

# Assignment Report

-Dinesh Metkari

## Contents

1. Problem Statement.....	3
2. Solution Overview.....	4
3. SOAP Service Operation: GetCitiesByCountry .....	5
3.1. Original Request & Response.....	5
3.2. Proxy Request & Response: GetCitiesByCountry.....	7
4. SOAP Service Operation: GetWeather.....	9
4.1. Original Request & Response.....	9
4.2. Proxy Request & Response .....	11
5. Additional Features that can be implemented .....	12

## 1. Problem Statement

Design a Microservice using Spring Boot and Apache Camel which will act as a REST proxy for a SOAP service hosted at the below URL. Package and deploy the service into a docker container.

<http://www.webserviceex.com/globalweather.asmx?WSDL>

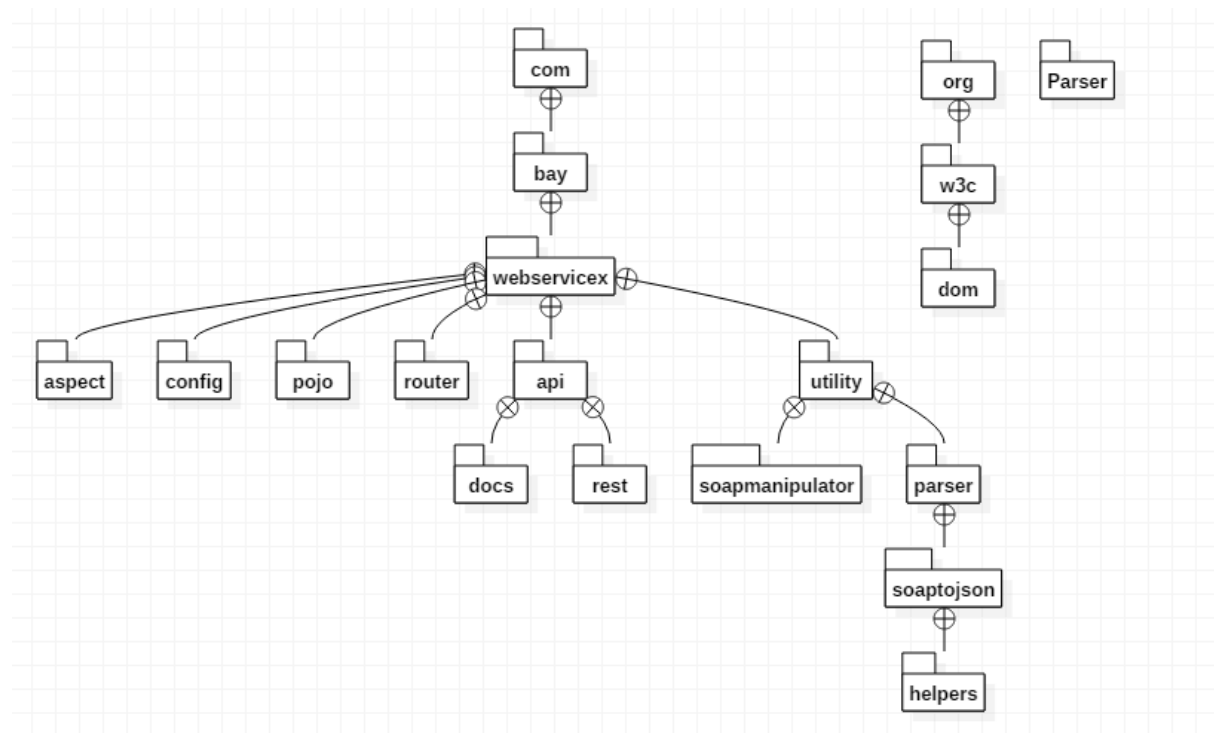
The following are the expected behaviors

- Accept input in the form of JSON or HTTP using a client such as postman
- Accept the name of the operation in the SOAP service to call
- Call the SOAP Service Operation and pass the Input
- Receive the Response
- Give out the JSON response on HTTP

Following are the constraints that will apply.

