Application of Microservice Architecture on B2B Processes

(IBM Watson Customer Engagement)

by

Vipin Dhonkaria

(Roll No. 2015274)

Supervisor(s):

External

Mr. Atul A. Gohad

(IBM ISL, Bangalore)

Internal

Dr. Manish Kumar Bajpai

(PDPM IIITDM Jabalpur)

Computer Science and Engineering

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, DESIGN AND MANUFACTURING JABALPUR

(1st June 2018 – 21st August 2018)

(Midterm Review)

Acknowledgement

The internship opportunity I had with IBM was a great chance for learning and professional development. Therefore, I consider myself as a very lucky individual as I was provided with an opportunity to be a part of it. I am also grateful for having a chance to meet so many wonderful people and professionals who led me though this internship period.

Bearing in mind previous I am using this opportunity to express my deepest gratitude and special thanks to the Mr. Atul A. Gohad(External Supervisor), who in spite of being extraordinarily busy with his duties, took time out to hear, guide and keep me on the correct path and allowing me to carry out my project at their esteemed organization and extending during the training.

I express my deepest thanks to Mrs. Rashmi Acharya, Advisory Software Developer for taking part in useful decision & giving necessary advices and guidance and arranged all facilities to make things easier. I choose this moment to acknowledge her contribution gratefully. My thanks and appreciation also goes to my colleague Mr. Arpit Jain in developing the project and helped me out with his abilities.

I would like to express my deepest gratitude and sincere thanks to Dr. Manish Kumar Bajpai(Internal Supervisor) who supported and helped me in all difficulties I have faced during my internship.

I perceive as this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement, in order to attain desired career objectives. Hope to continue cooperation with all of you in the future.

Date: 21 August, 2018

Warmest Regards

Vipin Dhonkaria

Table of Contents:

Introduction

The International Business Machines Corporation (IBM) is an American multinational technology company headquartered in Armonk, New York, United States, with operations in over 170 countries. IBM manufactures and markets computer hardware, middleware and software, and provides hosting and consulting services in areas ranging from mainframe computers to nanotechnology.

IBM aims to bring Businesses closer and smarter than ever with the help of their state of the art enterprise software product called B2B Sterling Integrator. IBM B2B Integrator helps companies integrate complex B2B (Business to Business) / EDI (Electronic Data Exchange) processes with their partner communities. IBM aims to transform the B2B Sterling product into Microservice architecture.

1.1 Brief Overview

Until now, I have completed several tasks pertaining to the B2Bi Sterling Integrator. Proceeding in stages, Firstly, I researched about the microservice architecture and Business-to-Business processes. Secondly, understood the basic Business process modelling language(BPML). BPML is an XML standard to write business process. To experiment further with the microservice software development architecture, I developed two Microservices – One on node.js (Backend Service) and other using Java JSP (Frontend Service). I gave a presentation regarding I established a REST API connection channel between both services. Lastly, I got myself acquainted with the Sterling Integrator product and deployed a Sample War file on this platform. I explored the code structure to build services and deploy onto this platform. Finally I deployed B2BiAPIs war file.

After deploying the B2BiAPIs war file on sterling integrator, I have created Rest Client Service to perform CRUD(Create, Read, Update, Delete) operations on these APIs using business processes. It also allows to perform multiple CRUD operations in a single business process. Clients can make multiple service instances and can call different services to perform different operations in a single business process.

1.2 Literature Survey

I have researched a lot for developing and deploying the Rest Client Service on Sterling Integrator. Firstly I have to understand how to create a business process for which I have gone through SI52_BP_Modeling_book which makes me understand all concepts of business process and business process modelling language(BPML).
Then I read several IBM developer blogs to get acquainted with Rest Concepts and previous researches done on this topic.

- REST API Proof of Concept - Automate Sterling FileGateway Community and Trading Partner creation
- REST API Client - upload message to Sterling B2B Integrator Mailbox
- SFG_B2Bi REST API Example Clients

1.3 Internship Plan

In these 3 months of my internship, I have completed several tasks. Firstly, I gave presentation and demo on Microservice Architecture, then I deployed B2BiAPIs war file on the sterling integrator and finally I created Rest Client Service to perform CRUD operations on these B2BiAPIs.

