Design decisions for GlobalWeatherAPI

While developing the GlobalWetherAPI the following things were considered.

* Number of operations in the GlobalWeather WSDL that were required to be exposed – 2
* Actions performed in these 2 operations of the WSDL – Both were GET actions
* The number of resources for our REST API – 2 resources, one each for an operation
* REST url format –
  + The GetCitiesByCountry operation returned s list of prominent cities for a given country. As the cities are a collection of data that can be individually queried further the input field country is passed as a query parameter.
  + The GetWeather operation returns the weather information of a given city in a given country. This information is specific and unique to that city, so the inputs to this operation country and city are passed as URI parameters.
* In case of user providing a url that is not defined within the RAML, the user would be notified saying “Object Not Found” and a standard 404 return code. As both the resources were simple Get methods there are no input validations I have considered here.
* There was no technical challenge I faced in the solution as it is a simple mapping from source to destination.
* However as the actual webservice was unavailable and I had to use the docker service. As a reason I could not test it for a wide range of scenarios as the docker image has been hardcoded with one response for each operation.