- No usage of WSDL to Java to convert WSDL types into Java Objects
- JSON to SOAP envelope conversion should be dynamic and on the fly. Hypothetically by changing the WSDL (and minor configuration) we should be able to call another SOAP service.
- The same proxy should be able to call different operations within the specified WSDL

## 2. Solution Overview



Package Structure Diagram

- Designed Proxy Service using apache camel, Spring Microservices,
- Technologies used: Camel, Spring, Spring Annotation, Spring Boot, Spring AOP, Swagger, Maven, Docker, SOAP UI libraries in pom file
- Identified soap service operations and their existing soap request and response.
- Done R&D with various implementations
- Selected appropriate implementation based on the constraints to satisfy.
- Designed the microservice corresponding to the soap service operations
- Camel proxy service runs on configurable server port (8080)
- When microservice is invoked from camel proxy, it has been configured to fetch corresponding soap service and required configuration items can be configured from application.properties
- It fetches SOAP Request XML dynamically
- It modifies SOAP request with JSON parameters dynamically
- It invokes SOAP Service call and receives response
- Response can be returned back as JSON or XML format dynamically, which is configurable in application.properties file
- The same proxy service can be used with other soap services with minimal changes.
- The proxy service containerized using docker.

### 3. SOAP Service Operation: GetCitiesByCountry

#### 3.1. Original Request & Response

URL	<a href="http://www.webserviceX.com/globalweather.aspx">http://www.webserviceX.com/globalweather.aspx</a>
SOAPAction	http://www.webserviceX.NET/GetCitiesByCountry
Request	<pre>&lt;soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://www.webserviceX.NET"&gt;   &lt;soapenv:Header/&gt;   &lt;soapenv:Body&gt;     &lt;web:GetCitiesByCountry&gt;       &lt;web:CountryName&gt;India&lt;/web:CountryName&gt;     &lt;/web:GetCitiesByCountry&gt;   &lt;/soapenv:Body&gt; &lt;/soapenv:Envelope&gt;</pre>
Response	<pre>&lt;?xml version="1.0" encoding="utf-8"?&gt; &lt;soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"&gt;   &lt;soap:Body&gt;     &lt;GetCitiesByCountryResponse xmlns="http://www.webserviceX.NET"&gt;       &lt;GetCitiesByCountryResult&gt;&amp;lt;NewDataSet&amp;gt;         &amp;lt;Table&amp;gt;           &amp;lt;Country&amp;gt;British Indian Ocean Territory&amp;lt;/Country&amp;gt;           &amp;lt;City&amp;gt;Diego Garcia&amp;lt;/City&amp;gt;         &amp;lt;/Table&amp;gt;         &amp;lt;Table&amp;gt;           &amp;lt;Country&amp;gt;India&amp;lt;/Country&amp;gt;           &amp;lt;City&amp;gt;Vellore&amp;lt;/City&amp;gt;         &amp;lt;/Table&amp;gt;       &amp;lt;/NewDataSet&amp;gt;&lt;/GetCitiesByCountryResult&gt;     &lt;/GetCitiesByCountryResponse&gt;   &lt;/soap:Body&gt; &lt;/soap:Envelope&gt;</pre>

Postman

File Edit View Collection History Help

New Import Runner Builder Team Library

Filter

History Collections

Clear all

Today

- POST http://www.webservices.com/globalweather.asmx
- POST http://localhost:8080/v1/globalweather/GetCitiesByCountry
- POST http://localhost:8080/v1/globalweather/GetCitiesByCountry
- POST http://localhost:8082/globalweather/GetCitiesByCountry
- POST http://localhost:8080/v1/globalweather/GetWeather
- POST http://localhost:8080/v1/globalweather/GetWeather
- POST http://localhost:8080/v1/globalweather/GetWeather
- POST http://localhost:8080/v1/globalweather/GetWeather

POST http://www.webservices.com/globalweather.asmx

Authorization Headers (2) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary XML (text/xml)

```
1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://www.webserviceX.NET">
2   <soapenv:Header/>
3   <soapenv:Body>
4     <web:GetCitiesByCountry>
5       <web:CountryName>India</web:CountryName>
6     </web:GetCitiesByCountry>
7   </soapenv:Body>
8 </soapenv:Envelope>
```

