Lesson 8 Introduction to Gosu

As a configuration developer, you have to be familiar with the basic Gosu syntax and language features. In this lab, you will write Gosu code in Gosu Scratchpad to print to the debug console details about specific contacts in TrainingApp.

8.1 Prerequisites

For this exercise, use TrainingApp, Guidewire Studio, and a supported web browser. `http://localhost:8880/ab/ContactManager.do` is the default URL for TrainingApp. To view, edit, and delete various contacts, log in to TrainingApp as Alice Applegate. The login/password for Alice Applegate is `aapplegate/gw`.

8.2 Lab: Coding with Gosu

In this exercise, you find the PublicIDs for various contacts. You will use these PublicIDs to retrieve contact objects in Gosu Scratchpad later in this exercise.

Guidewire API

PublicID

PublicID is a field on almost every Guidewire data model entity that stores a unique identifier value. Unlike the "ID" field, PublicID is a writeable string. It is usually used to identify business objects as they are known to external systems.

8.2.1 Write it down

1. **Question: Navigate to the Summary page of the following company: Burlingame Saab. On the Summary page, what is the Public ID for Burlingame Saab?**

2. **Question: Navigate to the Summary page of the following doctor: Rebecca Stevens. On the Summary page, what is the Public ID for Rebecca Stevens?**

8.2.2 Configuration

In the second part of the exercise, you will open Gosu Scratchpad. Then, you will write and run Gosu code to print to the debug console details about specific contacts in TrainingApp.

1. **Open Gosu Scratchpad in Studio**

   a) Verify that the *Run in Debug Process* icon is available in Gosu Scratchpad.

2. **Write Gosu code**

   a) Use the `trainingapp.base.QueryUtil.findContact()` method to retrieve the contact using the PublicID of the contact.

   b) Write Gosu code that prints to the debug console details about the contact based on the following logic:

   - If the contact found for the PublicID
     - The display name and create date of the contact

       Format: <contactname> was created <date>.

     - If the contact has a Primary Address, the State of the primary address

       Format: Primary address state is <state>.

     - The type of the contact and whether the contact is or is not a strategic partner

       Format: The contact is of the subtype <subtypename> and is [NOT] a strategic partner.

     - If the contact is of the type ABDoctor, print the doctor category and the doctor specialty.

       Format: Doctor category is <category> and doctor specialty is <specialty>.

       (If the doctor category or doctor specialty is null then the following: Doctor category is NOT set and doctor specialty is NOT set.)

     - Else print that the contact is NOT of the type ABDoctor

       Format: Contact is NOT of the type <ABDoctorTypeName>.

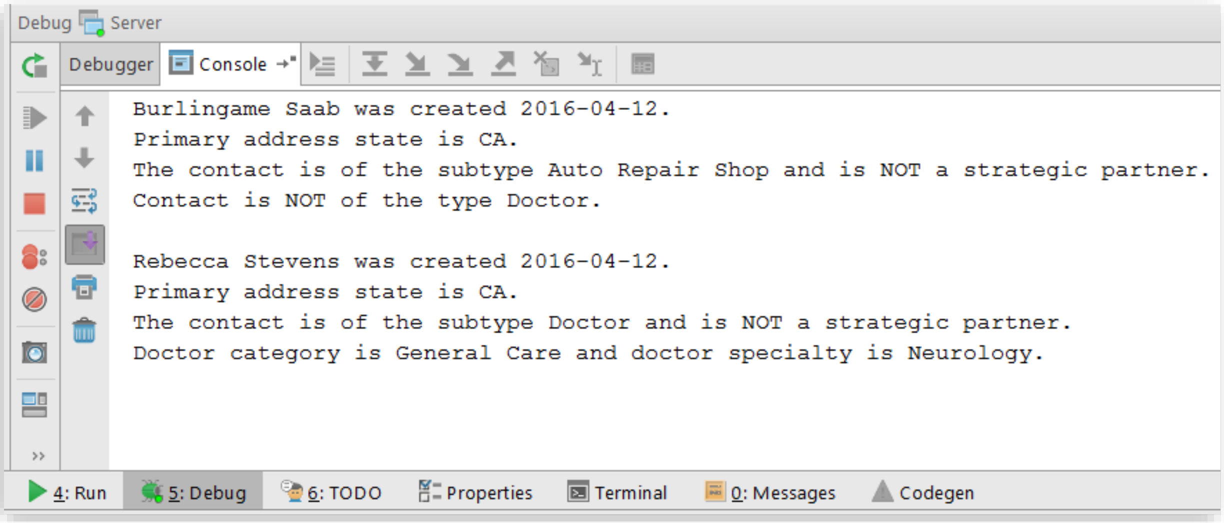
   - If the contact is NOT found for the PublicID
     - **Format**: No contact found for PublicID: <publicID>.

8.2.3 Verification

Activity

Verify the work you have done

1. **Execute your Gosu Code in Gosu Scratchpad**
   a) Using the PublicID for Burlingame Saab, verify the console output.
   b) Using the PublicID for Rebecca Stevens, verify the console output.



```
Burlingame Saab was created 2016-04-12.
Primary address state is CA.
The contact is of the subtype Auto Repair Shop and is NOT a strategic partner.
Contact is NOT of the type Doctor.

Rebecca Stevens was created 2016-04-12.
Primary address state is CA.
The contact is of the subtype Doctor and is NOT a strategic partner.
Doctor category is General Care and doctor specialty is Neurology.
```

Stop

8.3 Lab: Working with arrays

In this exercise, you will write Gosu code in Gosu Scratchpad to retrieve the objects in an entity data model array. Finally, you will print different details of the bank accounts to the server log.

8.3.1 Write it down

1. **Question: Navigate to the Summary page of the following contact: Eric Andy. On the Summary page, what is the Public ID for Eric Andy? Go to the Bank accounts card on the Details page and verify that the contact has 2 bank accounts: ACME Credit Union – checking and savings accounts.**

2. **Question: Navigate to the Summary page of the following contact: William Andy. On the Summary page, what is the Public ID for William Andy? Go to the Bank accounts card on the Details page and verify that the contact has 2 bank accounts: SUCCEED Credit Union – checking and National Bank -**

savings accounts.

8.3.2 Configuration

In the second part of the exercise, you will open Gosu Scratchpad. Then, you will write and run Gosu code to print to the debug console details about specific contacts in TrainingApp.

1. **Open Gosu Scratchpad in Studio**

   a) Verify that the *Run in Debug Process* icon is available in Gosu Scratchpad.

2. **Write Gosu code**

   a) Use the trainingapp.base.QueryUtil.findPerson() method to retrieve each contact.

   b) Write Gosu code that prints bank account details to the server log per contact, based on the following requirements:

   - If the contact found for the PublicID
     - The number of bank accounts

       Format: *<contactname> has <numberofaccounts> bank accounts.*
     - If the contact has bank accounts, a numbered list of the bank accounts starting from 1

       Format: *<contactname>* has:

       <bankname> : *<accounttype>* -- *<accountnumber>*

       <bankname> : *<accounttype>* -- *<accountnumber>*
     - If the contact has at least one account with the bank name “National Bank”

       Format: This contact (*<contactname>*) has an account at *<bankname>*.

       Otherwise:

       Format: This contact (*<contactname>*) does not have an account at *<bankname>*.
   - If the contact is NOT found for the PublicID
     - **Format**: No person found for PublicID: <publicID>

8.3.3 Verification

Activity

Verify the work you have done

1. **Execute you Gosu Code in Gosu Scratchpad**

   a) Verify the console output.