My plan of next 3 months is to deploy User Account Service and Trading Partner Service APIs using Spring Boot. Currently all the APIs are deployed using java, but due to large amount of requests clients are facing problem in accessing APIs taking longer execution time.
After completing it, if I will have sufficient time then I will be working on other projects too.

1. Report on the Present Investigation

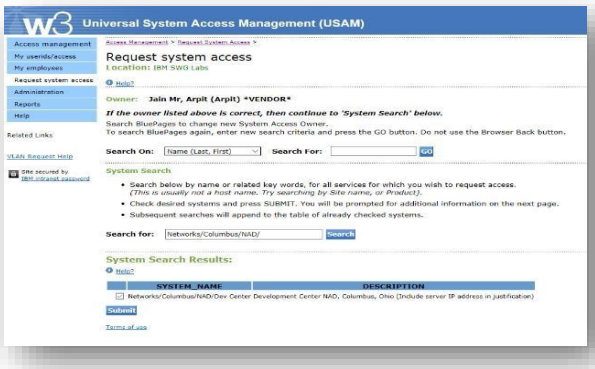
(Progress until the Midterm review)

2.1 Task 1: Researching about Microservice Architecture and B2B Processes

During this duration, I researched and studied about these topics and gave a presentation meeting to the entire B2B Team regarding my study and findings.

2.2 Task 2: Getting access to the B2B sterling Integrator server

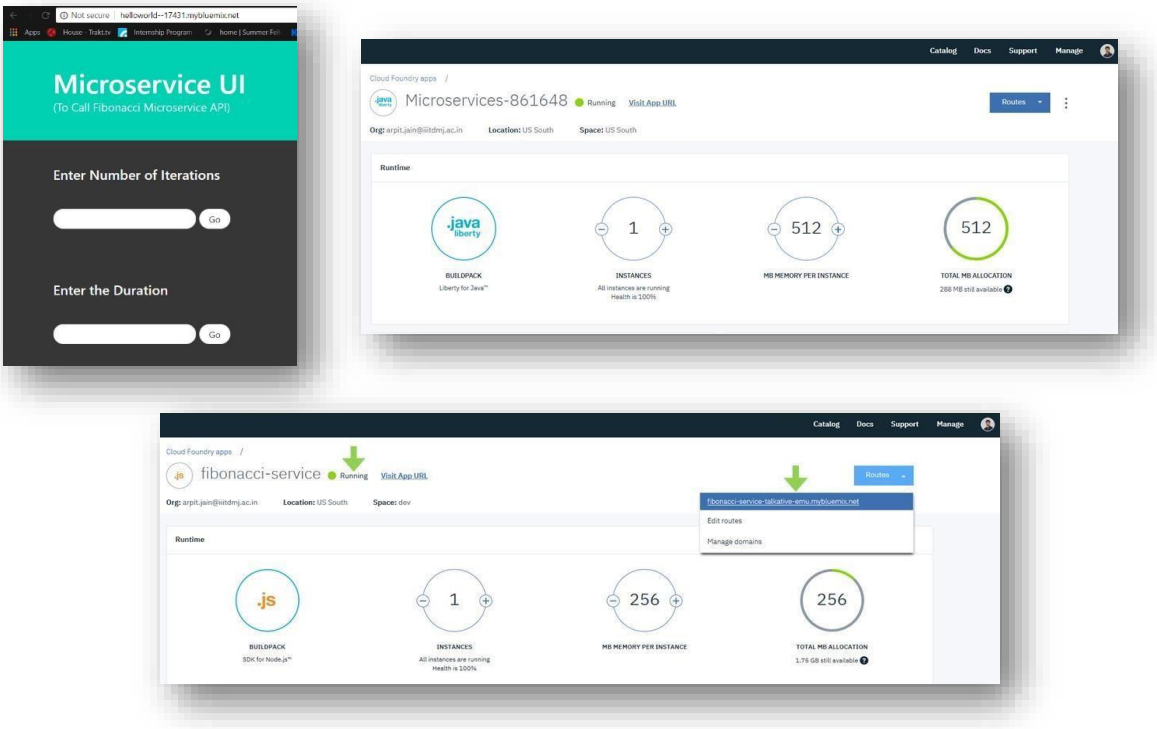
To work on the B2B Sterling Integrator product, access to the servers of IBM on which the product is currently deployed and is running live was to be acquired. The access to the servers and the B2B Project requested and was granted within a week.



