

06/04/2022:

Agenda:

- What is web Service?
- Basic Concept Behind Web Service
- Why we use it?

Project Application:

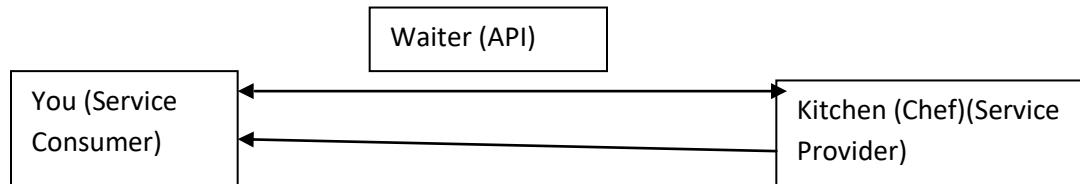
- Front End(UI)
- Backend(Database)
- Service(API)

Web Service:

- Service available over the web

Scenario:

- You Speak in English and the Chef who is present inside kitchen speaks German & you don't know/understand each other in that case here is a person who make sure two applications communicate with each other & whatever response you have to get from the server or whatever response a server has to get from a client is in a format where two entities can understand & communicate with each other



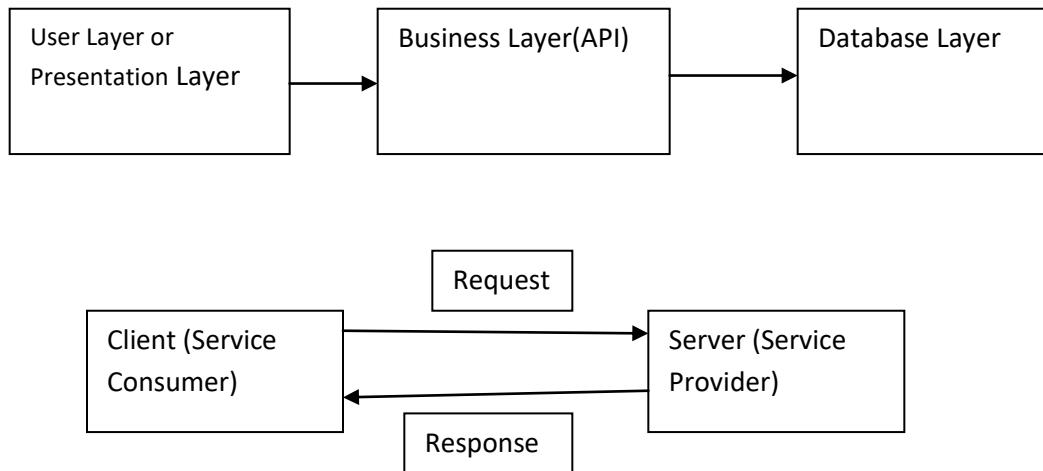
- **Waiter:** Who is doing communication between you and Kitchen
- **Waiter:** Waiter is performing role of Web Service through an API
- We can Say he(waiter) is communicating between two applications and making sure that the communication between two applications in Proper.

Web Service:

- Service available over web
- Enables Communication between applications over the web
- Platform independent Communication

- Using Web Service two different applications can talk to each other & exchange data/information.
- An API is the acronym for Application Programming Interface, which is a software intermediary that allows two applications to talk to each other.
- Each time you use an APP Like Facebook, send an instant message or check the weather on your phone, you are using an API.

What is API?



Medium: HTTP

Format: XML/JSON

Two Types of Web Services:

- SOAP Service
- REST Service

08/04/2022:

Agenda: Components of Web Services (WSDL, UDDI) Types of Web Services(SOAP)

WSDL: Web Service description Language

- WSDL Stands for Web Service description Language

- WSDL is standard format for describing a web Service.
- WSDL is pronounced as ‘Wiz-dull’ and spelled as ‘W-S-D-L’
- WSDL definition describes how to access a web service and what operations it will Perform
- WSDL is often used in combination with **SOAP** and XML schema to provide web services over the Internet
- A client program connecting to web service can read the WSDL to determine what functions are available on the server.
- WSDL is an XML document with a <definitions> at the root and the child elements,
<types>,
<message>,
<portType>,
<binding>.

<definition>:

- Element must be the root element of all WSDL documents
- It defines the name of the web service
- The definitions element is container for all the Other elements

<types>:

- WSDL <types> element take care of defining the data types that are used by the web service
- WSDL allows the types to be defined in separate elements so that the types are reusable with multiple web services

```

▼<wsdl:types>
  ▼<s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
    ▼<s:element name="Add">
      ▼<s:complexType>
        ▼<s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="intA" type="s:int"/>
          <s:element minOccurs="1" maxOccurs="1" name="intB" type="s:int"/>
        </s:sequence>
      </s:complexType>
    </s:element>
  
```



<message>:

- The <message> element describes the data being exchanged between the web service providers and consumers
- Each Web Service has two Messages: Input and Output

```

▼<wsdl:message name="AddSoapIn">
  <wsdl:part name="parameters" element="tns:Add"/>
</wsdl:message>
▼<wsdl:message name="AddSoapOut">
  <wsdl:part name="parameters" element="tns:AddResponse"/>
</wsdl:message>

```

<portType>:

- <portType> can combine one request and one response message into a single request/response operation. This is most commonly used in SOAP Services.
- A PortType can define multiple operations.

```

<wsdl:portType name="CalculatorSoap">
  ▼<wsdl:operation name="Add">
    <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Adds two i
    Online</wsdl:documentation>
    <wsdl:input message="tns:AddSoapIn"/>✓
    <wsdl:output message="tns:AddSoapOut"/>✓
  </wsdl:operation>
  ▼<wsdl:operation name="Subtract">
    <wsdl:input message="tns:SubtractSoapIn"/> ✓
    <wsdl:output message="tns:SubtractSoapOut"/> ✓
  </wsdl:operation>
  ▼<wsdl:operation name="Multiply">
    <wsdl:input message="tns: MultiplySoapIn"/> ✓
    <wsdl:output message="tns: MultiplySoapOut"/> ✓
  </wsdl:operation>
  ▼<wsdl:operation name="Divide">
    <wsdl:input message="tns: DivideSoapIn"/> ✓
    <wsdl:output message="tns: DivideSoapOut"/> ✓
  </wsdl:operation>
</wsdl:portType>

```

<binding>:

- The <binding> element provides specific details on how a portType operation will actually be transmitted over the web.
- The bindings can be made available via multiple transports including HTTP GET, HTTP POST or SOAP.

```

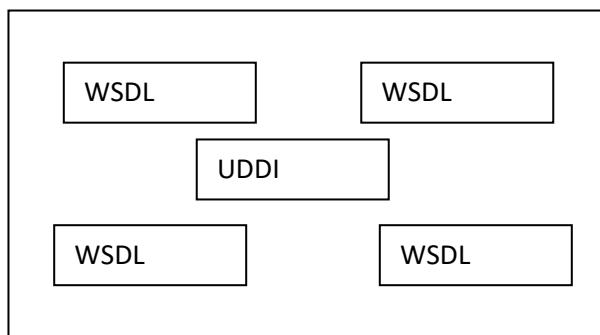
    ▼<wsdl:binding name="CalculatorSoap12" type="tns:CalculatorSoap">
      <soap12:binding transport="http://schemas.xmlsoap.org/soap/http"/>
      ▼<wsdl:operation name="Add">
        <soap12:operation soapAction="http://tempuri.org/Add" style="document"/>
        ▼<wsdl:input>
          <soap12:body use="literal"/>
        </wsdl:input>
        ▼<wsdl:output>
          <soap12:body use="literal"/>
        </wsdl:output>
      </wsdl:operation>
      ▼<wsdl:operation name="Subtract">
        <soap12:operation soapAction="http://tempuri.org/Subtract" style="document"/>
        ▼<wsdl:input>
          <soap12:body use="literal"/>
        </wsdl:input>
        ▼<wsdl:output>

```

WSDL File/Service: <definitions>, <types>, <message>, <port Type>, <binding>

UDDI: Universal Description Discovery and Integration

- Universal description discovery integration is an XML based standard for publishing and finding the web service.
- A web Service Providers publishes his web service (through WSDL) on an online directory from where consumers can query & search the web services
- This is online registry/directory is called UDDI.



Web Services:

- SOAP Service
- REST Service

SOAP Web Service:

- A web service that complies to a SOAP web service specifications is a SOAP Service

What are these Specifications?

- Basic Standards/Specifications: SOAP,WSDL,UDDI

Who Defines and dictates these Specifications?

- W3C(World Wide Web Consortium): An international community that develops open standards for World Wide Web: <https://www.w3c.org>

09/04/2022:

Agenda: SOAP Protocol

SOAP: Simple Object Access Protocol

- SOAP is XML based protocol for exchanging information between the web services
- SOAP is a communication Protocol
- SOAP Provides data transport for Web Services.
- All the information/message exchange happens over a common format: **XML**
- SOAP has defined Structure/Format: **SOAP Message**
- SOAP Message Consists of : **Envelope, Header, Body and Fault**

<https://www.soapui.org/downloads/soapui/>

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
    xmlns:tem="http://tempuri.org/">  
  
    <soapenv:Header/>  
  
    <soapenv:Body>  
        <tem:Add>  
            <tem:intA>?</tem:intA>  
            <tem:intB>?</tem:intB>  
        </tem:Add>  
    </soapenv:Body>  
  
</soapenv:Envelope>
```

Envelope: (Mandatory)

- Every SOAP message has a root envelope element
- Envelope is mandatory Part of SOAP Message

- The SOAP Envelope indicates the start and the end of the message so that the receiver knows when entire message has been received.
- The SOAP envelope solves the problem of knowing when you are done receiving a message and are ready to Process it.

Header: (Optional)

- Contains any optional attributes of the message used in Processing the message

Body: (Mandatory)

- Contains the XML data comprising the message being sent
- The SOAP Body is a mandatory element which contains application defined XML data being exchanged in the SOAP Message.
- The body must be contained within the envelope and must follow any headers that might be defined for the message

Fault: (Optional)

- An optional fault element that provides information about errors that occurred while processing the message.
- When an error occurs during processing the response to a SOAP Message is called a SOAP fault element in the body of the message, and the fault is returned to sender of the SOAP message.

```

<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>
    <soap:Fault>
      <faultcode>soap:Client</faultcode>
      <faultstring>System.Web.Services.Protocols.SoapException: Server was unable to read request. --->
      System.InvalidOperationException: There is an error in XML document (5, 32). --->
      System.FormatException: Input string was not in a correct format.

      at System.Number.StringToNumber(String str, NumberStyles options, NumberBuffer& number,
      NumberFormatInfo info, Boolean parseDecimal)

      at System.Number.ParseInt32(String s, NumberStyles style, NumberFormatInfo info)

      at System.Xml.XmlConvert.ToInt32(String s)

      at Microsoft.Xml.Serialization.GeneratedAssembly.XmlSerializationReader1.Read1_Add()
  
```

at
Microsoft.Xml.Serialization.GeneratedAssembly.ArrayOfObjectSerializer.Deserialize(XmlSerializationReader reader)

at System.Xml.Serialization.XmlSerializer.Deserialize(XmlReader xmlReader, String encodingStyle, XmlDeserializationEvents events)

--- End of inner exception stack trace ---

at System.Xml.Serialization.XmlSerializer.Deserialize(XmlReader xmlReader, String encodingStyle, XmlDeserializationEvents events)

at System.Xml.Serialization.XmlSerializer.Deserialize(XmlReader xmlReader, String encodingStyle)

at System.Web.Services.Protocols.SoapServerProtocol.ReadParameters()

--- End of inner exception stack trace ---

at System.Web.Services.Protocols.SoapServerProtocol.ReadParameters()

at System.Web.Services.Protocols.WebServiceHandler.CoreProcessRequest()</faultstring>

<detail/>

</soap:Fault>

</soap:Body>

</soap:Envelope>

The screenshot shows the SoapUI.org website with a dark header bar. The header includes the SoapUI logo, navigation links for Tools, Learn, Docs, and Resources, a search icon, a shopping cart icon, and a yellow button labeled "Try ReadyAPI Free".

ReadyAPI section:

- Icon: A yellow gear with blue lines.
- Section title: ReadyAPI
- Description: Get the most advanced functional testing tool for REST, SOAP and GraphQL APIs.
- Call-to-action: Download ReadyAPI
- Link: Learn More
- Features (list):
 - ✓ SOAP API Testing
 - ✓ REST API Testing
 - ✓ WSDL Coverage
 - ✓ Scripted Assertions
 - ✓ Largest Online API Testing

SoapUI Open Source section:

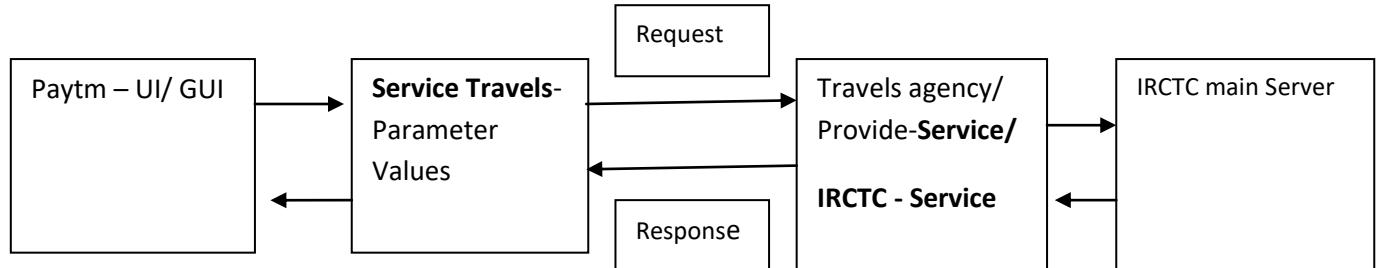
- Icon: A yellow gear with blue lines.
- Section title: SoapUI Open Source
- Description: Get the open source version of the most widely used API testing tool in the world.
- Call-to-action: Download SoapUI Open Source
- Link: Learn More
- Features (list):
 - ✓ SOAP API Testing
 - ✓ REST API Testing
 - ✓ WSDL Coverage
 - ✓ Scripted Assertions
 - ✓ Largest Online API Testing

A small orange box with a question mark icon and the text "Welcome! Do you need help with anything?" is visible on the right side of the page.

API Testing:

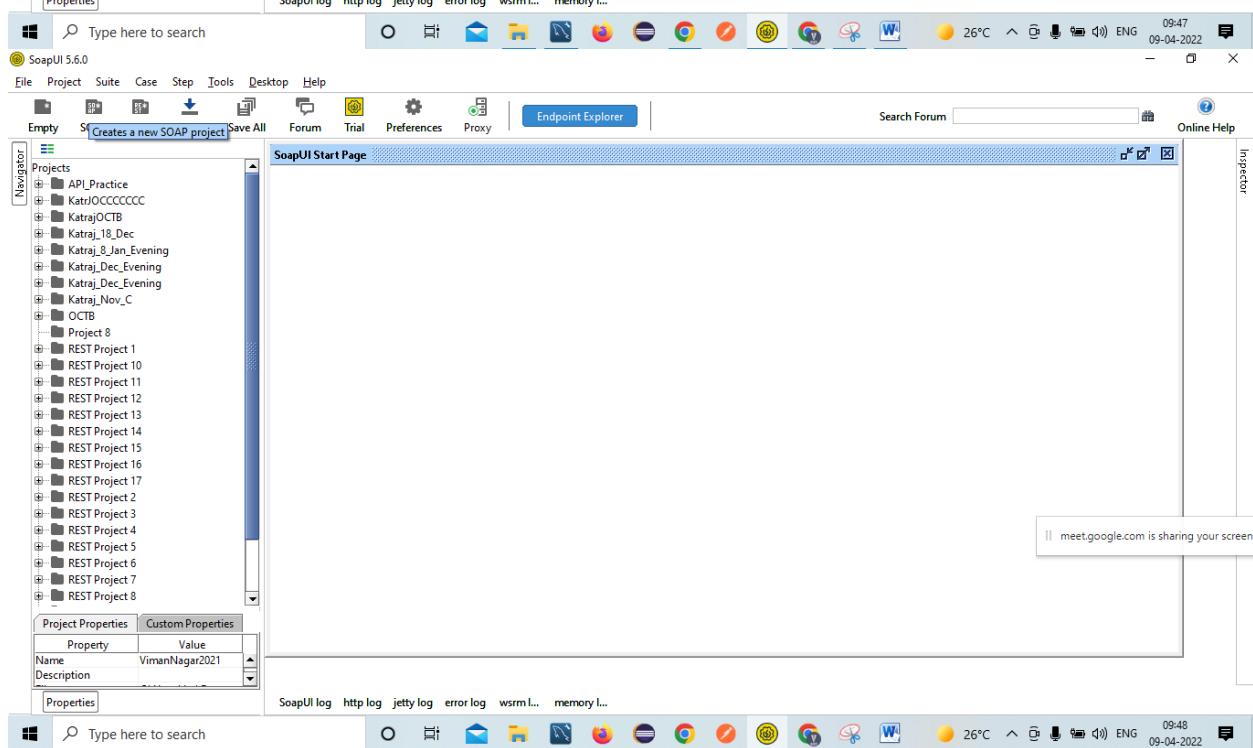
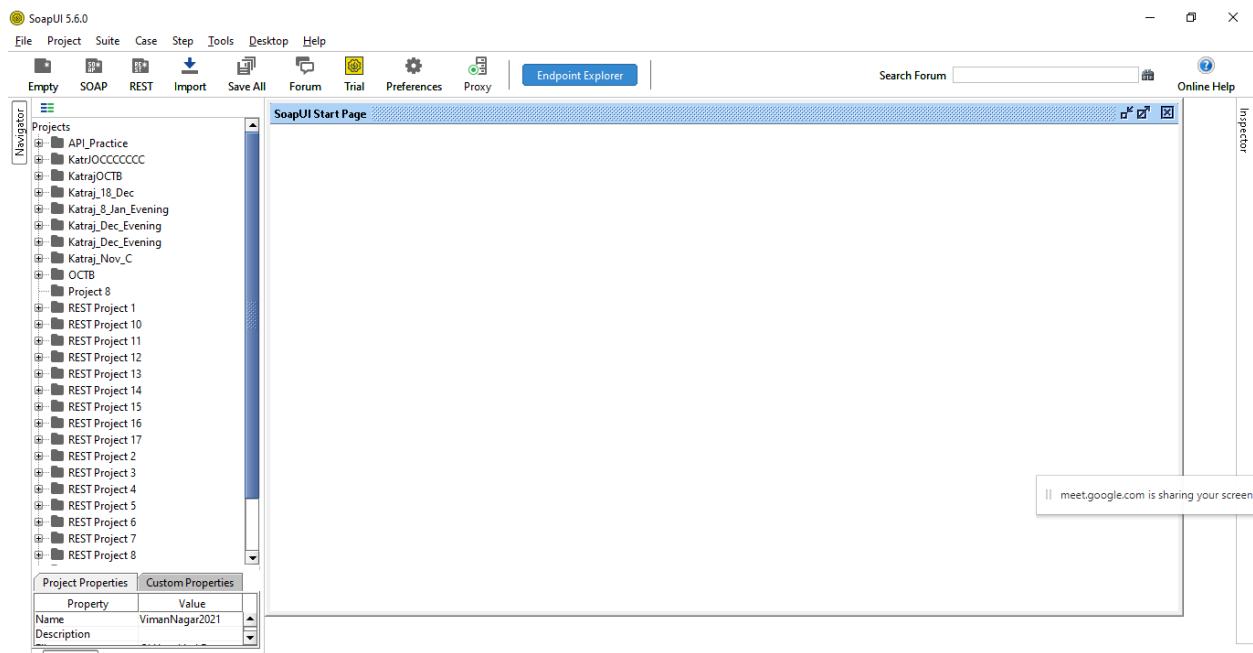
2 Types of Web Services

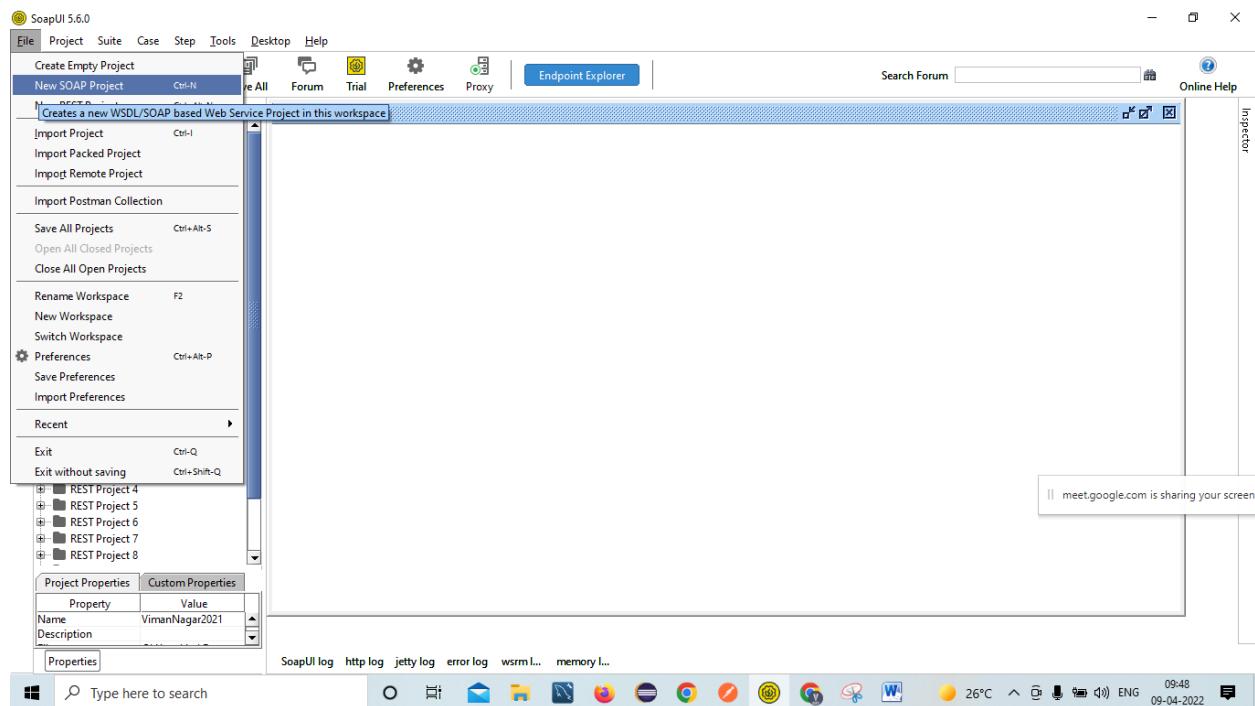
- **SOAP** Service(SOAPUI Tool)
- **REST** Service(Soapui and Postman)

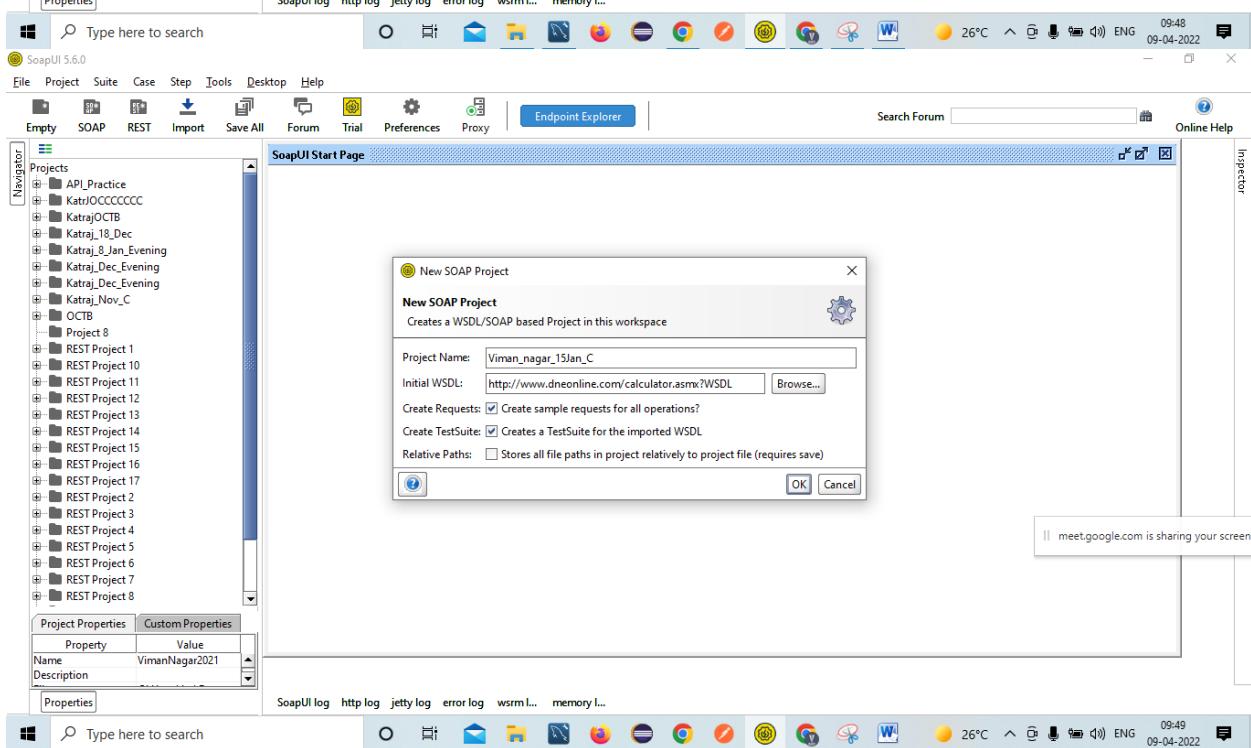
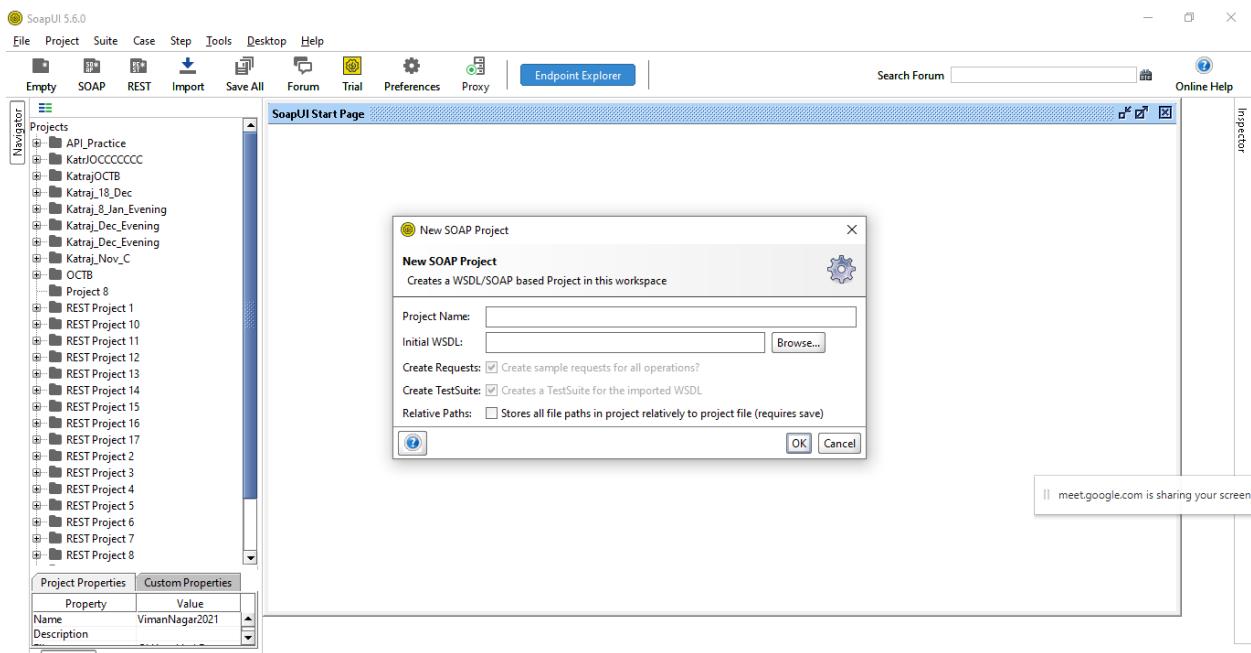


SOAP Service : (SOAPUI Tool)

- Developer will provide WSDL file: <http://www.dneonline.com/calculator.asmx?WSDL>
- Unit Testing Document
- Time taken for the response=5Secs







SoapUI Start Page

The SoapUI interface shows a project structure in the Navigator pane. A blue arrow points from the 'CalculatorSoap' node under 'Viman_nagar_15Jan_C' to the Request Editor pane.

Request 1

The Request Editor displays an XML message:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope">
  <soapenv:Header>
  <soapenv:Body>
    <tem:Add>
      <tem:intA>?</tem:intA>
      <tem:intB>?</tem:intB>
    </tem:Add>
  </soapenv:Body>
</soapenv:Envelope>

```

Properties

Property	Value
Name	Request 1
Description	
Message Size	291

Log

Logs show the status of the request:

- 09:52 - Request 1 sent to http://www.dneonline.com/calculator.asmx
- 09:52 - Request 1 received response
- 09:52 - Response status: OK

SoapUI 5.6.0

File Project Suite Case Step Tools Desktop Help

Empty SOAP REST Import Save All Forum Trial Preferences Proxy Endpoint Explorer Search Forum Online Help

SoapUI Start Page

SD API Request 1 Submit request to specified endpoint URL (Alt+Enter) = "http://www.dneonline.com/calculator.asmx"

Raw XML

```
<soapenv:Header>
<soapenv:Body>
<tem:Add>
<tem:intA>1</tem:intA>
<tem:intB>1</tem:intB>
<tem:Add>
</Soapenv:Body>
</soapenv:Envelope>
```

Raw XML

Request Properties

Property	Value
Name	Request 1
Description	
Message Size	291

Properties

SoapUI log http log jetty log error log wsrm... memory l...

