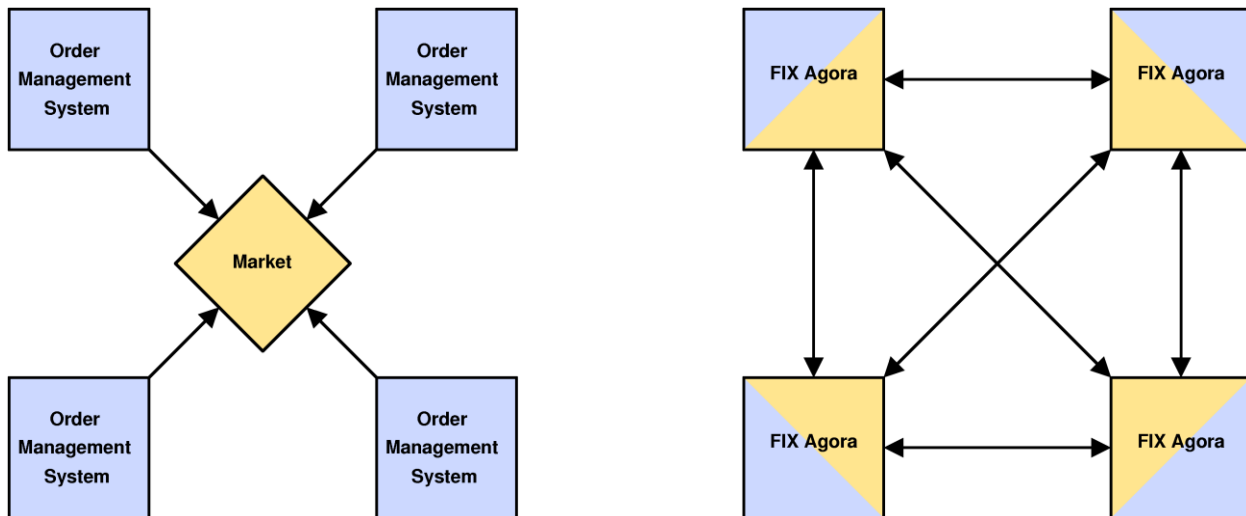
Objectives

This guide gives step-by-step instructions for adding custom spreadsheet functions into FIX Agora. The paper starts with an introduction of the Java Spreadsheet API and describes the implementation of a sample function that calculates the business days since the Lehman crash according to a bank calendar.

About FIX Agora

FIX Agora is a peer-to-peer OTC trading network for financial institutions with an integrated order management system. The application can act as buy side and sell side. FIX Agora supports indicative quoting and processing of order requests. It is equipped with multiuser spreadsheets for price calculations as market maker.

FIX Agora ships with the option for exporting trade reports into Excel® sheets and SAP® Financial Supply Chain Management (requires SAP® JCO). The application is client/server based, modular and extensible.



FIX Agora is working with the Financial Information eXchange (“FIX”) protocol. The FIX protocol is an industry driven standard for the electronic communication of trade related messages, released and maintained by FIX Protocol, Ltd.

FIX Agora is a 100% Java open source implementation running on Windows, Linux/UNIX and Mac OS (client only). It requires Oracle® JRE version of 1.6 or higher.

This software is published under the GNU Lesser General Public License, Version 2.1 (LGPLv2.1).

Prerequisites

Download the version of Eclipse that fits to your operating system from:

<http://www.eclipse.org>

Follow the installation instructions and create a new Java project named *FIXAgoraExtensionTest*.

FIX Agora spreadsheets are working internally with the Apache POI API. If you are not familiar with this API, it is a good idea to take a look at:

<http://poi.apache.org/spreadsheet/user-defined-functions.html>

The bank calendars used in FIX Agora are taken from JQuantlib. A few samples for this API can be found here:

<http://www.jquantlib.com>

The spreadsheet functions are integrated in FIX Agora by dependency injection with a service loader:

<http://docs.oracle.com/javase/6/docs/api/java/util/ServiceLoader.html>

Business Scenario

An additional function is needed in a spreadsheet. This function should calculate the business days since the Lehman crash on September 15, 2008 to a given date according to a specified bank calendar. The syntax of the function is:

LEHMAN(calendar, date)

calendar: the name of a bank calendar in FIX Agora

date: a calendar date after the Lehman crash as sequential serial number

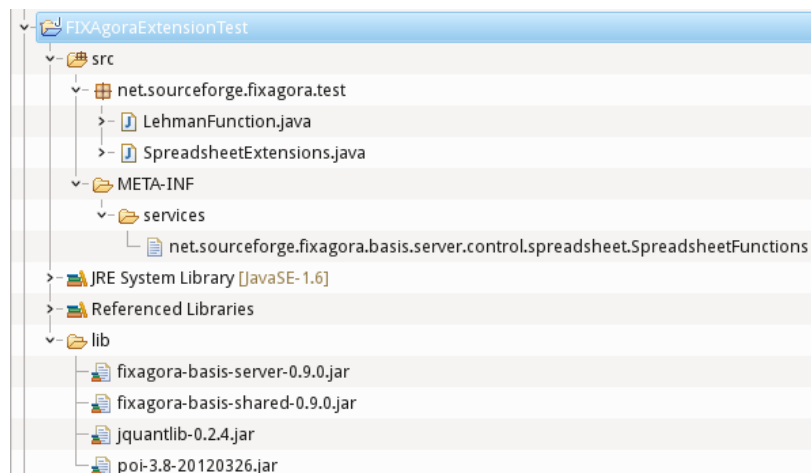
Example:

LEHMAN("United States (Government Bond Market)",DATE(2010,11,15))

would return 541.

