**GLOBAL WEATHER SERVICE API – TROUBLESHOOTING**

1. **COMPLIATION ,RUNTIME AND MUNIT ERRORS**

1.A) There is a missing Munit Library.

The error was more specific to the get\_city\_by\_country-test-suite.xml and this was found out from the problems tab on the configuration panel of Any point studio.

Steps taken: Added the Munit library to the Mule project using the Configure Build Path Option.

Added the Munit Runtime Option Library to the list of libraries that the project references.

**Outcome:** There were no errors displayed on the Problems tab and No errors (X mark) were displayed on the Project explorer panel on the left side of the Studio.

As a next step, ran the project and the start-up was not successful.

Looking at the console window, found that the project failed to run because of the following error:

Could not load properties; nested exception is java.io. FileNotFoundException: class path resource [properties/globalweather-${mule.envt}.properties] cannot be opened because it does not exist

1.B) The error indicated that a property mule.envt did not exist.

Steps: Browsed through the properties file reference by the Interface to know where this variable has been defined. identified that environment variable in the mule application was defined incorrectly as mule.env instead of mule.envt

* mule.envt=local

**Outcome:** The error related to environment variable reference did not appear anymore.

Upon re-running the application after fixing the error above, The process execution was still unsuccessful.

The error appeared on the console after the wsdl retrieve step.

Steps taken:

Going through the console, looking at the sequence of events it looked to be a problem specific to get:/city:weather-api-config.

1. To confirm the same, modified both the Main flows get:/city:weather-api-config and get:/weather:weather-api-config to reference the Get\_weather flow.

**Outcome: No errors were noticed and application started without errors. This indicated that problem is in the Get City by country flow.**

1. As a next step, to confirm that there are no issues with the WSDL in the Get City by Country flow,
2. Reloaded the wsdl in the Git city by country flow and ran the project. It still failed.
3. Created a new project with a web service consumer flow and imported the wsdl with the Get cities by country operation. The new project was started successfully with no errors.

**Outcome: No errors with the WSDL.**

Identified that the Until successful has been configured with default settings. However, since this is a web service call the scope should not be asynchronous and it needs to be synchronous. the main flow needs to wait for the web service consumer call processing within the until successful scope.

Added **synchronous="true" for the until successful scope** to the configuration XML

**Outcome: No errors observed and application was deployed successfully.**

Runtime errors – Get Weather Flow:

1. Typecast error – City (String) was mapped to CityName as Number.

To fix this, removed the number datatype in the output payload data weave mapper.

1. Mapping error (Generate Response (String))– The trim function works only on a string, whereas the output returned from the webservice call is an object.

To fix this, the Mule replace expression needs to be executed first and the output payload needs to be passed onto the Transform to trim the payload .

The data weave code was also modified from Trim payload. GetWeatherResponse to trim payload (based on the output returned from expression)

1. Mapping error (Transform Message)– The final elements in the JSON response mapped were mapped directly and they were missing the GetWeatherResponse element.

Updated the Data weave code to include GetWeatherResponse. <element name> in the JSON Output.

Runtime errors – Get Cities by Country Flow:

1. Mapping error (Extract Response (String))– The trim function works only on a string, whereas the output returned from the webservice call is an object.

To fix this, the Mule replace expression needs to be executed first and the output payload needs to be passed onto the Transform to trim the payload .

The data weave code was also modified from Trim payload. GetWeatherResponse to trim payload (based on the output returned from expression)

1. Mapping error (Generate Response)– The final elements in the JSON response mapped were mapped directly and they were missing the GetWeatherResponse element.

Updated the Data weave code to include GetCitiesByCountryResponse. <element name> in the JSON Output.