Solution: Arrays

In this lab, you will write Gosu code to retrieve objects in an entity data model array.

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. Working with Arrays

In this exercise, you will first create bank accounts for two contacts.[[1]](#endnote-1) Then, you will write Gosu code in Gosu Scratchpad to retrieve the objects in an entity data model array. For each task, write Gosu code in Gosu Scratchpad that accomplishes the task.

Configuration

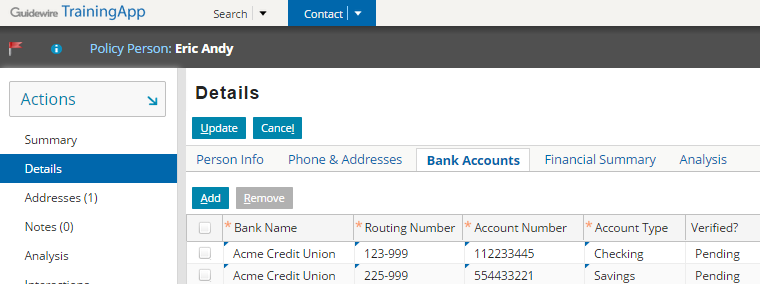
1. Log in to TrainingApp
2. Log in as Alice Applegate.
3. Create bank accounts for Eric Andy
4. Open the Eric Andy contact.

Write it down

On the Summary page, what is the Public ID for Eric Andy?

|  |
| --- |
| ab:98 |

1. Create two bank accounts:
   * Acme Credit Union -- Checking account
   * Acme Credit Union -- Savings account



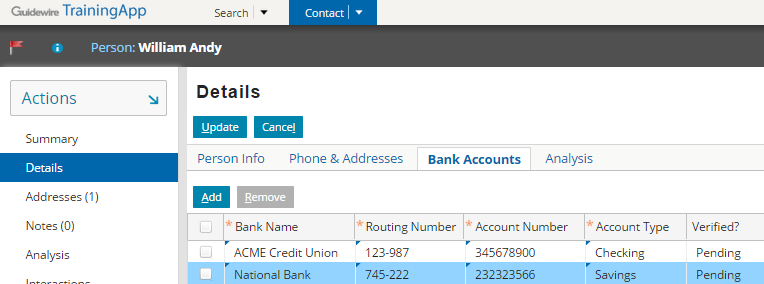
1. Create bank accounts for William Andy
2. Open the William Andy contact.

Write it down

On the Summary page, what is the Public ID for William Andy?

|  |
| --- |
| ab:5 |

1. Create two bank accounts:
   * ACME Credit Union -- Checking account
   * National Bank -- Savings account



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. Output the count of bank accounts per contact
7. Write Gosu code that outputs to the debug console the number of bank accounts for William Andy and Eric Andy.
8. Use the ta.QueryUtil.findPerson() method to retrieve each contact.
9. Output account details per contact
10. For William Andy and Eric Andy, print a numbered list starting from 1 where the list item consists of the bank, the account type, and the account number.
11. Use the ta.QueryUtil.findPerson() method to retrieve each contact.
12. Output account details only if contact has an account with the National Bank
13. If a there is at least one account in the array with the bank name "National Bank", print to the console: "This contact has a National Bank account".
14. Otherwise, print "This contact does not have an account with National Bank".
15. Use the ta.QueryUtil.findPerson() method to retrieve each contact.
16. Use an array method that requires a block as an argument.

uses ta.QueryUtil

//Find the person whose Bank Account to check using the ta.QueryUtil helper methods

var output: String = ""

var personIDs = new String[] {"98", "5"}

output += "\*\*\*\*\* Count the bank accounts per contact \n"

for (personID in personIDs) {

var person = QueryUtil.findPerson(personID)

if (person != null) {

output += person.DisplayName + " has " + person.BankAccounts.length + " bank accounts. \n"

} else {

output += "person not found for " + personID + "\n"

}

}

output += "\*\*\*\*\* Output account details per contact \n"

for (personID in personIDs) {

var person = QueryUtil.findPerson(personID)

if (person.BankAccounts.length > 0) {

output += person.DisplayName + " has " + "\n"

for (bankAccount in person.BankAccounts index i) {

output += i + 1 + ") "

output += bankAccount.BankName + " : " + bankAccount.AccountType + " -- " + bankAccount.AccountNumber + "\n"

}

} else {

output += "person " + personID + " has no bank accounts! \n"

}

}

output += "\*\*\*\*\* Output account details per contact only if with National Bank \n"

for (personID in personIDs) {

var person = QueryUtil.findPerson(personID)

var bankName: String = "National Bank"

if (person.BankAccounts.hasMatch(\account -> account.BankName == bankName)) {

output += "This contact (" +person.DisplayName + ") has an account at " + bankName + "\n"

} else {

output += "This contact (" +person.DisplayName + ") does not have an account at " + bankName + "\n"

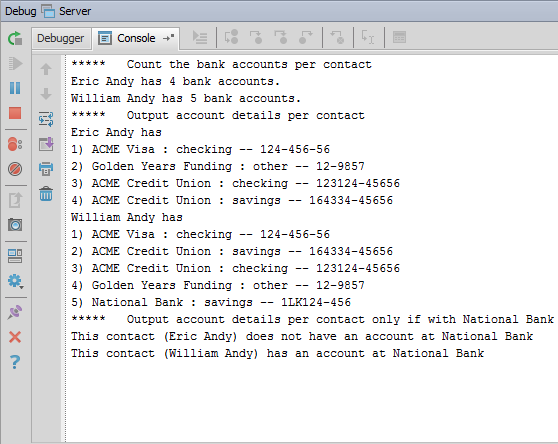
}

}

print(output)

Verification

1. Execute your Gosu Code in Gosu Scratchpad
2. Verify the console output.



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

1. [↑](#endnote-ref-1)