2.3 Task 3: Microservice Demo application

To demonstrate the usage of Microservices architecture for better application design, I implemented a two-microservice application. One microservice acted as a Backend API Fibonacci microservice which was called by another microservice which acted as an UI.

Link of the Demo : http://helloworld--17431.mybluemix.net/



2.4 Task 4: Deploying a sample WAR file on the B2B SI server

B2B Sterling Integrator works based on the Business processes. These business processes are used to automate the operation of the services in the business environment. There is a basic set of base services which form the core of the SI product. These services are written in Java and are prepackaged altogether in a WAR archive format. To execute and test these services, the B2Bi war file is to be deployed on the SI server. Before deploying the actual B2Bi war file, I deployed a sample WAR packaged JSP application to test out the dependencies.

2.5 Task 5: Deploying the B2BiAPIs WAR file on the B2B SI server

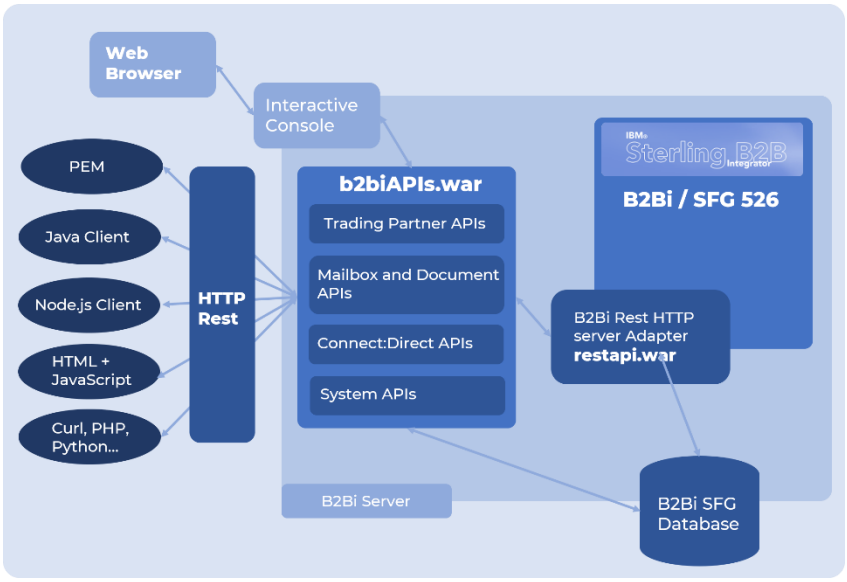
RESTful web services stand for **RE**presentational **S**tate **T**ransfer. Here, each URL is a representation of some object. Restful systems interface with external systems as web resources. Resources are distinctively identified by URIs (Uniform Resource Identifiers). **e.g. /user/12**. Also, RESTful web services support formats like JSON, XML.

REST Terminology

Sterling B2B Integrator 5.2.6.1 introduced a new REST API interface to provide support for the recently released Partner Engagement Manager (formerly Multi-Enterprise Relationship Management (MRM)). The REST API provides a more efficient mechanism for onboarding trading partners.

B2Bi REST implementation

It was introduced in B2Bi 526 and it allows user to programmatically Create, Read, Update and Delete resources in B2Bi. JSON and XML supported as input and output formats. It provides support for Partner Engagement Manager PEM. It is a more efficient mechanism for on boarding trading partners compared to Xapi (Prior to SFG 526 and REST API).



B2Bi REST API Resources

CA Digital Certificate Services
CodeList Services
CodeListCode Services
Community Services
CustomProtocol Services
Digital Certificate Duplicate Check Services
Document Services
External User Services
FgArrivedFile Services
FgRoute Services
Generated Password Services
JDBC Service Tracking Services
Mailbox Services
Mailbox Content Services
Mailbox Message Services
Message Batch Services
Partner Group Services
Permission Services
PGPKey Services
PGP Server Profile Services
Routing Channel Services
Routing Channel Duplicate Check Services
Routing Rule Services
Schedule Services
ServiceDefinition Services
ServiceInstance Services
SSH Authorized User Key Services
SSH Duplicate Check Services
SSH Known Host Key Services
SSH Remote Profile Services
SSH User Identity Key Services
Sterling Connect Direct Netmap Services
Sterling Connect Direct Netmap Xref Services
Sterling Connect Direct Node Services
Sterling Connect Direct Node Duplicate Check Services
Sterling Connect Direct XREF Duplicate Check Services
System Digital Certificate Services
TestSFGDeliveryStatus Services
Test Trading Partner Services
Trading Partner Services
Trusted Digital Certificate Services
User Account Services
User Group Services
UserVirtualRoot Services
Workflow Services
WorkFlowMonitor Services

After deploying the war file, now we can access the APIs by logging in to the sterling integrator and providing the right port and URI.