File Project Suite Case Step Tools Desktop Help

Empty SOAP REST Import Save All Forum Trial Preferences Proxy Endpoint Explorer Search Forum Online Help

SoapUI Start Page

SD API Request 1 Submit request to specified endpoint URL (Alt+Enter) = "http://www.dneonline.com/calculator.asmx"

Raw XML

```
<soap:Envelope 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">
<soap:Header>
<soap:Body>
<AddResponse xmlns="http://tempuri.org/">
<AddResult>2</AddResult>
</AddResponse>
</soap:Body>
</soap:Envelope>
```

Raw XML

Request Properties

Property	Value
Name	Request 1
Description	
Message Size	291

Properties

SoapUI log http log jetty log error log wsrm... memory l...

Test Cases:

1. Validating SOAP Response=Pass

The screenshot shows the SoapUI interface with the following details:

- Toolbar:** File, Project, Suite, Case, Step, Tools, Desktop, Help.
- Navigator:** REST Project 13, REST Project 14, REST Project 15, REST Project 16, REST Project 17, REST Project 2, REST Project 3, REST Project 4, REST Project 5, REST Project 6, REST Project 7, REST Project 8, REST Project 9, Sample, VimanNagar2021, Viman_nagar_15Jan_C, CalculatorSoap, Add, Divide, Multiply, Subtract, CalculatorSoap12, Add, Divide, Multiply, Subtract.
- Request Properties Table:**

Property	Value
Name	Request 1
Description	
Message Size	291
- Request 1:** Request type is SOAP. The XML request is:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<soapenv:Body>
<tem:Add>
<tem:intA>1</tem:intA>
<tem:intB>1</tem:intB>
</tem:Add>
</soapenv:Body>
</soapenv:Envelope>
```

- Response:** The XML response is identical to the request, indicating a successful response. A blue arrow points to the <AddResponse> tag in the response XML.
- Headers:** Headers (11), Attachments (0), SSL Info, WSS (0), JMS (0).
- Time:** response time: 839ms (325 bytes)
- Status:** 7:17

2. Validating data and count of data in the response=Pass

The screenshot shows the SoapUI interface with the following details:

- Toolbar:** File, Project, Suite, Case, Step, Tools, Desktop, Help.
- Navigator:** REST Project 13, REST Project 14, REST Project 15, REST Project 16, REST Project 17, REST Project 2, REST Project 3, REST Project 4, REST Project 5, REST Project 6, REST Project 7, REST Project 8, REST Project 9, Sample, VimanNagar2021, Viman_nagar_15Jan_C, CalculatorSoap, Add, Divide, Multiply, Subtract, CalculatorSoap12, Add, Divide, Multiply, Subtract.
- Request Properties Table:**

Property	Value
Name	Request 1
Description	
Message Size	291
- Request 1:** Request type is SOAP. The XML request is:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<soapenv:Body>
<tem:Add>
<tem:intA>1</tem:intA>
<tem:intB>1</tem:intB>
</tem:Add>
</soapenv:Body>
</soapenv:Envelope>
```

- Response:** The XML response is identical to the request, indicating a successful response.
- Headers:** Headers (11), Attachments (0), SSL Info, WSS (0), JMS (0).
- Time:** response time: 839ms (325 bytes)
- Status:** 4:35

3. Validating Tagname/Attributes Present in the response=EC=AddResult=>Pass

The screenshot shows the SoapUI interface with the following details:

- Request Properties:**

Property	Value
Name	Request 1
Description	Viman_nagar_15Jan_C
Message Size	291
- Response XML:**

```

<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:Header/>
  <soap:Body>
    <AddResponse xmlns="http://tempuri.org/">
      <AddResult>2</AddResult>
      <AddResult>2</AddResult>
    </AddResponse>
  </soap:Body>
</soap:Envelope>

```
- Headers:**

Header	Value
Cache-Control	private, max-age=0
Server	Microsoft-IIS/10.0
Content-Encoding	gzip
X-AspNet-Version	2.0.50727
Vary	
Status#	HTTP/1.1 200 OK
X-Powered-By	PleskWin
Content-Length	348

4. Validating different status codes=EC-200-OK=Pass

The screenshot shows the SoapUI interface with the following details:

- Request Properties:**

Property	Value
Name	Request 1
Description	Viman_nagar_15Jan_C
Message Size	291
- Response XML:**

```

<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:Header/>
  <soap:Body>
    <AddResponse xmlns="http://tempuri.org/">
      <AddResult>2</AddResult>
      <AddResult>2</AddResult>
    </AddResponse>
  </soap:Body>
</soap:Envelope>

```
- Headers:**

Header	Value
Cache-Control	private, max-age=0
Server	Microsoft-IIS/10.0
Content-Encoding	gzip
X-AspNet-Version	2.0.50727
Vary	
Status#	HTTP/1.1 200 OK
X-Powered-By	PleskWin
Content-Length	348

5. Validating time taken for the response=EC=5Secs=Pass

The screenshot shows the SoapUI interface with the following details:

- File Menu:** File, Project, Suite, Case, Step, Tools, Desktop, Help.
- Toolbar:** Empty, SOAP, REST, Import, Save All, Forum, Trial, Preferences, Proxy, Endpoint Explorer, Search Forum, Online Help.
- Navigator:** Shows a list of REST projects and a specific project named "VimanNagar15Jan_C" which contains a "CalculatorSoap" interface.
- Request View:** Request 1 is selected, showing the XML request and response. The request XML is:


```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope">
        <soapenv:Header>
        <soapenv:Body>
          <tem:Add>
            <tem:intA>1</tem:intA>
            <tem:intB>1</tem:intB>
          </tem:Add>
        </soapenv:Body>
      </soapenv:Envelope>
```

 The response XML is:


```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope">
        <soap:Header>
        <soap:Body>
          <AddResponse xmlns="http://tempuri.org/">
            <AddResult>2</AddResult>
          </AddResponse>
        </soap:Body>
      </soap:Envelope>
```
- Request Properties:** Name: Request 1, Description: , Message Size: 291.
- Bottom Status Bar:** response time: 839ms (325 bytes).

6. Applying Assertions for Verification.

The screenshot shows the SoapUI interface with the following details:

- File Menu:** File, Project, Suite, Case, Step, Tools, Desktop, Help.
- Toolbar:** Empty, SOAP, REST, Import, Save All, Forum, Trial, Preferences, Proxy, Endpoint Explorer, Search Forum, Online Help.
- Navigator:** Shows a list of REST projects and a specific project named "VimanNagar15Jan_C" which contains a "CalculatorSoap" interface.
- Request View:** Request 1 is selected, showing the XML request and response. The request XML is identical to the one in the previous screenshot.
- Context Menu (Open with right-click):**
 - Show Interface Viewer
 - Add JMS endpoint
 - Generate Code
 - Check WSDL Compliance
 - Launch TcpMon
 - Generate TestSuite** (highlighted)
 - Generate SOAP Mock Service
 - Generate Documentation
- Bottom Status Bar:** Headers (9) Attachments (..) SSL Info WSS (0) JMS (0) 1:1.

SoapUI Start Page

Request

```

<soap Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soap Header>
        <soap Body>
            <Item>
                <Add>
                    <item:intA>2</item:intA>
                    <item:intB>2</item:intB>
                </Add>
            </Item>
        </soap Body>
    </soap Envelope>

```

Response

```

<soap Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soap Body>
        <AddResponse xmlns="http://tempuri.org/">
            <AddResult>4</AddResult>
        </AddResponse>
    </soap Body>
</soap Envelope>

```

Headers (0) Attachments (0) SSL Info WSS (0) JMS (0)

Assertions (0) Request Log (1)

response time: 601ms (323 bytes)

Custom Properties

TestRequest Properties	
Property	Value
Name	Add

SoapUI 5.6.0

File Project Suite Case Step Tools Desktop Help

Empty SOAP REST Import Save All Forum Trial Preferences Proxy Endpoint Explorer Search Forum Online Help

Navigator

- REST Project 13
- REST Project 14
- REST Project 15
- REST Project 16
- REST Project 17
- REST Project 2
- REST Project 3
- REST Project 4
- REST Project 5
- REST Project 6
- REST Project 7
- REST Project 8
- REST Project 9
- Sample
- VimanNagar2021
- Viman_nagar_15Jan_C
- CalculatorSoap
- CalculatorSoap12
- CalculatorSoap12 TestSuite
- Add
- Divide
- Multiply
- Subtract
- Test Steps (1)
- Add

Custom Properties

TestRequest Properties	
Property	Value
Name	Add

Properties

SoapUI log http log jetty log Type here to search Stop sharing Hide 24°C ENG 09:12 11-04-2022

SoapUI 5.6.0

File Project Suite Case Step Tools Desktop Help

Empty SOAP REST Import Save All Forum Trial Preferences Proxy Endpoint Explorer Search Forum Online Help Inspector

Navigator

REST Project 13
REST Project 14
REST Project 15
REST Project 16
REST Project 17
REST Project 2
REST Project 3
REST Project 4
REST Project 5
REST Project 6
REST Project 7
REST Project 8
REST Project 9
Sample
VimanNagar2021
Viman_nagar_15Jan_C
CalculatorSoap
CalculatorSoap12
CalculatorSoap12 TestSuite
Add TestCase
Test Steps (1)
Add

Custom Properties

TestRequest Properties

Property	Value
Name	Add

Properties

SoapUI Start Page

Add Assertion

Add Assertion

Select the source property and which assertion to apply below

Assertions

Recently used

Property Content

Contains

Searches for the existence of a string token in the property value, supports regular expressions. Applicable to any property.

JsonPath Count

Uses an JsonPath expression to count the occurrences of an element. Applicable to any property containing JSON.

JsonPath Existence Match

Uses an JsonPath expression to select content from the target property and compares the result to an expected value. Applicable to any property containing JSON.

JsonPath Match

Uses an JsonPath expression to existence of a node and compares the result to an expected value. Applicable to any property containing JSON.

JsonPath RegEx Match

Uses an JsonPath expression to select content from the target property and compares the result to an specified RegEx. Applicable to any property containing JSON.

Add Cancel

Assertions (0) Request Log (1)

response time: 601ms (323 bytes)

SoapUI log http log jetty log Stop sharing Hide

Type here to search

09:12 11-04-2022

File Project Suite Case Step Tools Desktop Help

Empty SOAP REST Import Save All Forum Trial Preferences Proxy Endpoint Explorer Search Forum Online Help Inspector

Navigator

REST Project 13
REST Project 14
REST Project 15
REST Project 16
REST Project 17
REST Project 2
REST Project 3
REST Project 4
REST Project 5
REST Project 6
REST Project 7
REST Project 8
REST Project 9
Sample
VimanNagar2021
Viman_nagar_15Jan_C
CalculatorSoap
CalculatorSoap12
CalculatorSoap12 TestSuite
Add TestCase
Test Steps (1)
Add

Custom Properties

TestRequest Properties

Property	Value
Name	Add

Properties

SoapUI Start Page

Add Assertion

Add Assertion

Select the source property and which assertion to apply below

Assertions

Recently used

Property Content

Contains

Searches for the existence of a string token in the property value, supports regular expressions. Applicable to any property.

JsonPath Count

Uses an JsonPath expression to count the occurrences of an element. Applicable to any property containing JSON.

JsonPath Existence Match

Uses an JsonPath expression to select content from the target property and compares the result to an expected value. Applicable to any property containing JSON.

JsonPath Match

Uses an JsonPath expression to existence of a node and compares the result to an expected value. Applicable to any property containing JSON.

JsonPath RegEx Match

Uses an JsonPath expression to select content from the target property and compares the result to an specified RegEx. Applicable to any property containing JSON.

Add Cancel

Assertions (0) Request Log (1)

response time: 601ms (323 bytes)

SoapUI log http log jetty log Stop sharing Hide

Type here to search

09:13 11-04-2022

SoapUI 5.6.0

File Project Suite Case Step Tools Desktop Help

Empty SOAP REST Import Save All Forum Trial Preferences Proxy Endpoint Explorer Search Forum Online Help

Navigator:

- REST Project 13
- REST Project 14
- REST Project 15
- REST Project 16
- REST Project 17
- REST Project 2
- REST Project 3
- REST Project 4
- REST Project 5
- REST Project 6
- REST Project 7
- REST Project 8
- REST Project 9
- Sample
- VimanNagar2021
- Viman_nagar_15Jan_C
- CalculatorSoap
- CalculatorSoap12
 - Add
 - Request 1
 - Divide
 - Multiply
 - Subtract
- CalculatorSoap12 TestSuite
 - Add TestCase
 - Test Steps (1)

Custom Properties

TestRequest Properties	
Property Name	Value
Properties	Add

SoapUI Start Page

Add http://www.dneonline.com/calculator.asmx

Contains Assertion

Specify options

Content: <S> Contains Assertion

Ignore Case: Ignore case in comparison

Regular Expression: Use token as Regular Expression

OK Cancel

Headers (9) Attachments (0) SSL Info WSS (0) JMS (0)

Assertions (1) Request Log (1)

response time: 601ms (323 bytes)

SoapUI log http log jetty log Stop sharing Hide

Type here to search

09:14 11-04-2022 ENG

Navigator:

- REST Project 13
- REST Project 14
- REST Project 15
- REST Project 16
- REST Project 17
- REST Project 2
- REST Project 3
- REST Project 4
- REST Project 5
- REST Project 6
- REST Project 7
- REST Project 8
- REST Project 9
- Sample
- VimanNagar2021
- Viman_nagar_15Jan_C
- CalculatorSoap
- CalculatorSoap12
 - Add
 - Request 1
 - Divide
 - Multiply
 - Subtract
- CalculatorSoap12 TestSuite
 - Test Steps (1)

Custom Properties

TestRequest Properties	
Property Name	Value
Properties	Add

SoapUI Start Page

Add http://www.dneonline.com/calculator.asmx

Contains Assertion

Specify options

Content: <S> Contains Assertion

Ignore Case: Ignore case in comparison

Regular Expression: Use token as Regular Expression

OK Cancel

Headers (9) Attachments (0) SSL Info WSS (0) JMS (0)

Assertions (1) Request Log (1)

response time: 601ms (323 bytes)

SoapUI log http log jetty log Stop sharing Hide

Type here to search

09:14 11-04-2022 ENG

SoapUI 5.6.0

SoapUI Start Page

Endpoint Explorer

File Project Suite Case Step Tools Desktop Help

Empty SOAP REST Import Save All Forum Trial Preferences Proxy

Navigator

- REST Project 13
- REST Project 14
- REST Project 15
- REST Project 16
- REST Project 17
- REST Project 2
- REST Project 3
- REST Project 4
- REST Project 5
- REST Project 6
- REST Project 7
- REST Project 8
- REST Project 9
- Sample
- VimanNagar2021
- Viman_nagar_15Jan_C
- CalculatorSoap12
- CalculatorSoap12 TestSuite
- Add TestCase
- Test Steps (1)
- Add

Custom Properties

TestRequest Properties	
Name	Add
Properties	
Type here to search	

Search Forum Online Help

Raw XML Response

```
<soap Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  

<soap Header>  

<soap Body>  

<AddResponse xmlns="http://tempuri.org/">  

<AddResult>4</AddResult>  

</AddResponse>  

</soap Body>  

</soap Envelope>
```

Headers (9) Attachments (0) SSL Info WSS (0) JMS (0)

Contains - VALID

Assertions (1) Request Log (1)
response time: 601ms (323 bytes)

Online Help

SoapUI Start Page

Endpoint Explorer

File Project Suite Case Step Tools Desktop Help

Empty SOAP REST Import Save All Forum Trial Preferences Proxy

Navigator

- REST Project 13
- REST Project 14
- REST Project 15
- REST Project 16
- REST Project 17
- REST Project 2
- REST Project 3
- REST Project 4
- REST Project 5
- REST Project 6
- REST Project 7
- REST Project 8
- REST Project 9
- Sample
- VimanNagar2021
- Viman_nagar_15Jan_C
- CalculatorSoap
- CalculatorSoap12
- CalculatorSoap12 TestSuite
- Add TestCase
- Test Steps (1)
- Add

Custom Properties

TestRequest Properties	
Name	Add
Properties	
Type here to search	

Search Forum Online Help

Raw XML Response

```
<soap Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  

<soap Header>  

<soap Body>  

<AddResponse xmlns="http://tempuri.org/">  

<AddResult>4</AddResult>  

</AddResponse>  

</soap Body>  

</soap Envelope>
```

Contains Assertion

Contains Assertion
Specify options

Content: address

Ignore Case: Ignore case in comparison
Regular Expression: Use token as Regular Expression

OK Cancel

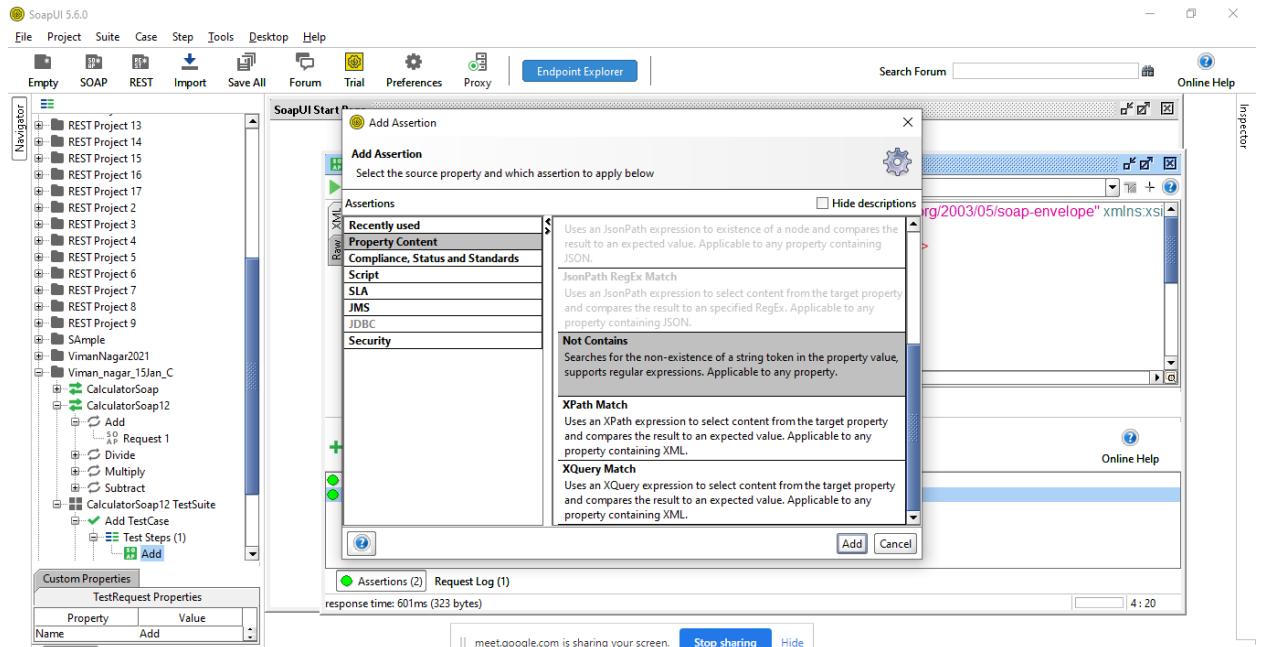
Assertions (2) Request Log (1)
response time: 601ms (323 bytes)

Stop sharing Hide

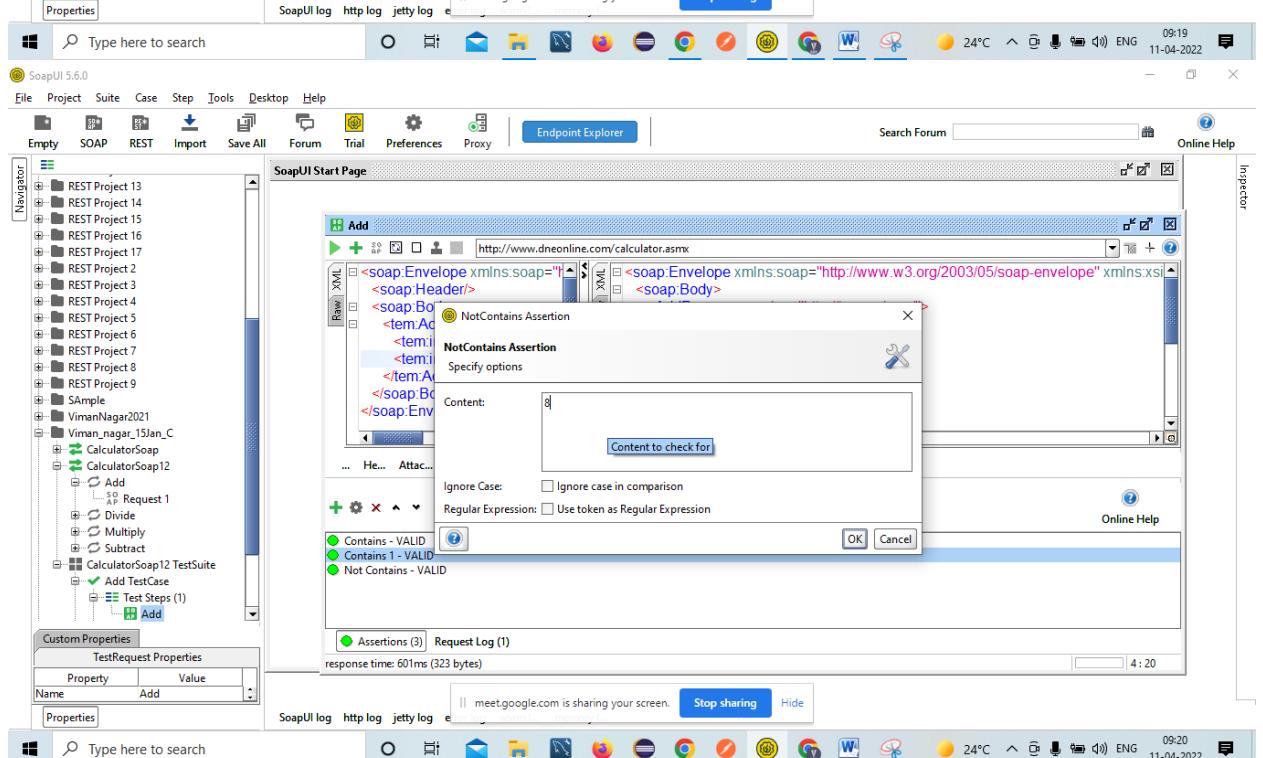
SoapUI log http log jetty log

meet.google.com is sharing your screen.

09:19 11-04-2022 ENG 24°C



7.



SoapUI Start Page

The SoapUI interface shows a project structure on the left under the Navigator tab, listing various REST and SOAP projects. The main workspace displays an 'Add' request for the URL <http://www.dneonline.com/calculator.asmx>. The XML request body is shown in the Raw tab:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<soap:Header/>
<soap:Body>
<item:Add>
<tem:intA>2</tem:intA>
<tem:intB>4</tem:intB>
</item:Add>
</soap:Body>
</soap:Envelope>
```

The Response tab shows the XML response body:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<soap:Header/>
<soap:Body>
<AddResponse xmlns="http://tempuri.org/">
<AddResult>4</AddResult>
</AddResponse>
</soap:Body>
</soap:Envelope>
```

The Assertions panel shows three assertions: 'Contains - VALID', 'Contains 1 - VALID', and 'Not Contains - VALID'. The 'Not Contains' assertion has a red checkmark.

Add Assertion Dialog

The 'Add Assertion' dialog is open, showing the 'Assertions' section. It lists several assertion types:

- Recently used
- Property Content
- Compliance, Status and Standards
- Script
- SLA
- JMS
- JDBC
- Security

The 'Contains' and 'Not Contains' options are expanded, with their descriptions visible. The 'Response SLA' option is also listed.

System Tray and Taskbar

The taskbar at the bottom shows the Windows Start button, a search bar, and icons for various applications like File Explorer, Mail, and Browser. The system tray shows battery status, network connection, and system information (24°C, ENG, 09:22, 11-04-2022).

SoapUI Start Page

Add Assertion

Select the source property and which assertion to apply below

Assertions

- Recently used
- Property Content
- Compliance, Status and Standards
- Script
- SLA
- JMS
- JDBC
- Security

SOAP TestSteps:

- SOAP Request
- SOAP Response
- Valid HTTP Status Codes
- WS-Addressing Request
- WS-Addressing Response

SOAP TestRequest Properties

Property	Value
Name	Add

Custom Properties

TestRequest Properties	
Property	Value
Name	Add

Response time: 746ms (323 bytes)

Stop sharing Hide

SoapUI log http log jetty log

File Project Suite Case Step Tools Desktop Help

Empty SOAP REST Import Save All Forum Trial Preferences Proxy Endpoint Explorer Search Forum Online Help

SoapUI 5.6.0

File Project Suite Case Step Tools Desktop Help

Empty SOAP REST Import Save All Forum Trial Preferences Proxy Endpoint Explorer Search Forum Online Help

Navigator

- REST Project 13
- REST Project 14
- REST Project 15
- REST Project 16
- REST Project 17
- REST Project 2
- REST Project 3
- REST Project 4
- REST Project 5
- REST Project 6
- REST Project 7
- REST Project 8
- REST Project 9
- Sample
- VimanNagar2021
- Viman_nagar_15Jan_C
- CalculatorSoap
- CalculatorSoap12
- CalculatorSoap12 TestSuite
- Add TestCase
- Test Steps (1)
- Add

Custom Properties

TestRequest Properties	
Name	Add
Properties	

SoapUI Start Page

Add

http://www.dneonline.com/calculator.asmx

Valid HTTP status codes Assertion

Specify codes

codes: 200

Value

ZU3J0UZT
HTTP/1.1 200 OK
PleskWin
222

OK Cancel

Online Help

Contains - VALID
Contains 1 - VALID
Not Contains - VALID
Response SLA - VALID
Valid HTTP Status Codes - FAILED
-> Response status code:200 is not in acceptable list of status codes

Assertions (5) Request Log (2)

meet.google.com is sharing your screen. Stop sharing Hide

response time: 746ms (323 bytes)

SoapUI log http log jetty log error log wsrm log memory log

09:27 11-04-2022

Type here to search

SoapUI 5.6.0

File Project Suite Case Step Tools Desktop Help

Empty SOAP REST Import Save All Forum Trial Preferences Proxy Endpoint Explorer Search Forum Online Help

Navigator

- REST Project 13
- REST Project 14
- REST Project 15
- REST Project 16
- REST Project 17
- REST Project 2
- REST Project 3
- REST Project 4
- REST Project 5
- REST Project 6
- REST Project 7
- REST Project 8
- REST Project 9
- Sample
- VimanNagar2021
- Viman_nagar_15Jan_C
- CalculatorSoap
- CalculatorSoap12
- CalculatorSoap12 TestSuite
- Add TestCase
- Test Steps (1)
- Add

Custom Properties

TestRequest Properties	
Name	Add
Properties	

SoapUI Start Page

Add

http://www.dneonline.com/calculator.asmx

soap Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

soap Body

item:Add

item:intA>2</item:intA>

item:intB>2</item:intB>

item:Add

soap Body

soap:Envelope

Header Value

x-asInvokerVersion ZU3J0UZT

#status# HTTP/1.1 200 OK

X-Powered-By-Plesk

Contact-Length 222

Headers (9) Attachments (...) SSL Info WSS (0) JMS (0)

Online Help

Contains - VALID
Contains 1 - VALID
Not Contains - VALID
Response SLA - VALID
Valid HTTP Status Codes - VALID

Assertions (5) Request Log (3)

meet.google.com is sharing your screen. Stop sharing Hide

response time: 625ms (323 bytes)

SoapUI log http log jetty log error log wsrm log memory log

09:27 11-04-2022

Type here to search

The screenshot displays two separate instances of the SoapUI application interface, version 5.6.0, illustrating different ways to configure assertions for a SOAP test step.

Top Instance (Assertion Configuration):

- Project Structure:** REST Project 13, REST Project 14, REST Project 15, REST Project 16, REST Project 17, REST Project 2, REST Project 3, REST Project 4, REST Project 5, REST Project 6, REST Project 7, REST Project 8, REST Project 9, Sample, VimanNagar2021, Viman_nagar_15Jan_C, CalculatorSoap, CalculatorSoap12, CalculatorSoap12 TestSuite (containing Add, Divide, Multiply, Subtract, and Test Steps (1)).
- Custom Properties:** TestRequest Properties (Property: Name, Value: Add).
- Assertions Dialog:** The "Add Assertion" dialog is open, showing the "Assertions" list with "Recently used" items: Property Content, Compliance, Status and Standards, Script, SLA, JMS, JDBC, and Security. The "Invalid HTTP Status Codes" assertion is selected, with its description and configuration options visible.
- Log Panel:** Shows a log entry for a successful response from meet.google.com sharing the screen.
- Toolbar:** Includes SoapUI log, http log, jetty log, error log, wsrm..., memory log, and a timestamp of 09:27 11-04-2022.

