Lab: Messaging Acknowledging

In this lab, you will build the final stage of the messaging integration point: acknowledging the message.

Requirements

This lab requires that you use TrainingApp 8.0, ExternalApp 8.0, Guidewire Studio 8.0, and a supported web browser. ExternalApp 8.0 serves as an external system that processes requests for fraud reports.

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.

To run ExternalApp, navigate to c:\Guidewire\ExternalApp and double-click the Start ExternalApp shortcut. The start-up process is complete when the message "\*\*\*\* ContactManager ready \*\*\*\*" appears in the console.

1. Modify the transport plugin

In this exercise, you will modify your transport plugin to respond to returned values with specific behaviors.

Configuration

1. Skip all messages
2. In TrainingApp, skip all messages.   
   Administration Tab 🡪 Event Messages
3. Clear the MessageHistory table
4. Delete all message histories.

Administration Tab 🡪 Training: Messages 🡪MessageHistory Table

Tasks

The following table describes the meaning of the return values of the checkForFraudReport() web services method:

|  |  |
| --- | --- |
| Integer | Meaning |
| 1 | Request processed, no fraud report found |
| 2 | Request processed, fraud report found! |
| 3 | Request contains improperly formatted string payload |
| 4 | Request contains malformed XML payload |
| 5 | Request could not be processed (database unavailable) |
| 6 | Request could not be processed (user authentication) |
| 9 | An unknown error occurred |

1. Modify your transport plugin to respond to the specific return values as follows:

a) Use the trainingapp.demo.messageack.MessageAckUtil class methods to simplify your code.

|  |  |
| --- | --- |
| Integer | Behavior |
| 1 | a) Acknowledge the message.  b) Create a contact note identifying a fraud check was completed.  Set the type to "general".   Set subject to "No fraud report found"  Set the body to "Fraud report request processed, no fraud report found"   Attach the note to the associated contact. |
| 2 | a) Acknowledge the message.  b) Create a contact note identifying a fraud check was completed.   Set the type to "general".   Set subject to "Fraud report found".  Set the body to "Fraud report request processed, fraud report found".   Attach the note to the associated contact.  c) Create a flag entry identifying that a fraud report was found.   Set the reason to "fraudulent\_activity".  Set the flag date to the current date.  Attach the flag entry to the associated contact. |
| 3,4,5,6,9 | a) Report a message error using the appropriate category. |

Verification

For this lab, you do not need to create test cases to check for improperly formatted strings and malformed XML. You already verified both of these test cases in the Sending Messages lab.

To verify your work, you will create several contacts in TrainingApp. For each contact, verify the behavior according to the tasks specifications.

The checkForFraudReport() API returns codes using the following logic:

|  |  |  |
| --- | --- | --- |
| Tax ID  starts with... | Return Code Integer | Meaning |
| 1 | 1 | Request processed, no fraud report found |
| 2 | 2 | Request processed, fraud report found! |
| 5 | 5 | Request could not be processed (network resources unavailable) |
| 6 | 6 | Request could not be processed (a permanent error occurred) |
| any other digit | 1 or 2 | Request processed. ExternalApp finds a fraud report half of the time and for the other half, there is no report found. The result is random. |

1. Tax ID starting with 1
2. Log in to TrainingApp as Super User (su/gw).
3. Create a new ABPerson.
4. Specify a Tax ID starting with 1.
5. Verify the following response behavior:
6. The ABContact has a ContactNote identifying a completed fraud check.
7. The ABContact has no flag entry pertaining to fraud.
8. The MessageHistory table shows for the message a status of "Acknowledged (10)".
9. Tax ID starting with 2
10. Create a new ABPerson.
11. Specify a Tax ID starting with 2.
12. Verify the following response behavior:
13. The ABContact has a ContactNote identifying a completed fraud check.
14. The ABContact has a flag entry pertaining to fraud.
15. The MessageHistory table shows for the message a status of "Acknowledged (10)".
16. Tax ID starting with 5
17. Create a new ABPerson.
18. Specify a Tax ID starting with 5.
19. Verify the following response behavior:
20. The Message table shows for the message a status of "Retryable error (4)".
21. The MessageHistory table shows messages with a status of "Error retried (12)".
22. Tax ID starting with 6
23. Create a new ABPerson.
24. Specify a Tax ID starting with 6.
25. Verify the following response behavior:
26. The MessageHistory table shows messages with a status of "Error retried (12)".

Restoring TrainingApp

After you have verified the behavior, you should TrainingApp may continue to try to send messages in the message table. If ExternalApp is not running, stack trace errors will print to the console.

The resulting stack traces may be disruptive to your future lab work.

To avoid the possible disruptions, skip any Fraud Report messages that remain in the Message table.

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