Lesson 8 RESTful Web Services

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 REST API

Exercise

Succeed Insurance wants to create a new contact API that exposes contact information defined in the Integration View created in the Integration View exercise.

8.1.1 Requirements

Spec 1 Define a contact API that exposes contact detail information defined in the Integration View and uses the contact_api filter.

Spec 2 The basePath must be /si/contact/v1.

Spec 3 Define an API handler class whose method is called getContactDetailInformation.

Spec 4 Make the API available to external resources.

8.1.2 Tasks

1. **Define the API schema.**
2. **Define the API handler class.**
3. **Publish the API.**
4. **Deploy code changes.**
5. **Perform verification steps.**

8.1.3 Verification steps

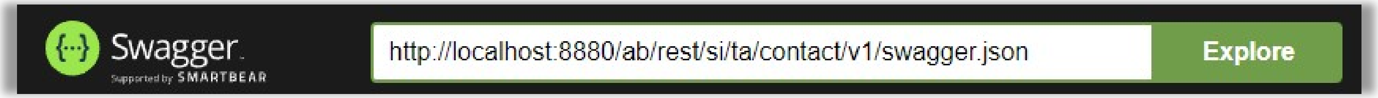
1. **Use swagger-ui distribution to test.**

   a) Open a browser and enter the following URL to launch swagger-ui:

   http://localhost:8880/ab/resources/swagger-ui

   b) Manually change the endpoint to the new API and click **Explore**.

   http://localhost:8880/ab/rest/si/ta/contact/v1/swagger.json



http://localhost:8880/ab/rest/si/ta/contact/v1/swagger.json

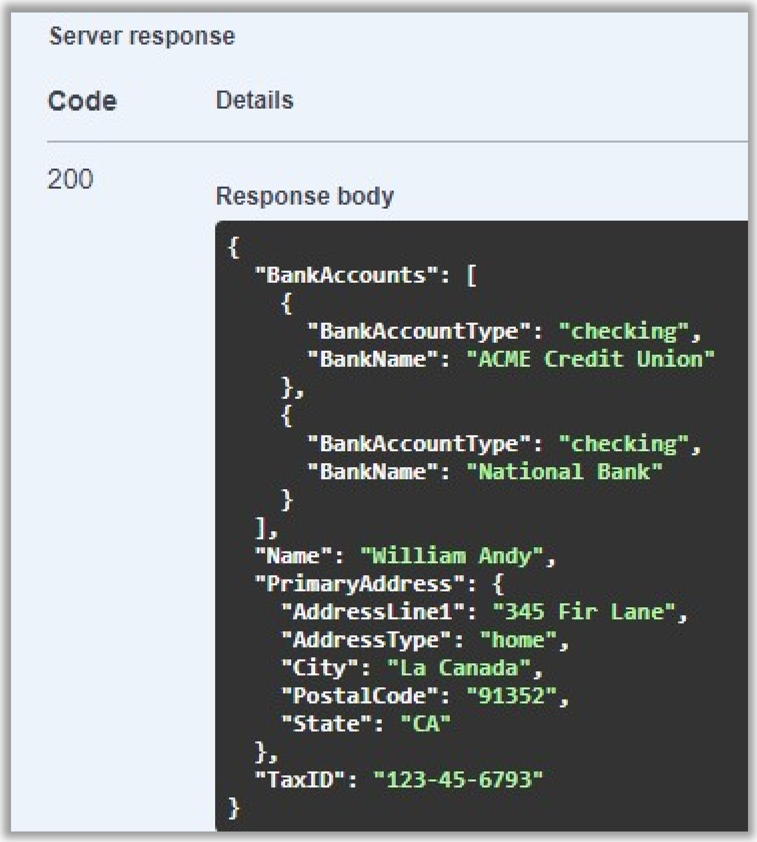
c) Authorize the external resource.

- Click the **Authorize** button.
- Enter **TrainingApp** credentials:
  - Username: su
  - Password: gw
- Click **Authorize** button and close window.

d) Test GET operation for **/contacts/{contactId}**.

