

Data Driven Virtual Service - How to Add More Horse Power to Your Virtual Service

This feature has born out of a very practical need. While recording is a great way to create a virtual service, it has limited fire power when it comes to the extent of data set (Request - Response pair) that the virtual service can hold this way.

Most often, you may want to LOOKUP or REFER to an external data set (an XLSX or CSV file) to get a response based on certain values in the request. This ability is what is offered by Data-Driving your virtual service.

This feature allows you refer to a .xlsx file or a .csv file for a valid response based on request parameters. The matching row is identified and the response is returned. For example, see below an example of a simple virtual service that is supposed to return certain basic details about a COUNTRY name which is passed as INPUT.

Country Name	Capital	Currency	Official Language	Head of Government
Afghanistan	Kabul	Afghani	Dari Persian; Pashto	President - Ashraf Gani
Albania	Tirane	Lek	Albanian	Prime Minister – Edi Rama
Algeria	Algiers	Dinar	Arabic; Tamazight; French	Prime Minister – Ahmed Ouyahia
Andorra	Andorra la Vella	Euro	Catalan	Antoni Martí
Angola	Luanda	New Kwanza	Portuguese	President – João Lourenço

Assume that your backend service accepts the following request and response format.

Request	Response
<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://www.webservicex.net/"> <soapenv:Header/> <soapenv:Body> <web:GetDetailsByCountry> <web:Country>India</web:Country> </web:GetDetailsByCountry> </soapenv:Body> </soapenv:Envelope></pre>	<pre><?xml version="1.0" encoding="UTF-8" standalone="no"?> <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"> <soap:Body> <GetDetailsByCountryResponse xmlns="http://www.webservicex.net/"> <Country>India</Country> <Capital>New Delhi</Capital> <Currency>Rupee</Currency> <Language>Hindi</Language> <HeadOfGovt>Prime Minister – Narendra Modi</HeadOfGovt> </GetDetailsByCountryResponse> </soap:Body> </soap:Envelope></pre>

In order to fully virtualize your backend, you have few options:

Approach	Pros	Cons
Record with all the possible request parameters	Easy to set up recording Works well for limited set of data	Doesn't scale for large volume of varied requests Takes very long time to record all possible inputs and responses
Start with a basic virtual service using an RR pair and then let the VS be deployed in learning mode	Easy to create the VS and deploy	May take a very long time to capture all possible request-response pairs to be recorded and have the VS fully ready to replace the live system

<p>Data Drive your virtual service</p>	<p>Quick - All the possible request inputs and corresponding responses can be exported from a Database to a .xlsx or .csv file and powers your VS</p> <p>Flexible - Incremental updates to your request-response data store could be handled very easily just by adding new set of rows to the data set and re-deploy the VS</p> <p>Efficient - This is by far the most efficient way to deal with large set of request-response pairs</p> <p>Closer to reality - By having support for large data sets readily available, the Virtual Service very closely mimics the real service</p>	<p>This can be setup only from PO</p>
--	---	---------------------------------------