Body Cookies Headers (9) Test Results

Status: 200 OK Time: 1539 ms Size: 1.53 KB

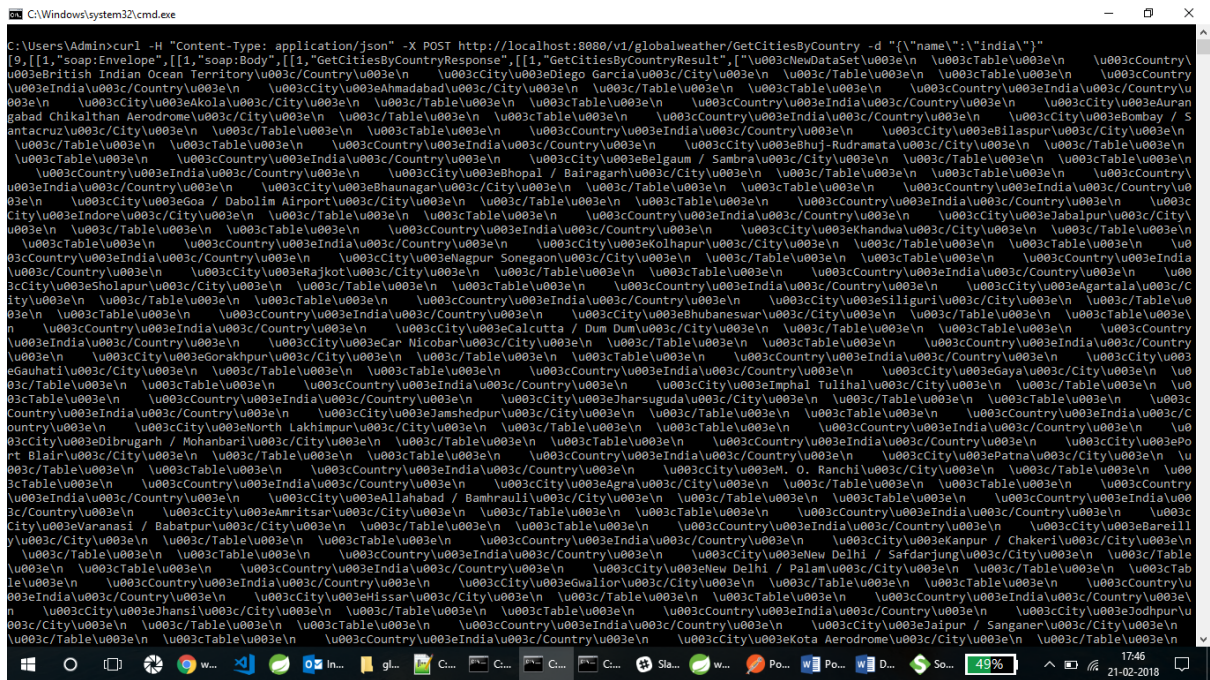
Pretty Raw Preview XML

```
4 <GetCitiesByCountryResponse xmlns="http://www.webserviceX.NET">
5   <GetCitiesByCountryResult>&lt;NewDataSet&gt;
6     &lt;Table&gt;
7       &lt;Country&gt;British Indian Ocean Territory&lt;/Country&gt;
8       &lt;City&gt;Diego Garcia&lt;/City&gt;
9     &lt;/Table&gt;
10    &lt;Table&gt;
11      &lt;Country&gt;India&lt;/Country&gt;
12      &lt;City&gt;Ahmadabad&lt;/City&gt;
13    &lt;/Table&gt;
14    &lt;Table&gt;
15      &lt;Country&gt;India&lt;/Country&gt;
```