URI: URI provided in the step 10. Ex: /B2BiAPIs i.e. if we want to perform Read request on CA Digital Certificate Services then we can perform it by giving

URI: /B2BiAPIs/CAdigitalcertifcateservices

2.6 Task 6: Researching about Business Processes and Service Instances

Service Instance

The service instance defines an instance of the service. The serviceinstances.xml file contains the service instance. It has information like the display name and the target of the service. Because you can manually define a service through the user interface, this file is optional.

```
<?xml version="1.0" encoding="UTF-8"?>
<services>
<service parentdefname="genericrest" name="genericrest"
      displayname="genericrest"
      description="Acts as a generic rest API client to the REST APIs"
      targetenv="all"
      activestatus="1"
      systemservice="0"
      parentdefid="-1">
</service>
</services>
```

Business process

A business process is any goal-driven, ordered flow of activities that accomplishes a business objective. In Sterling B2B Integrator, business process refers to the automated implementation of business objectives. Invoicing, order fulfilment, and updating employee information are all examples of business processes

2.7 Task 7: Developing Rest Client Service for the B2B APIs

Building services and adapters requires java programming knowledge and skills, as well as a solid understanding of the sterling integrator system. The following list includes the types of prerequisite knowledge and experience necessary for successfully creating custom services and adapters:

- Java (J2SE) programming knowledge
- General operational and architectural knowledge of the SI system.
- Eclipse IDE programming experience.
- XML Understanding specifically for writing custom Business process definitions.
- API endpoint usage and general understanding of JSON/XML data resources.
- Understanding of RESTful services.
- Multi-threaded programming experience in Java
- Exception Handling in Java
- Ability to write custom APIs and user exits

So, let us summarize what you need to write a service in ISBI. This list contains all the necessary elements and will be explained in the following article.

- A working directory where you place all the files. It will have a mandatory filesystem layout that must be followed.
- An editor that can be used to edit java source files, text files and XML files.
- A Java JDK with compiler. Recommendation: Please use the exact same level of the JDK that you also use to run ISBI. If you develop/compile on the same box where ISBI is installed, you might consider using the ISBI built-in JDK.
- A set of Jar files from your ISBI Installation. Since the custom service should run with your version of SI, you should also use ISBIs libs to compile against. It is a good practice to recompile the service, every time you apply a patch to ISBI.

Abstract

On a very basic level, to write a custom service for SI platform, there are two ways to deploy a service.

1. Create a pre-packaged Jar Service package consisting of all the service resources and install it later using InstallService shell script file.
2. Manually, copy the required service definition java files and Business Processes to their respective product asset folders. Now, building the Assets and finally install the patch.

The service structure for our REST API client is described as shown –

The genericrest.jar file contains:

```
META-INF
        MANIFEST.MF
genericrest
        jars
           genericrest
               1_0
                 genericrest.jar
        bpml
           genericrest.bpml.in
        servicedefs
            genericrest.xml
genericrest.java
RestRequestHandler.java
RestAPIClientGET.java
RestAPIClientPOST.java
RestAPIClientPUT.java
RestAPIClientDELETE.java
genericrestserver.java
genericrestserverImpl.java
serviceinstances.xml
```

Class definition

The file genericrest.jar provides the class definition. This JAR file contains following three files (In the proper package hierarchy)–

```
com
   sterlingcommerce
       woodstock
```

```
services
    genericrest
     genericrest.java
       RestRequestHandler.java
       RestAPIClientGET.java
       RestAPIClientPOST.java
       RestAPIClientPUT.java
  RestAPIClientDELETE.java
       genericrestserver.java
       genericrestserverImpl.java
```

When installed, these files will be appended to the <Install_Directory>/install/properties/dynamicclasspath.cfg file.

genericrest.java – The service which actually acts as a gateway to handle the upcoming requests depending upon the input parameters from the process data.
RestRequestHandler.java – The java file which distributes the current request depending upon the request type.
RestAPIClientGET.java – The java file which executes the GET request.
RestAPIClientPOST.java - The java file which executes the POST request.
RestAPIClientPUT.java - The java file which executes the PUT request.
RestAPIClientDELETE.java - The java file which executes the DELETE request.
genericrestserver.java – The java file that is responsible for restarting, shutdown; etc of the genericrest service.
genericrestserverImpl.java – The java file that implements the genericrestserver class.