Bottom Instance (Raw XML Response):

- Project Structure:** REST Project 13, REST Project 14, REST Project 15, REST Project 16, REST Project 17, REST Project 2, REST Project 3, REST Project 4, REST Project 5, REST Project 6, REST Project 7, REST Project 8, REST Project 9, Sample, VimanNagar2021, Viman_nagar_15Jan_C, CalculatorSoap, CalculatorSoap12, CalculatorSoap12 TestSuite (containing Add, Divide, Multiply, Subtract, and Test Steps (1)).
- Custom Properties:** TestRequest Properties (Property: Name, Value: Add).
- Endpoint Explorer:** Shows the URL <http://www.dneonline.com/calculator.asmx>.
- Raw XML Response:** Displays the raw XML response from the server, showing the structure of the SOAP envelope and its contents.
- Headers:** Shows the following headers:
 - X-Aspver: 2.0.20727
 - #status#
 - X-Powered-By: PleskWin
 - Content-Length: 222
- Assertions:** The Assertions panel shows 6 assertions: Contains - VALID, Contains 1 - VALID, Not Contains - VALID, Response SLA - VALID, Valid HTTP Status Codes - VALID, and Invalid HTTP Status Codes - FAILED.
- Log Panel:** Shows a log entry for a successful response from meet.google.com sharing the screen.
- Toolbar:** Includes SoapUI log, http log, jetty log, error log, wsrm..., memory log, and a timestamp of 09:27 11-04-2022.

7. Validating the SOAP Service by passing Test data=Re-Testing

7.1 Verify the SOAP Service by passing one digit number → EC=one digits numbers should be added=Pass

The screenshot shows the SoapUI interface with the 'Endpoint Explorer' tab selected. In the center, there's an 'Add' dialog box for a SOAP request. The XML payload is set to:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
<soap:Header>
<soap:Body>
<item:Add>
<item:intA>2</item:intA>
<item:intB>2</item:intB>
</item:Add>
</soap:Body>
</soap:Envelope>
```

The response pane shows the raw XML response:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soap:Header>
<AddResponse xmlns="http://tempuri.org/">
<AddResult>4</AddResult>
</AddResponse>
</soap:Header>
<soap:Body>
<AddResponse>
<AddResult>4</AddResult>
</AddResponse>
</soap:Body>
</soap:Envelope>
```

Below the response, the 'Header' section displays:

Header	Value
X-AspNet-Version	2.0.50727
#status#	HTTP/1.1 200 OK
X-Powered-By-Plesk	PleskWin

The status bar at the bottom indicates a response time of 784ms (324 bytes) and a timestamp of 4:22.

7.2 Verify the SOAP Service by passing two digit number → EC=Two digits numbers should be added=Pass

This screenshot is similar to the previous one but shows a different test step. The 'Add' dialog XML payload is now:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
<soap:Header>
<soap:Body>
<item:Add>
<item:intA>33</item:intA>
<item:intB>22</item:intB>
</item:Add>
</soap:Body>
</soap:Envelope>
```

The response pane shows the raw XML response:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soap:Header>
<AddResponse xmlns="http://tempuri.org/">
<AddResult>55</AddResult>
</AddResponse>
</soap:Header>
<soap:Body>
<AddResponse>
<AddResult>55</AddResult>
</AddResponse>
</soap:Body>
</soap:Envelope>
```

The 'Header' section is identical to the first screenshot.

The status bar at the bottom indicates a response time of 784ms (324 bytes) and a timestamp of 4:22.

7.3 Verify the SOAP Service by passing Four/Five digit number → EC=Four/Five digits numbers should be added=Pass

The screenshot shows the SoapUI interface with the following details:

- Toolbar:** File, Project, Suite, Case, Step, Tools, Desktop, Help.
- Navigator:** REST Project 13, REST Project 14, REST Project 15, REST Project 16, REST Project 17, REST Project 2, REST Project 3, REST Project 4, REST Project 5, REST Project 6, REST Project 7, REST Project 8, REST Project 9, Sample, ViranNagar2021, Viran_nagar_15Jan_C, CalculatorSoap, CalculatorSoap12.
- Request Editor:** URL: http://www.dneonline.com/calculator.asmx. The XML payload is:


```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Header>
  <soap:Body>
    <tem:Add>
      <tem:intA>3333</tem:intA>
      <tem:intB>2222</tem:intB>
    </tem:Add>
  </soap:Body>
</soap:Envelope>
```
- Response Editor:** XML pane shows the response with status code 5555. Headers include:

Header	Value
X-AspNet-Version	2.0.50727
#status#	HTTP/1.1 200 OK
X-Powered-By-Plesk	PleskWin
Content-Length	326
Date	Mon, 11 Apr 2022 04:10:44 GMT
- Assertions:** 6 assertions passed.

7.4 Verify the SOAP Service by passing one digit number at 1st place and two digit number at 2nd place → one digit number at 1st place and two digit number at 2nd place should be added=Pass

The screenshot shows the SoapUI interface with the following details:

- Toolbar:** File, Project, Suite, Case, Step, Tools, Desktop, Help.
- Navigator:** REST Project 13, REST Project 14, REST Project 15, REST Project 16, REST Project 17, REST Project 2, REST Project 3, REST Project 4, REST Project 5, REST Project 6, REST Project 7, REST Project 8, REST Project 9, Sample, ViranNagar2021, Viran_nagar_15Jan_C, CalculatorSoap, CalculatorSoap12.
- Request Editor:** URL: http://www.dneonline.com/calculator.asmx. The XML payload is:


```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Header>
  <soap:Body>
    <tem:Add>
      <tem:intA>1</tem:intA>
      <tem:intB>11</tem:intB>
    </tem:Add>
  </soap:Body>
</soap:Envelope>
```
- Response Editor:** XML pane shows the response with status code 12. Headers include:

Header	Value
X-AspNet-Version	2.0.50727
#status#	HTTP/1.1 200 OK
X-Powered-By-Plesk	PleskWin
- Assertions:** 6 assertions failed, including "Contains - FAILED" and "Missing token [4] in Response".

7.5 Verify the SOAP Service by Passing zero at both place → EC =Zero number should be added=Pass

The screenshot shows the SoapUI interface with the following details:

- Project:** estSuite
- Test Case:** 1)
- Request XML (Left):**

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soap:Header>
<soap:Body>
<tem:Add>
<tem:intA>0</tem:intA>
<tem:intB>0</tem:intB>
</tem:Add>
</soap:Body>
</soap:Envelope>
```
- Response XML (Right):**

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soap:Header>
<soap:Body>
<AddResponse xmlns="http://tempuri.org/">
<AddResult>0</AddResult>
</AddResponse>
</soap:Body>
</soap:Envelope>
```
- Headers:**

Header	Value
X-AspNet-Version	2.0.50727
#status#	HTTP/1.1 200 OK
X-Powered-By-Plesk	PleskWin
- Assertions:** Contains - FAILED (highlighted in red)
- Request Log:** 10 entries

7.6 Verify the SOAP Service by Passing both digits as -ve numbers → EC=Negative numbers should be added=Pass

The screenshot shows the SoapUI interface with the following details:

- Project:** estSuite
- Test Case:** 1)
- Request XML (Left):**

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soap:Header>
<soap:Body>
<tem:Add>
<tem:intA>-2</tem:intA>
<tem:intB>-23</tem:intB>
</tem:Add>
</soap:Body>
</soap:Envelope>
```
- Response XML (Right):**

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soap:Header>
<soap:Body>
<AddResponse xmlns="http://tempuri.org/">
<AddResult>25</AddResult>
</AddResponse>
</soap:Body>
</soap:Envelope>
```
- Headers:**

Header	Value
X-AspNet-Version	2.0.50727
#status#	HTTP/1.1 200 OK
X-Powered-By-Plesk	PleskWin
- Assertions:** Contains - FAILED (highlighted in red)
- Request Log:** 11 entries

8. Validating Negative Test cases

8.1 Verify the SOAP Service by passing decimal digit numbers → EC=Decimal digits
numbers should not be added=Pass

The screenshot shows the SoapUI interface with the following details:

- Toolbar:** File, Project, Suite, Case, Step, Tools, Desktop, Help.
- Navigator:** REST Project 13, REST Project 14, REST Project 15, REST Project 16, REST Project 17, REST Project 2, REST Project 3, REST Project 4, REST Project 5, REST Project 6, REST Project 7, REST Project 8, REST Project 9, Sample, VimanNagar2021, Viman_nagar_15Jan_C, CalculatorSoap, CalculatorSoap12, Add, Divide, Multiply, Subtract, CalculatorSoap12 TestSuite, Add TestCase, Test Steps (1), Add.
- Custom Properties:** TestRequest Properties, Property, Value.
- SOAP Request (Raw XML):**

```
<soap Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
<soap Header>
<soap Body>
<tem:Add>
<tem:intA>5.90</tem:intA>
<tem:intB>-8.38</tem:intB>
</tem:Add>
</soap Body>
</soap Envelope>
```
- SOAP Response (Raw XML):**

```
<soap Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
<soap Body>
<soap Fault>
<soap Code>
<tem:intA>5.90</tem:intA>
<tem:intB>-8.38</tem:intB>
</tem:Add>
<soap Value>soap:Sender</soap:Value>
</soap Code>
<soap Reason>
<soap Text xml:lang="en">System.Web.Services.Protocols.SoapException: at System.Number.StringToNumber(String str, NumberStyles options, NumberBuffer& number, NumberFormatInfo info, Boolean parseDecimal)</soap Text>
<soap Text xml:lang="en"> at System.Number.ParseInt32(String s, NumberStyles style, NumberFormatInfo info, Boolean parseDecimal)</soap Text>
<soap Text xml:lang="en"> at System.Xml.Convert.ToInt32(String s)</soap Text>
<soap Text xml:lang="en"> at Microsoft.Xml.Serialization.GeneratedAssembly.XmlSerializationReader1.Read</soap Text>
<soap Text xml:lang="en"> at Microsoft.Xml.Serialization.GeneratedAssembly.ArrayOfObjectSerializer.Deserialize</soap Text>
<soap Text xml:lang="en"> at System.Xml.Serialization.XmlSerializer.Deserialize(XmlReader xmlReader, String name, XmlTypeCode typeCode, Boolean ignoreTypeCode)</soap Text>
```
- Header:** X-AspNet-Version: 2.0.50727, HTTP/1.1 500 Internal Server Error.
- Assertions:** Contains - FAILED (Missing token [4] in Response), Contains 1 - FAILED (Missing token [address] in Response), Not Contains - VALID, Response SLA - VALID, Valid-HTTP-Header-Code - FAILED.
- Request Log:** Assertions (6), Request Log (14).
- Response Time:** 686ms (1812 bytes).
- Bottom Status Bar:** 5 : 29.

8.2 Verify the SOAP Service by passing String/Fractional digit number → EC=Number should not be added=Pass

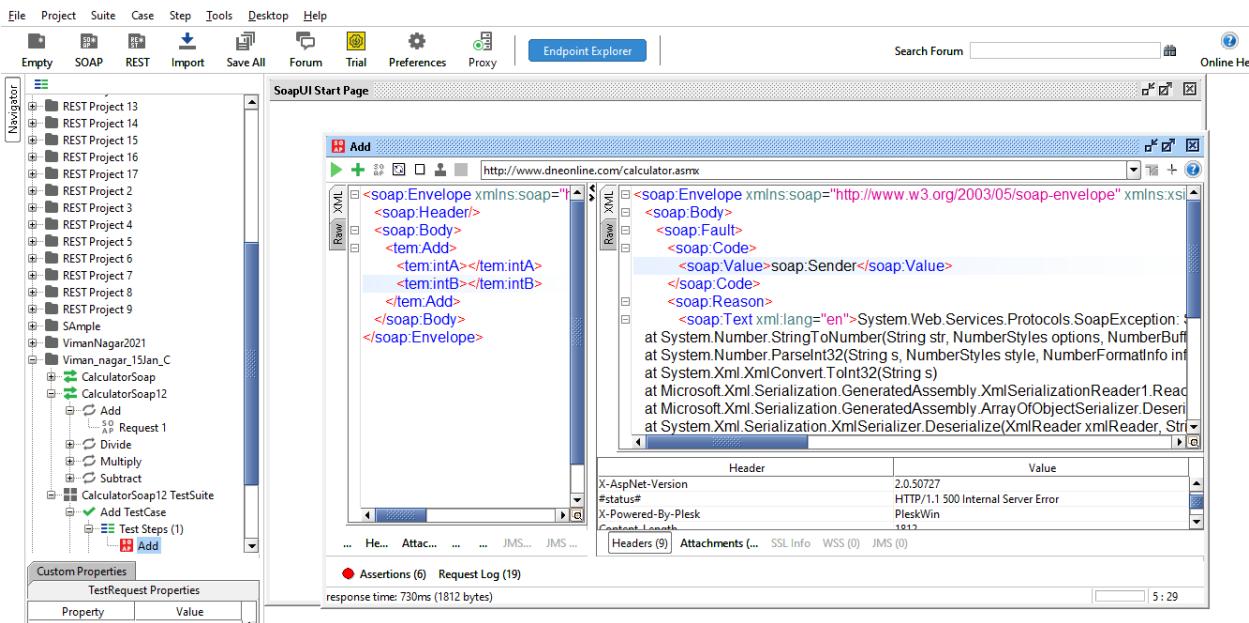
The screenshot shows the SoapUI interface with the following details:

- Toolbar:** File, Project, Suite, Case, Step, Tools, Desktop, Help.
- Navigator:** REST Project 13, REST Project 14, REST Project 15, REST Project 16, REST Project 17, REST Project 2, REST Project 3, REST Project 4, REST Project 5, REST Project 6, REST Project 7, REST Project 8, REST Project 9, Sample, VimanNagar2021, Viman_nagar_15Jan_C, CalculatorSoap, CalculatorSoap12, Add, Divide, Multiply, Subtract, CalculatorSoap12 TestSuite, Add TestCase, Test Steps (1), Add.
- Custom Properties:** TestRequest Properties, Property, Value.
- SOAP Request (Raw XML):**

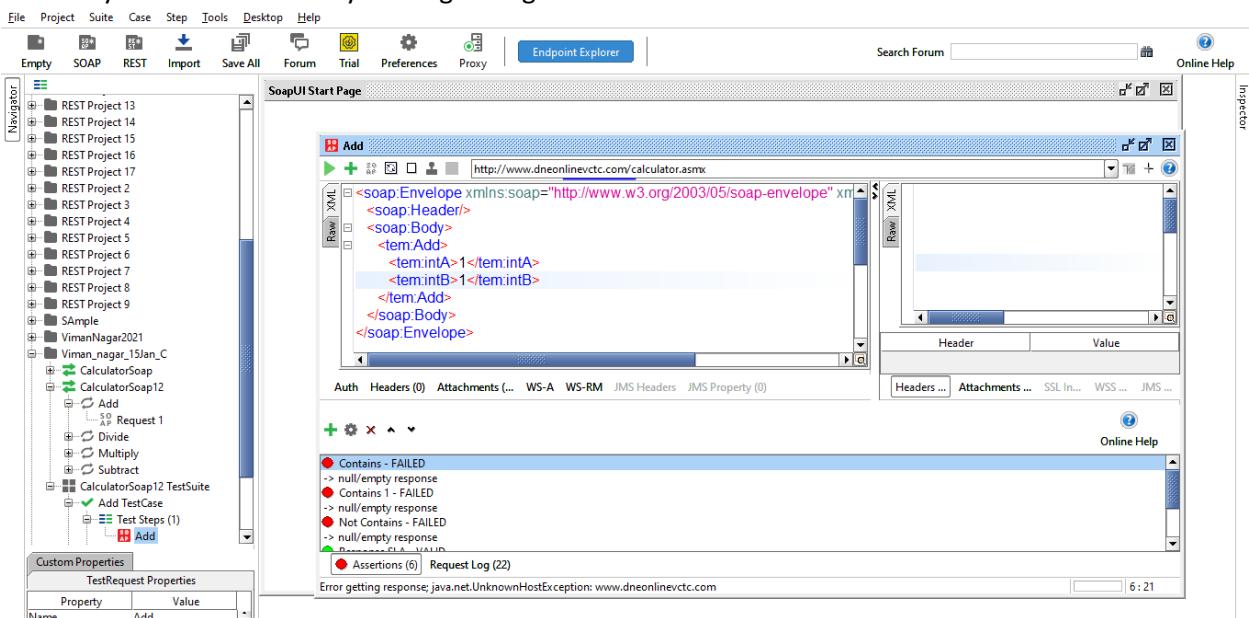
```
<soap Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
<soap Header>
<soap Body>
<tem:Add>
<tem:intA>Five</tem:intA>
<tem:intB>Two</tem:intB>
</tem:Add>
</soap Body>
</soap Envelope>
```
- SOAP Response (Raw XML):**

```
<soap Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
<soap Body>
<soap Fault>
<soap Code>
<tem:intA>Five</tem:intA>
<tem:intB>Two</tem:intB>
</tem:Add>
<soap Value>soap:Sender</soap:Value>
</soap Code>
<soap Reason>
<soap Text xml:lang="en">System.Web.Services.Protocols.SoapException: at System.Number.StringToNumber(String str, NumberStyles options, NumberBuffer& number, NumberFormatInfo info, Boolean parseDecimal)</soap Text>
<soap Text xml:lang="en"> at System.Number.ParseInt32(String s, NumberStyles style, NumberFormatInfo info, Boolean parseDecimal)</soap Text>
<soap Text xml:lang="en"> at System.Xml.Convert.ToInt32(String s)</soap Text>
<soap Text xml:lang="en"> at Microsoft.Xml.Serialization.GeneratedAssembly.XmlSerializationReader1.Read</soap Text>
<soap Text xml:lang="en"> at Microsoft.Xml.Serialization.GeneratedAssembly.ArrayOfObjectSerializer.Deserialize</soap Text>
<soap Text xml:lang="en"> at System.Xml.Serialization.XmlSerializer.Deserialize(XmlReader xmlReader, String name, XmlTypeCode typeCode, Boolean ignoreTypeCode)</soap Text>
```
- Header:** X-AspNet-Version: 2.0.50727, status# HTTP/1.1 500 Internal Server Error, X-Powered-By-Plesk, Content-Length: 1017.
- Assertions:** Assertions (6), Request Log (15).
- Response Time:** 778ms (1812 bytes).
- Bottom Status Bar:** 5 : 29.

8.3 Verify the SOAP Service by passing Null/Blank values → EC=Should not be added=Pass



8.4 Verify the SOAP service by Passing wrong WSDL file → EC=Pass



Few Important Points:

1. How to create new SOAP Project in SOAP UI.
2. Create Test Suite and Test cases
3. Run Test cases
4. How to add Assertions(Validation Points)
 - Contains
 - Not Contains

- Xpath Match
- Xpath Query Match
- Valid HTTP Status Codes
- InValid HTTP Status Codes
- SOAP Fault
- NOT SOAP Fault
- Schema Compliance
- SOAP Response
- Response SLA
- Script Assertion
- JSON Path Count
- JSON Path Existance
- JSON Path Match

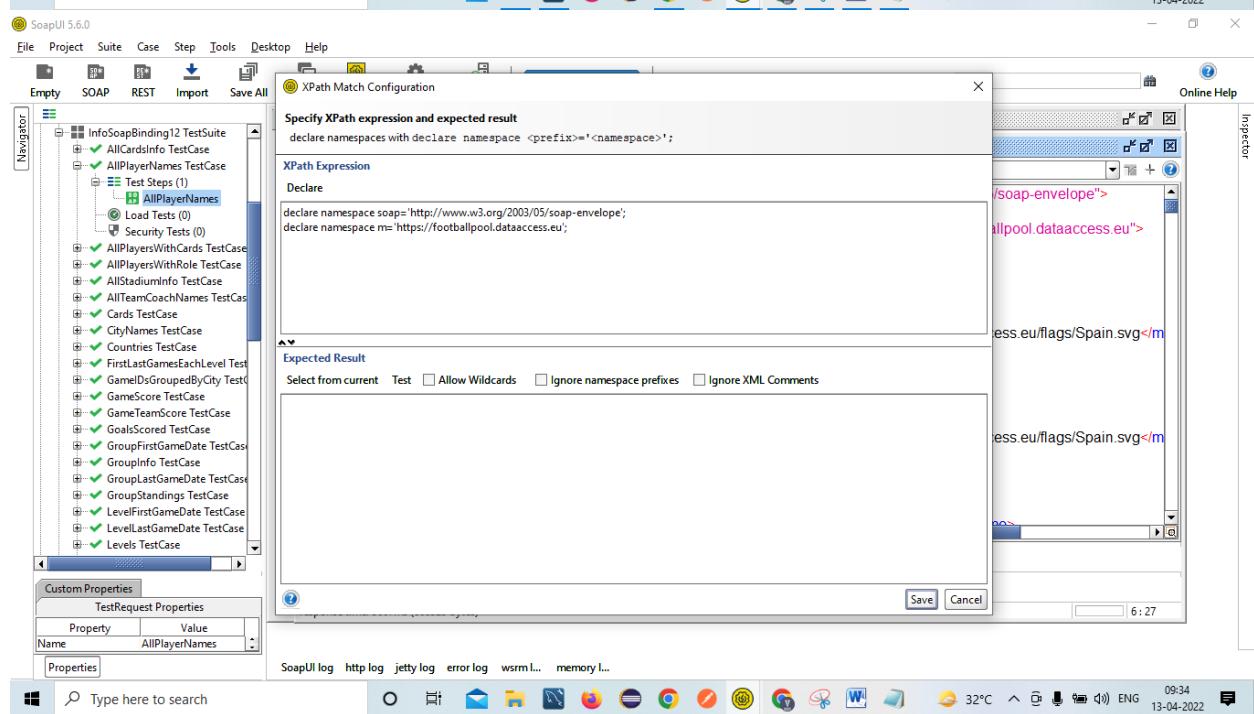
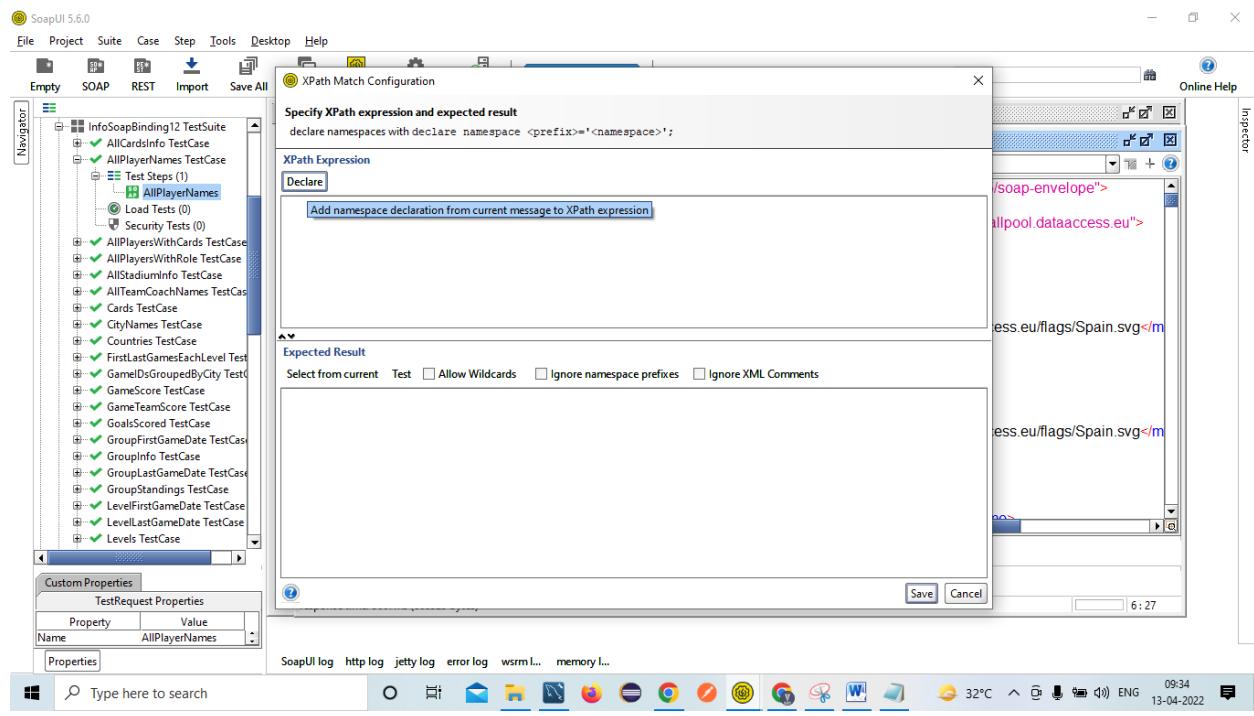
13/04/2022:

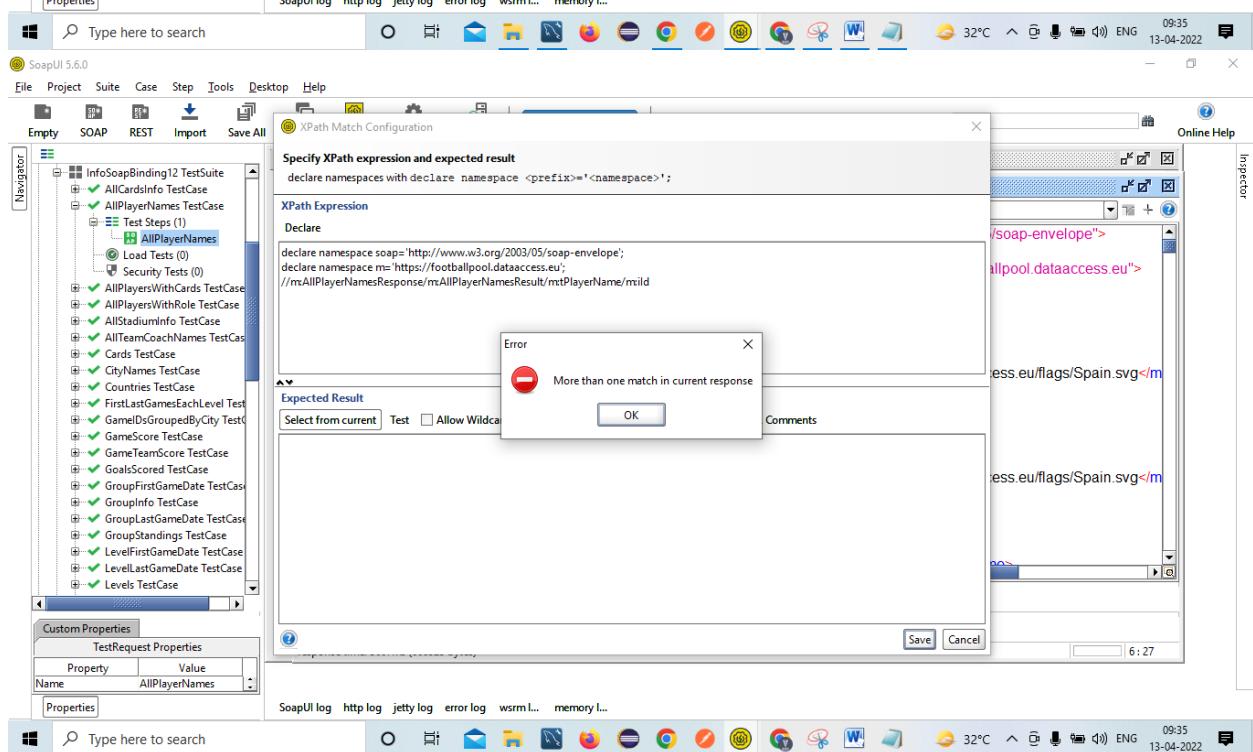
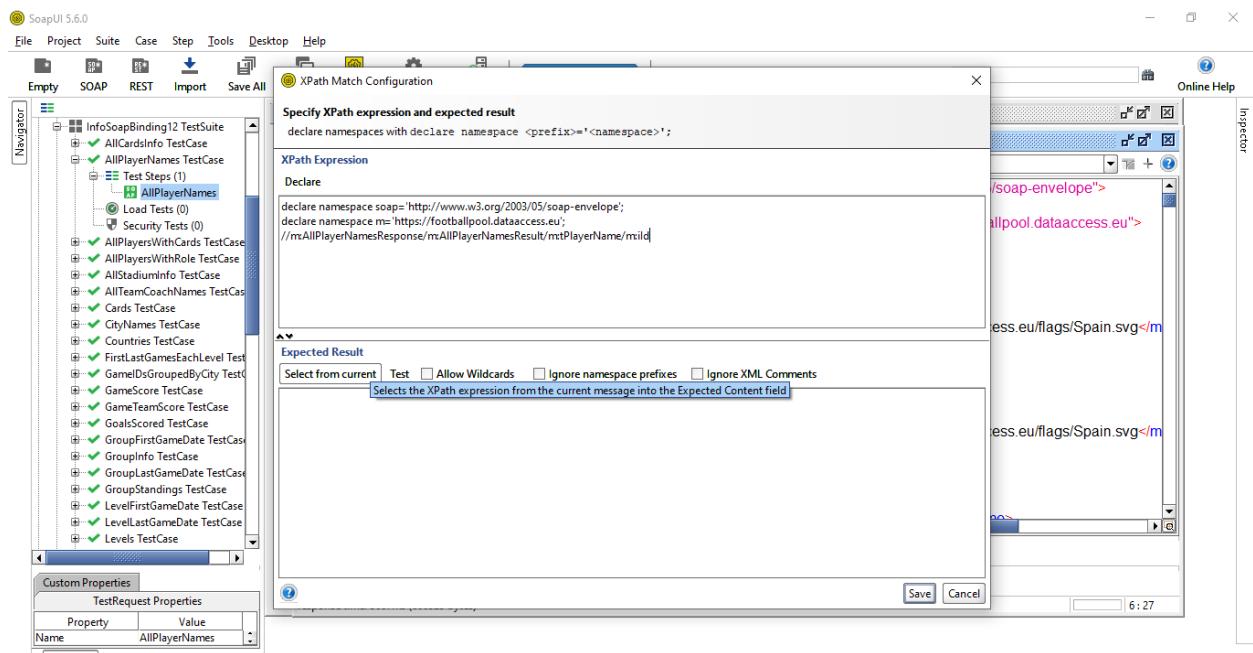
Agenda: Xpath Assertion, REST Service