Windows taskbar: 17:47 21-02-2018

### 3.2. Proxy Request & Response: GetCitiesByCountry

URL	<a href="http://www.webserviceX.com/globalweather.asmx">http://www.webserviceX.com/globalweather.asmx</a>
SOAPAction	http://www.webserviceX.NET/GetCitiesByCountry
Request	<pre>{   "name": "India" }</pre>
Response	<pre>[9,[[1,"soap:Envelope",[[1,"soap:Body",[[1,"GetCitiesByCountryResponse",[[1,"GetCitiesByCountryResult",["\u003cNewDataSet\u003e\n \u003cTable\u003e\n \u003cCountry\u003eBritish Indian Ocean Territory\u003c/Country\u003e\n \u003cCity\u003eDiego Garcia\u003c/City\u003e\n \u003c/Table\u003e\n \u003cTable\u003e\n \u003cCountry\u003eIndia\u003c/Country\u003e\n \u003cCity\u003eAhmadabad\u003c/City\u003e\n \u003c/Table\u003e\n \u003cTable\u003e\n \u003cCountry\u003eIndia\u003c/Country\u003e\n \u003cCity\u003eAkola\u003c/City\u003e\n \u003c/Table\u003e\n \u003cTable\u003e\n \u003cCountry\u003eIndia\u003c/Country\u003e\n \u003cCity\u003eAurangabad Chikalthan Aerodrome\u003c/City\u003e\n \u003c/Table\u003e\n \u003cCountry\u003eIndia\u003c/Country\u003e\n \u003cCity\u003eVellore\u003c/City\u003e\n \u003c/Table\u003e\n\u003c/NewDataSet\u003e"]]]],["xmlns","http://www.webserviceX.NET"]]]]]],["xmlns:soap","http://schemas.xmlsoap.org/soap/envelope/"],["xmlns:xsd","http://www.w3.org/2001/XMLSchema"],["xmlns:xsi","http://www.w3.org/2001/XMLSchema-instance"]]]]]]</pre>





## 4. SOAP Service Operation: GetWeather

### 4.1. Original Request & Response

URL	<a href="http://www.webserviceX.com/globalweather.aspx">http://www.webserviceX.com/globalweather.aspx</a>
SOAPAction	http://www.webserviceX.NET/GetWeather
Request	<pre>&lt;soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://www.webserviceX.NET"&gt;   &lt;soapenv:Header/&gt;   &lt;soapenv:Body&gt;     &lt;web:GetWeather&gt;       &lt;!--Optional:--&gt;       &lt;web:CityName&gt;Mumbai&lt;/web:CityName&gt;       &lt;!--Optional:--&gt;       &lt;web:CountryName&gt;India&lt;/web:CountryName&gt;     &lt;/web:GetWeather&gt;   &lt;/soapenv:Body&gt; &lt;/soapenv:Envelope&gt;</pre>
Response	<pre>&lt;?xml version="1.0" encoding="utf-8"?&gt; &lt;soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"&gt;   &lt;soap:Body&gt;     &lt;GetWeatherResponse xmlns="http://www.webserviceX.NET"&gt;       &lt;GetWeatherResult&gt;Data Not Found&lt;/GetWeatherResult&gt;     &lt;/GetWeatherResponse&gt;   &lt;/soap:Body&gt; &lt;/soap:Envelope&gt;</pre>

Postman

File Edit View Collection History Help

New Import Runner Team Library

Filter

History Collections

Clear all

Today

- POST http://www.webservices.com/globalweather.asmx
- POST http://www.webservices.com/globalweather.asmx
- POST http://www.webservices.com/globalweather.asmx
- POST http://www.webservices.com/globalweather.asmx
- POST http://localhost:8080/v1/globalweather/GetCitiesB
- POST http://localhost:8080/v1/globalweather/GetCitiesB
- POST http://localhost:8080/globalweather/GetCitiesByCo
- POST http://localhost:8080/v1/globalweather/GetCitiesByCo

Builder

POST http://www.webservices.com/globalweather.asmx

Authorization Headers (2) Body Pre-request Script Tests

Key	Value	Description
Content-Type	text/xml	
SOAPAction	http://www.webserviceX.NET/GetWeather	