Details of the Java service files

A service in Sterling Integrator is at least one Java class that implements an interface from Sterling Integrator. In addition to that you have a deployment descriptor that tells the installer during runtime which service you are trying to install. It also tells the system the name of the class that is the entry point into your custom service.

In the starting class you will have to have a method called processData(). The signature of this method comes from the com.sterlingcommerce.woodstock.services.IService Interface. In that method you receive the com.sterlingcommerce.woodstock.workflow.WorkFlowContext from the business process. It does contain references to the documents and you can access the process data elements.

Installing and using the genericrest service

Depending upon the way you chose to build the service in the Abstract section, you have two different ways of installing the service.

1. Pre-package the application (jar) In the directory format specified above. Use the InstallService.sh or Install Service.cmd file to install the genericrest.jar file.

2. Or, you can put the resource and the source files of your service in their respective folders inside the assets/product/src and assets/product/resources directory.

Now, after placing these files as shown above, follow these set of steps

1. Go to assets directory and then run `./build.sh`
   This will build and package all the newly copied files (new service files) into jar files. These jar files will be created in the assets/dist/ directory.
2. Now, go to …/<Install_Directory>/install/bin/ and run the following command –
   `./InstallService.sh <path of the newly created asset jar>`
   In our case, this command is –
   `./InstallService.sh /asset/dist/patch/patch_asset_4060603.jar`
3. In the same directory, run `./hardstop.sh` followed by `./run.sh`
   This will restart the ISBI server.

CRUD Operations

Clients can perform CRUD(Create, Read, Update ,Delete) requests by creating business process and calling the generic rest(Rest Client Service) through bp. They can create multiple business processes, multiple service instances and can call different operations through different service instances.
It also allows clients to pass response of one service instance to another using XML Xpath. Different types of options are available for the clients to create and update the instance of an API. They can pass json input in business process, can pass the response of one operation to other and can also pass json input through file.

Multiple GET, DELETE, GET Operation

A single business process that calls two different service instances of generic rest. In first operation, GET and DELETE operations are carried out in a comma seperated url and request type to fetch and delete single instance of an API. In second service, GET request to retreive all the instances of an API are carried out.

Business Process

Business Process Definition:

```
<process name="genericrest">
  <sequence>
        <operation name="Request">
          <participant name='genericrest'/>
          <output message='xout'>
            <assign to='url'>http://9.55.53.11:51665/B2BAPIs/svc/cadigitalcertificates/b2bidemo1/,http://9.55.53.11:51665/B2BAPIs/svc/cadigitalcertificates/b2bidemo1/</assign>
            <assign to='restoperation'>GET,DELETE</assign>
            <assign to='auth'>admin:password</assign>
            <assign to='.' from='*' />
          </output>
         <input message="xin">
           <assign to='service1' from='*' />
         </input>
        </operation>
        <operation name="Request1">
          <participant name='genericrest1'/>
          <output message='xout'>
            <assign to='url'>http://9.55.53.11:51665/B2BAPIs/svc/cadigitalcertificates/</assign>
            <assign to='restoperation'>GET</assign>
            <assign to='auth'>admin:password</assign>
          </output>
         <input message="xin">
           <assign to='service2' from='*'/>
         </input>
        </operation>
  </sequence>
</process>
```

