Solution 1: Create an exchange rate plugin

1. **Create a plugin package.**

   a) Right-click on **si.ta** package and select **New ➔ Package**.

   b) Enter **plugin.exchangerate** as the new package name.

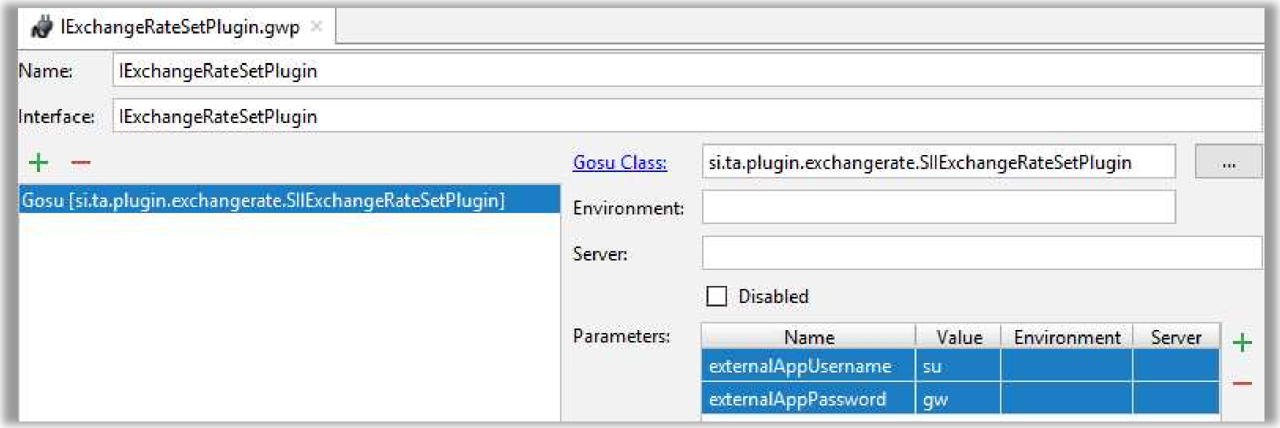
2. **Create a custom rate exchange Gosu class using best practice naming convention.**

   a) Right-click on the **exchangerate** package and select **New ➔ Gosu Class**.

   b) Enter **SIIExchangeRateSetPlugin** as the new Gosu class name.

```
package si.ta.plugin.exchangerate

uses gw.plugin.InitializablePlugin
uses gw.plugin.exchangerate.IExchangeRateSetPlugin

class SIIExchangeRateSetPlugin implements IExchangeRateSetPlugin, InitializablePlugin{
  // authentication parameters
  private static final var USERNAME = "externalAppUsername"
  private static var _username : String
  private static final var PASSWORD = "externalAppPassword"
  private static var _password : String

  override function createExchangeRateSet() : ExchangeRateSet {
    // 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 API properties
    var CurrencyAPI = new trainingapp.webservice.currency.exchangeratewsc.currencyapi.CurrencyAPI()
    CurrencyAPI.Config.Http.Authentication.Basic.Username = _username
    CurrencyAPI.Config.Http.Authentication.Basic.Password = _password
    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
  }

  override property set Parameters(map : Map<Object, Object>) {
    _username = map.get(USERNAME) as String
    _password = map.get(PASSWORD) as String
  }

}
```

3. **Modify the IExchangeRateSetPlugin registry.**


IExchangeRateSetPlugin.gwp

Name: IExchangeRateSetPlugin

Interface: IExchangeRateSetPlugin

Gosu [si.ta.plugin.exchangerate.SIIExchangeRateSetPlugin]

Gosu Class: si.ta.plugin.exchangerate.SIIExchangeRateSetPlugin

Environment:

Server:

□ Disabled

Parameters:

Name	Value	Environment	Server
externalAppUsername	su		
externalAppPassword	gw		

4. **Deploy code changes.**

   a) From the **Studio** menu, **Restart** the server.

5. **Perform verification steps.**