iNTERFACEWARE Inc.
672 Dupont Street, Suite 505
Toronto, ON, M6G 1Z6
interfaceware.com

How to Rapidly Scale Your Integrations Using API's Exercise

For IUC2019

September 20th, 2019

Purpose

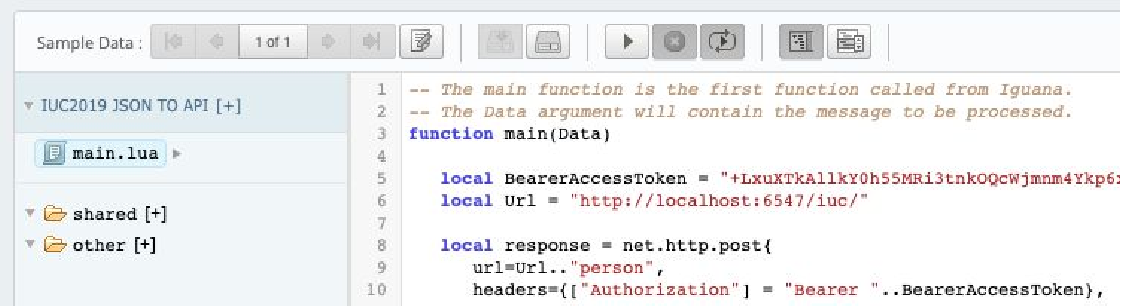
This document describes the exercise accompanying the *How to Rapidly Scale Your Integrations Using API's* presentation at IUC2019.

Included is a GitHub repository that will contain six channels with *Exercise3:* prepended to the channel name, that can be imported into any Iguana instance (see Add/Configure Repositories and Import Channels).

If you have any questions or concerns, please contact us at support@interfaceware.com and CC paul.le@interfaceware.com or amandeep.aujla@interfaceware.com in the email.

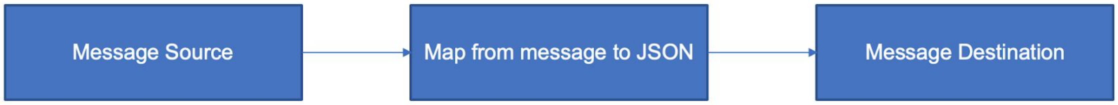
Overview

1. Using the GitHub repository, import all six channels into any Iguana instance.
2. Ensure that the *Exercise2: API Server* channel has also been imported as well from the GitHub repository, and that there is an access key available (see *Juggling Internal & External APIs with Iguana Exercise* document for more information). In the *Exercise3: 6 - JSON to API* channel, add the access key to the *BearerAccessToken* variable and ensure the URL is pointing to the correct *Exercise2: API Server* channel URL:

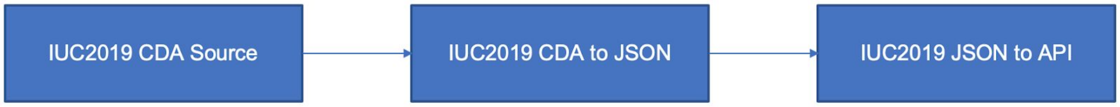


```
-- The main function is the first function called from Iguana.
-- The Data argument will contain the message to be processed.
function main(Data)

   local BearerAccessToken = "+LxuXTkAllkY0h55MRi3tnkOQcWjmnm4Ykp6:
   local Url = "http://localhost:6547/iuc/"

   local response = net.http.post{
      url=Url.."person",
      headers={["Authorization"] = "Bearer "..BearerAccessToken},
```

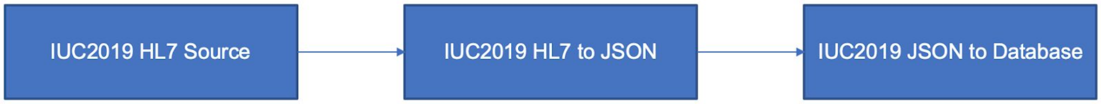
3. There are two workflows included in these six channels, both of which have the following structure:



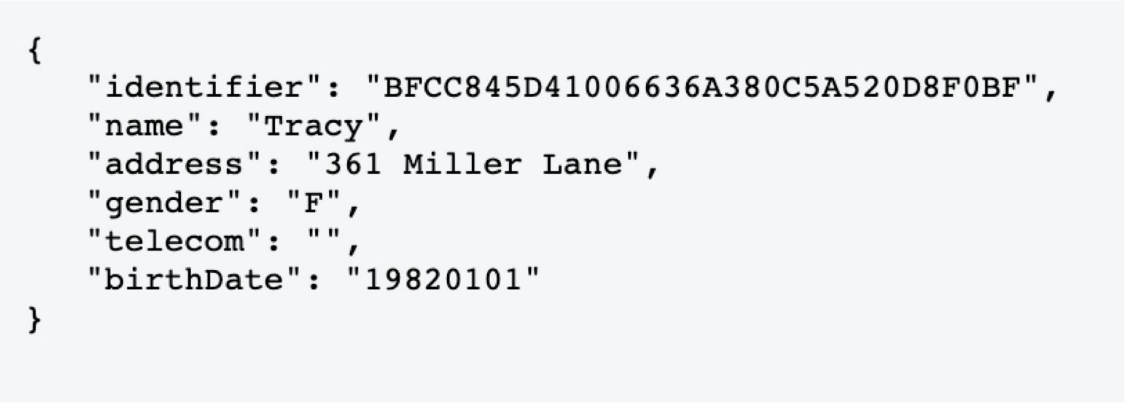
4. The first workflow involves a CDA source that sends CDA documents to a filtering channel that maps CDA to a JSON message, which then send the JSON message to another channel that uses the JSON message to make an API request to the *Exercise2: API Server* channel:



5. The second workflow involves an HL7 source that sends HL7 messages to a filtering channel that maps HL7 to a JSON message, which then sends the JSON message to another channel that writes the data to a SQLite database:



6. Both workflows use the same JSON message, where the filter channel is used to map either CDA or HL7 to the JSON message. The *Exercise3: 5 - JSON to DB* and *Exercise3: 6 - JSON to API* channels both accept the same JSON message to perform the workflow:



```
{
   "identifier": "BFCC845D41006636A380C5A520D8F0BF",
   "name": "Tracy",
   "address": "361 Miller Lane",
   "gender": "F",
   "telecom": "",
   "birthDate": "19820101"
}
```