Business Process Execution

Execute Business Process

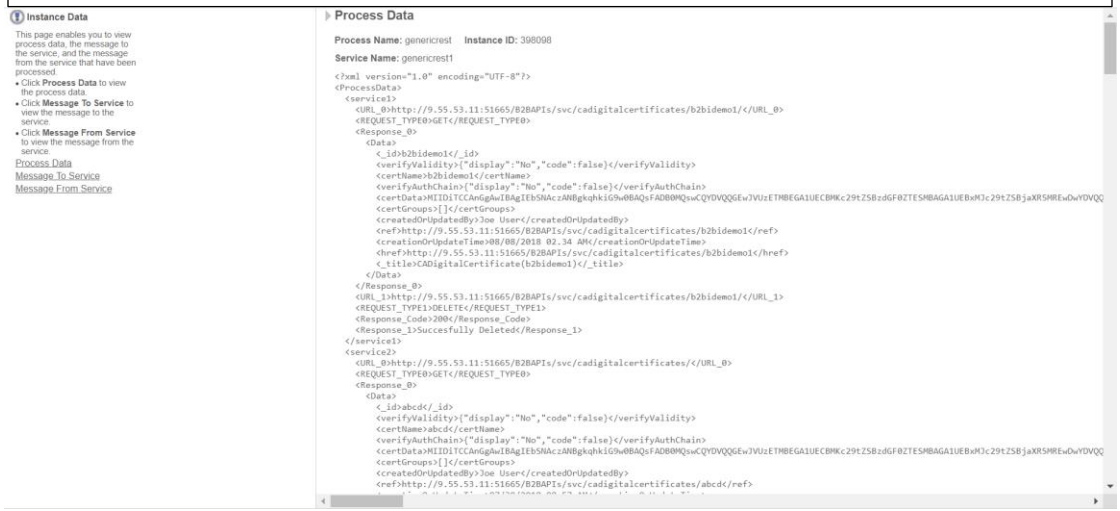
Name: genericrest **Instance ID:** 398098 **User:** admin

Completed
Status: Success

Step	Service	Status	Advanced Status	Started	Ended	Execution Node	Status Report	Document	Instance Data
2	genericrest1	Success	None	08/20/2018 7:27:19 AM EDT	08/20/2018 7:27:19 AM EDT	node1 [-]	None	None	Info
1	genericrest	Success	None	08/20/2018 7:27:19 AM EDT	08/20/2018 7:27:19 AM EDT	node1 [-]	None	None	Info
0	INITIATING_CONTEXT from user 'admin'	Success	None	08/20/2018 7:27:18 AM EDT	08/20/2018 7:27:18 AM EDT	node1 [-]	None	None	None

* Inline Invocation
[-] Load Balanced, [p] Preferred Node, [m] Mandatory Node, [r] Execution Role

Output



Multiple GET, POST, GET Operation

In this business process, multiple CRUD operations are performes using single business process. First GET operation retreive all the instances of an API. Then POST operation is carried out by passing json object input through file. Then again GET operation fetches all he instances including the POST instance of an API.

Business Process

Business Process Definition:

```
<process name="genericrest">
<sequence>
<operation name="Request">
<participant name='genericrest'/>
<output message='xout'>
<assign to='url'>http://9.55.53.11:51665/B2BAPIs/svc/cadigitalcertificates/,http://9.55.53.11:51665/B2BAPIs/svc/cadigitalcertificates/,http://9.55.53.11:51665/B2BAPIs/svc/cadigitalcertificates/</assign>
<assign to='restoperation'>GET,POST,GET</assign>
<assign to='auth'>admin:password</assign>
<assign to='.' from='PrimaryDocument/@SCIObjectID'></assign>
<assign to='.' from='*' />
</output>
<input message="xin">
<assign to='.' from='*' />
</input>
</operation>
</sequence>
</process>
```

Output

Multiple GET, PUT, GET Operation

In this business process, two service different instances of generic rest are called. In the first operation, GET operation is carried out to retreive the instance of b2bi_demo. Then PUT operation is carried out by passing json input in the business process itself.

Business Process

Business Process Definition:

```
<process name="genericrest">
<sequence>
<operation name="Request">
<participant name='genericrest'/>
<output message='xout'>
<assign to='url'>http://9.55.53.11:51665/B2BAPIs/svc/cadigitalcertificates/b2bidemo4,http://9.55.53.11:51665/B2BAPIs/svc/cadigitalcertificates/b2bidemo4,http://9.55.53.11:51665/B2BAPIs/svc/cadigitalcertificates/b
<assign to='restoperation'>GET,PUT,GET</assign>
<assign to='auth'>admin:password</assign>
<assign to='jsoninput1'>"certName":"b2bidemo4","verifyAuthChain":true,"verifyValidity":true</assign>
<assign to='.' from='*' />
</output>
<input message="xin">
<assign to='.' from='*' />
</input>
</operation>
</sequence>
</process>
```

Multiple GET POST(using Xpath) Operation

There are two operations in the example mentioned below. In the first operation, GET request is used to fetch certificate data using **cadigitalcertificate API.** The second operation retrieves the data for POST input from the Process Data of the first operation using assign statements in the format mentioned below.

