Lab: Publishing Guidewire Web Services

In this lab, you will develop, publish, and test a simple Guidewire web service.

Requirements

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

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. Create a web service

In this exercise, you will create a WS-I web service named CompanyAPI. Your will create a web service that contains three web methods related to the ABCompany entity. One method will verify a tax identification (tax ID) value to see a company is associated with that value. Another method will create a contact note. The third method will generate an Employee Summary.

Tasks

1. Create the web service package and class for CompanyAPI
2. If not already created, in Guidewire Studio, create a package called acmelab.ta.webservices.
3. Create a CompanyAPI class in the appropriate package.
4. Create a method that verifies a company by tax ID
5. Define an input parameter for the tax ID as a String.
6. Identify if a company exists with the given tax ID.
7. For a company that exists with a given tax id, the method must return true as Boolean value.
8. Create a method that creates a ContactNote for a given company
9. Define two input parameters: a String representing a tax ID and a String identifying the body of the note.
10. If the tax ID does not corresponds to a company, the method does nothing.
11. If a company exists with the tax ID, creates a note whose ContactNoteType is "general", whose subject is "External note", and whose body is specified by the input parameter.
12. Add the note to the company's ContactNotes array and save the changes.
13. The method does not return a value.
14. Create a method that takes a tax ID for a company and returns an EmployeeSummary object
15. Define an input parameter for the tax ID as a String.
16. If the tax ID does not corresponds to a company, the method does nothing.
17. If a company exists with the tax ID, return an EmployeeSummary object.
18. The EmployeSummary object must contain the following fields:

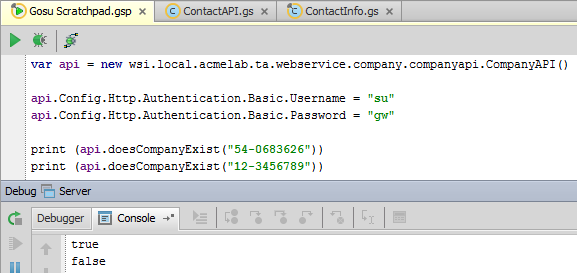
* Number of employees as an integer.
* Employee score as an integer.
* Headquarters location as a string created by concatenating the city, state, and country of the company's primary address.

Verification

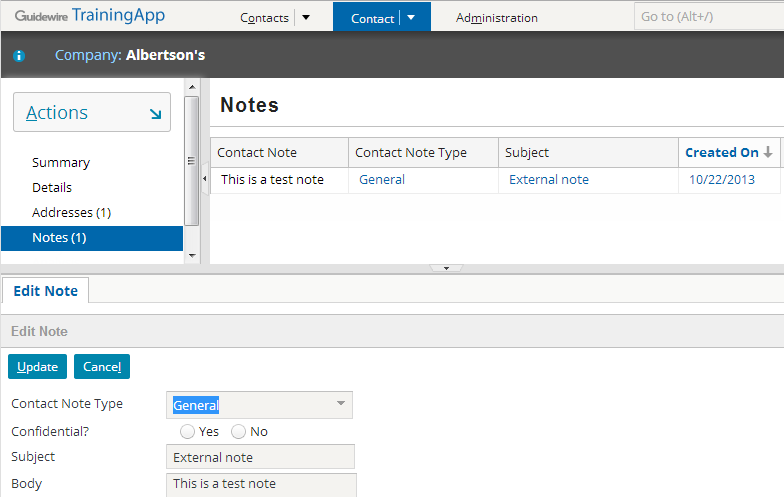
First, create a local test WSDL for your web service. Then, Debug the Server. Next, open Gosu Scratchpad. Write Gosu code to verify the CompanyAPI methods. Run in Debug Process.

1. Verify the results of the company exists method with the following tax identification numbers:
2. Verify that the tax ID of 54-0683626 returns true. This is the tax ID for Albertson's.

Verify that the tax ID of 12-3456789 returns false.

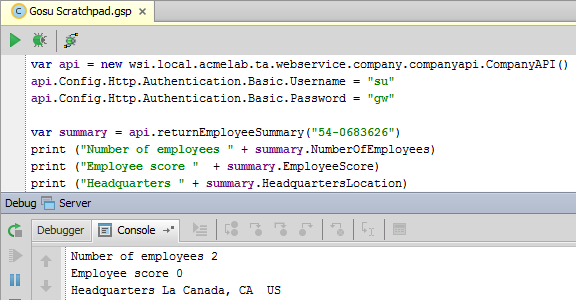


1. Create a contact note
2. Create a note for Albertson's. The tax ID is 54-0683626.
3. Verify that the note appear in Training App for the Company.



1. View the Employee Summary details
2. In TrainingApp, view the details of an ABPerson such as William Andy.
3. For the given person, set the employer to Albertson's.
4. Gosu Scratchpad, retrieve an employee summary for the Albertson's company.
5. Use the tax ID of 54-0683626.
6. Verify the following:

* Number of employees should be at least 1.
* Employee score may or may not be 0.
* Headquarters shows as "La Canada, CA US".



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