Solution: Introduction to Gosu

In this lab, you will write Gosu code in Gosu Scratchpad to print to the debug console details about specific contacts in TrainingApp.

Requirements

This lab requires that you use TrainingApp 8.0, Guidewire Studio 8.0, and a supported web browser. To view, edit, and delete various contacts, log in to TrainingApp as Alice Applegate. The default URL for TrainingApp is <http://localhost:8880/ab/ContactManager.do>. The login/password for Alice Applegate is aapplegate/gw.

1. Coding in Gosu Scratchpad

In this exercise, you first find the public ids for various contacts. You will use these public ids to retrieve contact objects in Gosu Scratchpad later in this exercise. Then, you will open Gosu Scratchpad. Next, you will write and run Gosu code to print to the debug console details about specific contacts in TrainingApp.

Configuration

1. Log in to TrainingApp
2. Log in as Alice Applegate.
3. Search for Burlingame Saab
4. View the Summary page for Burlingame Saab.

Write it down

On the Summary page, what is the Public ID for Burlingame Saab?

|  |
| --- |
| ab:78 |

1. Search for Rebecca Stevens
2. View the Summary page for Rebecca Stevens.

Write it down

On the Summary page, what is the Public ID for Rebecca Stevens?

|  |
| --- |
| ab:70 |

Tasks

1. Open Guidewire Studio for TrainingApp
2. From Studio, if your server is not already running, start the server using Debug 'Server'.
3. Review the Debug console for errors and verify that the application is running in the Debug console.
4. Open Gosu Scratchpad
5. Verify that the Run in Debug Process icon is available in Gosu Scratchpad.  
   
6. Write Gosu code
7. Use the ta.QueryUtil.findContact() method to retrieve the contact using the public id of the contact.

  
uses ta.QueryUtil  
var publicID = "ab:70"   
var contact = QueryUtil.findContact(publicID)

1. Write Gosu code that prints to the debug console the following details about the contact:
   * The display name and create date of the contact
   * If the contact has a Primary Address, the State of the primary address
   * The type of the contact and whether the contact is or is not a strategic partner
   * If the contact is of the type ABDoctor, print the doctor category and the doctor specialty.

/\*

TA80\_CONF180\_GosuIntro\_Solution.txt

Copy and paste Gosu code in Gosu Scratchpad

Run in Debug 'Server' Process

\*/

uses ta.QueryUtil

var output: String = ""

var publicID: String = "" // execute code with public id values, so specific a string here

var contact = QueryUtil.findContact(publicID) // use the QueryUtil helper methods

if (contact != null) {

// display name and create date

output += contact.DisplayName + " was created " + contact.CreateTime + ". \n"

// primary address state

if (contact.PrimaryAddress.State != null) {

output += "Primary address state is " + contact.PrimaryAddress.State + ". \n"

}

// type of contact and whether a strategic partner or not

output += "The contact is of the type " + contact.Subtype.DisplayName

output += contact.IsStrategicPartner\_Ext ? " and is a strategic partner" : " and is NOT a strategic partner"

output += ". \n"

if (contact typeis entity.ABDoctor) {

output += "Doctor category is " + contact.Category.DisplayName + " and specialty is " + contact.Specialty.DisplayName + ". \n"

}

else {

output += "Contact is not of the type " + entity.ABDoctor.Type.DisplayName + ".\n"

}

} else {

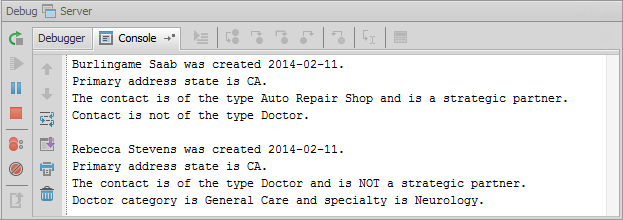
output += "No contact found for public id = " + publicID + "! \n"

}

print(output)

Verification

1. Execute you Gosu Code in Gosu Scratchpad
2. Using the public id for Burlingame Saab, verify the console output.
3. Using the public id for Rebecca Stevens, verify the console output.



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