1. **Introduction:** This document describes the design of RestAPI’s which invokes webservice to return the results to the user. It involves calling of 2 webservice

GetWeather: This service will return the weather details for a give city and country

GetCitiesByCountry: This service will return all the cities for a particular country.

1. **Scope:** The scope of the document will involve

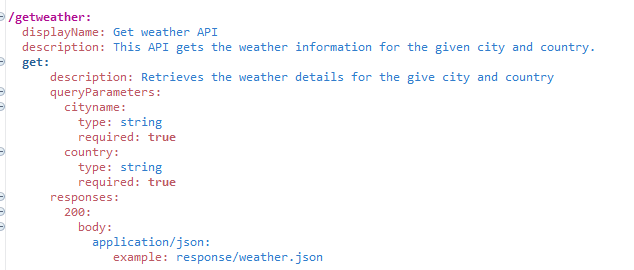
* Development of rest API’s
* Development of MUnit test cases
* Deploying webservice as a docker image

1. **Development of Rest API’s**

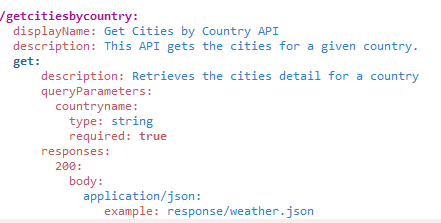
2 API’s were developed to invoke 2 webservice. The RAML file provides the definition of API’s

GetWeather abd GetCitiesByCountry

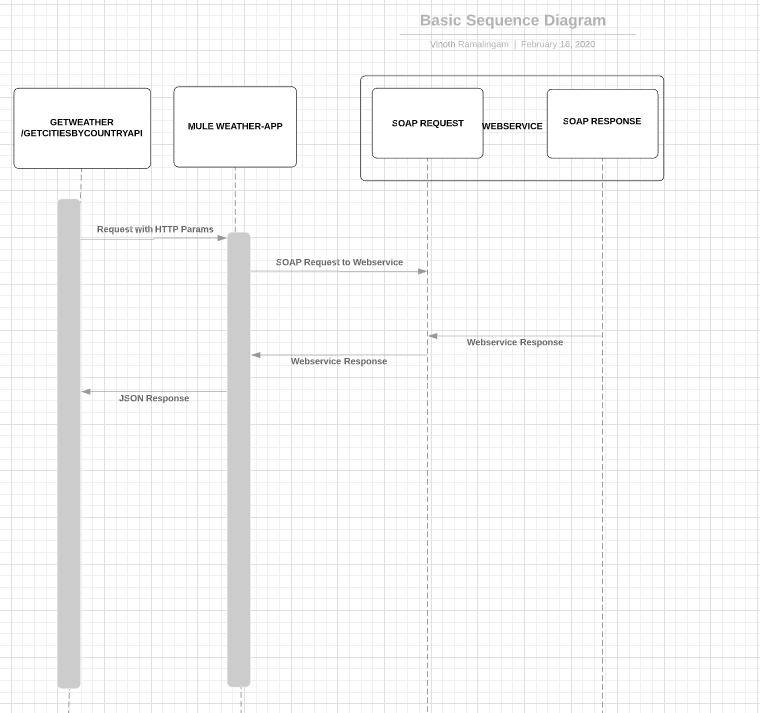
* 1. Getweather API: Below is the snippet of RAML file definition for this API



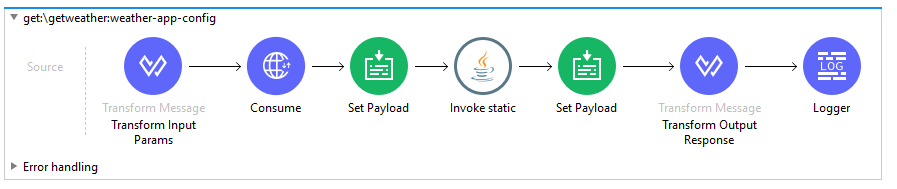
* 1. GetCitiesByCountries: Below is the snippet of RAML file definition for this API

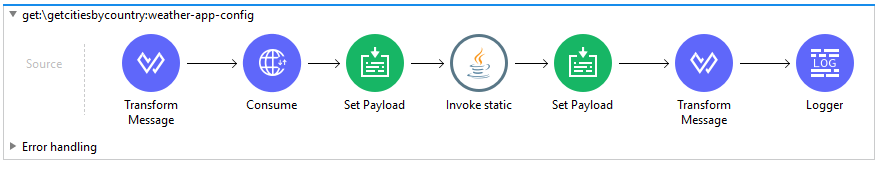


1. **Basic Sequence Diagram**



1. **Mulesoft Flow**
   1. Getweather/GetCitiesByCountry API





1. Transform Input Params: This will transform the input HTTP pramas into and XML payload which will be sent to webservice
2. Consume: This will invoke the GetWeather/GetCitiesByCountry operation of the webservice and return the response
3. Java Function: This will transform the raw CDATA response into and XML based output
4. Transform Output response: This will return eh structured JSON response as output.
5. **MUnit Testcases:**
   1. Weather-app-test-suite

Two Munit Testcases have been developed for each of the API’s



1. Set Event: This will set the payload as input. In this it will set the HTTP params as input
2. Sub Flow: This will invoke the sub process flow in the mule flow
3. Mock When: This will mock the webservice call and return the response
4. Assert Equals: This will verify the response returned against the expected response
5. **Deploy Webservice as docker image**
   1. Docker image was built using the given DockerFile and stored in the docker hub

docker build -t <your username>/weather-app .

* 1. Docker image was deployed from the docker hub

docker run -p 49160:8080 -d <your username>/weather-ap

Once the docker image is deployed the webservice is available to serve the traffic.