Lab: Gosu for Integration

In this lab, you will implement an exchange rate plugin that retrieves rates from an external system.

Requirements

This lab requires that you use TrainingApp 8.0, ExternalApp 8.0, Guidewire Studio 8.0, and a supported web browser. Start Guidewire Studio for Training App. Debug the Server.

To run ExternalApp, navigate to c:\Guidewire\ExternalApp and double-click the Start ExternalApp shortcut. The command window of ExternalApp will read "\*\*\*\* ContactManager ready \*\*\*\*" when the application is running.

To view, edit, and delete various contacts, log in to TrainingApp as Super User. The default URL for TrainingApp is <http://localhost:8880/ab/ContactManager.do>. The login/password for Super User is su/gw.

1. Exchange Rate plugin

In this exercise, you will create an implementation of the exchange rate plugin that meets specific requirements.

Tasks

1. Create packages and the exchange rate class
2. In Guidewire Studio, create a package called acmelab.ta.plugin.exchangerate.
3. Create a class that conforms to Guidewire recommended naming conventions for predefined plugins.
4. Implement the IExchangeRateSetPlugin
5. For the class, implement the gw.plugin.exchangerate.IExchangeRateSetPlugin interface.
6. Implement any methods.

Gosu code to copy

1. Retrieve exchange rates from the ExternalApp currency exchange rate web service
2. Copy and paste the following code snippet inside the createExchangeRateSet() method stub

*// START THE CUT-AND-PASTE HERE*

*// Create and initialize new exchange rate set*

**var** erSet = **new** ExchangeRateSet()

erSet.Name = **"Lab ExchangeRateSet "** + gw.api.util.DateUtil.currentDate()

erSet.Description = **"Lab ExchangeRateSet"**

erSet.MarketRates = **true**

erSet.EffectiveDate = gw.api.util.DateUtil.currentDate()

*// Create external web service object and set authentication properties*

**var** CurrencyAPI = **new** acme.ta. webservice.currency.exchangeratewsc.currencyapi.CurrencyAPI()

CurrencyAPI.Config.Http.Authentication.Basic.Username = **"su"**

CurrencyAPI.Config.Http.Authentication.Basic.Password = **"gw"**

**var** baseCurrencies = Currency.getTypeKeys(**true**)

**var** priceCurrencies = Currency.getTypeKeys(**true**)

*// For each base/price currency pair, get exchange rate and add it to set*

**for** (currentBaseCurrency **in** baseCurrencies) {

**for** (currentPriceCurrency **in** priceCurrencies) {

**var** newExchangeRate = **new** ExchangeRate()

newExchangeRate.BaseCurrency = currentBaseCurrency

newExchangeRate.PriceCurrency = currentPriceCurrency

newExchangeRate.Rate = CurrencyAPI.getConversionRate(

currentBaseCurrency **as** java.lang.String,

currentPriceCurrency **as** java.lang.String)

erSet.addToExchangeRates(newExchangeRate)

}

}

**return** erSet

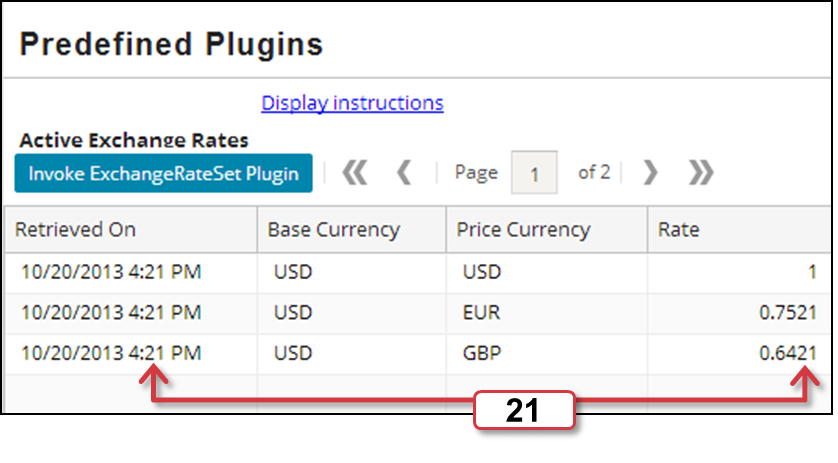
*// END THE CUT-AND-PASTE HERE*

Verification

You will verify the behavior of the Exchange Rate plugin. You will compare the last two digits of the USD to non-USD rates. The rate value should match the Retrieved On time stamp minutes.

1. Invoke the Exchange Rate Set Plugin
2. In Training App, navigate to the Administration tab.
3. In the side menu, open the Training: Plugins page.
4. Click Invoke ExchangeRateSetPlugin
5. Verify the data

Verify that the last two decimal places (thousands and ten thousands) are the same value as the retrieved on minute value.

1. 

|  |  |
| --- | --- |
|  | Stop and ask your instructor to review your completed lab. |