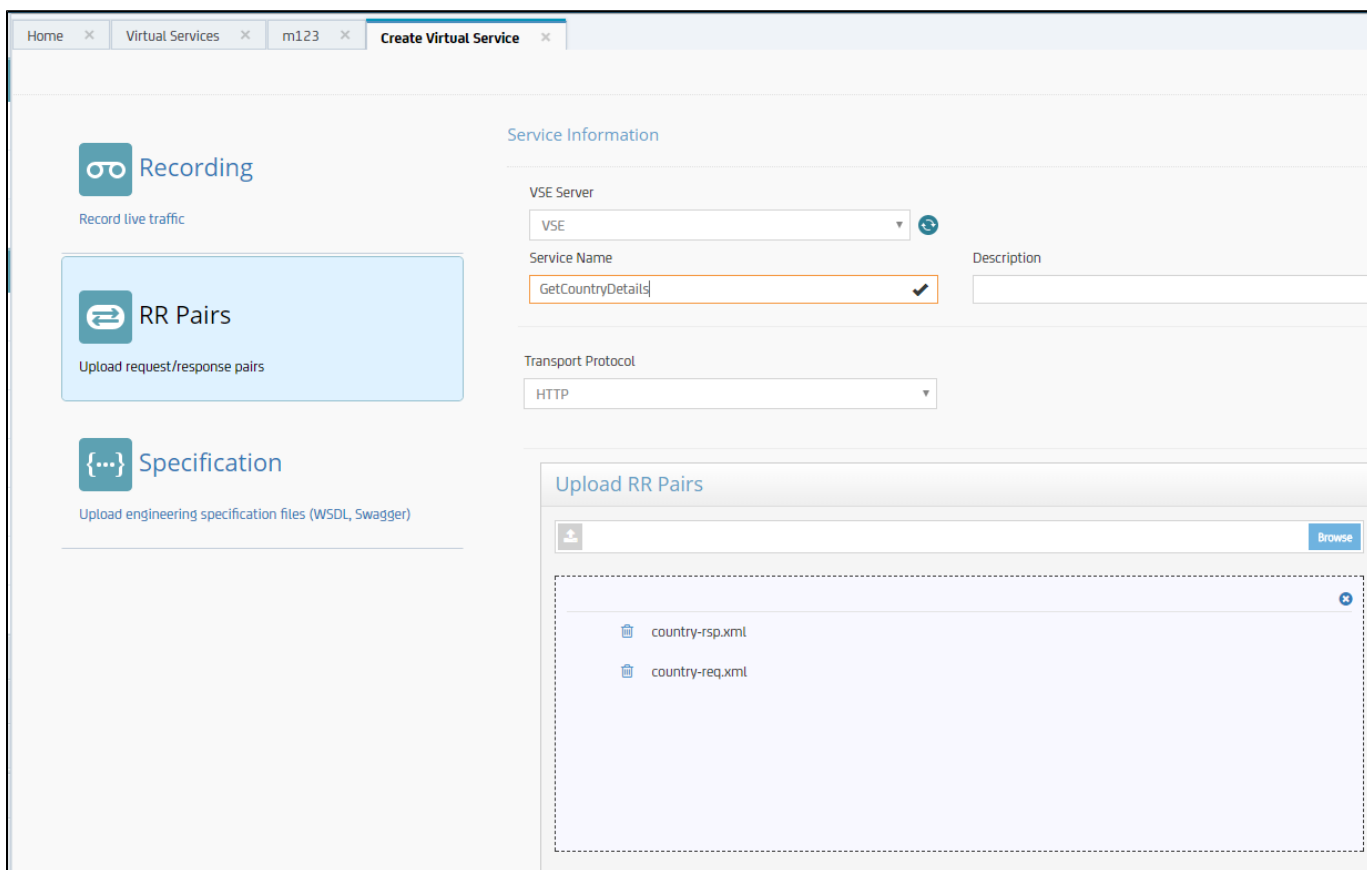
How to create Data Driven Virtual Service GetDetailsByCountry ?

Step 1 Create a basic virtual service using RR pairs

Login to Portal

Create VS using RR pair files attached here: [country-req.xml](#) [country-rsp.xml](#)

Start your VS



Virtual Services | m123 | **GetCountryDetails**

Create Virtual Service

Service Name	GetCountryDetails	VSE Server	VSE
Description	Enter Description	Transport Protocol	HTTP

HTTP Transport Protocol

1 Collect Transactions → 2 Configure → 3 Configure DPH

Request/Response Data Protocol

Request Data Protocols

Request - Available Data Protocols	Request - Selected Data Protocols
Web Service (SOAP) CICS Copybook Data Protocol REST Data Protocol XML Data Protocol Copybook Data Protocol	Web Service (SOAP)

Response Data Protocols

Response - Available Data Protocols	Response - Selected Data Protocols
CICS Copybook Data Protocol Copybook Data Protocol	

Create Virtual Service

Service Name	GetCountryDetails	VSE Server	VSE
Description	Enter Description	Transport Protocol	HTTP

HTTP Transport Protocol

1 Collect Transactions → 2 Configure → 3 Configure DPH

[Req] Web Services (SOAP) This Data Protocol does not require any configuration.

← Prev Next →

← Prev Revert Save Save and Next →

Step 2 Test your Deployed VS

Using WS, create a test case and add a single step - Web service execution (XML) and hit the end point of the VS deployed from step 1. Use the XML request from country-req.xml file. The VS would promptly respond with the recorded response.

