Exercise 24.2 –JMS for the Bank Application

The Setup:

The goal in this exercise is to implement the actual sending of JMS messages in the bank application where we previously only had stubs.

Start by copying the **exercise23_2** project to **exercise24_2**, and then create a separate project called **exercise24_2-receiver**.

Add the Spring context and the JMS related dependencies shown in the previous exercise to both projects. You will probably also need to add exercise24_2 to the dependencies of exercise24_2-receiver.

Remember that the HornetQ JMS server needs to be running in order to do this exercise.

The Exercise:

Create a simple TaxService application in the receiver project which can receive JMS messages and outputs the contents of the message to the console.

Then change the bank application in such a way that whenever a deposit of 10,000 Euros or greater is made, an actual JMS message is sent to the to the TaxService application by a TaxServiceJMSSender.

Integration Exercises

JVM

RMIClient

RMIApplication

+main()

RMI

JVM

bank

Application

+main()

bank.service

<<interface>>
IAccountService

+createAccount()
+deposit()
+withdraw()
+depositEuros()
+withdrawEuros()
+getAccount()
+getAllAccounts()
+transferFunds()

AccountService

+createAccount()
+deposit()
+withdraw()
+depostEuros()
+withdrawEuros()
+getAccount()
+getAllAccounts()
+transferFunds()

DI

<<interface>>
ICurrencyConverter

+euroToDollars()
+dollarsToEuros()

CurrencyConverter

+euroToDollars()
+dollarsToEuros()

bank.domain

<<entity>>
Account

+accountnumber

+deposit()
+withdraw()
+getBalance()
+transferFunds()

<<entity>>
Customer

+name

1

<<entity>>
AccountEntry

+date
+amount
+description
+fromAccountNumber
+fromPersonName

bank.dao

<<interface>>
IAccountDAO

+saveAccount()
+updateAccount()
+loadAccount()
+getAccounts()

AccountDAO

+saveAccount()
+updateAccount()
+loadAccount()
+getAccounts()

AccountDAOSpringHibernate

+saveAccount()
+updateAccount()
+loadAccount()
+getAccounts()

bank.service.aop

StopWatchAdvice

+methodTimer()

JMSLogAdvice

+logJMSMessage()

DAOLogAdvice

+logDAOCall()

bank.jms

<<interface>>
IJMSSender

+sendJMSMessage()

JMSSender

+sendJMSMessage()

TaxServiceJMSSender

+sendJMSMessage()

JMS

JVM

TaxService

+onMessage()

bank.logging

<<interface>>
ILogger

+log()

Logger

+log()