Xpath Assertion:

- Xpath is a query language for selecting Nodes(Tagnames) from an XML document
- Xpath can also used to compute values from the content of an xml document
- To Find something is present in the response

Ex: <https://ws.footballpool.dataaccess.eu/info.wso?WSDL>





Suite View:

- InfoSoapBinding12 TestSuite
 - AllCardsInfo TestCase
 - AllPlayerNames TestCase
 - Test Steps (1)
 - AllPlayerNames
 - Load Tests (0)
 - Security Tests (0)
 - AllPlayersWithCards TestCase
 - AllPlayersWithRole TestCase
 - AllStadiumInfo TestCase
 - AllTeamCoachNames TestCase
 - Cards TestCase
 - CityNames TestCase
 - Countries TestCase
 - FirstLastGamesEachLevel Test
 - GamelsGroupedByCity Test
 - GameScore TestCase
 - GameTeamScore TestCase
 - GoalsScored TestCase
 - GroupFirstGameDate TestCase
 - GroupInfo TestCase
 - GroupLastGameDate TestCase
 - GroupStandings TestCase
 - LevelFirstGameDate TestCase
 - LevelLastGameDate TestCase
 - Levels TestCase

Custom Properties:

TestRequest Properties	
Property	Value
Name	AllPlayerNames

Properties:

Property	Value
Name	AllPlayerNames

XPath Match Configuration Dialog:

Specify XPath expression and expected result

```
declare namespace with declare namespace <prefix='<namespace>'>;
```

XPath Expression

Declare

```
declare namespace soap='http://www.w3.org/2003/05/soap-envelope';
declare namespace m='https://footballpool.dataaccess.eu';
//m:AllPlayerNamesResponse/m:AllPlayerNamesResult/m:tPlayerName[1]/m:id
```

Expected Result

Select from current Test Allow Wildcards Ignore namespace prefixes Ignore XML Comments

3413

Output Preview:

```
/soap-envelope>
  /allpool.dataaccess.eu">
    /ess.eu/flags/Spain.svg</m
```

Suite View:

- InfoSoapBinding12 TestSuite
 - AllCardsInfo TestCase
 - AllPlayerNames TestCase
 - Test Steps (1)
 - AllPlayerNames
 - Load Tests (0)
 - Security Tests (0)
 - AllPlayersWithCards TestCase
 - AllPlayersWithRole TestCase
 - AllStadiumInfo TestCase
 - AllTeamCoachNames TestCase
 - Cards TestCase
 - CityNames TestCase
 - Countries TestCase
 - FirstLastGamesEachLevel Test
 - GamelsGroupedByCity Test
 - GameScore TestCase
 - GameTeamScore TestCase
 - GoalsScored TestCase
 - GroupFirstGameDate TestCase
 - GroupInfo TestCase
 - GroupLastGameDate TestCase
 - GroupStandings TestCase
 - LevelFirstGameDate TestCase
 - LevelLastGameDate TestCase
 - Levels TestCase

Custom Properties:

TestRequest Properties	
Property	Value
Name	AllPlayerNames

Properties:

Property	Value
Name	AllPlayerNames

XPath Match Configuration Dialog:

Specify XPath expression and expected result

```
declare namespace with declare namespace <prefix='<namespace>'>;
```

XPath Expression

Declare

```
declare namespace soap='http://www.w3.org/2003/05/soap-envelope';
declare namespace m='https://footballpool.dataaccess.eu';
count(/msName)
```

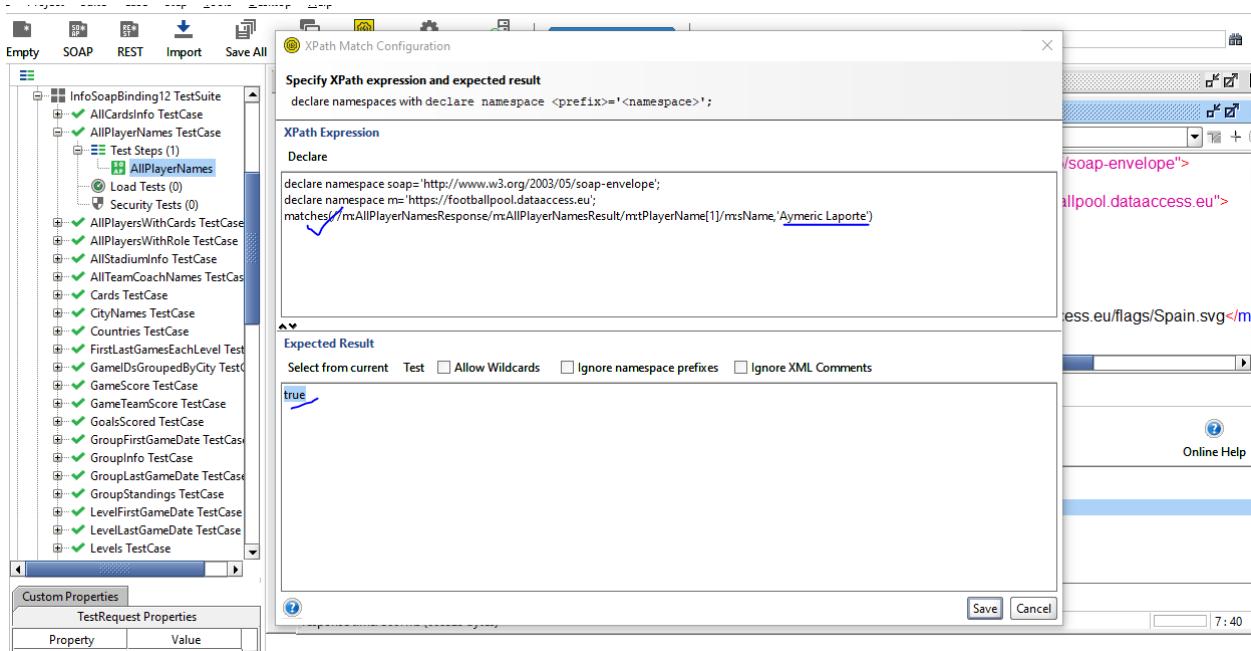
Expected Result

Select from current Test Allow Wildcards Ignore namespace prefixes Ignore XML Comments

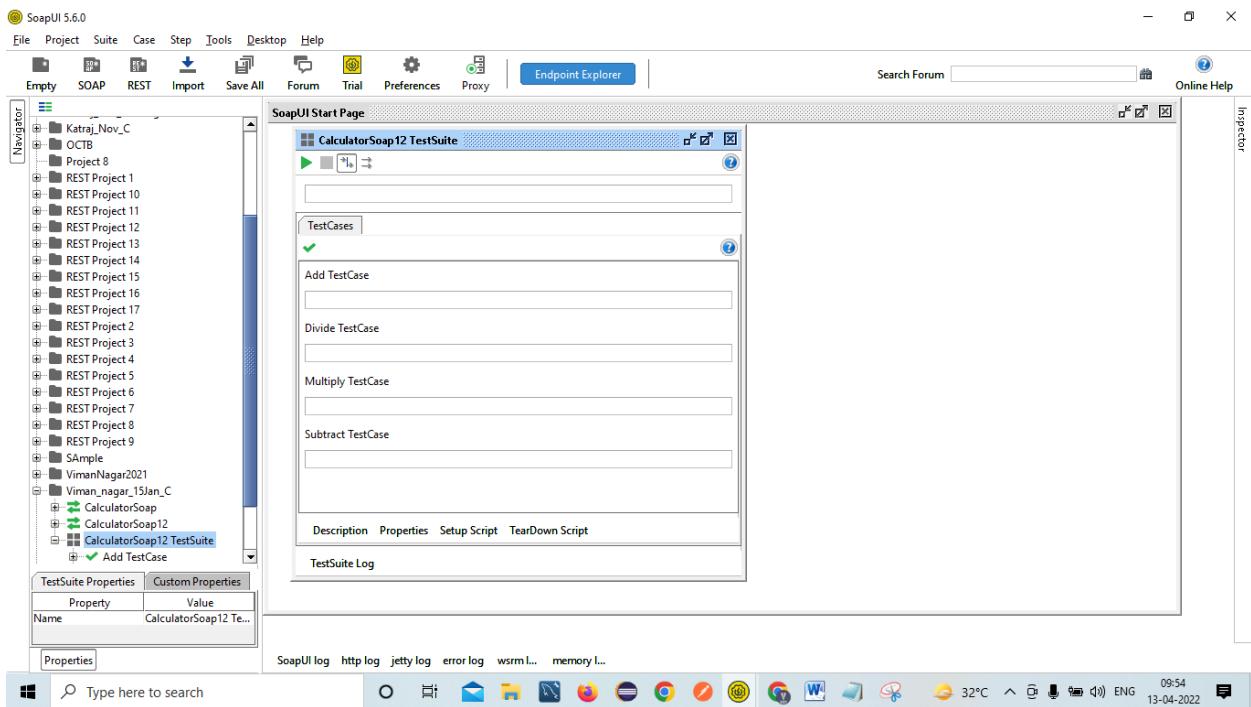
2386

Output Preview:

```
yName>
  /ess.eu/flags/Czech Repub
  /ess.eu/flags/Slovakia.svg<
```



How to Run Test Suite in SOAP UI?



Sequential:

SoapUI 5.6.0

File Project Suite Case Step Tools Desktop Help

Empty SOAP REST Import Save All Forum Trial Preferences Proxy Endpoint Explorer Search Forum Online Help

SoapUI Start Page

CalculatorSoap12 TestSuite

Runs the selected TestCases

TestCases

- Add TestCase
- Divide TestCase
- Multiply TestCase
- Subtract TestCase

Description Properties Setup Script TearDown Script TestSuite Log

Properties

TestSuite Properties Custom Properties

Property	Value
Name	CalculatorSoap12 Te...

Properties

SoapUI log http log jetty log error log wsrm... memory l... 09:56 13-04-2022

Type here to search

Inspector

SoapUI 5.6.0

File Project Suite Case Step Tools Desktop Help

Empty SOAP REST Import Save All Forum Trial Preferences Proxy Endpoint Explorer Search Forum Online Help

SoapUI Start Page

CalculatorSoap12 TestSuite

The selected TestCases are run in sequence.

TestCases

- Add TestCase
- Divide TestCase
- Multiply TestCase
- Subtract TestCase

Description Properties Setup Script TearDown Script TestSuite Log

Properties

TestSuite Properties Custom Properties

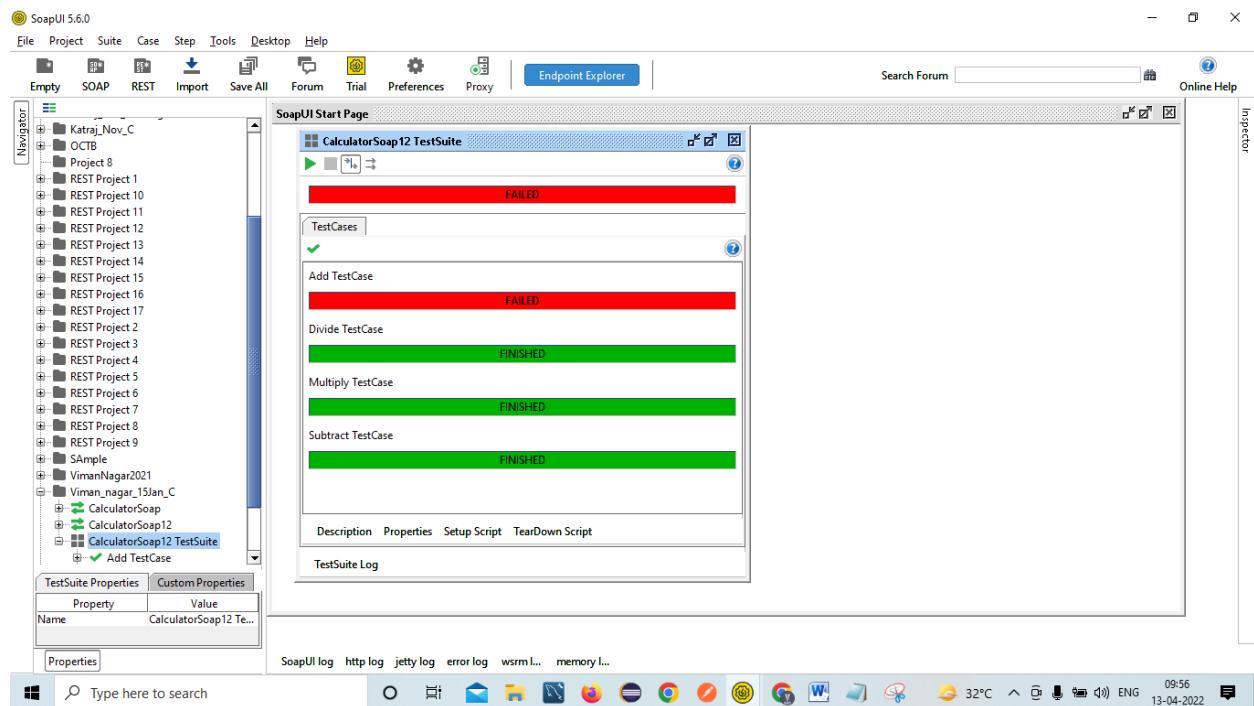
Property	Value
Name	CalculatorSoap12 Te...

Properties

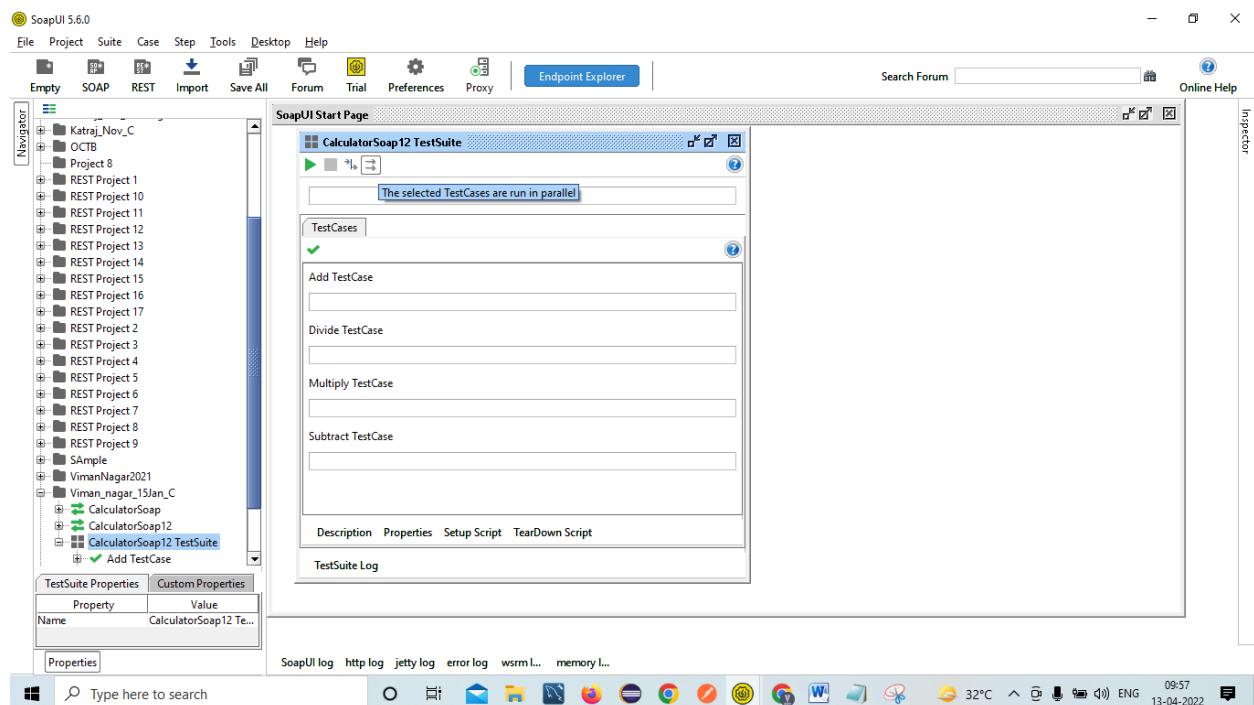
SoapUI log http log jetty log error log wsrm... memory l... 09:56 13-04-2022

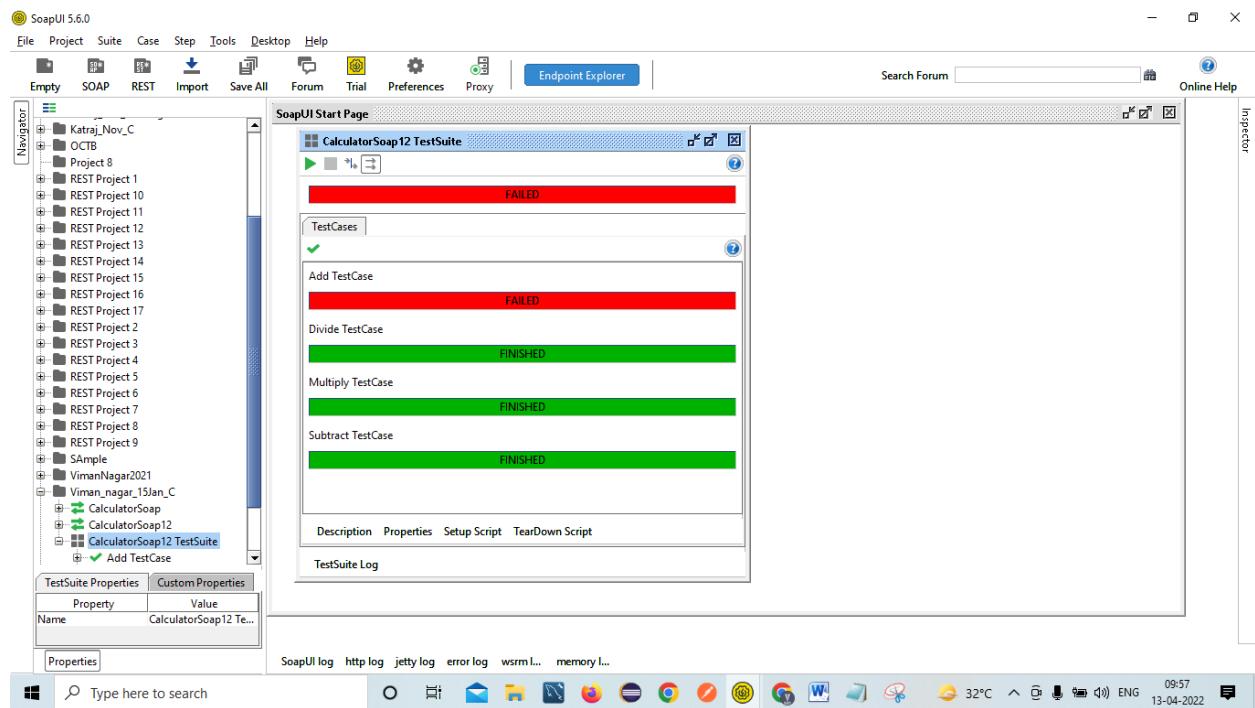
Type here to search

Inspector

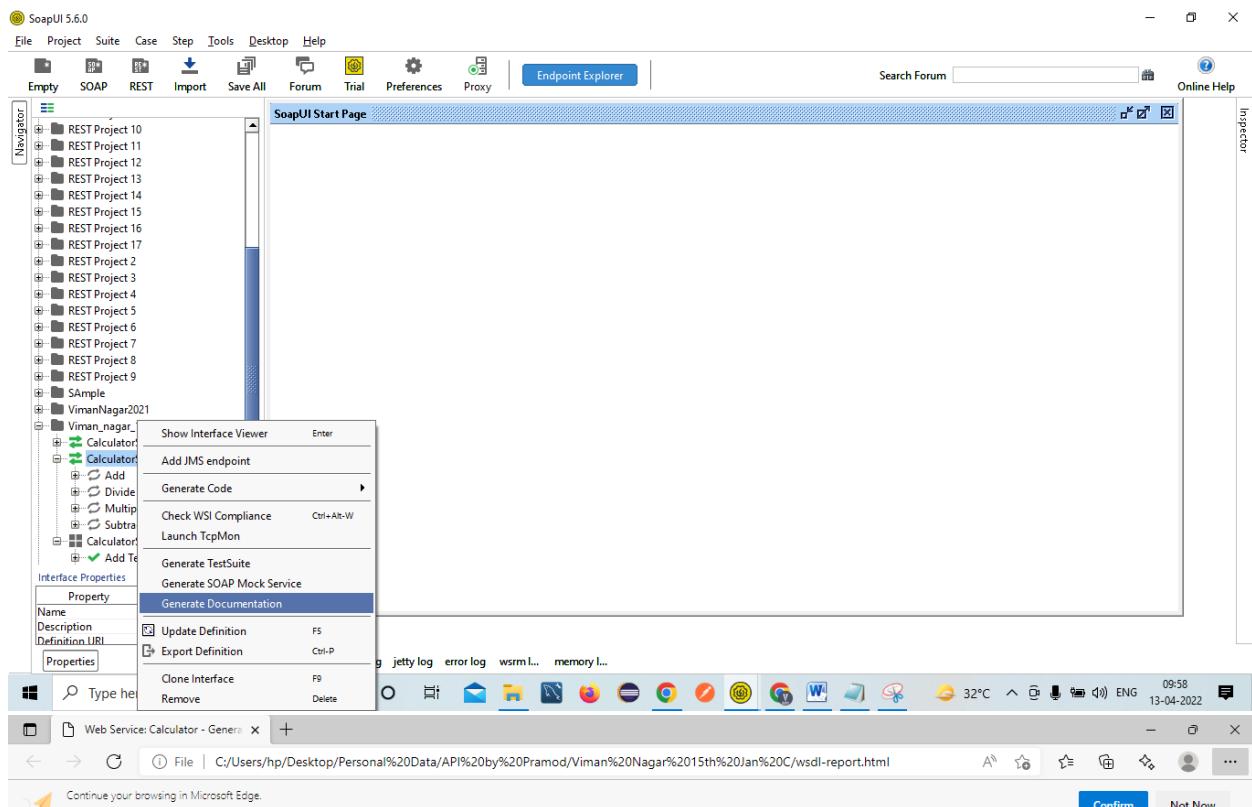


Parallel:





How to generate Documentation in SOAP UI?



The screenshot shows the Microsoft Edge browser displaying the 'Web Service: Calculator' documentation page. The title bar says 'Web Service: Calculator'. The page content includes:

- Web Service: Calculator**
- Target Namespace: <http://tempuri.org/>
- Port CalculatorSoap** [Port type](#) [Source code](#)
 - Location: <http://www.dneonline.com/calculator.asmx>
 - Protocol: SOAP
 - Transport protocol: SOAP over HTTP
 - Operations:
 1. Add [Detail](#) [Source code](#)
 2. Divide [Detail](#) [Source code](#)
 3. Multiply [Detail](#) [Source code](#)
 4. Subtract [Detail](#) [Source code](#)
- Port CalculatorSoap12** [Port type](#) [Source code](#)
 - Location: <http://www.dneonline.com/calculator.asmx>
 - Protocol: SOAP
 - Transport protocol: SOAP over HTTP
 - Operations:
 1. Add [Detail](#) [Source code](#)
 2. Divide [Detail](#) [Source code](#)
 3. Multiply [Detail](#) [Source code](#)
 4. Subtract [Detail](#) [Source code](#)

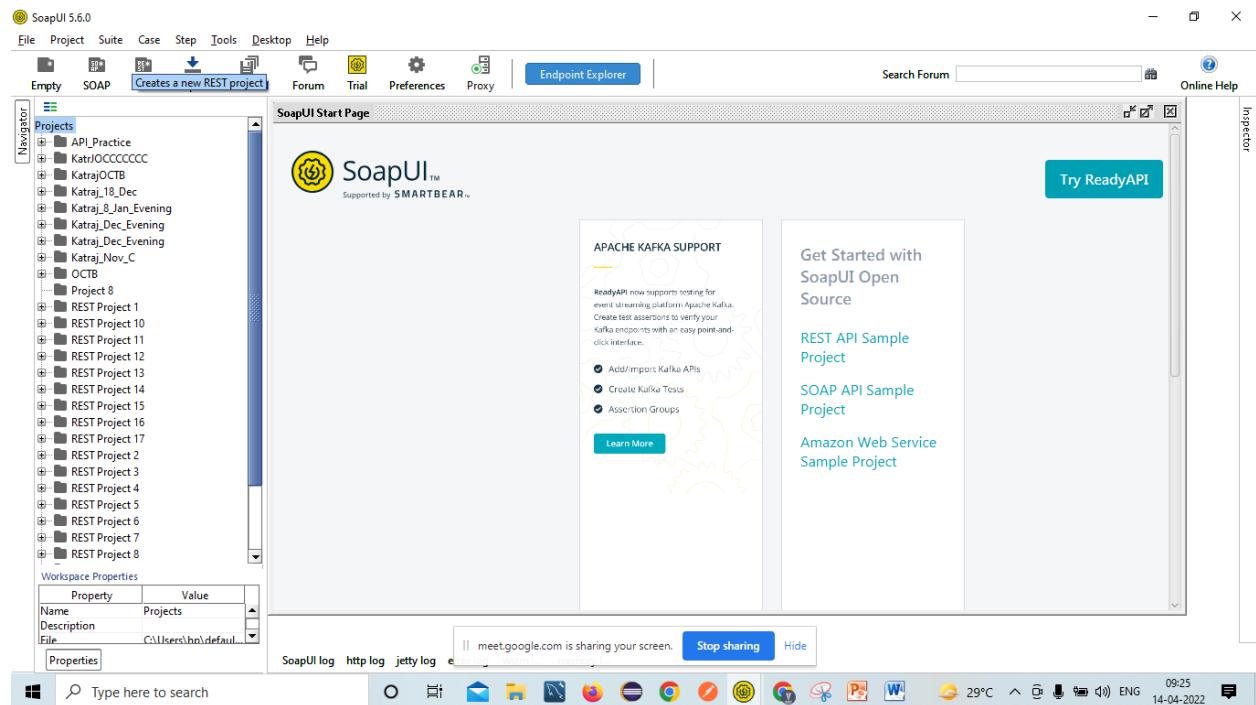
The taskbar at the bottom shows the Microsoft Edge browser window titled 'Web Service: Calculator - Gener...' with the URL 'C:/Users/hp/Desktop/Personal%20Data/API%20by%20Pramod/Viman%20Nagar%2015th%20Jan%20C/wsdl-report.html'. The system tray shows the date as 13-04-2022 and the time as 10:00.