TEST CASE: m123_20190404-154425895.tst

Virtual Services	Total	Running	Offline	Fail
	30	3	27	0

Actions	Category	Name	Resource/Type	Status	Txn Count
		BLOG1	17865 : http://	Running	3
		m123	44414 : http://	Running	6
		GetCountryDetails	56581 : http://	Running	3

Step 3 - Data Drive your Virtual Service

The next step is to data drive the VS so that the VS gets more horse power and makes itself useful. The VS as of now knows only about India. It must be fed with knowledge about remaining 202 countries of the world. Let's see how to add more horse power to your VS with some simple steps.

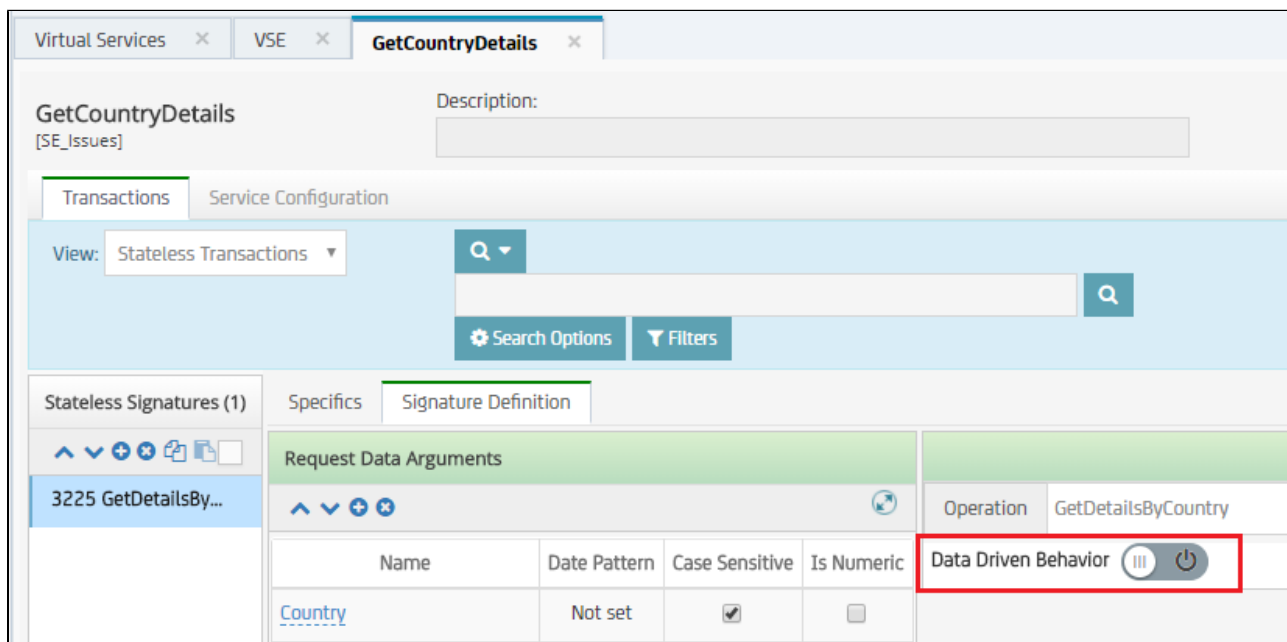
Prepare your data set as an .xlsx or .csv file with the column names occupying first row. For this specific example, I quickly prepared the sheet by taking all the data that is available at <https://www.bankexamstoday.com/2018/05/countries-capital-currency-and-languages.html>

My data sheet looks like what is shown below. The same has been attached here: [countries.xlsx](#)