- Click **GET** button.
- Click **Try it out** button.
- Enter **ab:5** as the **contactId**.
- Click **Execute**.

e) Server response code is **200** along with contact detail information.



Server response

Code	Details
200	Response body

```
{
  "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"
}
```

Solution 1: Create a Contact REST API

1. **Define the API schema.**

   a) Create a new package.

   - Right-click on **integration** package and click **New ➔ Package**.
   - Enter **apis.si.ta** as the new package name.

   b) Create a new API schema file.

   - Right-click on **apis.si.ta** package and click **New ➔ File**.
   - Enter **contact-1.0.swagger.yaml** as the new file name.

   c) Add schema header information and new paths.

```
swagger: '2.0'
info:
  version: '1.0'
  title: "Contact API"
  description: "Contact API"
basePath: /si/ta/contact/v1
x-gw-schema-import:
 contact: si.ta.contact-1.0
x-gw-apihandlers:
- si.ta.restapi.ContactAPIHandler
produces:
- application/json
consumes:
- application/json
paths:
  /contacts/{contactId}:
    get:
      summary: "Retrieves a contact details"
      operationId: getContactDetailInformation
      parameters:
      - $ref: "#/parameters/contactId"
      responses:
        '200':
          description: "Contact details returned"
          schema:
            $ref: "contact#/definitions/ContactDetails"
parameters:
  contactId:
    name: contactId
    in: path
    type: string
    required: true
```

2. **Define the API handler class.**

   a) Create a new package.

   - Right-click on **gsrc** package and click **New ➔ Package**.

- Enter **si.ta.restapi** as the new package name.

b) Create a new Gosu class.

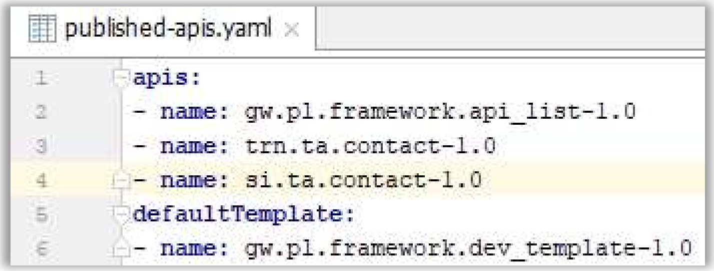
- Right-click on **restapi** package and click **New ➔ Gosu Class**.
- Enter **ContactAPIHandler** as the new Gosu class name.

c) Configure the API method.

```
package si.ta.restapi

uses gw.api.database.Query
uses gw.api.database.Relop
uses gw.api.json.JsonConfigAccess
uses gw.api.json.mapping.JsonMappingOptions
uses gw.api.json.mapping.TransformResult

class ContactAPIHandler {

  function getContactDetailInformation(contactId : String) : TransformResult {
    // Query for contact
    var user = findContactById(contactId)
    // Create JsonMapper object
    var jsonMapper = JsonConfigAccess.getMapper("si.ta.contact-1.0", "ContactDetails")
    var mappingOpts = new JsonMappingOptions().withFilter("si.ta.contact_api-1.0")
    return jsonMapper.transformObject(user, mappingOpts)
  }

  private function findContactById(id : String) : ABContact {
    // Query for contact
    var queryObj = Query.make(ABContact)
    queryObj.compare(ABContact#PublicID, Relop.Equals, id)
    var targetObj = queryObj.select().AtMostOneRow
    return targetObj
  }

}
```

3. Publish the API.

a) Update the **published-apis.yaml** file with the fully qualified name of the new API.



```
apis:
  - name: gw.pl.framework.api_list-1.0
  - name: trn.ta.contact-1.0
  - name: si.ta.contact-1.0
defaultTemplate:
  - name: gw.pl.framework.dev_template-1.0
```

4. Deploy code changes.

a) From the **Studio** menu, **Restart the server**.

5. Perform verification steps.