**GLOBAL WEATHER SERVICE API –Description and Technical Specification**

Global weather service is a Mule based REST API that provides two different weather services

1. Weather information for Major cities in the world.
2. Provides a list of major cities for a country.

API Details:

[http://localhost:8085/v1/\*](http://localhost:8085/v1/*)

**Operations/HTTP methods and their purpose:**

1. GET /weather

Example Request: GET <http://localhost:8085/v1/weather?country=Australia> &&city=Melbourne

Description: This API Call retrieves weather information for all major cities around the world.

Response:

1. If the API call is successful, a HTTP status code of 200 would be returned and the response would contain the weather information of a city in a country that includes Wind, visibility, Sky conditions, Temperature and Relative Humidity in a JSON format.



1. If unsuccessful, an JSON error response would be returned containing error code and corresponding error message describing the reason for failure.
2. GET /city

Example Request: GET <http://localhost:8085/v1/city?country=Australia>

Description: This API Call retrieves list of major cities in a country.

Response:

1. If the API call is successful, a HTTP status code of 200 would be returned and the response would contain list of major cities in the country specified in the HTTP request URI.



1. If unsuccessful, an JSON error response would be returned containing error code and corresponding error message describing the reason for failure.

**TECHNICAL Description and Purpose of the Mule Application and various flows:**

**The Main Mule flow in the project is the weather-api flow and the flow executes the below actions:**

1. Receives the request from the user on a HTTP listener.
2. The HTTP listener is connected to a API KIT router that sends the request to the appropriate Resource specified in the weather-api RAML file.
3. There are two flows – one each for method/resource pairs. That is one flow for the Get Weather and the other flow for Get City.
4. The Get Weather Flow has a flow reference to another Sub-flow sf\_get\_weather\_request and the Get City flow has a reference to the sub flow sf\_get\_cityFlow\_request.
5. The main flow also has a global exception mapping strategy for the APIKit that returns HTTP Status codes and error messages depending on the error messages returned from the Web service.
6. Lastly there is a API console flow for simulating/testing API calls that receives request sent on the path http:/localhost:8085/v1/console/

**The get\_weather flow --- > that references to sf\_get\_weather\_request**

Receives the city and country values from the HTTP request and makes call to the Web Service consumer (WS\_Consumer\_Weather) GetWeather operation.

If the Web service call is success, then it passes the payload to the response sub-flow (sf\_get\_weather\_response). The response sub-flow receives the payload and converts the SOAP response to a JSON format and sends the weather information parameters about the city as a JSON response message back to the requestor.

If the Web service call returns an empty payload (or) an invalid response, then the sf\_commonBusinessException flow is called and a business exception message and HTTP status code is sent as a response.

**The get\_city\_by\_country flow --- > that references to sf\_get\_cityFlow\_request**

Receives the country value from the HTTP request and makes call to the Web Service consumer (WS\_Consumer\_Cities) GetCitiesByCountry operation.

If the Web service call is success, then it passes the payload to the response sub-flow (sf\_get\_cityFlow\_response). The response sub-flow receives the payload and converts the SOAP response to a JSON format and sends the list of cities in a country as a JSON response message back to the requestor.

If the Web service call returns an empty payload (or) an invalid response, then the sf\_commonBusinessException flow is called and a business exception message and HTTP status code is sent as a response.

**Common Mule flow:**

This flow consists of the following 3 flows that are referenced in the sf\_get\_weather\_request/response and sf\_get\_cityFlow\_request/response flows for logging and exception handling

* sf\_commonFlowLogRequest– to log the HTTP requests received in the request
* sf\_commonErrorHandling– returns JSON error responses with HTTP statuses, status codes and error messages for the strategies defined in Global exception handling.
* sf\_commonBusinessException– returns a HTTP status code 422 – Business Exception – if the web service is null (or) the Web Service consumer was unable to process the response received from SOAP Web Service.

**global-exception-handler flow:**

This flow catches the exceptions that occur in the get\_weather and get\_city\_by\_country flows and returns the error message and a HTTP status code as a json response.