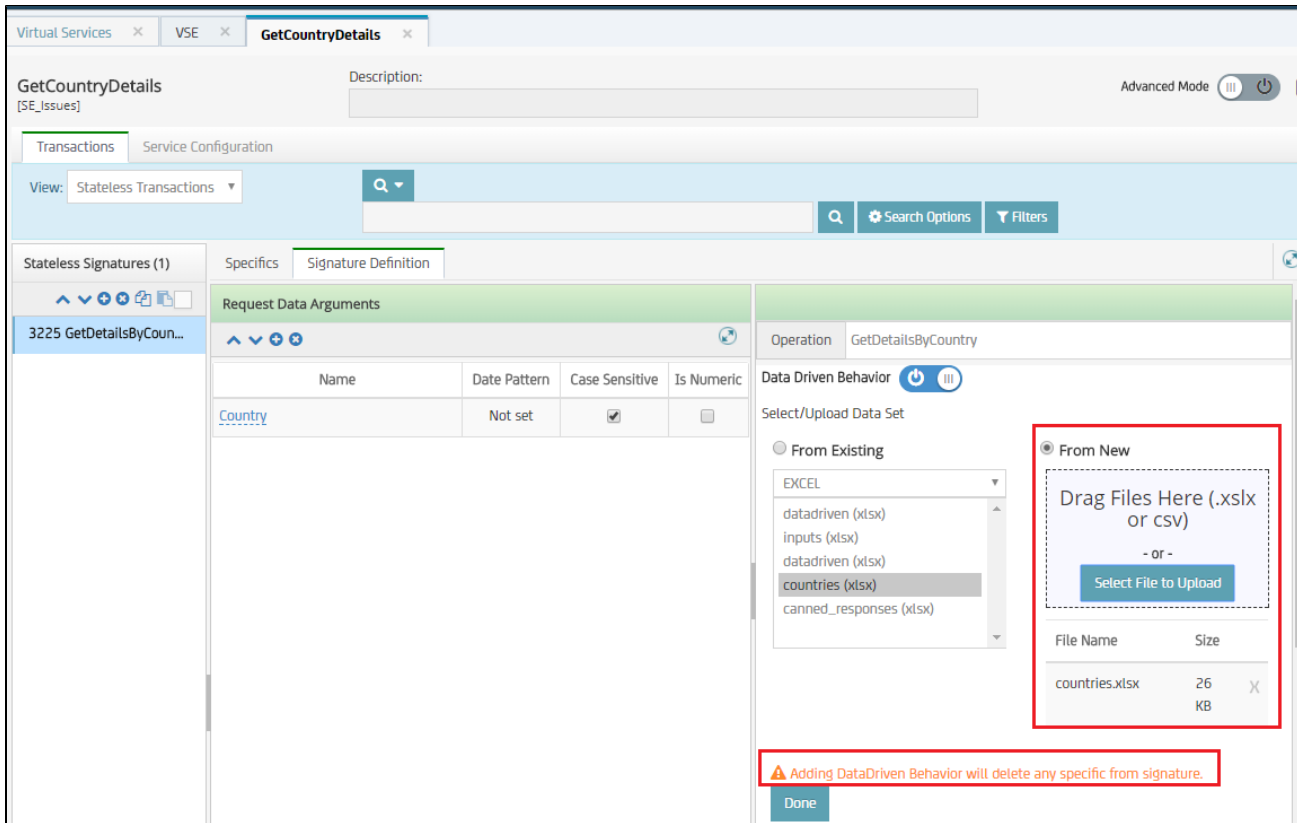
	A	B	C	D	E
1	Country	Capital	Currency	Language	Head
2	Afghanistan	Kabul	Afghani	Dari Persian; Pashto	President - Ashraf Gani
3	Albania	Tirane	Lek	Albanian	Prime Minister – Edi Rama
4	Algeria	Algiers	Dinar	Arabic; Tamazigh t; French	Prime Minister – Ahmed Ouyahia
5	Andorra	Andorra la Vella	Euro	Catalan	Antoni Martí
6	Angola	Luanda	New Kwanza	Portugues e	President – João Lourenço
	Antigua and Barbuda	Saint John's	East Caribbean dollar	English	Prime Minister – Gaston

Once the data sheet is ready, its time to re-wire our VS by adding Data-Driven behavior. You do this by following simple steps:

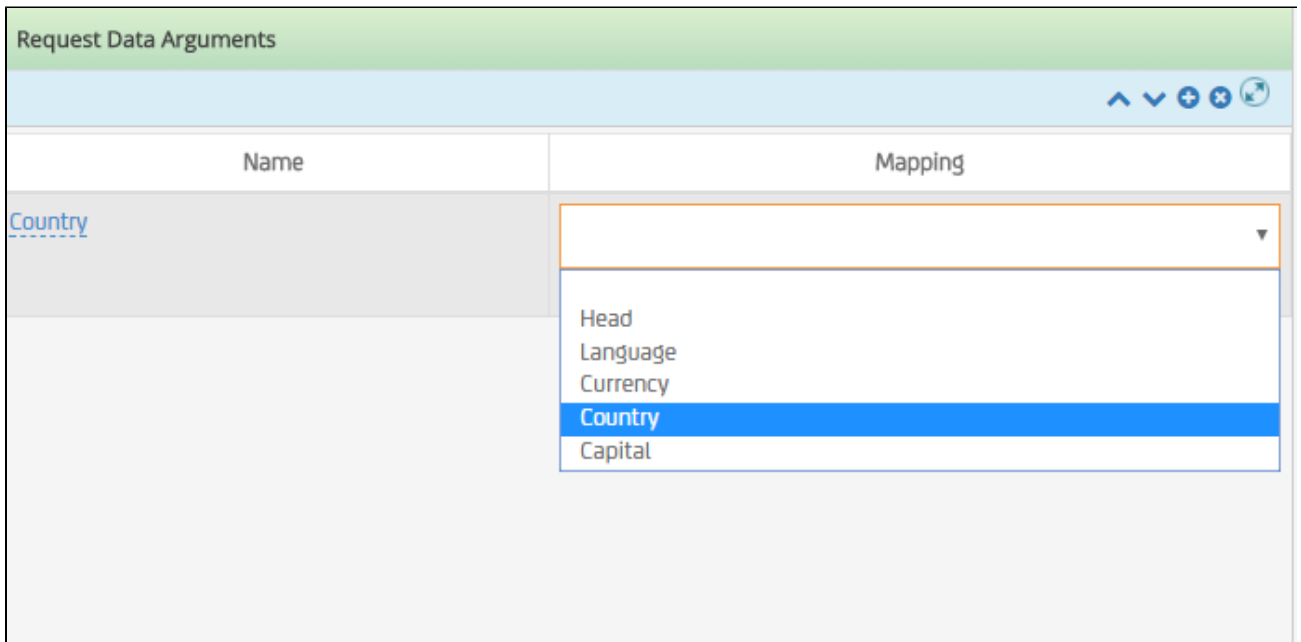
1. Go to portal
2. Go to Manage VS
3. Locate the VS that we created in Step 1
4. Edit the VS
5. Go to 'Signature Definition' tab and Turn ON 'Data Driven Behavior' by gently moving the slider to right.



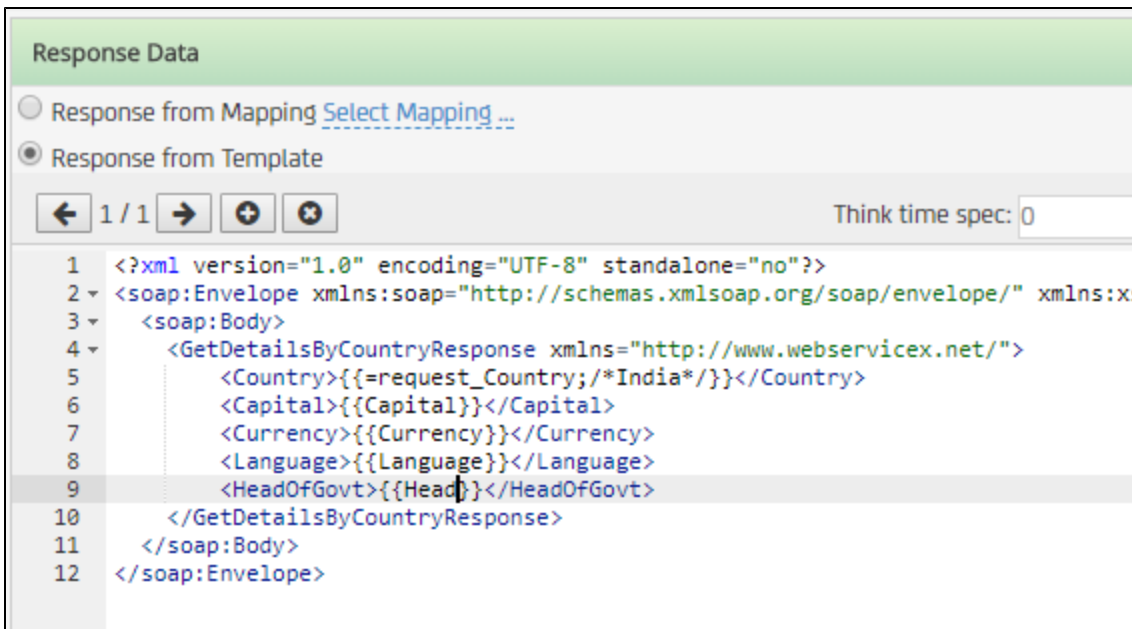
6. You would be shown an option to data drive using existing files known to the system or upload a new file. I uploaded the countries.xlsx file and click **Done**. **This will upload the file and read the column headers so that we can map different columns for input and response**