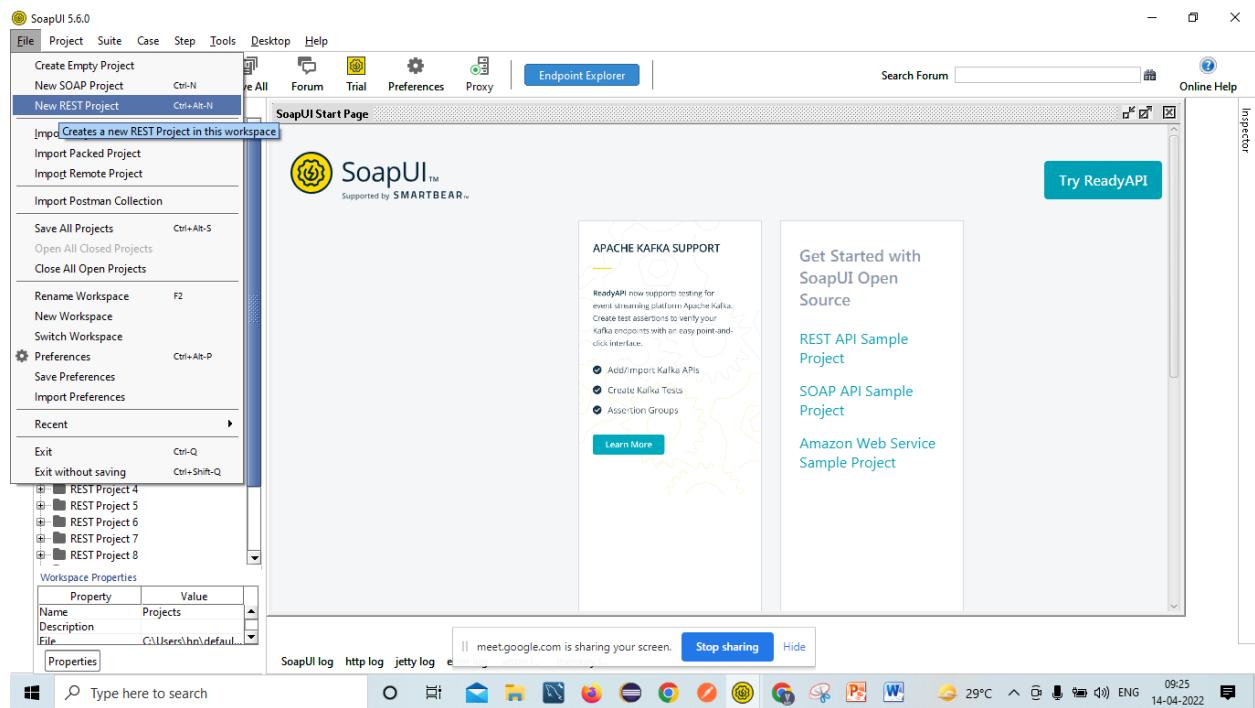
14/04/2022:

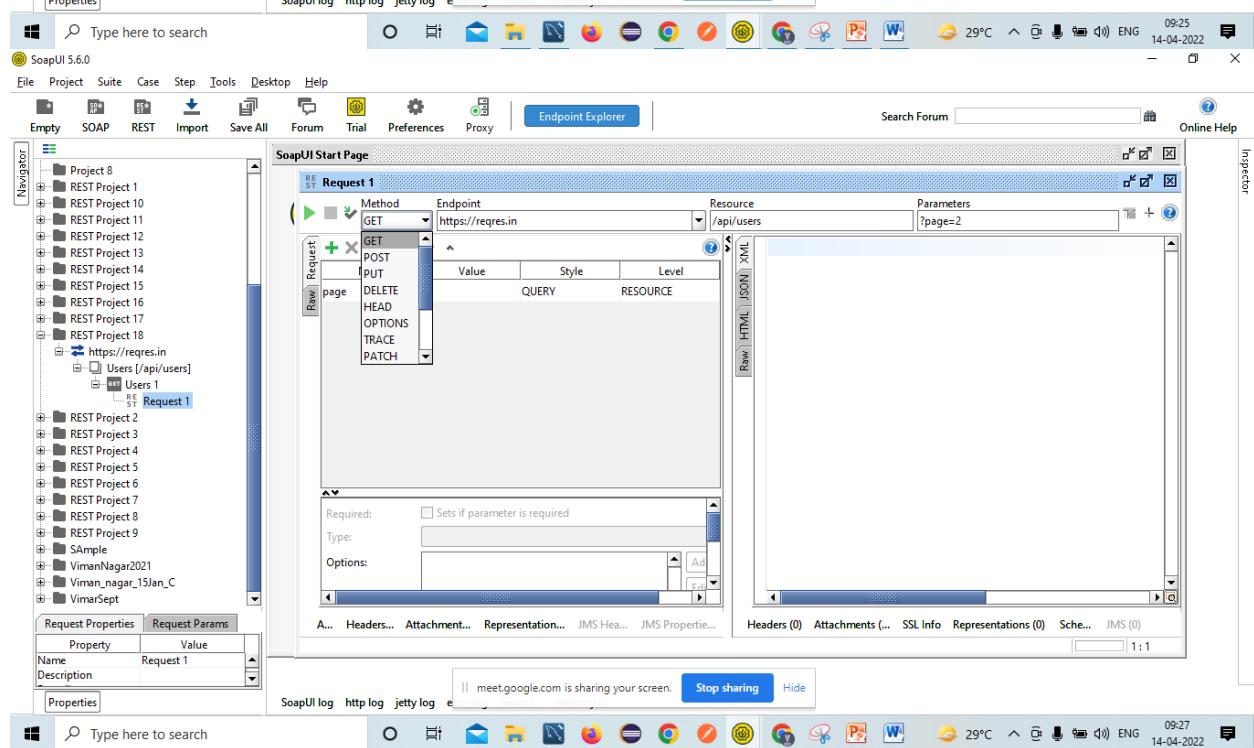
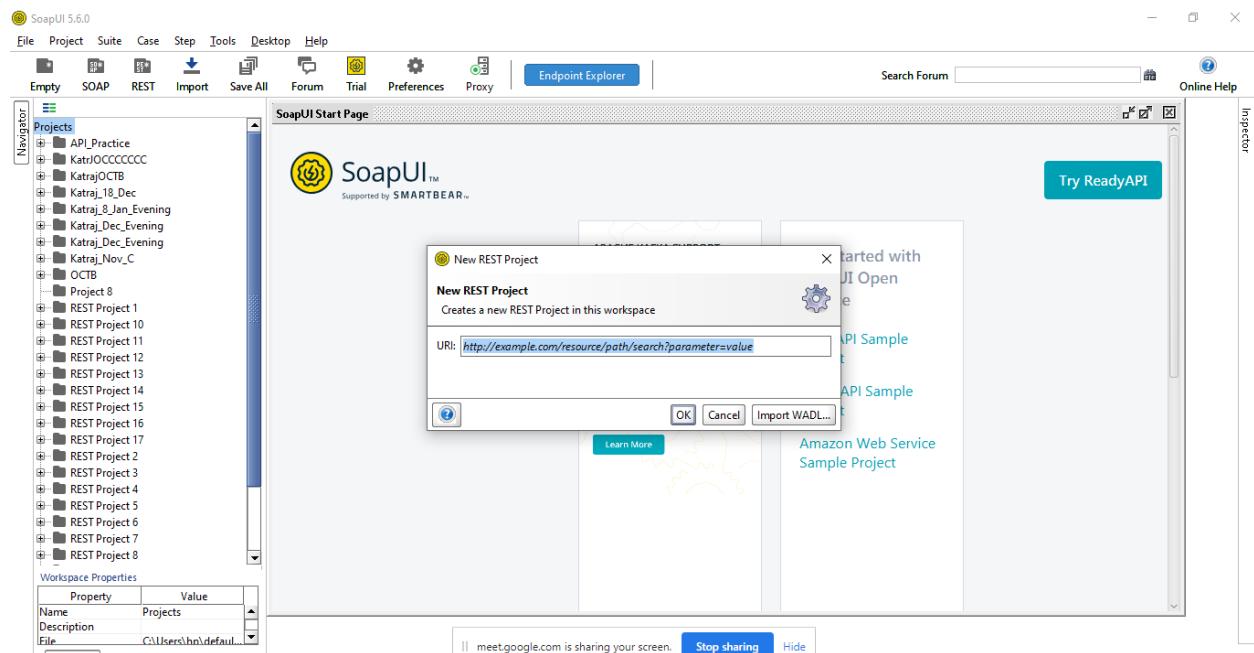
Agenda: Status Codes, REST Service (GET Method)

GET Method:

- GET Method Service-URL/URI=https://reqres.in/api/users?page=2
- Username & Password(Basic Auth)/API Key/Bearer Token—aadjajbkdbkdb
- Time taken for the response=5 Secs
- Unit Testing Documents







SoapUI 5.6.0

File Project Suite Case Step Tools Desktop Help

Empty SOAP REST Import Save All Forum Trial Preferences Proxy Endpoint Explorer Search Forum Online Help Inspector

SoapUI Start Page

Request 1

Method: GET Endpoint: https://reqres.in Resource: /api/users Parameters: ?page=2

Submit request to specified endpoint URL (Alt-Enter)

Request

Name	Value	Style	Level
page	2	QUERY	RESOURCE

Raw XML JSON

Required: Sets if parameter is required
Type:
Options:

Headers (0) Attachments (0) SSL Info Representations (0) Schema (0) JMS (0)

1:1

SoapUI log http log jetty log Stop sharing Hide

Type here to search

09:27 14-04-2022 ENG

SoapUI 5.6.0

File Project Suite Case Step Tools Desktop Help

Empty SOAP REST Import Save All Forum Trial Preferences Proxy Endpoint Explorer Search Forum Online Help Inspector

SoapUI Start Page

Request 1

Method: GET Endpoint: https://reqres.in Resource: /api/users Parameters: ?page=2

The content you are trying to view cannot be viewed as XML

Request

Name	Value	Style	Level
page	2	QUERY	RESOURCE

Raw XML JSON

Required: Sets if parameter is required
Type:
Options:

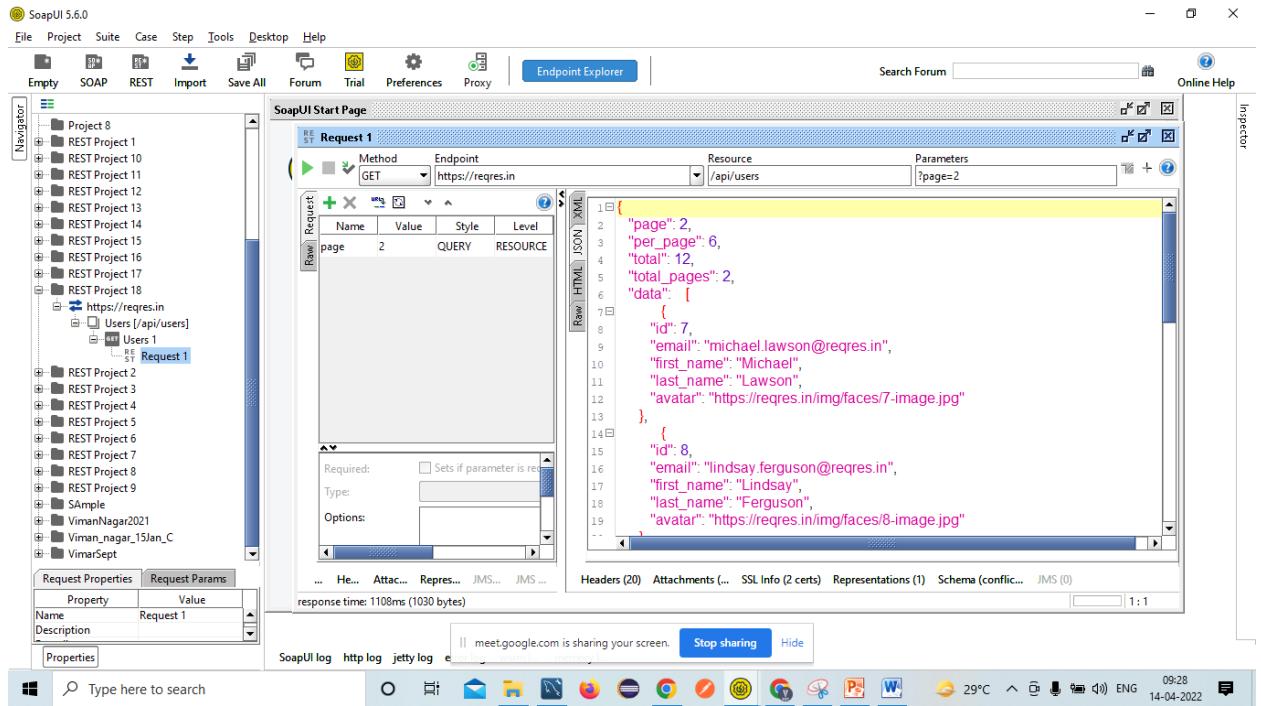
Headers (20) Attachments (0) SSL Info (2 certs) Representations (1) Schema (conflict...) JMS (0)

response time: 1108ms (1030 bytes)

SoapUI log http log jetty log Stop sharing Hide

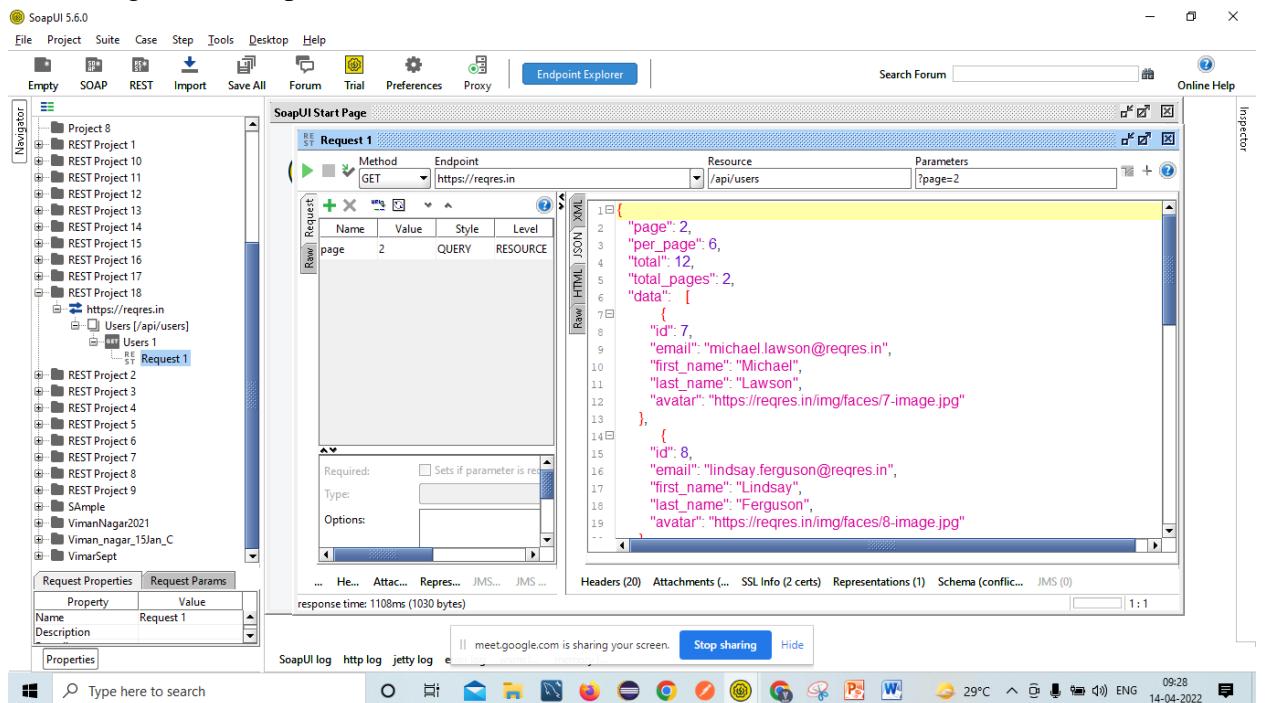
Type here to search

09:28 14-04-2022 ENG



Test Cases:

1. Validating REST Response=Pass



2. Validating Data and Count of Data in the REST Response=Pass

SoapUI Start Page

Request 1

Method: GET, Endpoint: https://reqres.in, Resource: /api/users, Parameters: ?page=2

Raw Request:

Name	Value	Style	Level
page	2	QUERY	RESOURCE

Raw Response (JSON):

```

page : 2,
"per_page": 6,
"total": 12,
"total_pages": 2,
"data": [
    {
        "id": 7,
        "email": "michael.lawson@reqres.in",
        "first_name": "Michael",
        "last_name": "Lawson",
        "avatar": "https://reqres.in/img/faces/7-image.jpg"
    },
    {
        "id": 8,
        "email": "lindsay.ferguson@reqres.in",
        "first_name": "Lindsay",
        "last_name": "Ferguson",
        "avatar": "https://reqres.in/img/faces/8-image.jpg"
    }
]
  
```

Annotations in red and blue highlight the following fields in the JSON response:

- Red highlights:** "data", "id", "email", "first_name", "last_name", "avatar".
- Blue highlights:** "id", "email", "first_name", "last_name", "avatar".

3. Validating tagnames/attributes present in the response → EC=Id, Email, first_name, last_name, Avatar=Pass

Request 1

Method: GET, Endpoint: https://reqres.in, Resource: /api/users, Parameters: ?page=2

Raw Request:

Name	Value	Style	Level
page	2	QUERY	RESOURCE

Raw Response (JSON):

```

page : 2,
"per_page": 6,
"total": 12,
"total_pages": 2,
"data": [
    {
        "id": 7,
        "email": "michael.lawson@reqres.in",
        "first_name": "Michael",
        "last_name": "Lawson",
        "avatar": "https://reqres.in/img/faces/7-image.jpg"
    },
    {
        "id": 8,
        "email": "lindsay.ferguson@reqres.in",
        "first_name": "Lindsay",
        "last_name": "Ferguson",
        "avatar": "https://reqres.in/img/faces/8-image.jpg"
    }
]
  
```

Annotations in red and blue highlight the following fields in the JSON response:

- Red highlights:** "data", "id", "email", "first_name", "last_name", "avatar".
- Blue highlights:** "id", "email", "first_name", "last_name", "avatar".

4. Validating different Status Codes → EC=200-OK=Pass

The screenshot shows the Postman interface with a successful API call. The request details are as follows:

- Method:** GET
- Endpoint:** https://reqres.in/api/users
- Parameters:** ?page=2

The response body is displayed in JSON format:

```

    "page": 2,
    "per_page": 6,
    "total": 12,
    "total_pages": 2,
    "data": [
        {
            "id": 7,
            "email": "michael.lawson@reqres.in",
            "first_name": "Michael",
            "last_name": "Lawson",
            "avatar": "https://reqres.in/img/faces/7-image.jpg"
        }
    ]
  
```

The response headers include:

- Content-Encoding: gzip
- Vary: Accept-Encoding
- #status#: HTTP/1.1 200 OK
- alt-svc: h3=":443"; ma=86400, h3-29=":443"; ma=86400, 2193
- Age: 2193
- Content-Type: application/json; charset=utf-8
- X-Powered-By: Express

5. Validating time taken for the response → EC=5secs=Pass

The screenshot shows the Postman interface with a successful API call. The request details are identical to the previous one.

The response body is identical to the previous one:

```

    "page": 2,
    "per_page": 6,
    "total": 12,
    "total_pages": 2,
    "data": [
        {
            "id": 7,
            "email": "michael.lawson@reqres.in",
            "first_name": "Michael",
            "last_name": "Lawson",
            "avatar": "https://reqres.in/img/faces/7-image.jpg"
        }
    ]
  
```

The response headers are also identical to the previous one. Additionally, the status bar at the bottom of the interface shows the response time: 1108ms (1030 bytes).

6. Applying assertion for verification

SoapUI 5.6.0

File Project Suite Case Step Tools Desktop Help

Empty SOAP REST Import Save All Forum Trial Preferences Proxy Endpoint Explorer Search Forum Online Help Inspector

SoapUI Start Page

Request 1

Method: GET Endpoint: https://reqres.in Resource: /api/users Parameters: ?page=2

Request Name Value Style Level

Raw: page 2 QUERY RESOURCE

Raw XML JSON HTML

```

1 {
2   "page": 2,
3   "per_page": 6,
4   "total": 12,
5   "total_pages": 2,
6   "data": [
7     {
8       "id": 7,
9       "email": "michael.lawson@reqres.in",
10      "first_name": "Michael",
11      "last_name": "Lawson",
12      "avatar": "https://reqres.in/img/faces/7-image.jpg"
13    }
  
```

Header

Header	Value
Content-Encoding	gzip
Vary	Accept-Encoding
status#	HTTP/1.1 200 OK
alt-svc	h3=":443"; ma=86400, h3-29=":443"; ma=86400
Age	2193
Content-Type	application/json; charset=utf-8
X-Powered-By	Express

Headers (20) Attachments (...) SSL Info (2 certs) Representations (1) Schema (conflict...) JMS (0)

response time: 1108ms (1030 bytes)

|| meet.google.com is sharing your screen. Stop sharing Hide

SoapUI log http log jetty log error log wsm... memory l...

Properties

Type here to search

SoapUI 5.6.0

File Project Suite Case Step Tools Desktop Help

Empty SOAP REST Import Save All Forum Trial Preferences Proxy Endpoint Explorer Search Forum Online Help Inspector

SoapUI Start Page

Request 1

Endpoint: https://reqres.in

Resource/Method: /api/users?page=1

Raw Request

Name Value Style Level

Raw XML JSON HTML

```

1 {
2   "page": 1,
3   "per_page": 6,
4   "total": 12,
5   "total_pages": 2,
6   "data": [
7     {
8       "id": 1,
9       "email": "michael.lawson@reqres.in",
10      "first_name": "Michael",
11      "last_name": "Lawson",
12      "avatar": "https://reqres.in/img/faces/1-image.jpg"
13    }
  
```

Headers (20) Attachments (...) SSL Info (2 certs) Representations (1) Schema (conflict...) JMS (0)

response time: 878ms (1030 bytes)

|| meet.google.com is sharing your screen. Stop sharing Hide

Assertions (0) Request Log (1)

SoapUI log http log jetty log error log wsm... memory l...

Properties

Type here to search

SoapUI 5.6.0

File Project Suite Case Step Tools Desktop Help

Empty SOAP REST Import Save All Forum Trial Preferences Proxy Endpoint Explorer Search Forum Online Help

Navigator

- Katraj_Nov_C
 - OCTB
 - Project 8
 - REST Project 1
 - REST Project 10
 - REST Project 11
 - REST Project 12
 - REST Project 13
 - REST Project 14
 - REST Project 15
 - REST Project 16
 - REST Project 17
 - REST Project 18
 - https://reqres.in
 - https://reqres.in TestSuite
 - Users TestCase
 - Test Steps (1)
 - Request 1
 - Load Tests (0)
 - Security Tests (0)
 - REST Project 2
 - REST Project 3
 - REST Project 4
 - REST Project 5
 - REST Project 6
 - REST Project 7
 - REST Project 8

Custom Properties

Property	Value
Name	Request 1

Properties

Type here to search

SOAP REST Import Save All Forum Trial Preferences Proxy Endpoint Explorer Search Forum Online Help

09:38 14-04-2022 ENG

SoapUI Start Page

Request 1

Endpoint: https://reqres.in

Resource/Method: [/api/users?page=2]

Valid HTTP status codes Assertion

Specify codes: 200

Comma-separated acceptable status codes

Assertions (1) Request Log (1)

response time: 870ms (1030 bytes)

SoapUI log http log jetty log error log wsm... memory log

meet.google.com is sharing your screen. Stop sharing Hide

SoapUI Start Page

Request 1

Endpoint: https://reqres.in

Resource/Method: [/api/users?page=2]

Raw Request

page 2

Header

#status# HTTP/1.1 200 OK

alt-svc h3=":443"; ma=86400, h3-29=":443"; ma=86400

Age 3277

Headers (20) Attachments (..) SSL Info (2 certs) Representations (1) Schema (conflict...) JMS (0)

Valid HTTP Status Codes - VALID

Assertions (1) Request Log (1)

response time: 870ms (1030 bytes)

The screenshot shows the SoapUI interface with a successful API request. The request details are as follows:

- Endpoint:** https://reqres.in
- Resource/Method:** GET Users -> Users 1 [/api/users?page=2]
- Request Body (Raw XML):**

```
1 <[{"page": 2, "per_page": 2}]>
```
- Headers:**

Header	Value
Accept-Encoding	HTTP/1.1 200 OK
Content-Type	application/json; charset=utf-8
Age	2895
Alt-Svc	h3=":443"; ma=86400, h3-29=":443"; ma=86400
#status#	200
- Assertions:** Valid HTTP Status Codes - VALID, Response SLA - VALID
- Request Log:** response time: 758ms (1030 bytes)

7. Validating the functionality by passing Test data(Re-Testing)

7.1 Verify by Passing Parameter as 1 in the GET Method ➔ EC=Status Code-200-Ok & GET Method will show response=Pass

SoapUI Start Page

Request 1

Method: GET Endpoint: https://reqres.in Resource: /api/users Parameters: ?page=2

Raw Request:

```
page 2 QUERY RESOURCE
```

Raw Response:

```
[{"page": 2, "per_page": 6, "total": 12, "total_pages": 2, "data": [{"id": 1, "email": "george.bluth@reqres.in", "first_name": "George", "last_name": "Bluth", "avatar": "https://reqres.in/img/faces/1-image.jpg"}, {"id": 2, "email": "janet.weaver@reqres.in", "first_name": "Janet", "last_name": "Weaver", "avatar": "https://reqres.in/img/faces/2-image.jpg"}]}
```

Headers (20) Attachments (...) SSL Info (2 certs) Representations (1) Schema (conflict...) JMS (0)

SoapUI Start Page

Request 1

Method: GET Endpoint: https://reqres.in Resource: /api/users Parameters: ?page=1

Raw Request:

```
page 1 QUERY RESOURCE
```

Raw Response:

```
[{"page": 1, "per_page": 6, "total": 12, "total_pages": 2, "data": [{"id": 1, "email": "george.bluth@reqres.in", "first_name": "George", "last_name": "Bluth", "avatar": "https://reqres.in/img/faces/1-image.jpg"}, {"id": 2, "email": "janet.weaver@reqres.in", "first_name": "Janet", "last_name": "Weaver", "avatar": "https://reqres.in/img/faces/2-image.jpg"}]}
```

Headers (20) Attachments (...) SSL Info (2 certs) Representations (1) Schema (conflict...) JMS (0)

7.2 Verify by Passing Parameter as 2 in the GET Method → EC=Status Code-200-Ok & GET Method will show response=Pass

The screenshot shows the SoapUI interface with the following details:

- Endpoint Explorer:** Shows a project named "Katraj_Nov_C" with various REST projects listed.
- Request 1:** A GET request to <https://reqres.in> with the resource path `/api/users`.
- Parameters:** A parameter `?page=2` is added.
- Request View:** The raw JSON response is displayed, showing two user objects (Michael and Lindsay) on page 2.

```

1 {
  "page": 2,
  "per_page": 6,
  "total": 12,
  "total_pages": 2,
  "data": [
    {
      "id": 7,
      "email": "michael.lawson@reqres.in",
      "first_name": "Michael",
      "last_name": "Lawson",
      "avatar": "https://reqres.in/img/faces/7-image.jpg"
    },
    {
      "id": 8,
      "email": "lindsay.ferguson@reqres.in",
      "first_name": "Lindsay",
      "last_name": "Ferguson",
      "avatar": "https://reqres.in/img/faces/8-image.jpg"
    }
  ]
}
  
```

7.3 Verify by Passing Parameter as 3 in the GET Method → EC=Status Code-200-Ok & GET Method will show response=Pass

The screenshot shows the SoapUI interface with the following details:

- Endpoint Explorer:** Shows a project named "Katraj_Nov_C" with various REST projects listed.
- Request 1:** A GET request to <https://reqres.in> with the resource path `/api/users`.
- Parameters:** A parameter `?page=3` is added.
- Request View:** The raw JSON response is displayed, showing two user objects (Lindsay and Michael) on page 3.

```

1 {
  "page": 3,
  "per_page": 6,
  "total": 12,
  "total_pages": 2,
  "data": [
    {
      "id": 8,
      "email": "lindsay.ferguson@reqres.in",
      "first_name": "Lindsay",
      "last_name": "Ferguson",
      "avatar": "https://reqres.in/img/faces/8-image.jpg"
    },
    {
      "id": 7,
      "email": "michael.lawson@reqres.in",
      "first_name": "Michael",
      "last_name": "Lawson",
      "avatar": "https://reqres.in/img/faces/7-image.jpg"
    }
  ]
}
  
```

7.4 Verify by Passing Parameter as 10/100/1000 in the GET Method → EC=Status Code-200-Ok & GET Method will show response=Pass

8. Validating negative test cases!

8.1 Verify by passing decimal values in GET Method → EC=Status Code-400-Bad Request & GET Method will not show the response=Fail-Defect

The screenshot shows the SoapUI interface with a project named "Katraj_Nov_C". The "Request 1" tab is active, displaying a GET request to the endpoint <https://reqres.in> with the resource path `/api/users`. The "Parameters" section contains a parameter `page=1.5`. The "Raw" tab shows the JSON response:

```

1 {
  "page": 1,
  "per_page": 6,
  "total": 12,
  "total_pages": 2,
  "data": [
    {
      "id": 1,
      "email": "george.bluth@reqres.in",
      "first_name": "George",
      "last_name": "Bluth",
      "avatar": "https://reqres.in/img/faces/1-image.jpg"
    }
  ]
}

```

The "Header" section includes standard headers like Content-Type, Accept-Encoding, and Vary.

8.2 Verify by passing parameter as Null/Blank value in GET Method → EC=Status Code-400-Bad Request & GET Method will not show the response=Fail-Defect

This screenshot is identical to the one above, showing the same SoapUI setup and endpoint. The difference is in the "Parameters" section where the `page` parameter is set to `null` instead of `1.5`. The response JSON remains the same:

```

1 {
  "page": 1,
  "per_page": 6,
  "total": 12,
  "total_pages": 2,
  "data": [
    {
      "id": 1,
      "email": "george.bluth@reqres.in",
      "first_name": "George",
      "last_name": "Bluth",
      "avatar": "https://reqres.in/img/faces/1-image.jpg"
    }
  ]
}

```

8.3 Verify by passing parameter as Character/String value in GET Method → EC=Status Code-400-Bad Request & GET Method will not show the response=Fail-Defect

The screenshot shows the SoapUI interface with the following details:

- File Bar:** File, Project, Suite, Case, Step, Tools, Desktop, Help.
- Toolbar:** Empty, SOAP, REST, Import, Save All, Forum, Trial, Preferences, Proxy, Endpoint Explorer.
- Search Bar:** Search Forum.
- Navigator:** Shows a project structure with various REST projects and a specific test case named "Request 1".
- Request Panel:** Method: GET, Endpoint: https://reqres.in/api/users?page=Two, Resource: /api/users, Parameters: ?page=Two.
- Response View:**
 - Raw (JSON):**

```
{
  "page": 1,
  "per_page": 6,
  "total": 12,
  "total_pages": 2,
  "data": [
    {
      "id": 1,
      "email": "george.bluth@reqres.in",
      "first_name": "George",
      "last_name": "Bluth",
      "avatar": "https://reqres.in/img/faces/1-image.jpg"
    }
  ]
}
```
 - Header:**

Header	Value
NEL	{"success_fraction":0,"report_to":"cf-nel","max_age":604800}
Report-To	{"endpoints":[{"url": "https://a.nel.cloudflare.com/report..."}]}
Content-Encoding	gzip
Vary	Accept-Encoding
#status#	HTTP/1.1 200 OK
alt-svc	h3=":443"; ma=86400, h3-29=":443"; ma=86400
Content-Type	application/json; charset=utf-8
- Bottom Status:** Headers (19), Attachments (...), SSL Info (2 certs), Representations (1), Schema (config...), JMS (0).

