Lesson 7 Integration Views

This exercise requires that you use **TrainingApp**, **Guidewire Studio**, and a supported web browser. Start **Guidewire Studio for TrainingApp**. Start the server as **Debug** 'Server'.

The default URL for **TrainingApp** is: http://localhost:8880/ab/ContactManager.do. Log in to **TrainingApp** as Alice Applegate User whose login/password is **aapplegate/gw**.

Exercise 1: Create a Contact Integration View

Exercise

Succeed Insurance has an integration point that needs to create JSON output based on the existing **TrainingApp** contact Integration View. They want to add new fields and the ability to export only full name and tax id.

7.1.1 Requirements

Spec 1 Extend the existing **TrainingApp** Integration View located at integration.schemas.trn.ta package.

Spec 2 Add the following new fields to the existing output:

- AssignedUser
  - Name
- BankAccounts
  - BankName
  - BankAccountType
- MaritalStatus
- SenderRefID
- TaxID

Spec 3 Add a filter called **fraud_check** that exports the following fields:

- Name
- Tax ID
- SenderRefID

Important!

The **fraud_check** filter is used in the Creating Messages exercise.

End of important information.

Spec 4 Add a filter called **contact_api** that exports the following fields:

- AssignedUser
  - Name
- BankAccounts
  - BankAccountType
  - BankName
- Name
- PrimaryAddress
  - AddressLine1
  - AddressType
  - City
  - PostalCode
  - State
- TaxID

Important!

The **contact_api** filter is used in the RESTful Web Services exercise.

End of important information.

7.1.2 Tasks

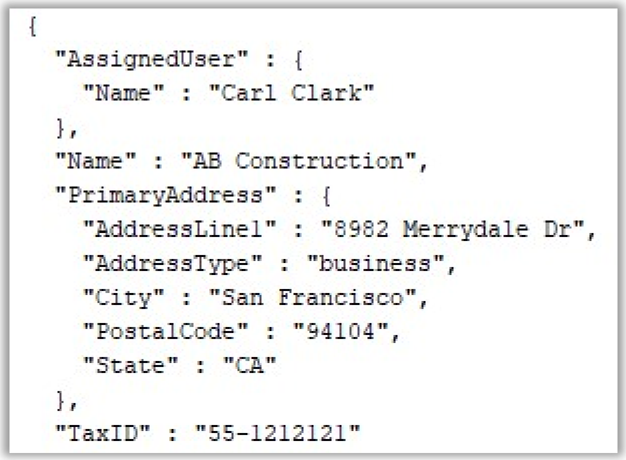
1. **Extend the integration.schemas.trn.ta.contact-1.0.schema.json file and add new fields.**
2. **Create an Enhancement for the class KeyableBean. Add a property getter called SenderRefIDPlaceholder_Ext which returns “@@SenderRefID@@”.**
3. **Extend the integration.mappings.trn.ta.contact-1.0.mapping.json file and add new fields. Map the SenderRefID field using ABContact.SenderRefIDPlaceholder_Ext.**
4. **Create the new filters.**
5. **Generate wrapper classes.**
6. **Deploy code changes.**
7. **Perform verification steps.**

7.1.3 7.1.3 Verification steps

1. **Generate debug console output using Gosu Scratchpad.**

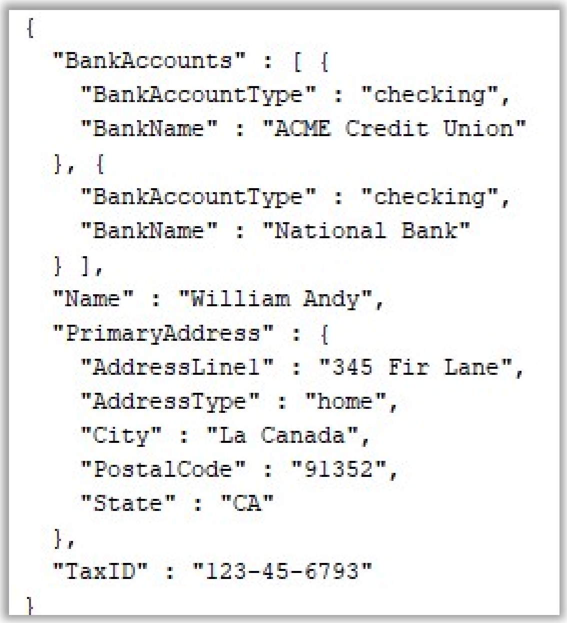
   a) In Studio, open Gosu Scratchpad by clicking **Tools** ➔ **Gosu Scratchpad**.

2. **Write code that will generate JSON output for ABConstruction (PublicID = absample:3) using filter contact_api.**



```
{
  "AssignedUser" : {
    "Name" : "Carl Clark"
  },
  "Name" : "AB Construction",
  "PrimaryAddress" : {
    "AddressLine1" : "8982 Merrydale Dr",
    "AddressType" : "business",
    "City" : "San Francisco",
    "PostalCode" : "94104",
    "State" : "CA"
  },
  "TaxID" : "55-1212121"
```

3. **Write code that will generate JSON output for William Andy (PublicID = ab:5) using filter contact_api.**



```
{
  "BankAccounts" : [ {
    "BankAccountType" : "checking",
    "BankName" : "ACME Credit Union"
  }, {
    "BankAccountType" : "checking",
    "BankName" : "National Bank"
  } ],
  "Name" : "William Andy",
  "PrimaryAddress" : {
    "AddressLine1" : "345 Fir Lane",
    "AddressType" : "home",
    "City" : "La Canada",
    "PostalCode" : "91352",
    "State" : "CA"
  },
  "TaxID" : "123-45-6793"
}
```

4. **Write code that will generate JSON output for William Andy (PublicID = ab:5) using filter fraud_check.**