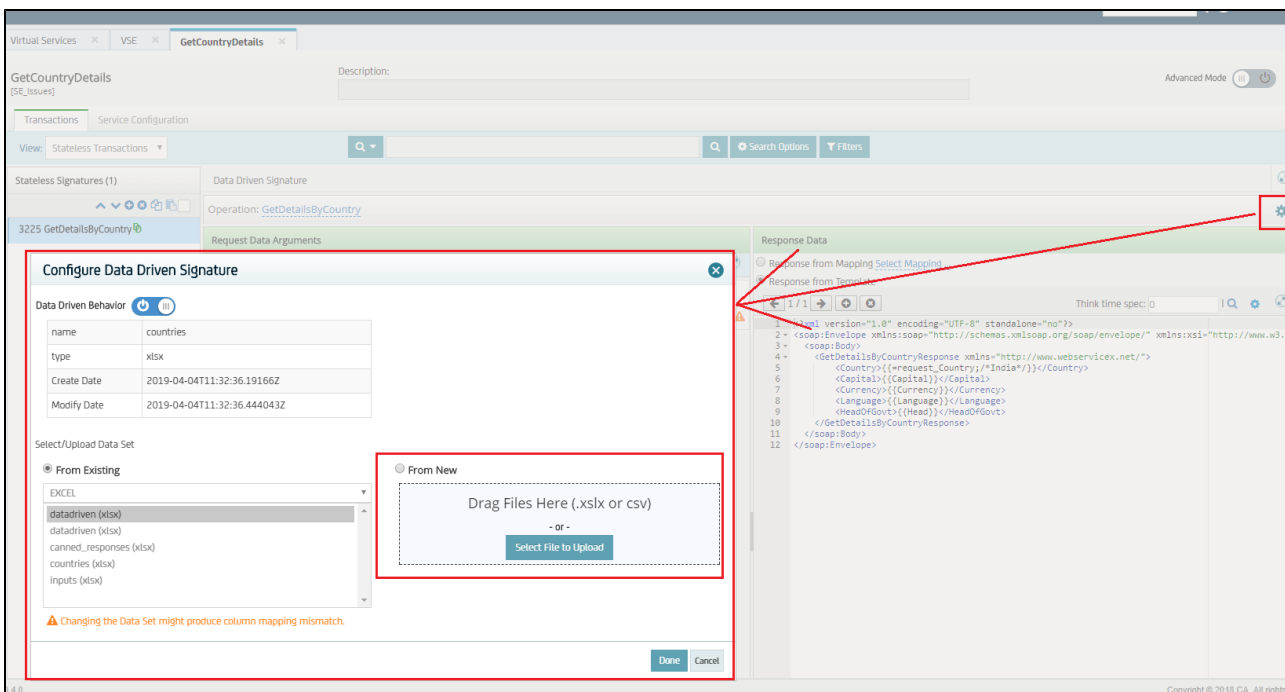
- Once the .xlsx file is uploaded, we move on to define how to map request and response arguments. For request, based on the XML request, the editor shows the argument and we need to identify the right column header from the .xlsx file. For simplicity, I named the columns the same as that of XML tag name. You are free to have column names as you wish. There is no restriction as such.



- Map the response arguments as next step and you have a choice here. You could either map a specific column as the response or you could fill in a template using values from various columns as shown below. The first method of mapping a single column as response would make sense if the entire XML response is provided in one column. However in the countries.xlsx file, we have separated the details of a country in separate columns. This page auto saves. Once you are done just close the tab.