Business Process

Business Process Definition:

```
<process name="genericrest">
<sequence>
<operation name="Request">
<participant name='genericrest'/>
<output message='xout'>
<assign to='url'>http://9.55.53.11:51665/B2BAPIs/svc/cadigitalcertificates/b2bi_demo/</assign>
<assign to='restoperation'>GET</assign>
<assign to='auth'>admin:password</assign>
<assign to='.' from='*' />
</output>
<input message="xin">
<assign to='service1' from='*' />
</input>
</operation>
<operation name="Request1">
<participant name='genericrest1'/>
<output message='xout'>
<assign to='url'>http://9.55.53.11:51665/B2BAPIs/svc/cadigitalcertificates/</assign>
<assign to='restoperation'>POST</assign>
<assign to='auth'>admin:password</assign>
<assign to="jsoninput1" from='concat(&apos;"certName":"&apos;,service1/Response_0/Data/certName/text(),&apos;10"&apos;)'></assign>
<assign to="jsoninput2" from='concat(&apos;"certData":"&apos;,service1/Response_0/Data/certData/text(),&apos;"&apos;)'></assign>
</output>
<input message="xin">
<assign to='service2' from='*'/>
</input>
</operation>
</sequence>
</process>
```

Note:
Currently, this service reports the following Runtime errors

- Authentication Error
- URL Validation Error
- Invalid Request Types
- HTTP Connection Errors
- Bad Request Errors

Since the code will run as a service you have to pay special attention to the fact that your code will run inside the ISBI virtual machine. Special attention must be paid to prevent

- Memory Leaks
- Thread Leaks
- File Handle Leaks
- Socket Leak

3. Results and Discussions

Microservices are an effective and fault-tolerant manner of designing and developing an application. B2Bi is a large monolithic application which needs to be divided into microservices. The heart of B2Bi is composed of several business processes which are used to automate the operation of B2B services (Mailbox, Invoice, Order; etc). Separate microservices can be deployed on the B2B SI platform server to enhance the functionality.

Rest Client Service allows to perform CRUD(Create, Read, Update, Delete) operations on B2BAPIs. Client can create multiple business processes and can call multiple service instances in a single business process. POST and PUT operation can be performed in multiple ways i.e. by passing json input in business process, by inputting json object through document and also passing the response of one operation to another using Xpath.

4. Conclusions

During past few weeks, I got acquainted to the Microservice architecture of software development, researched about the B2B processes and learned to code them. For implementing these ideas into practical uses, I had to understand about how the Sterling Integrator code functions. Using this knowledge, I deployed a sample WAR service to the server of SI.

After developing Rest Client for CRUD operations, it provides an efficient way to clients to call the desired API instance from business process and get the response in process data. Now they can make multiple business processes, make multiple CRUD operations and create multiple service instances .

4.1 Next Target

My plan of next 15 days is to get acquainted with Spring Boot and deploy the User Account Service and Trading Partner Service APIs using Spring Boot. Currently all the APIs are deployed using java, but due to large amount of requests clients are facing problem in accessing APIs taking longer execution time.

5. References

[1] https://www.ibm.com/in-en/marketplace/b2b-gateway-software

[2] https://www.ibm.com/developerworks/community/blogs/SterlingB2B/entry/

[3] Automate Sterling FileGateway Community and Trading Partner creation

[4] 'Amazon S3' Integration with IBM B2B Sterling Integrator

6. Publications

Rest Client Service: B2B Integrator

Rest Client Service is an essential and important integration in IBM Sterling that solves the major problem of clients in performing CRUD(Create, Read, Update, Delete) operations on B2BAPIs. Clients can create their own business processes as per requirement and can also create multiple service instances accordingly.

My work on Rest Client Service was published in **IBM Developer Works Article**.

Link: Rest Client Service