Java Implementation

Open Eclipse and create a project structure that looks like this:



The class `LehmanFunction` is implementing the business logic and contains the following code:

```
package net.sourceforge.fixagora.test;

import java.util.Date;
import java.util.Map;

import net.sourceforge.fixagora.basis.server.control.spreadsheet.AbstractSpreadSheetFunction;
import net.sourceforge.fixagora.basis.shared.model.persistence.BankCalendar;

import org.apache.poi.ss.formula.OperationEvaluationContext;
import org.apache.poi.ss.formula.eval.ErrorEval;
import org.apache.poi.ss.formula.eval.EvaluationException;
import org.apache.poi.ss.formula.eval.NumberEval;
import org.apache.poi.ss.formula.eval.OperandResolver;
import org.apache.poi.ss.formula.eval.ValueEval;
import org.apache.poi.ss.usermodel.DateUtil;
import org.jquantlib.time.Calendar;

public class LehmanFunction extends AbstractSpreadSheetFunction {

    @Override
    public String getFunctionName() {

        return "LEHMAN";
    }

    @Override
    public ValueEval evaluate(ValueEval[] args, OperationEvaluationContext ec) {

        if (args.length !=2) {

            return ErrorEval.VALUE_INVALID;
        }

        try {

            Map<String, BankCalendar> map = businessComponentHandler.getBankCalendarMap();

            ValueEval valueEval1 = OperandResolver.getSingleValue(args[0],
                                                                    ec.getRowIndex(),
                                                                    ec.getColumnIndex());

            String bankCalendarName = OperandResolver.coerceValueToString(valueEval1);

            if(bankCalendarName==null)
                return ErrorEval.VALUE_INVALID;

            BankCalendar bankCalendar = map.get(bankCalendarName);

            if(bankCalendar==null)
                return ErrorEval.VALUE_INVALID;

            ValueEval valueEval2 = OperandResolver.getSingleValue(args[1],
                                                                    ec.getRowIndex(),
                                                                    ec.getColumnIndex());

            int daycount = 0;

            org.jquantlib.time.Date lehmanCrash = new org.jquantlib.time.Date(15,9,2008);

            Date javaDate = DateUtil.getJavaDate(OperandResolver.coerceValueToDouble(valueEval2));

            org.jquantlib.time.Date date = new org.jquantlib.time.Date(javaDate);

            Calendar calendar = bankCalendar.getCalendar();
```

```

        while(lehmanCrash.lt(date)) {
            if(calendar.isBusinessDay(lehmanCrash))
                daycount++;

            lehmanCrash.addAssign(1);
        }

        return new NumberEval(daycount);
    }
    catch (Exception e) {
        if (e instanceof EvaluationException)
            return ((EvaluationException) e).getErrorEval();
        else
            return ErrorEval.VALUE_INVALID;
    }
}
}
}

```

Each spreadsheet function must extend the abstract class `AbstractSpreadSheetFunction` and implement the methods `getFunctionName` and `evaluate`. The method `getFunctionName` returns the unique name of the spreadsheet function. The `evaluate` method is doing the calculation.

First, the `LEHMAN` function is checking the number of arguments. `AbstractSpreadSheetFunction` contains a member named `businessComponentHandler`. A map containing all business calendars is retrieved from this member. The function is looking for a `JQuantlib BankCalendar` that matches the first parameter of the `LEHMAN` function. The second parameter of the `LEHMAN` function is firstly converted into a `Java Date` and then converted into a `JQuantlib Date`. Another `JQuantlib Date` named `lehmanCrash` is created and initialized as September 15, 2008. The variable `lehmanCrash` is incremented by a day as long as it is less than the second parameter of the `LEHMAN` function. Each day that represents a business day of the bank calendar is incrementing the variable `daycount`. Finally, the value of `daycount` is converted into an instance of `NumberEval` and returned as calculation result of the function.

The class `SpreadsheetExtensions` is implementing the interface `SpreadsheetFunctions` and looks like this:

```

package net.sourceforge.fixagora.test;

import java.util.ArrayList;
import java.util.List;

import net.sourceforge.fixagora.basis.server.control.spreadsheet.AbstractSpreadSheetFunction;
import net.sourceforge.fixagora.basis.server.control.spreadsheet.SpreadsheetFunctions;

public class SpreadsheetExtensions implements SpreadsheetFunctions {

    @Override
    public List<AbstractSpreadSheetFunction> getSpreadsheetFunctions() {

        List<AbstractSpreadSheetFunction> functions = new ArrayList<AbstractSpreadSheetFunction>();
        functions.add(new LehmanFunction());
        return functions;
    }
}

```

This class is just returning a list of all spreadsheet functions defined in the project. In this case the list only contains an instance of the `LehmanFunction`.