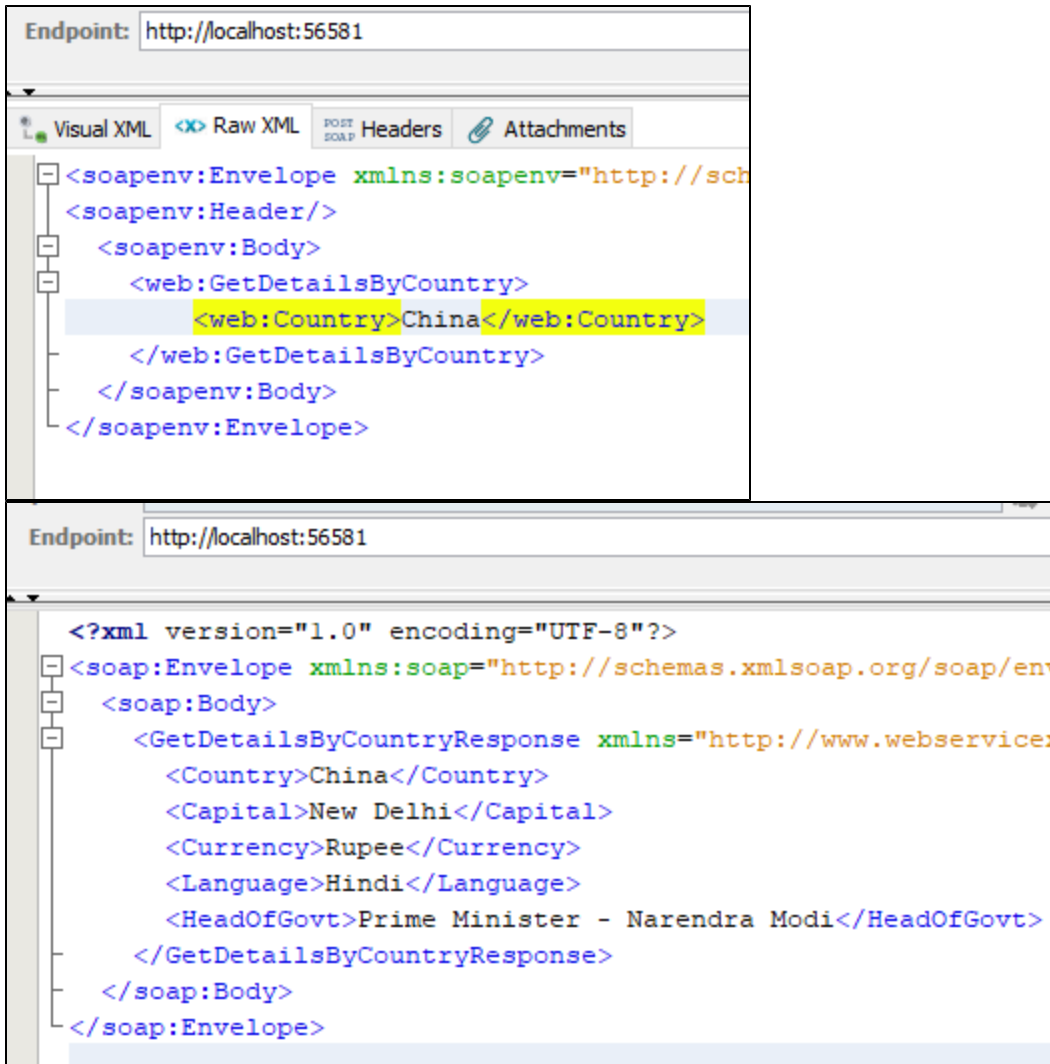
9. We are done with adding more horse power to our VS. Lets check and make sure that the VS indeed is configured correctly. Just open the VS in edit mode and check the xlsx file that it uses. You can also use this gear wheel to change the input file at a later point if needed.



10. Next step is to TEST the new Data Driven behavior and validate the newly enhanced horse power. For this, we go back to Step 2 and pass different country name to the input. This time, I am passing China as input and expecting the following values as response.

China	Beijing	Yuan/Renminbi	Mandarin	Prime Minister – Li Keqiang
-------	---------	---------------	----------	--------------------------------

But, what I got from the running VS was bizarre...I got all the details of India with just the country name changed...What went wrong ???



- The bizarre behavior is because we haven't re-deployed the VS after data driving it. Let's go ahead and re-deploy the VS. You do this from Manage VS



- Once done, check the new behavior and this time, the details of China are correct (as on 4th April 2019). Check few more countries too.


```
Endpoint: http://localhost:56581

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetDetailsByCountryResponse xmlns="http://www.webservices.com/GetDetailsByCountryResponse">
      <Country>China</Country>
      <Capital>Beijing</Capital>
      <Currency>Yuan/Renminbi</Currency>
      <Language>Mandarin</Language>
      <HeadOfGovt>Prime Minister - Li Keqiang</HeadOfGovt>
    </GetDetailsByCountryResponse>
  </soap:Body>
</soap:Envelope>
```

Hope you are as successful as I was and even happier than me when I got this thing working well. Good luck.

MAR file that contains Data driven VS along with required .xlsx file has been attached here: [GetCountryDetails.mar](#)