8.4 Verify by Passing wrong URL/URI in GET Method → EC=Status Code-404- Not Found & GET Method will not show the response=Pass

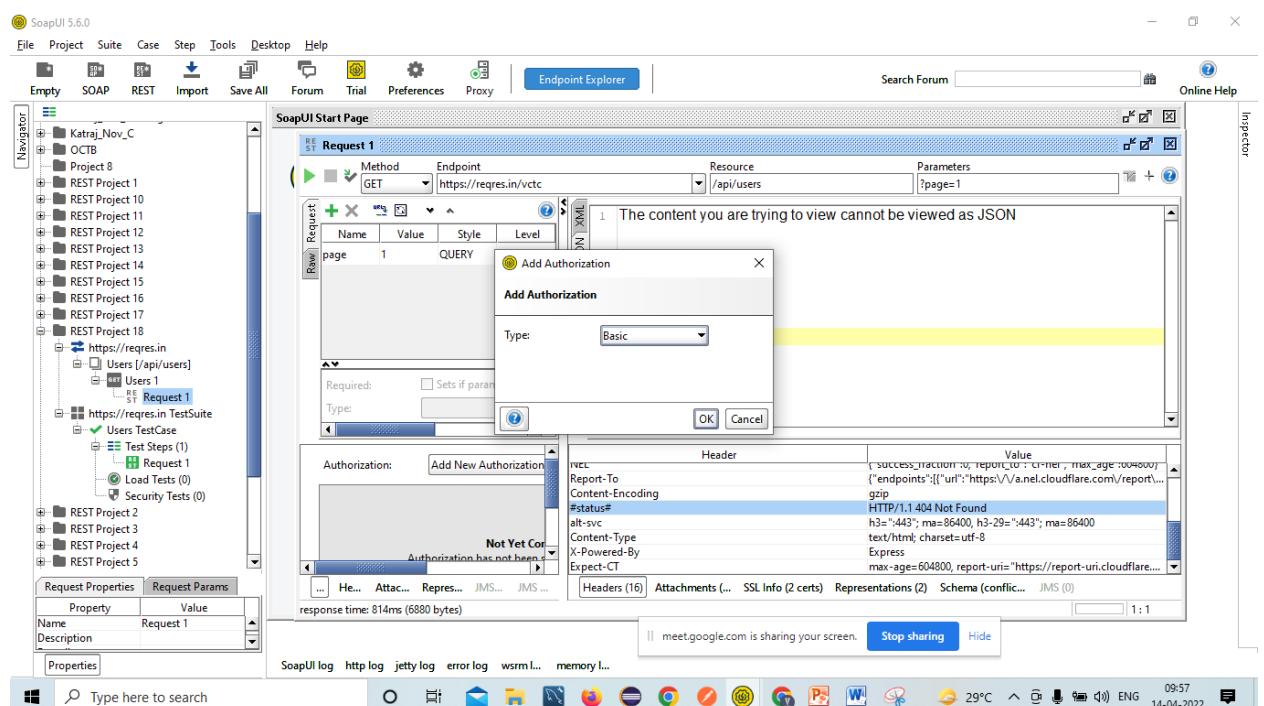
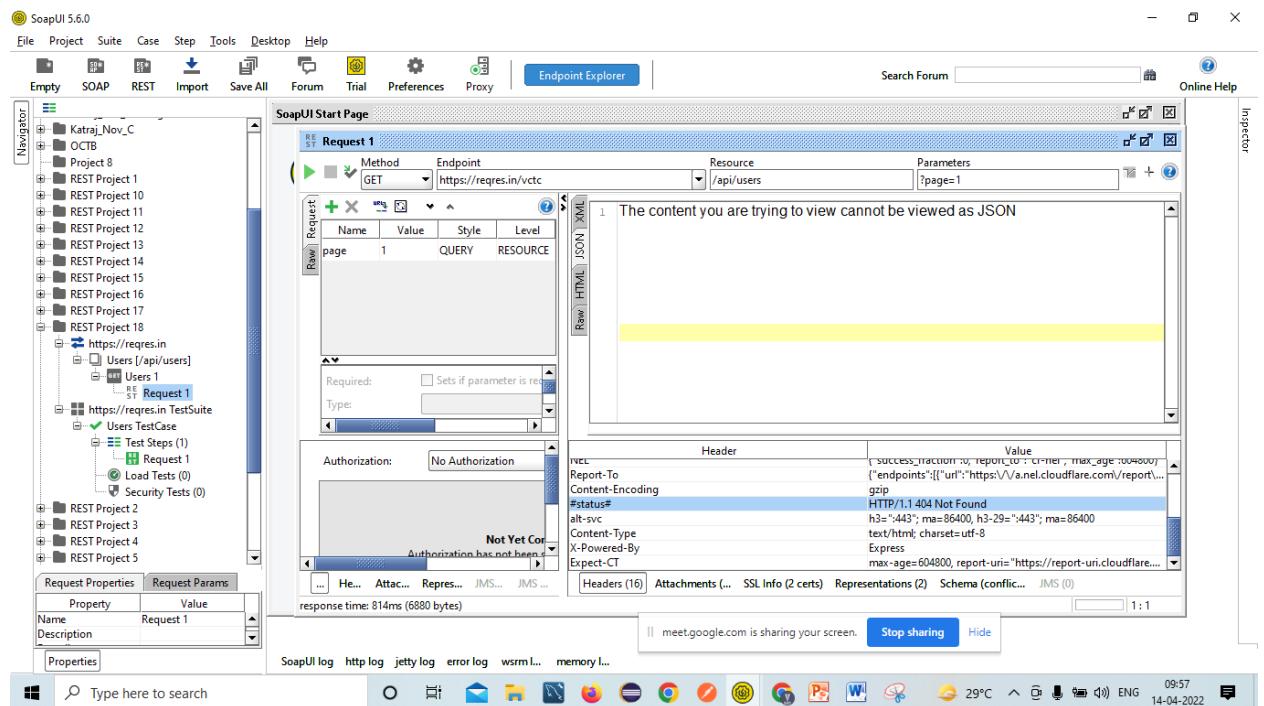
The screenshot shows the SoapUI interface with the following details:

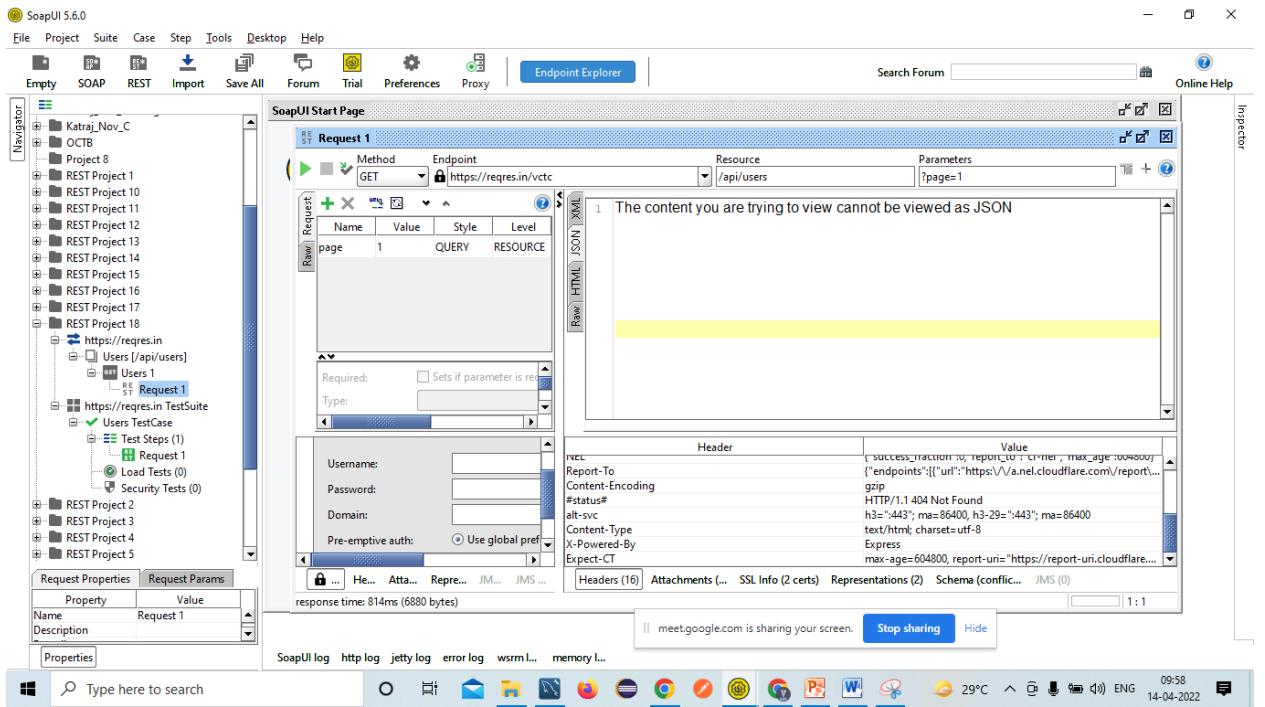
- File Bar:** File, Project, Suite, Case, Step, Tools, Desktop, Help.
- Toolbar:** Empty, SOAP, REST, Import, Save All, Forum, Trial, Preferences, Proxy, Endpoint Explorer.
- Search Bar:** Search Forum.
- Navigator:** Shows a project structure with various REST projects and a specific test case named "Request 1".
- Request Panel:** Method: GET, Endpoint: https://reqres.in/vtc, Resource: /api/users, Parameters: ?page=1.
- Response View:**
 - Raw (HTML):**

The content you are trying to view cannot be viewed as JSON
 - Header:**

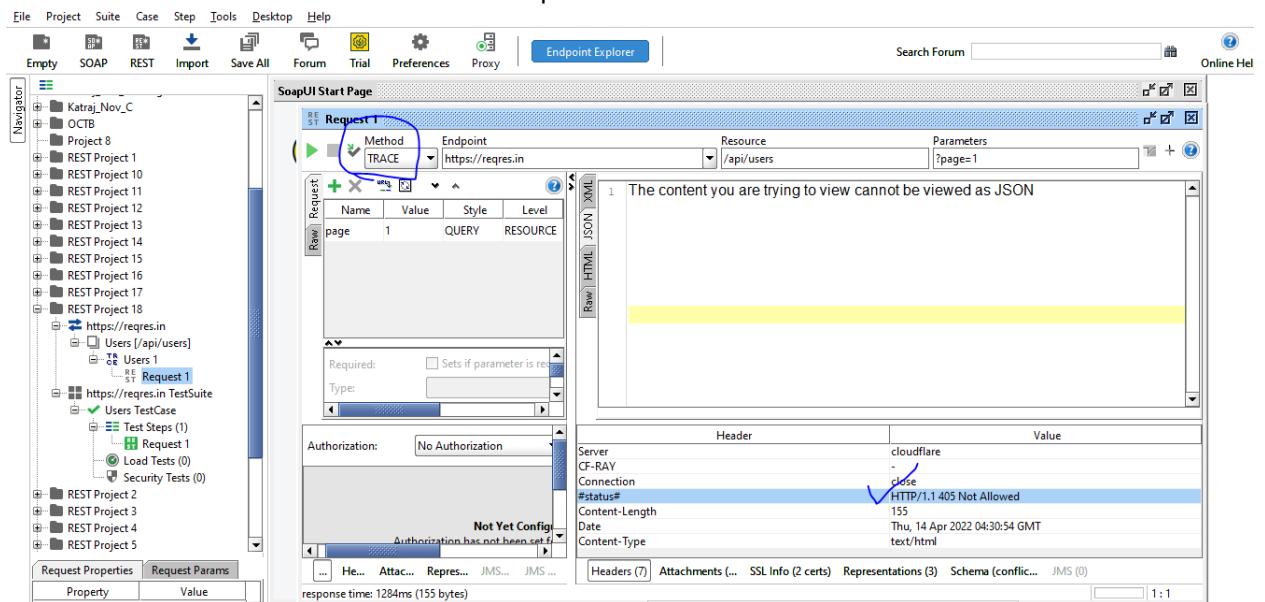
Header	Value
NEL	{"success_fraction":0,"report_to":"cf-nel","max_age":604800}
Report-To	{"endpoints":[{"url": "https://a.nel.cloudflare.com/report..."}]}
Content-Encoding	gzip
#status#	HTTP/1.1 404 Not Found
alt-svc	h3=":443"; ma=86400, h3-29=":443"; ma=86400
Content-Type	text/html; charset=utf-8
X-Powered-By	Express
Expect-CT	max-age=604800, report-uri="https://report-uri.cloudflare...
- Bottom Status:** Headers (16), Attachments (...), SSL Info (2 certs), Representations (2), Schema (config...), JMS (0).

8.5 Verify by passing wrong Authorisation/bearer Token/API Key/Basic Auth(Username & Password) in GET Method → EC=Status Code-401-UnAuthorization & GET Method will not show the response.





8.6 Verify by changing GET Method into another method → EC=Status Code-405-Method Not Allowed & GET Method will not show the response=Pass



15/04/2022:

Agenda: REST Terminologies, POST Method

REST Terminologies:

End Point/BaseURL: Address where API is hosted on the server

HTTP Methods which are commonly used to communicate with REST API's are: GET, POST, PUT, PATCH, DELETE

GET: The GET is used to extract information from the given server using a given URI. While using GET request, it should only extract data and should have no other effect on data. No Payload/Body required.

How to send input data in GET Method?

Ans: Using Query Parameters

POST: A post request is used to send data to the server, for example, Customer information, File Upload, etc using HTML forms.

How to send input data in POST Method?

Ans: Using Body Payload

PUT: Replaces all current representations of the target resource with uploaded content

DELETE: Remove the contents from the target resource

Resources:

Resources represents API/Collection which can be accessed from the servers

Google.com/maps

Google.com/images

Google.com/search

Path Parameters: are variable parts of the URL Path. They are typically used to point to a specific resource within a collection, such as user id identified by ID

<https://www.google.com/images/1123343>

<https://www.google.com/docs/1123343>

<https://www.amazon.com/orders/112>

Query Parameters:

Query Parameter is used to sort/filter the resources

Query Parameter is identified with “?”

https://www.amazon.com/orders?sort_by=2/20/2022

End Point Request URL can be constructed as below:

URI=(Base URL/End Point)/Resource/(Query/Path)Parameters

URI= **URL**+Resource+Parameter Value

URL & URI-

URL – uniform resource locator

URI- uniform resource identifier

EX.

<https://reqres.in/api/users?page=2>

where

Base URL/Domain/Endpoint – <https://reqres.in>

URI- <https://reqres.in/api/users?page=2>

URI contains: URL & other part

contains

1. parameters/Resource→/api/users

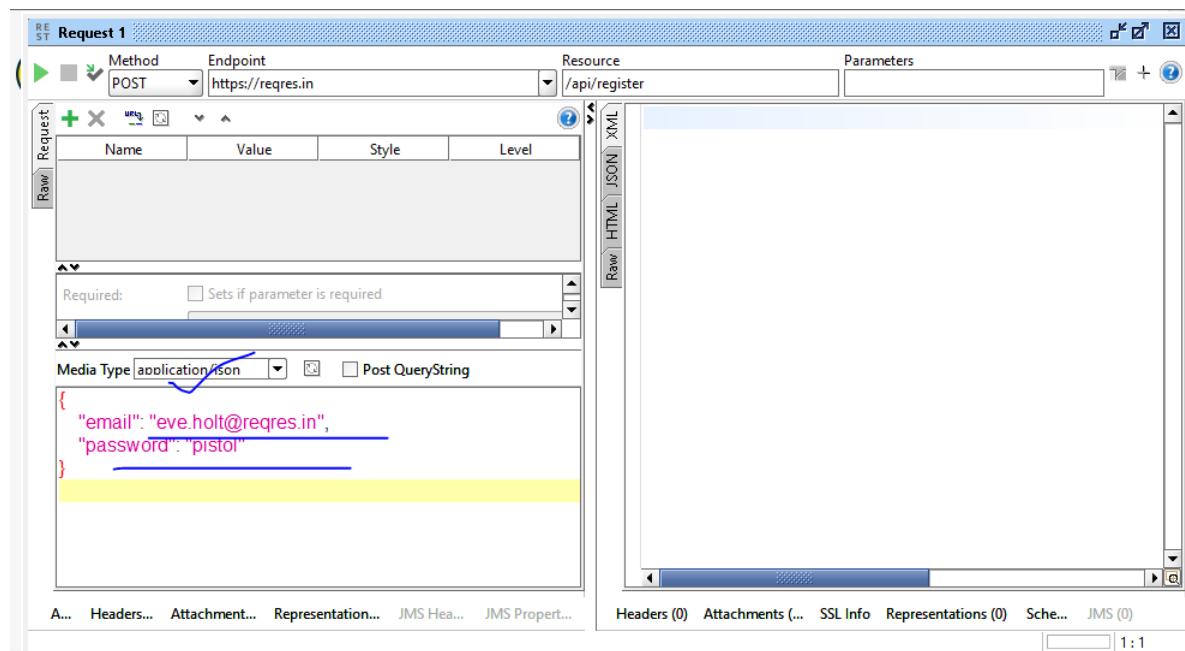
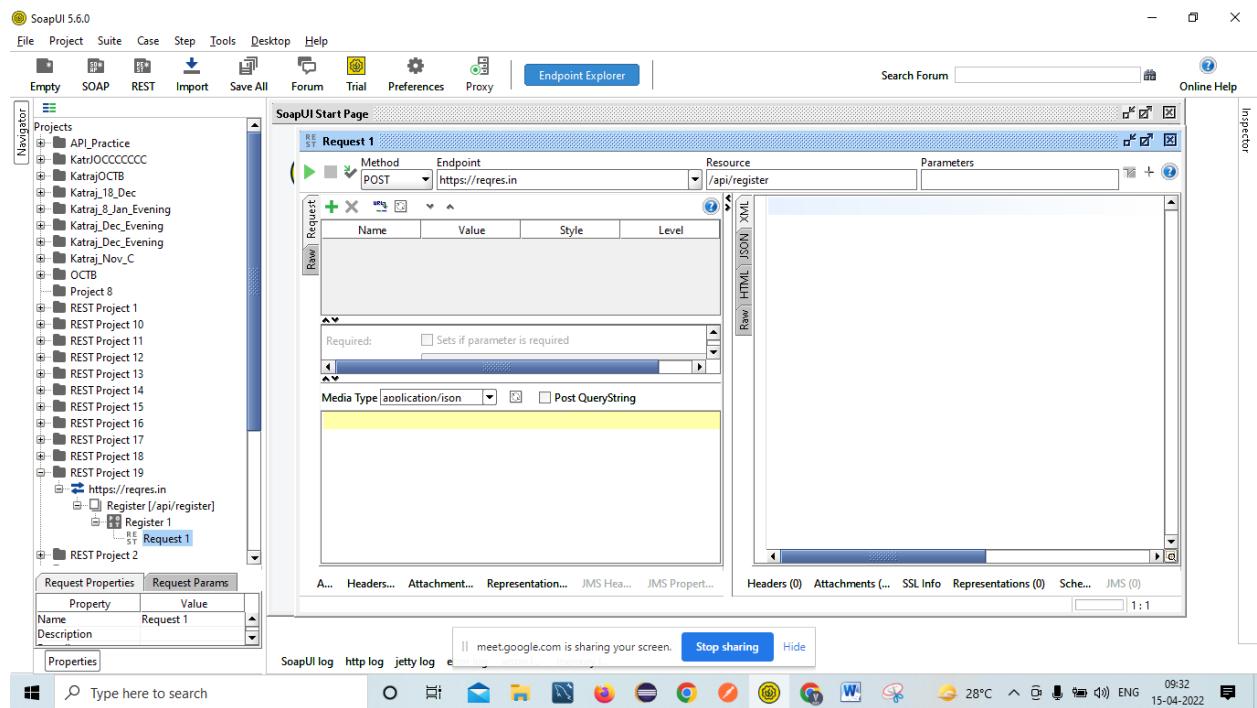
2. Parameter values/ Query String→after ? mark ex. page=2

URI= **URL** +Resource +Parameter value

POST Method/Request:

- POST Method Service-URL/URI=<https://reqres.in/api/register>
- Body Payload:

```
{  
    "email": "eve.holt@reqres.in",  
    "password": "pistol"  
}
```
- Unit Testing Document
- Time Taken for the Response=5Sec
- Username & Password(Basic Auth)/API Key/Bearer Token-dalablfblf



Test Cases:

1. Validating REST Response=Pass

The screenshot shows a REST client interface with the following details:

- Request 1:** Method: POST, Endpoint: https://reqres.in, Resource: /api/register, Parameters: None.
- Request Body:** Media Type: application/json. The JSON payload is:

```
{
  "email": "eve.holt@reqres.in",
  "password": "pistol"
}
```
- Response Body:** Raw tab shows the response JSON:

```
1曰{
2  "id": 4,
3  "token": "QpwL5tke4Pnpja7X4"
4 }
```
- Bottom Status:** Headers (16), Attachments (...), SSL Info (2 certs), Representations (1), Schema (conflic...), JMS (0). response time: 1236ms (36 bytes).

2. Validating Data and Count of data in the Response=Pass

The screenshot shows a REST client interface with the following details, similar to Request 1 but with validation highlights:

- Request 1:** Method: POST, Endpoint: https://reqres.in, Resource: /api/register, Parameters: None.
- Request Body:** Media Type: application/json. The JSON payload is:

```
{
  "email": "eve.holt@reqres.in",
  "password": "pistol"
}
```
- Response Body:** Raw tab shows the response JSON. A blue oval highlights the first two fields: "id": 4 and "token": "QpwL5tke4Pnpja7X4".

```
1曰{
2  "id": 4,
3  "token": "QpwL5tke4Pnpja7X4"
4 }
```
- Bottom Status:** Headers (16), Attachments (...), SSL Info (2 certs), Representations (1), Schema (conflic...), JMS (0). response time: 1236ms (36 bytes).

3. Validating Tagname/Attributes Present in the Response=EC=>Id,Token=Pass

The screenshot shows the SoapUI interface with the following details:

- File Bar:** File, Project, Suite, Case, Step, Tools, Desktop, Help.
- Toolbar:** Empty, SOAP, REST, Import, Save All, Forum, Trial, Preferences, Proxy, Endpoint Explorer, Search Forum, Online Help.
- Navigator:** Projects section listing various projects like API_Practice, KatrajOCCCCCCC, KatrajOCB, etc.
- Request Panel:**
 - Method: POST
 - Endpoint: https://reqres.in/api/register
 - Resource: /api/register
 - Parameters: None
- Request Body:** Raw, XML, JSON, HTML. The JSON tab shows the following payload:


```
{
        "email": "eve.holt@reqres.in",
        "password": "pistol"
      }
```
- Response Body:** Raw, XML, JSON, HTML. The JSON tab shows the response body:


```
{
        "id": 4,
        "token": "QpwL5tke4Pnpja7X4"
      }
```
- Response Headers:**

Header	Value
Cache-Control	W/24-4fP0za1geN2he+ohu8F0FhQJLks
Etag	NEL
Report-To	#status#
Status	HTTP/1.1 200 OK
Content-Length	36
Alt-Svc	h3=":443": ma=86400, h3-29=":443"; ma=86400
Content-Type	application/json; charset=utf-8

4. Validating Different Status Codes=>EC=Status Code-201-Created=Fail-Defect

The screenshot shows the SoapUI interface with the following details:

- File Bar:** File, Project, Suite, Case, Step, Tools, Desktop, Help.
- Toolbar:** Empty, SOAP, REST, Import, Save All, Forum, Trial, Preferences, Proxy, Endpoint Explorer, Search Forum, Online Help.
- Navigator:** Projects section listing various projects like API_Practice, KatrajOCCCCCCC, KatrajOCB, etc.
- Request Panel:**
 - Method: POST
 - Endpoint: https://reqres.in/api/register
 - Resource: /api/register
 - Parameters: None
- Request Body:** Raw, XML, JSON, HTML. The JSON tab shows the following payload:


```
{
        "email": "eve.holt@reqres.in",
        "password": "pistol"
      }
```
- Response Body:** Raw, XML, JSON, HTML. The JSON tab shows the response body:


```
{
        "id": 4,
        "token": "QpwL5tke4Pnpja7X4"
      }
```
- Response Headers:**

Header	Value
Cache-Control	W/24-4fP0za1geN2he+ohu8F0FhQJLks
Etag	NEL
Report-To	#status#
Status	HTTP/1.1 200 OK
Content-Length	36
Alt-Svc	h3=":443": ma=86400, h3-29=":443"; ma=86400
Content-Type	application/json; charset=utf-8

5. Validating Time Taken for the response=5Sec=Pass

The screenshot shows the SoapUI interface with a project named "API_Practice". A request named "Request 1" is selected, which is a POST request to the endpoint <https://reqres.in> with the resource path `/api/register`. The request body contains the following JSON:

```
{
  "id": 4,
  "token": "QpwL5tke4Pnpja7X4"
}
```

The response header includes:

- Content-Type: application/json; charset=utf-8
- Content-Length: 36
- Alt-Svc: h3=":443"; ma=86400, h3-29="443"; ma=86400
- Report-To: #status#
- Etag: W/24-4IP0za1geN2he+ohu8OFhGJLs
- Cache-Control: no-store, no-cache, must-revalidate, private
- Set-Cookie: NEL={\"success_fraction\":0,\"report_to\":\"cf-nel\",\"max_age\":604800}; endpoints=[{"url": "https://V.a.net.cloudflare.com/report..."}]
- HTTP/1.1 200 OK

The response body is a JSON object with the same structure as the request body, indicating a successful registration.

6. Applying Assertion for Verifications

7. Validating Functionality by passing Test data(Re-Testing)

7.1 Verify the POST Method/Request by Passing george.bluth@reqres.in with Password=>EC=For this email ID and Password will be inserted in the server.=Pass

The screenshot shows the SoapUI interface with a project named "API_Practice". A request named "Request 1" is selected, which is a POST request to the endpoint <https://reqres.in> with the resource path `/api/register`. The request body contains the following JSON:

```
{
  "email": "george.bluth@reqres.in",
  "password": "pistol"
}
```

The response header includes:

- Content-Type: application/json; charset=utf-8
- Content-Length: 36
- Alt-Svc: h3=":443"; ma=86400, h3-29="443"; ma=86400
- Report-To: #status#
- Etag: W/24-4IP0za1geN2he+ohu8OFhGJLs
- Cache-Control: no-store, no-cache, must-revalidate, private
- Set-Cookie: NEL={\"success_fraction\":0,\"report_to\":\"cf-nel\",\"max_age\":604800}; endpoints=[{"url": "https://V.a.net.cloudflare.com/report..."}]
- HTTP/1.1 200 OK

The response body is a JSON object with the same structure as the request body, indicating a successful registration. Handwritten annotations with blue checkmarks are present next to the "id" and "token" fields in the response body JSON.

7.2 Verify the POST Method/Request by Passing janet.weaver@regres.in with Password=>EC=For this email ID and Password will be inserted in the server.=Pass

The screenshot shows the SoapUI interface with the following details:

- File Bar:** File, Project, Suite, Case, Step, Tools, Desktop, Help.
- Toolbar:** Empty, SOAP, REST, Import, Save All, Forum, Trial, Preferences, Proxy, Endpoint Explorer.
- Navigator:** Projects section showing various projects like API_Practice, KatrajOCB, Katraj_18_Dec, etc.
- Request Panel:**
 - Method: POST
 - Endpoint: https://regres.in
 - Resource: /api/register
 - Parameters: None
- Request Details:**
 - Raw tab: Shows the JSON payload with 'email': 'janet.weaver@regres.in' and 'password': 'abcdef'.
 - JSON tab: Shows the same JSON payload.
 - XML tab: Shows the XML representation of the payload.
- Request Properties:** Shows Request 1 with properties Name: Request 1 and Description: None.

7.3 Verify the POST Method/Request by Passing janet.weaver@regres.in with Password=>EC=For this email ID and Password will be inserted in the server.=Pass

7.4 Verify the POST Method/Request by Passing charles.morris@regres.in with Password=>EC=For this email ID and Password will be inserted in the server.=Pass

The screenshot shows the SoapUI interface with the following details:

- File Bar:** File, Project, Suite, Case, Step, Tools, Desktop, Help.
- Toolbar:** Empty, SOAP, REST, Import, Save All, Forum, Trial, Preferences, Proxy, Endpoint Explorer.
- Navigator:** Projects section showing various projects like API_Practice, KatrajOCB, Katraj_18_Dec, etc.
- Request Panel:**
 - Method: POST
 - Endpoint: https://regres.in
 - Resource: /api/register
 - Parameters: None
- Request Details:**
 - Raw tab: Shows the JSON payload with 'email': 'charles.morris@regres.in' and 'password': 'abcdef'.
 - JSON tab: Shows the same JSON payload.
 - XML tab: Shows the XML representation of the payload.
- Request Properties:** Shows Request 1 with properties Name: Request 1 and Description: None.