New key Value Description

Body Cookies Headers (9) Test Results

Status: 200 OK Time: 626 ms Size: 598 B

Pretty Raw Preview XML

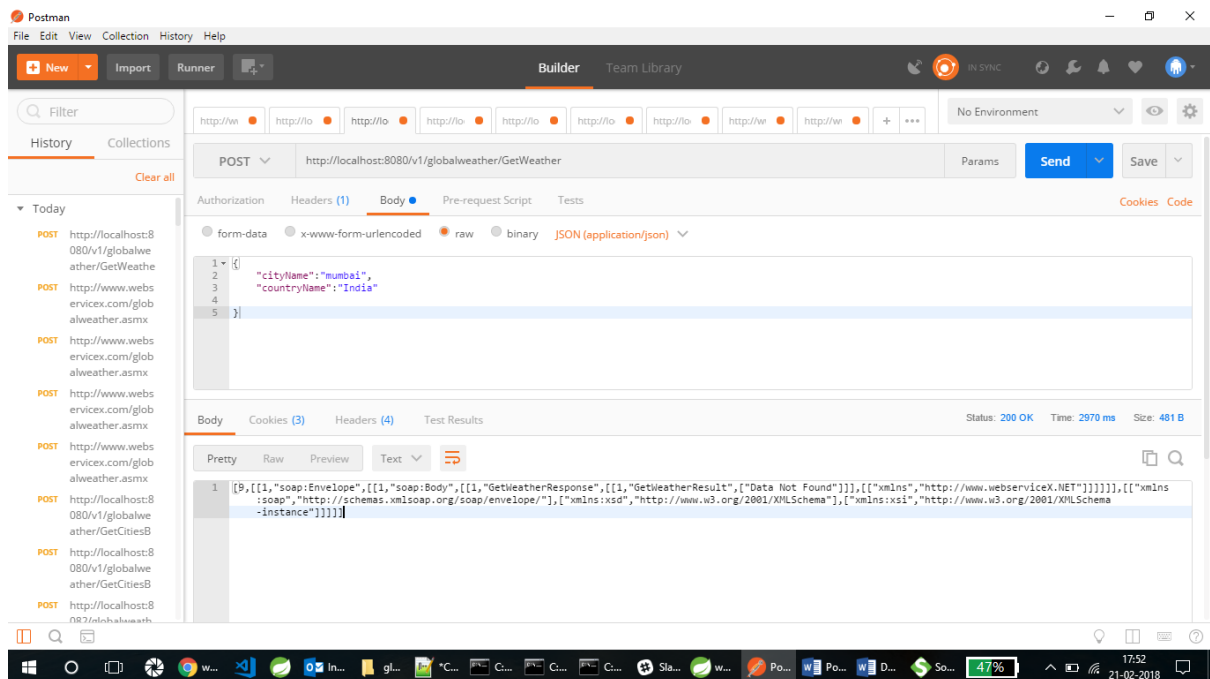
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
3   <soap:Body>
4     <GetWeatherResponse xmlns="http://www.webserviceX.NET">
5       <GetWeatherResult>Data Not Found</GetWeatherResult>
6     </GetWeatherResponse>
7   </soap:Body>
8 </soap:Envelope>
```

47%

17:51 21-02-2018

## 4.2. Proxy Request & Response

URL	<a href="http://www.webserviceX.com/globalweather.aspx">http://www.webserviceX.com/globalweather.aspx</a>
SOAPAction	http://www.webserviceX.NET/GetWeather
Request	<pre>{   "cityName":"mumbai",   "countryName":"India" }</pre>
Response	<pre>[9,[[1,"soap:Envelope",[[1,"soap:Body",[[1,"GetWeatherResponse", [[1,"GetWeatherResult",["Data Not Found"]]]],[{"xmlns":"http://www.webserviceX.NET"}]]]]],[{"xmlns:s oap","http://schemas.xmlsoap.org/soap/envelope/"},"{"xmlns:xsd"," http://www.w3.org/2001/XMLSchema"},"{"xmlns:xsi","http://www. w3.org/2001/XMLSchema-instance"}]]]]]</pre>



## 5. Additional Features that can be implemented

# Consul - <https://www.consul.io/api/agent.html>

Used for Service Discovery, Health Checking, Multidatcenter, DevOps

,

# Central Configuration Service

Used for central configuration which is shared across other services

# DevOps

Automated CI & CD using Jenkins, Vagrant, Ansible, Selenium Testing etc.