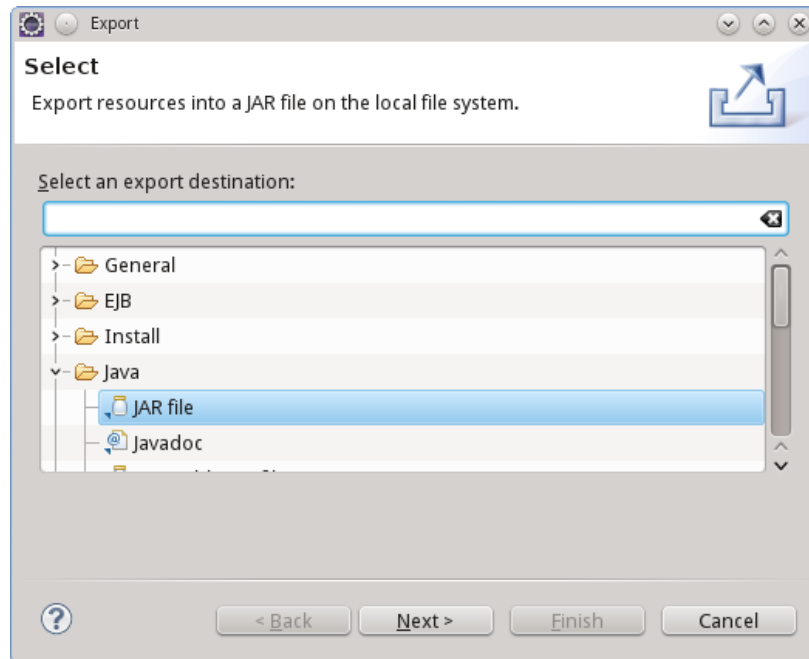
The file `net.sourceforge.fixagora.basis.server.control.spreadsheet.SpreadsheetFunctions` in the folder `services` is used by the service loader to identify the implementation of the java interface

SpreadsheetFunctions. This file must contain the full qualified name of the implementing class. In this case the file is containing this line with the name of the class SpreadsheetExtensions:

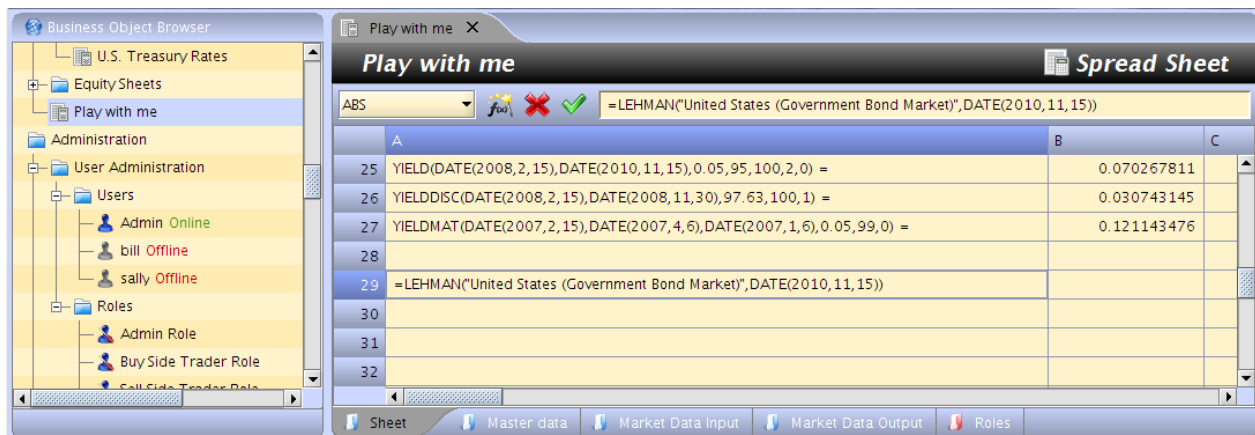
```
net.sourceforge.fixagora.test.SpreadsheetExtensions
```

The lib folder is containing all libraries required by the project. All these jar-archives can be found in the lib folder of the FIX Agora server.

Select the project FIXAgoraExtensionTest in the package explorer of Eclipse and export it by right clicking the project and choosing Export... :



Add the generated jar file to the extlib folder of the FIX Agora server. Now you can start the demo system containing the new spreadsheet function. Open the spreadsheet named Play with me as Admin and add the LEHMANN function.



If your function is requiring additional jar-archives, you can add them into the `extlib` folder of the FIX Agora server. Check that the `lib` folder of the FIX Agora server not already contains the libraries! You can add multiple functions into one project. The name of each function must be unique and should not be a reserved name of an Excel function.