7.5 Verify the POST Method/Request by Passing lindsay.ferguson@reqres.in with Password=>EC=For this email ID and Password will be inserted in the server.=Pass

The screenshot shows the SoapUI interface with a project named "API_Practice". A request named "Request 1" is selected, which is a POST method to the endpoint <https://reqres.in/api/register>. The request body is set to application/json and contains the following JSON payload:

```
{
  "id": 8,
  "token": "QpwL5tke4Pnpja7X8",
  "email": "lindsay.ferguson@reqres.in",
  "password": "abcdef"
}
```

7.6 Verify the POST Method/Request by Passing rachel.howell@reqres.in with Password=>EC=For this email ID and Password will be inserted in the server.=Pass

The screenshot shows the SoapUI interface with a project named "API_Practice". A request named "Request 1" is selected, which is a POST method to the endpoint <https://reqres.in/api/register>. The request body is set to application/json and contains the following JSON payload:

```
{
  "id": 12,
  "token": "QpwL5tke4Pnpja7X12",
  "email": "rachel.howell@reqres.in",
  "password": "abcdef"
}
```

8. Validating Negative Test cases!

8.1 Verify POST Method/Request by passing invalid email id [vctc@reqres.in](#)=>EC=400-Bad Request=Pass

The screenshot shows the SoapUI interface with a REST Request 1 configuration. The endpoint is set to <https://reqres.in> and the resource is `/api/register`. The request body is a JSON object:

```
{  
  "email": "vctc@reqres.in",  
  "password": "abcdef"  
}
```

The response status code is 400 Bad Request. The response body is:

```
{"error": "Note: Only defined users succeed registration"}
```

The response headers include:

Header	Value
NEL	{"success_fraction":0,"report_to":"cf-nel","url":"/api/register"}
Report-To	#status#
Content-Length	57
alt-svc	h3=":443"; ma=86400, h3-29=":443"; ma=86400
Content-Type	application/json; charset=utf-8
X-Powered-By	Express
Expect-CT	max-age=604800, report-uri="https://report-uri.com/report/v1/00000000000000000000000000000000"

8.2 Verify the POST Method by Passing Null/Blank email id=>EC=400-Bad Request=Pass

The screenshot shows the SoapUI interface with a REST Request 1 configuration. The endpoint is set to <https://reqres.in> and the resource is `/api/register`. The request body is a JSON object:

```
{  
  "email": "",  
  "password": "abcdef"  
}
```

The response status code is 400 Bad Request. The response body is:

```
{"error": "Note: Only defined users succeed registration"}
```

The response headers include:

Header	Value
NEL	{"success_fraction":0,"report_to":"cf-nel","url":"/api/register"}
Report-To	#status#
Content-Length	57
alt-svc	h3=":443"; ma=86400, h3-29=":443"; ma=86400
Content-Type	application/json; charset=utf-8
X-Powered-By	Express
Expect-CT	max-age=604800, report-uri="https://report-uri.com/report/v1/00000000000000000000000000000000"

8.3 Verify the POST Method/Request by Passing only emailID [rachel.howell@reqres.in](#)

Without Password=>EC=400-Bad Request=Pass

The screenshot shows the SoapUI interface with a POST request to <https://reqres.in> to the resource `/api/register`. The request body is:

```
{
  "email": "rachel.howell@reqres.in"
}
```

The response status is **HTTP/1.1 400 Bad Request**, and the error message is `{"error": "Missing password"}`.

8.4 Verify POST Method /Request by Passing invalid email domain rachel.howell@gmail.com =EC=400-Bad Request=Pass

The screenshot shows the SoapUI interface with a POST request to <https://reqres.in> to the resource `/api/register`. The request body is:

```
{
  "email": "rachel.howell@gmail.com",
  "password": "abcdef"
}
```

The response status is **HTTP/1.1 400 Bad Request**, and the error message is `{"error": "Note: Only defined users succeed registration"}`.

8.5 Verify POST Method/Request by Passing valid id(id=1) with Password=EC=400-Bad Request=Pass

File Project Suite Case Step Tools Desktop Help

Empty SOAP REST Import Save All Forum Trial Preferences Proxy Endpoint Explorer Search Forum Online Help

SoapUI Start Page

Request 1

Method: POST Endpoint: https://reqres.in Resource: /api/register Parameters:

Name	Value	Style	Level

Raw Request: Required: Sets if parameter is required

Media Type: application/json Post QueryString:

```
{
    "email": "lindsay.ferguson@reqres.in",
    "password": "abcde"
}
```

Header:

Header	Value
NEL	{"success_fraction":0,"report_to":"cf-nel","..."} {"endpoints":[{"url":"https://a.nel.cloudflare.com..."}]
Report-To	#status#
Content-Length	57
alt-svc	h3=":443"; ma=86400, h3-29=":443"; ma=8...
Content-Type	application/json; charset=utf-8
X-Powered-By	Express
Expect-CT	max-age=604800, report-uri="https://re...

Response:

```
{"error": "Note: Only defined users succeed registration"}
```

A... Headers... Attachment... Representation... JMS He... JMS Prop... response time: 687ms (57 bytes)

Request Properties Request Params

Property	Value
Name	Request 1

8.6 Verify POST Method Request/Request by Passing wrong URL/URI=>EC=404-Not Found=Pass

File Project Suite Case Step Tools Desktop Help

Empty SOAP REST Import Save All Forum Trial Preferences Proxy Endpoint Explorer Search Forum Online Help

SoapUI Start Page

Request 1

Method: POST Endpoint: https://reqres.com Resource: /api/register Parameters:

Name	Value	Style	Level

Raw Request: Required: Sets if parameter is required

Media Type: application/json Post QueryString:

```
{
    "email": "lindsay.ferguson@reqres.in",
    "password": "abcde"
}
```

Header:

Header	Value
Set-Cookie	apoc__page_must_=1; path=/; httpOnly; sameSite=Lax; Secure
Set-Cookie	apbc_cookies_test=%257B%2522cookies_...; path=/; httpOnly; sameSite=Lax; Secure
Vary	Accept-Encoding
Expires	Wed, 11 Jan 1984 05:00:00 GMT
#status#	HTTP/1.1 404 Not Found
X-XSS-Protection	1; mode=block
Link	<https://www.reqres.com/wp-json/>; rel="..."
Content-Type	text/html; charset=UTF-8

The content you are trying to view cannot be viewed as JSON.

A... Headers... Attachment... Representation... JMS He... JMS Prop... response time: 1223ms (63480 bytes)

Request Properties Request Params

Property	Value
Name	Request 1

8.7 Verify by changing POST Method/Request into Another Method/Request=EC=405-Method Not Allowed=Pass

The screenshot shows the SoapUI interface with the following details:

- Navigator:** Projects list includes API_Practice, KatrajOCCCCCCC, KatrajOCTB, Katraj_18_Dec, Katraj_8_Jan_Evening, Katraj_Dec_Evening, Katraj_Nov_C, OCTB, Project 8, REST Project 1, REST Project 10, REST Project 11, REST Project 12, REST Project 13, REST Project 14, REST Project 15, REST Project 16, REST Project 17, REST Project 18, and REST Project 19.
- Request Panel:** Method is set to TRACE, Endpoint is https://reqres.in, Resource is /api/register. A note says "The content you are trying to view cannot be viewed as JSON".
- Response Panel:** Status code is 405, reason is "Method Not Allowed". Headers include Server: cloudflare, CF-RAY: -, Connection: close, #status#: HTTP/1.1 405 Not Allowed, Content-Length: 155, Date: Fri, 15 Apr 2022 04:24:01 GMT, and Content-Type: text/html.

PUT Method/ Request-

- PUT method Service –URL/ URI= <https://reqres.in/api/users/2>
- Body/ Payload-

```
{
    "name": "morpheus",
    "job": "zion resident"
}
```

- Username & Password/ API Key/ Barrier token - tdcgbshdnjhgfvbhdhcnd
- Time take to response = 60 sec
- Unit Testing documents-

Delete method/ request-

- GET method Service –URL/ URI= <https://reqres.in/api/users/2>
- Username & Password/ API Key/ Barrier token - tdcgbshdnjhgfvbhdhcnd
- Time take to response = 60 sec
- Unit Testing documents-

17/04/2022:

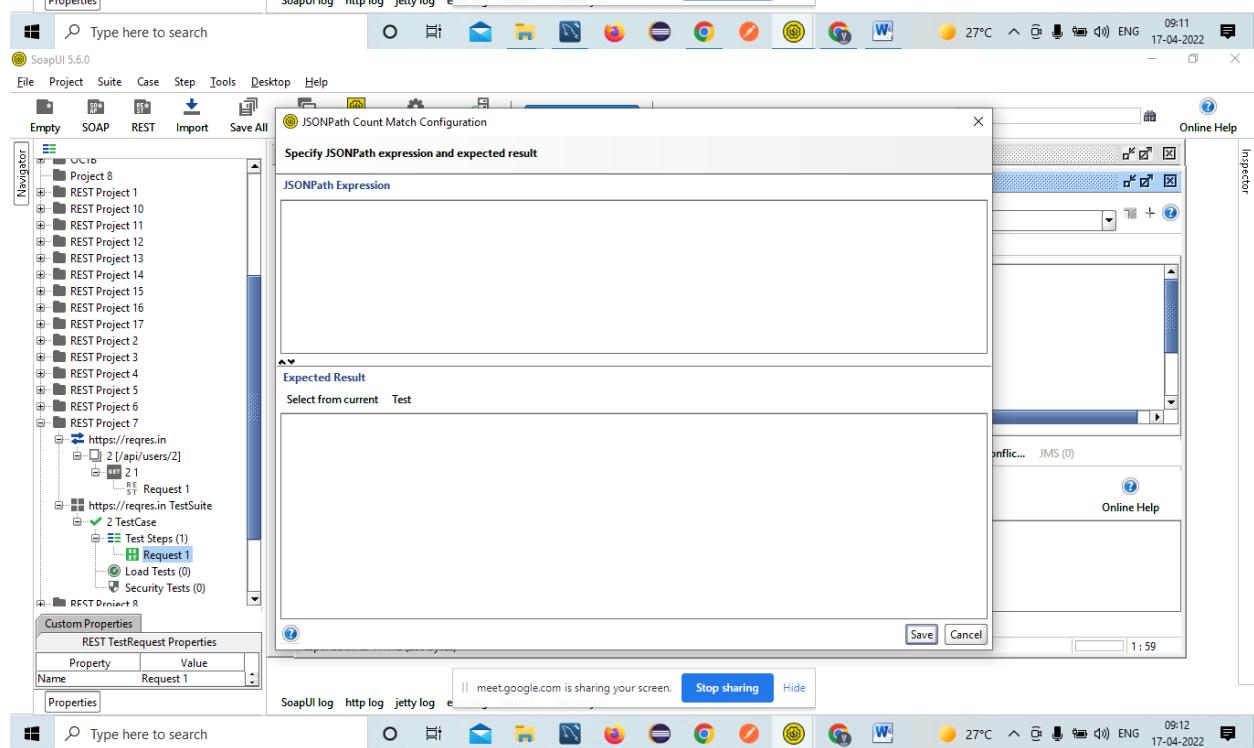
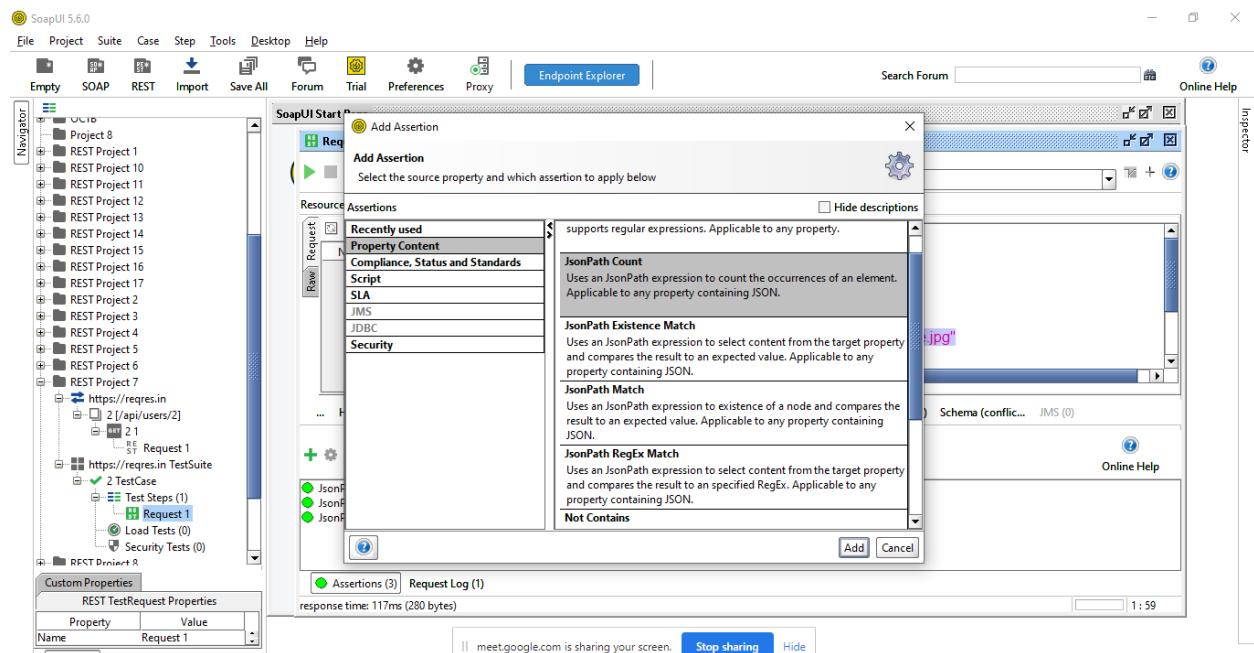
Agenda: JSON Path Count, JSON Path Existence, JSON Path Match, Postman

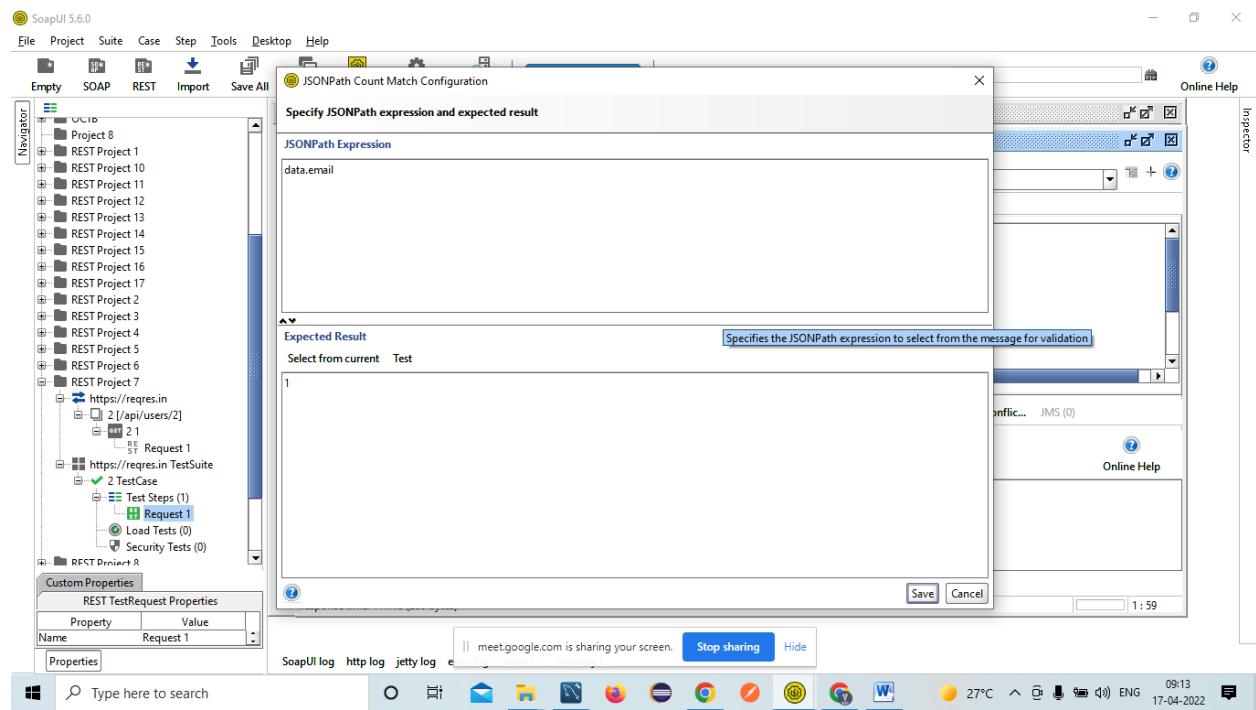
GET METHOD: <https://reqres.in/api/users/2>

Xpath(XML document-SOAP Service) similarly JSON Path(JSON document-REST Service)

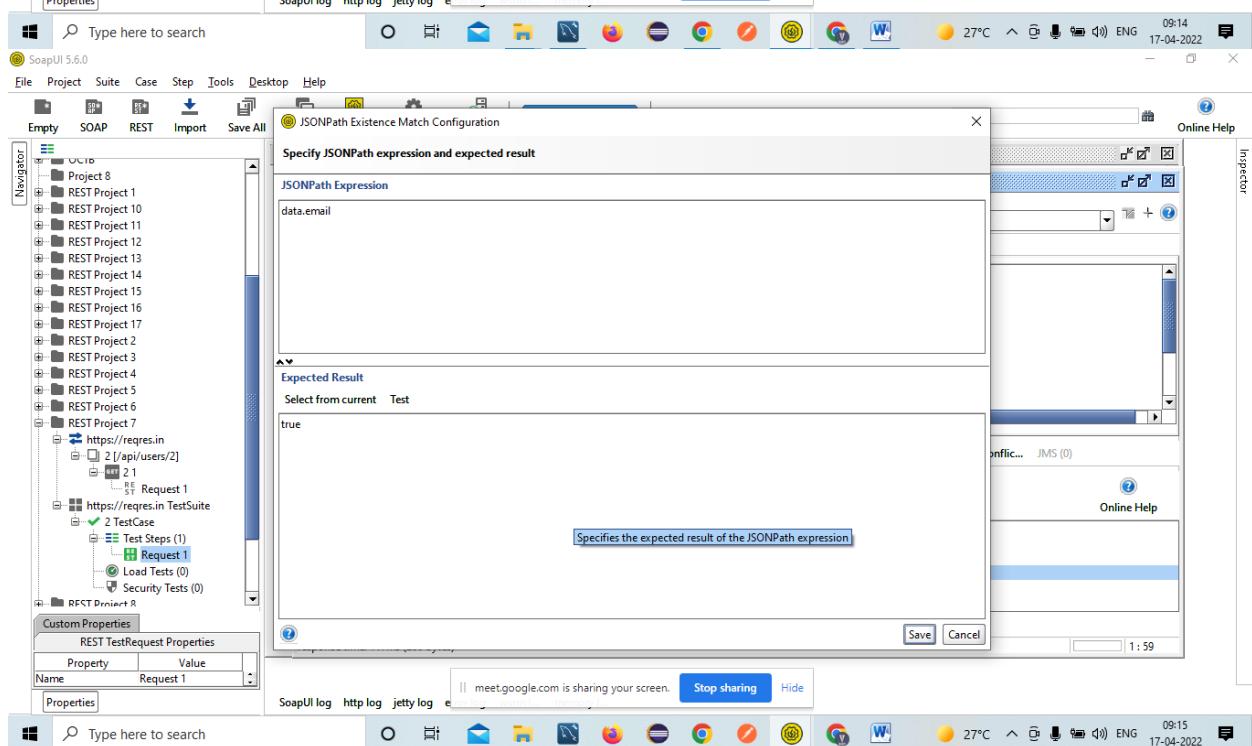
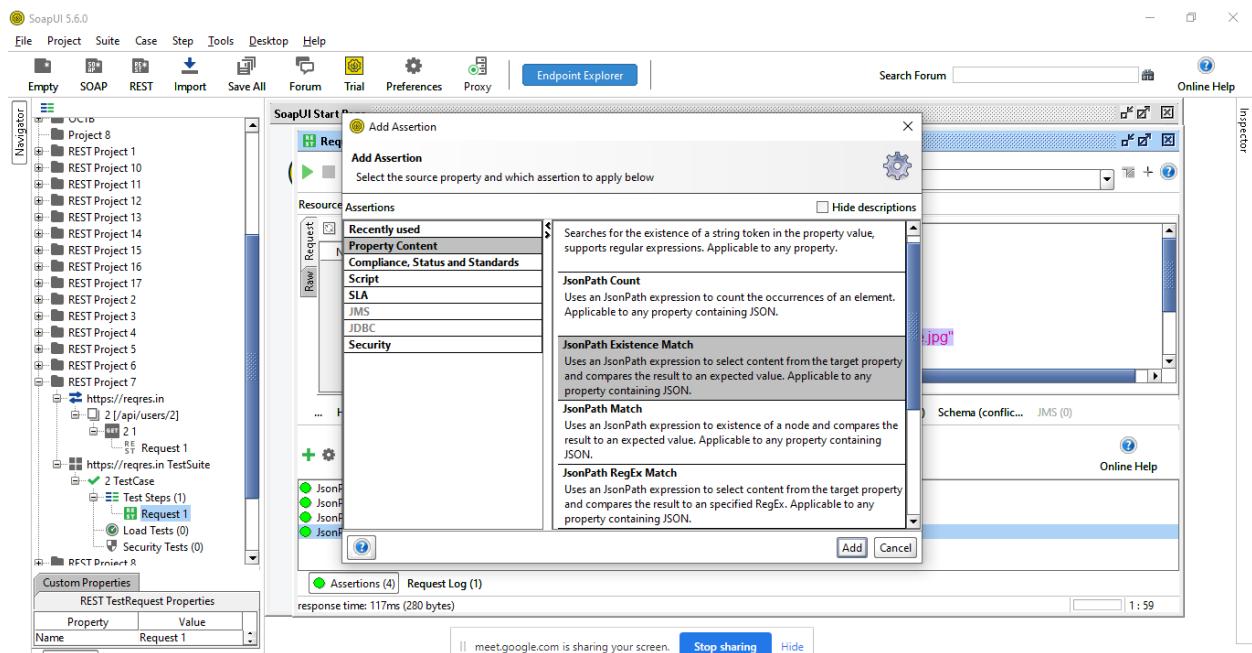
JSON Path Count:

<https://jsonpathfinder.com/>

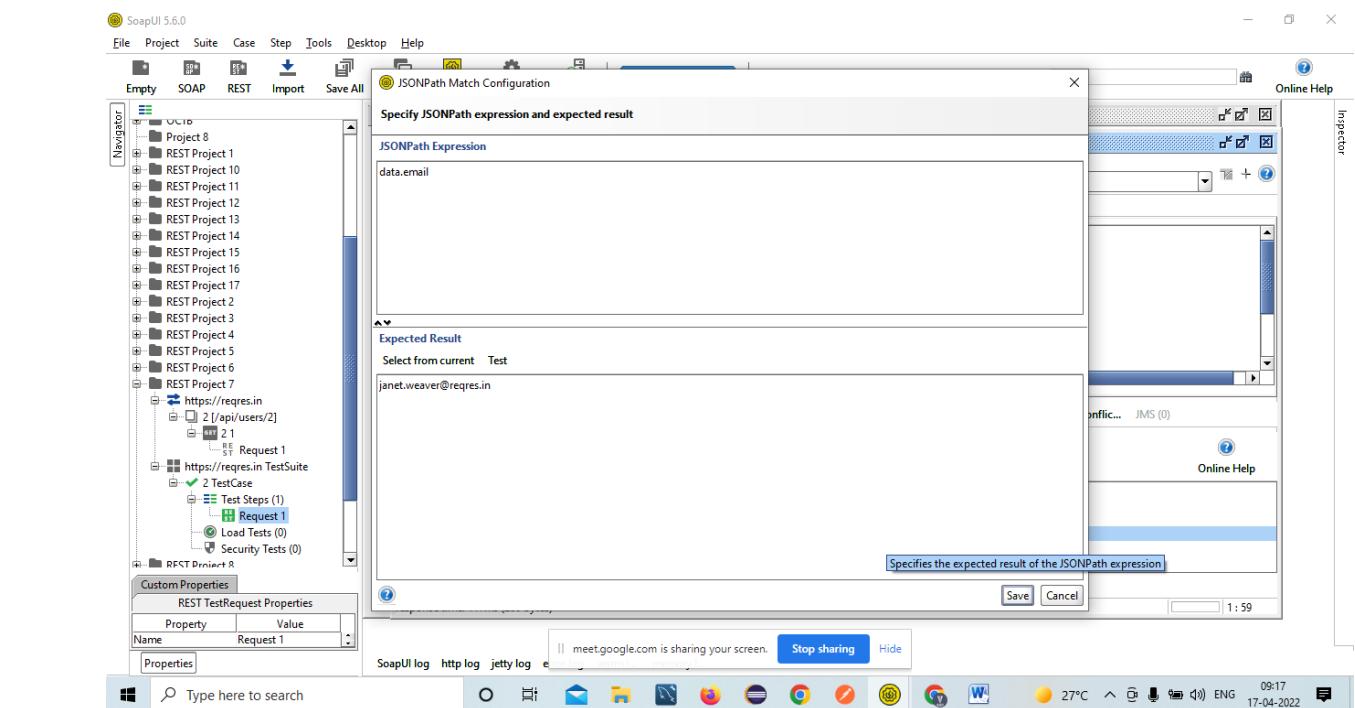
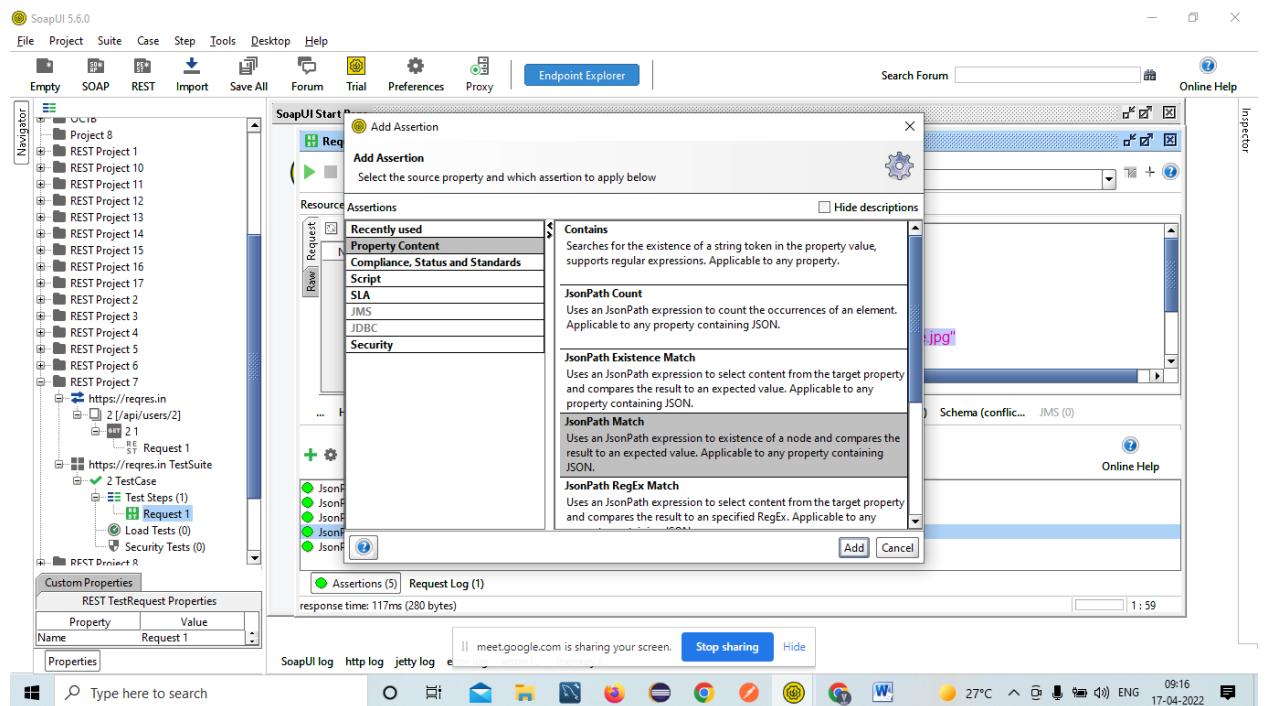




JSON Path Existence: Boolean O/P

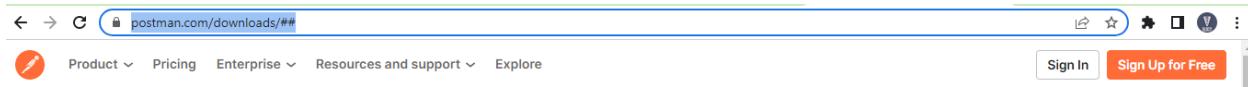


JSON Path Match:



Postman Tool: for Testing REST Services

<https://www.postman.com/downloads/#>

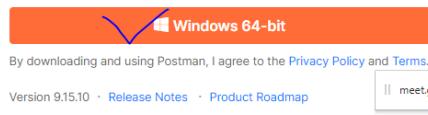


Download Postman

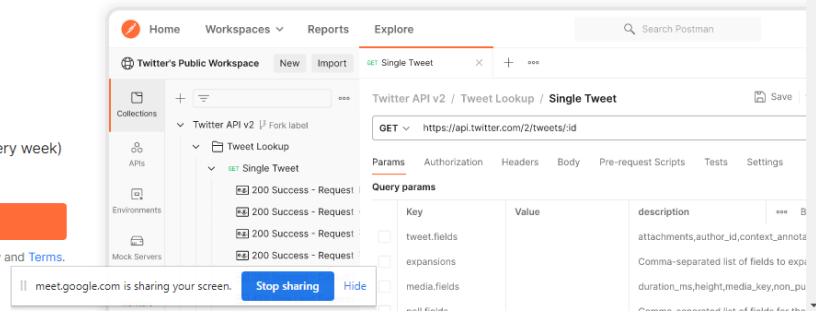
Download the app to quickly get started using the Postman API Platform. Or, if you prefer a browser experience, you can try the new web version of Postman.

The Postman app

The ever-improving Postman app (a new release every week) gives you a full-featured Postman experience.



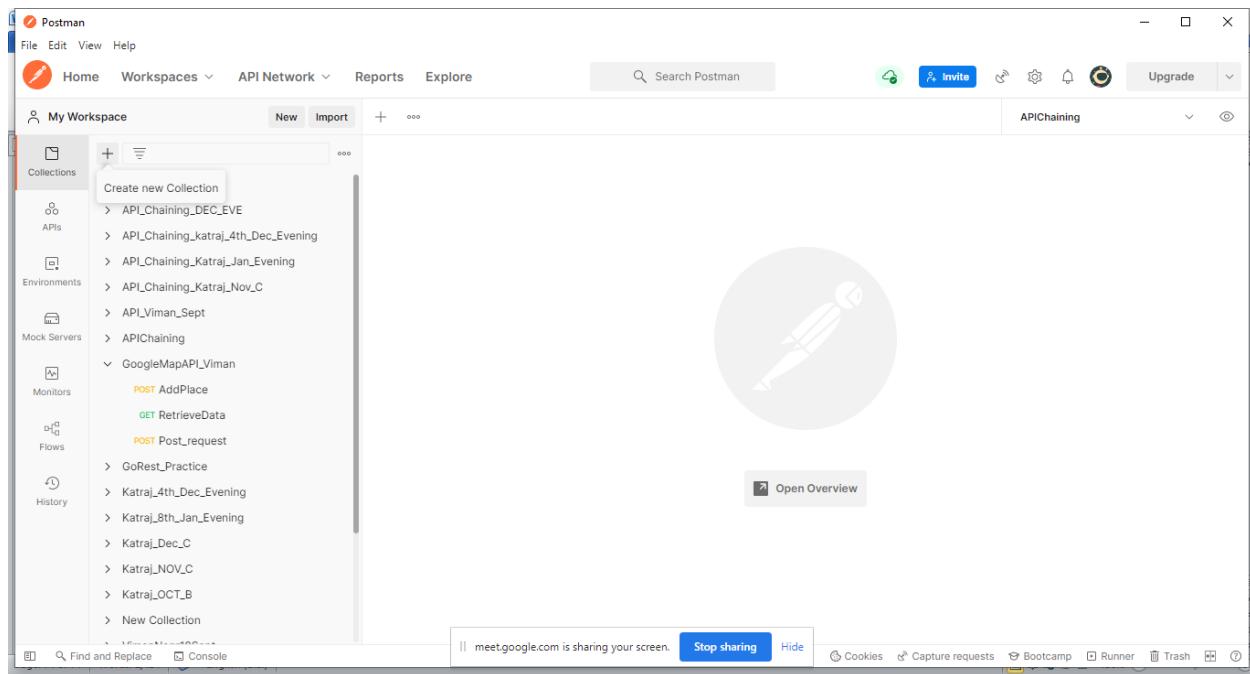
Version 9.15.10 · [Release Notes](#) · [Product Roadmap](#)



The screenshot shows the Postman web interface. The top navigation bar includes Home, Workspaces, Reports, and Explore. The current workspace is "Twitter's Public Workspace". The main area displays a collection named "Twitter API v2" under "Tweet Lookup". A specific request titled "Single Tweet" is selected, showing its details. The request method is GET, and the URL is https://api.twitter.com/2/tweets/:id. The "Query params" section contains several parameters: "description" (with value "B"), "tweet.fields" (unchecked), "expansions" (unchecked), and "media.fields" (unchecked). The right side of the interface shows detailed documentation for the "Single Tweet" endpoint.

What is Collection?

Answer: Group of requests grouped into a folder

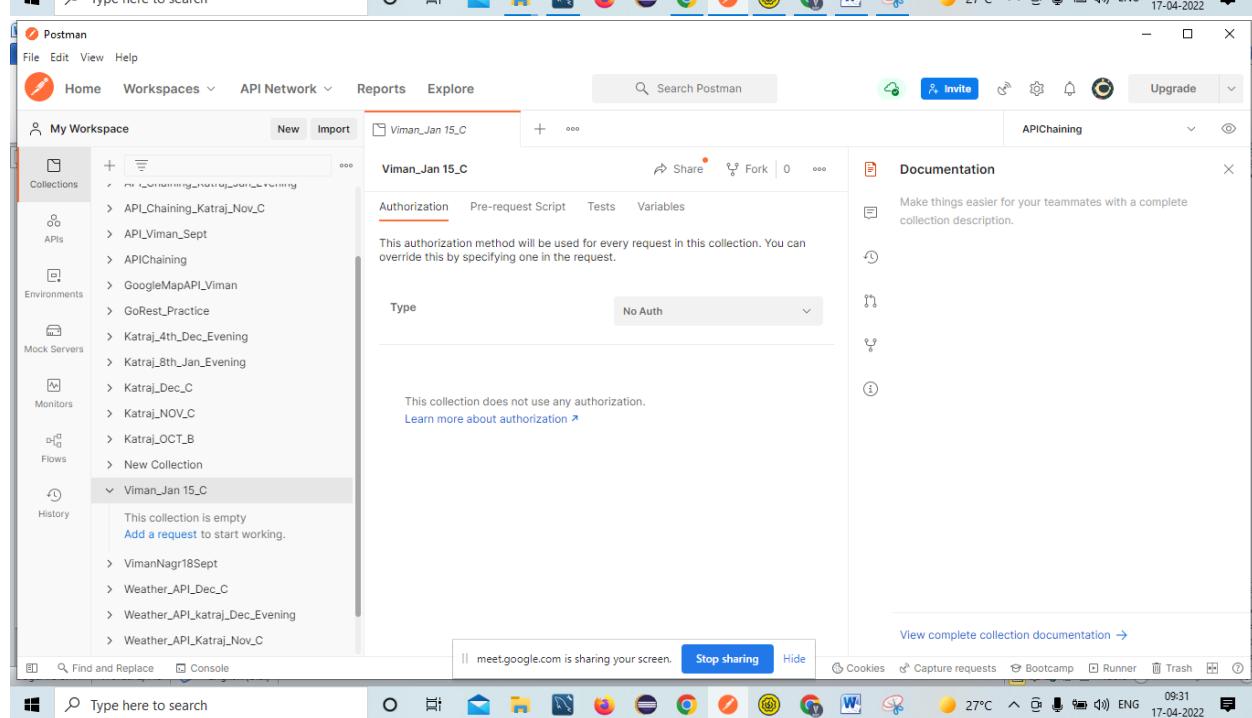
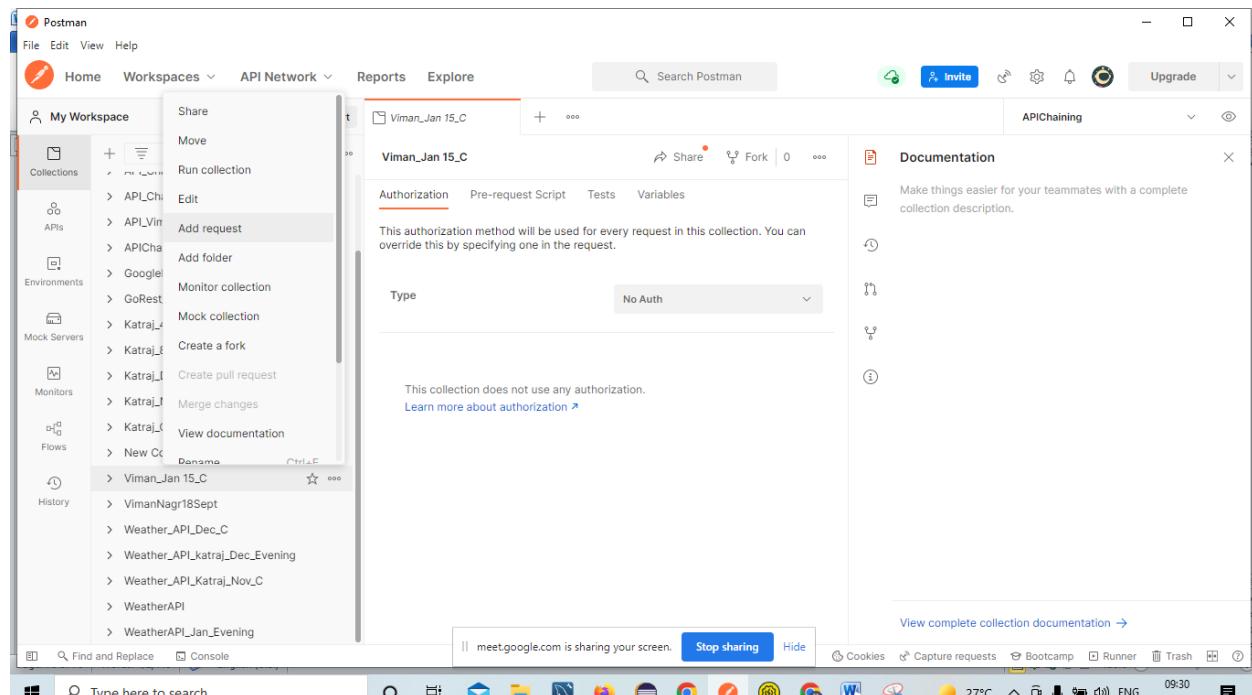


This screenshot shows the same Postman interface, but the collection "Viman_Jan 15_C" is selected in the sidebar. The main area displays the collection's details, including its name, a "Share" button, and a "Fork" button. Below this, there are tabs for "Authorization", "Pre-request Script", "Tests", and "Variables". A note states: "This authorization method will be used for every request in this collection. You can override this by specifying one in the request." It also notes: "Type" and "No Auth". To the right, there is a "Documentation" section with the heading "Documentation" and a sub-section "Make things easier for your teammates with a complete collection description." It includes a "View complete collection documentation →" link. The bottom toolbar and status bar are identical to the first screenshot.

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'File', 'Edit', 'View', 'Help', 'Home', 'Workspaces', 'API Network', 'Reports', and 'Explore'. A search bar says 'Search Postman' with a magnifying glass icon. On the right, there are icons for 'Invite', 'Upgrade', and settings. Below the navigation is a sidebar titled 'My Workspace' with sections for 'Collections', 'APIs', 'Environments', 'Mock Servers', 'Monitors', 'Flows', and 'History'. Under 'Collections', a tree view shows 'GoogleMapAPI_Viman' expanded, containing 'POST AddPlace', 'GET RetrieveData', and 'POST Post_request'. Other collapsed collections include 'GoRest_Practice', 'Katraj_4th_Dec_Evening', 'Katraj_8th_Jan_Evening', 'Katraj_Dec_C', 'Katraj_NOV_C', 'Katraj_OCT_B', 'New Collection', 'Viman_Jan 15_C' (which is highlighted with a blue box), 'VimanNagr18Sept', 'Weather_API_Dec_C', 'Weather_API_Katraj_Dec_Evening', 'Weather_API_Katraj_Nov_C', and 'WeatherAPI'. The main content area shows the details for 'Viman_Jan 15_C'. It has tabs for 'Authorization', 'Pre-request Script', 'Tests', and 'Variables'. The 'Authorization' tab is selected, showing 'Type' set to 'No Auth'. A note says 'This collection does not use any authorization.' and a link 'Learn more about authorization'. To the right, there's a 'Documentation' panel with a note 'Make things easier for your teammates with a complete collection description.' and a 'View complete collection documentation' button.

GET Method/Request:

- GET Method Service-URL/URI=https://reqres.in/api/users?page=2
- Username& Password(Basic Auth)/API Key/Bearer Token-dbjsbfksgfjf
- Time taken for the response=5Secs
- Unit Testing documents



Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

My Workspace

New Import Vim Jan 15_C GET New Request + ... APIChaining

Collections

- APIs
- Environments
- Mock Servers
- Monitors
- Flows
- History

GET_List_Users

GET Enter request URL

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

Response

Enter the URL and click Send to get a response

meet.google.com is sharing your screen. Stop sharing Hide

Find and Replace Console

Stop sharing Hide

Cookies Capture requests Bootcamp Runner Trash

09:31 17-04-2022

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

My Workspace

New Import Vim Jan 15_C GET GET_List_Users + ... APIChaining

Collections

- APIs
- Environments
- Mock Servers
- Monitors
- Flows
- History

Viman_Jan 15_C / GET_List_Users

GET Enter request URL

POST
PUT
PATCH
DELETE
COPY
HEAD
OPTIONS
LINK
UNLINK
PURGE
LOCK
UNLOCK
PROPFIND

Headers (6) Body Pre-request Script Tests Settings Cookies

Enter the URL and click Send to get a response

meet.google.com is sharing your screen. Stop sharing Hide

Find and Replace Console

Stop sharing Hide

Cookies Capture requests Bootcamp Runner Trash

09:31 17-04-2022

File Edit View Help

Home Workspaces API Network Reports Explore Search Postman Invite Upgrade

My Workspace

Collections APIs Environments Mock Servers Monitors Flows History

Viman_Jan 15_C / GET_List_Users

GET https://reqres.in/api/users?page=2

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
<input checked="" type="checkbox"/> page	2	Description	
Key	Value	Description	

Response

Click Send to get a response



File Edit View Help

Home Workspaces API Network Reports Explore Search Postman Invite Upgrade

My Workspace

Collections APIs Environments Mock Servers Monitors Flows History

Viman_Jan 15_C / GET_List_Users

GET https://reqres.in/api/users?page=2

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
<input checked="" type="checkbox"/> page	2	Description	
Key	Value	Description	

Body Cookies Headers (19) Test Results Status: 200 OK Time: 390 ms Size: 1.89 KB Save Response

Pretty Raw Preview Visualize JSON

```

1 "page": 2,
2 "per_page": 6,
3 "total": 12,
4 "total_pages": 2,
5 "data": [
6   {
7     "id": 7,
8     "email": "michael.lawson@reqres.in",
9   }
]

```

File Edit View Help

Home Workspaces API Network Reports Explore Search Postman

My Workspace New Import Vimani_Jan 15_C GET GET_List_Users

APIChaining

Collections APIs Environments Mock Servers Monitors Flows History

Viman_Jan 15_C / GET_List_Users

GET https://reqres.in/api/users?page=2

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Type API Key

Key Value

Body Cookies Headers (19) Test Results

Pretty Raw Preview Visualize JSON

```
1
2
3
4
5
6
7
8
9
```

Status: 200 OK Time: 390 ms Size: 1.89 KB Save Response

Send

APIChaining

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables.

Key Value

Value

Body Cookies Headers (19) Test Results

Pretty Raw Preview Visualize JSON

```
1
2
3
4
5
6
7
8
9
```

Status: 200 OK Time: 390 ms Size: 1.89 KB Save Response

Send

File Edit View Help

Home Workspaces API Network Reports Explore Search Postman

My Workspace New Import Vimani_Jan 15_C GET GET_List_Users

APIChaining

Collections APIs Environments Mock Servers Monitors Flows History

Viman_Jan 15_C / GET_List_Users

GET https://reqres.in/api/users?page=2

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Type Basic Auth

Username Password

Collaborative environment, we recommend using variables. Learn more about variables.

Body Cookies Headers (19) Test Results

Pretty Raw Preview Visualize JSON

```
1
2
3
4
5
6
7
8
9
```

Status: 200 OK Time: 390 ms Size: 1.89 KB Save Response

Send

APIChaining

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables.

Username Password

Show Password

Body Cookies Headers (19) Test Results

Pretty Raw Preview Visualize JSON

```
1
2
3
4
5
6
7
8
9
```

Status: 200 OK Time: 390 ms Size: 1.89 KB Save Response

Send

The screenshot shows the Postman application interface. On the left, there's a sidebar with navigation links like File, Edit, View, Help, Home, Workspaces, API Network, Reports, Explore, and a search bar. Below these are sections for Collections, APIs, Environments, Mock Servers, Monitors, Flows, and History. A 'My Workspace' section is also present.

The main area displays a collection named 'Viman_Jan 15_C'. Under this collection, there is a single item named 'GET GET_List_Users'. The request details show a GET method pointing to 'https://reqres.in/api/users?page=2'. The 'Authorization' tab is selected, showing a 'Bearer Token' type with a placeholder 'Token' and a 'Set as variable' button. The 'Headers' tab is also selected, listing 'Accept-Encoding', 'Connection', 'Accept', and 'Content-Type' with their respective values: 'gzip, deflate, br', 'keep-alive', 'application/json', and 'application/json'. The 'Body' tab is selected, showing a JSON response with a page number of 2 and a total of 12 users. The status bar at the bottom indicates a 200 OK response with a time of 390 ms and a size of 1.89 KB.

Tests in POSTMAN:

- Test is putting validations in your request to make sure API is working as Expected
- Request can have many validations
- Tests are written in JavaScript in Postman
- Test can be added at Request, Collection, Folder etc

File Edit View Help

Home Workspaces API Network Reports Explore Search Postman invite Upgrade

My Workspace

Collections APIs Environments Mock Servers Monitors Flows History

Viman_Jan 15_C / GET_List_Users

GET https://reqres.in/api/users?page=2

Params Authorization Headers (8) Body Pre-request Script Tests Settings

1

Test scripts are written in JavaScript, and are run after the response is received. Learn more about tests scripts

SNIPPETS Get an environment variable Get a global variable Get a variable

Body Cookies Headers (19) Test Results

Pretty Raw Preview Visualize JSON

Status: 200 OK Time: 390 ms Size: 1.89 KB Save Response

Find and Replace Console

File Edit View Help

Home Workspaces API Network Reports Explore Search Postman invite Upgrade

My Workspace

Collections APIs Environments Mock Servers Monitors Flows History

Viman_Jan 15_C / GET_List_Users

GET https://reqres.in/api/users?page=2

Params Authorization Headers (8) Body Pre-request Script Tests Settings

1 pm.test("Expected Status Code is: 200", function () {
2 pm.response.to.have.status(200);
3});

Test scripts are written in JavaScript, and are run after the response is received. Learn more about tests scripts

SNIPPETS Send a request Status code: Code is 200 Response body: Contains string

Body Cookies Headers (19) Test Results

All Passed Skipped Failed

PASS Expected Status Code is: 200

Status: 200 OK Time: 112 ms Size: 1.88 KB Save Response

The screenshot shows the Postman application interface. On the left, the sidebar lists 'My Workspace' with various collections, APIs, environments, mock servers, monitors, flows, and history. In the center, a specific collection named 'Viman_Jan 15_C' is selected. Under this collection, a 'GET' request named 'GET_List_Users' is shown. The request URL is 'https://reqres.in/api/users?page=2'. The 'Tests' tab is active, displaying a JavaScript code snippet:

```

1 pm.test("Expected Status Code is: 200", function () {
2   pm.response.to.have.status(200);
3 });
4 pm.test("Response time is less than 200ms", function () {
5   pm.expect(pm.response.responseTime).to.be.below(200);
6 });

```

The 'Test Results' section shows two green 'PASS' status indicators: 'Expected Status Code is: 200' and 'Response time is less than 200ms'. At the bottom right, the status bar indicates 'Status: 200 OK Time: 39 ms Size: 1.88 KB Save Response'.

Test Cases:

1. Validate REST Response=Pass

This screenshot shows the same Postman interface as the previous one, but with more detailed configurations. The 'Query Params' section for the 'GET_List_Users' request shows a table with a single row for 'page':

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> page	2	

The 'Body' section shows the JSON response from the API:

```

1 {
2   "page": 2,
3   "per_page": 6,
4   "total": 12,
5   "total_pages": 2,
6   "data": [
7     {
8       "id": 7,
9       "email": "michael.lawson@reqres.in",

```

2. Validating data and count of data present in the REST response=Pass

Postman screenshot showing a successful API call to `https://reqres.in/api/users?page=2`. The response body is a JSON object with the following structure:

```

{
  "page": 2,
  "per_page": 6,
  "total": 12,
  "total_pages": 2,
  "data": [
    {
      "id": 7,
      "email": "michael.lawson@reqres.in",
      "first_name": "Michael",
      "last_name": "Lawson",
      "avatar": "https://reqres.in/img/faces/7-image.jpg"
    }
  ]
}

```

3. Validating tagnames/Attributes present in the response → EC=Tagname should contain id,email(first_name,last_name,avatar)=Pass

Postman screenshot showing a successful API call to `https://reqres.in/api/users?page=2`. The response body is a JSON object with the following structure:

```

{
  "page": 2,
  "per_page": 6,
  "total": 12,
  "total_pages": 2,
  "data": [
    {
      "id": 7,
      "email": "michael.lawson@reqres.in",
      "first_name": "Michael",  
      "last_name": "Lawson",  
      "avatar": "https://reqres.in/img/faces/7-image.jpg"
    }
  ]
}

```

4. Validating different status codes present in the response → EC-200-OK=Pass

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists various collections, environments, and histories. The main workspace shows a 'GET List_Users' request. The 'Params' tab contains a 'page' parameter set to 2. The 'Body' tab displays a JSON response object. A tooltip for the response body indicates it's a standard success response. The status bar at the bottom shows 'Status: 200 OK'.

```

{
  "total_pages": 2,
  "data": [
    {
      "id": 7,
      "email": "michael.lawson@reqres.in",
      "first_name": "Michael",
      "last_name": "Lawson",
      "avatar": "https://reqres.in/img/faces/7-image.jpg"
    }
  ]
}
  
```

5. Validating time taken for the response → EC=5Secs=Pass

This screenshot is identical to the previous one, showing the same API request and response. However, a blue oval highlights the 'Time: 63 ms' entry in the status bar at the bottom of the screen.

6. Applying assertions for verifications

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists various collections, environments, and histories. The main workspace displays a collection named 'Viman_Jan 15_C' containing a single GET request for the URL `https://reqres.in/api/users?page=2`. The 'Tests' tab is currently active, showing an empty test script area:

```
1 pm.test("Status code is 200", function () {  
2     pm.response.to.have.status(200);  
3 });
```

Below the script, the 'Test Results' section indicates 'There are no tests for this request'. The status bar at the bottom shows 'Status: 200 OK'.

This screenshot shows the same Postman session as the previous one, but with a test script added to the 'Tests' tab:

```
1 pm.test("Status code is 200", function () {  
2     pm.response.to.have.status(200);  
3 });
```

The 'Test Results' section now shows a single green 'PASS' result with the message 'Status code is 200'. A blue arrow points from the 'Status code is 200' message in the results table back up to the corresponding line in the test script.

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists various collections, environments, and histories. The main workspace displays a collection named 'Viman_Jan 15_C' containing a single test step named 'GET_List_Users'. The request method is 'GET' to the URL 'https://reqres.in/api/users?page=2'. The 'Tests' tab contains the following JavaScript code:

```
1 pm.test("Status code is 200", function () {  
2 | pm.response.to.have.status(200);  
3 });  
4 pm.test("Response time is less than 200ms", function () {  
5 | pm.expect(pm.response.responseTime).to.be.below(200);  
6 });
```

The results section shows one test result: 'Status code is 200' with a green 'PASS' status and a checkmark. The status bar at the bottom indicates a successful POST request with a status of 200 OK, a response time of 33 ms, and a size of 1.88 KB.

This screenshot is identical to the one above, showing the same Postman interface and API call. However, it includes several blue checkmarks overlaid on the status bars and text areas, specifically highlighting the 'Status code is 200' and 'Response time is less than 200ms' sections in the results table and the status bar message.

Screenshot of Postman interface showing a collection named "Viman_Jan 15_C".

The main panel displays a GET request to `https://reqres.in/api/users?page=2`. The "Tests" tab contains the following JavaScript code:

```
3 });
4 pm.test("Response time is less than 200ms", function () {
5   |   pm.expect(pm.response.responseTime).to.be.below(200);
6 });
7 pm.test("Your test name", function () {
8   |   var jsonData = pm.response.json();
9   |   pm.expect(jsonData.value).to.eql(100);
10 });


```

The "Test Results" section shows two successful tests:

- PASS Status code is 200
- PASS Response time is less than 200ms

Details pane on the right shows:

- Test scripts are written in JavaScript, and are run after the response is received.
- SNIPPETS Response body: Contains string
- Response body: JSON value check
- Response body: Is equal to a string

Screenshot of Postman interface showing the same collection "Viman_Jan 15_C".

The main panel displays the same GET request to `https://reqres.in/api/users?page=2`. The "Tests" tab contains the following JavaScript code:

```
3 });
4 pm.test("Response time is less than 200ms", function () {
5   |   pm.expect(pm.response.responseTime).to.be.below(200);
6 });
7 pm.test("Test Expected Output:", function () {
8   |   var jsonData = pm.response.json();
9   |   pm.expect(jsonData.total_pages).to.eql(2);
```

The "Test Results" section shows three successful tests:

- PASS Status code is 200
- PASS Response time is less than 200ms
- PASS Test Expected Output:

Details pane on the right shows:

- Test scripts are written in JavaScript, and are run after the response is received.
- SNIPPETS Response body: Contains string
- Response body: JSON value check
- Response body: Is equal to a string

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** My Workspace, Collections, APIs, Environments, Mock Servers, Monitors, Flows, History.
- Current Collection:** Vimani_Jan 15_C
- Request:** GET /GET_List_Users
- URL:** https://reqres.in/api/users?page=2
- Tests Tab (JavaScript code):**

```

3  });
4 pm.test("Response time is less than 200ms", function () {
5   pm.expect(pm.response.responseTime).to.be.below(200);
6 });
7 pm.test("Test Expected Output:", function () {
8   var jsonData = pm.response.json();
9   pm.expect(jsonData.data[0].first_name).to.eql("Michael");
10 });
    
```
- Test Results:** 3/3
 - PASS Status code is 200
 - PASS Response time is less than 200ms
 - PASS Test Expected Output:
- Bottom Status:** Status: 200 OK, Time: 51 ms, Size: 1.88 KB, Save Response

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** My Workspace, Collections, APIs, Environments, Mock Servers, Monitors, Flows, History.
- Current Collection:** Vimani_Jan 15_C
- Request:** GET /GET_List_Users
- URL:** https://reqres.in/api/users?page=2
- Tests Tab (JavaScript code):**

```

3  });
4 pm.test("Response time is less than 200ms", function () {
5   pm.expect(pm.response.responseTime).to.be.below(200);
6 });
7 pm.test("Test Expected Output:", function () {
8   var jsonData = pm.response.json();
9   pm.expect(jsonData.data[0].email).to.eql("michael.lawson@reqres.in");
10 });
    
```
- Test Results:** 3/3
 - PASS Status code is 200
 - PASS Response time is less than 200ms
 - PASS Test Expected Output: ✓
- Bottom Status:** Status: 200 OK, Time: 57 ms, Size: 1.88 KB, Save Response

7. Validating functionality by passing Test data (Re-Testing)

7.1 Verify by Passing parameter as 1 in GET Method=>EC=Status Code-200-OK & GET Method will show the response=Pass

The screenshot shows the Postman interface with a successful API call. The URL is `https://reqres.in/api/users?page=1`. In the 'Params' tab, there is a checked checkbox for 'page' with a value of '1'. The 'Body' tab shows a JSON response with a page number of 1. A blue arrow points from the 'page' parameter in the Params table to the 'page' field in the JSON response.

```

1
2   "page": 1,
3   "per_page": 6,
4   "total": 12,
5   "total_pages": 2,
6   "data": [
7     {
8       "id": 1,
9       "email": "george.bluth@reqres.in",

```

7.2 Verify by Passing parameter as 2 in GET Method=>EC=Status Code-200-OK & GET Method will show the response=Pass

The screenshot shows the Postman interface with a successful API call. The URL is `https://reqres.in/api/users?page=2`. In the 'Params' tab, there is a checked checkbox for 'page' with a value of '2'. The 'Body' tab shows a JSON response with a page number of 2. A blue arrow points from the 'page' parameter in the Params table to the 'page' field in the JSON response.

```

1
2   "page": 2,
3   "per_page": 6,
4   "total": 12,
5   "total_pages": 2,
6   "data": [
7     {
8       "id": 7,
9       "email": "michael.lawson@reqres.in",

```

7.3 Verify by Passing parameter as 3 in GET Method=>EC=Status Code-200-OK & GET Method will show the response=Pass

The screenshot shows the Postman interface with a collection named "Viman_Jan 15_C". A specific test case named "GET GET_List_Users" is selected. The request method is set to "GET" and the URL is "https://reqres.in/api/users?page=3". In the "Query Params" section, there is a single entry for "page" with a value of "3". The "Body" tab shows the JSON response, which includes a "page" key with a value of 3, a "per_page" key with a value of 6, a "total" key with a value of 12, a "total_pages" key with a value of 2, and a "data" array containing one user object. The user object has an "id" of 1, an "email" of "george.bluth@reqres.in", a "first_name" of "George", a "last_name" of "Bluth", and an "avatar" URL of "https://reqres.in/img/faces/1-image.jpg". The status bar at the bottom indicates a 200 OK status with a time of 212 ms and a size of 1.06 KB.

7.4 Verify by Passing parameter as 10/100/1000 in GET Method=>EC=Status Code-200-OK & GET Method will show the response=Pass

8. Validating Negative Test Cases!

8.1 Verify by Passing parameter as a decimal values in GET Method=>EC=Status Code-400-Bad Request & GET Method will not show the response=Fail-Defect

The screenshot shows the Postman interface with the same collection and test case setup as the previous screenshot. The URL is now "https://reqres.in/api/users?page=1.55". The "Query Params" section still has a "page" entry with a value of "1.55". The "Body" tab shows the JSON response, which is identical to the previous successful request, indicating that the decimal value was accepted despite being invalid. The status bar at the bottom indicates a 200 OK status with a time of 293 ms and a size of 1.84 KB.

8.2 Verify by Passing parameter as Null/Blank Values in GET Method=>EC=Status Code-400-Bad Request & GET Method will not show the response=Fail-Defect

8.3 Verify by Passing parameter as a Character/String Values in GET Method=>EC=Status Code-400-Bad Request & GET Method will not show the response= Fail-Defect

8.4 Verify by passing WRONG URL/URI in GET Method=> EC=Status Code-404-Not Found & GET Method will not show the response=Pass

The screenshot shows the Postman interface with a collection named 'Viman_Jan 15_C'. A specific test case named 'GET_LIST_Users' is selected. The request method is set to 'GET' with the URL 'https://reqres.in/vtctc/api/users?page=2'. In the 'Query Params' section, there is a single entry for 'page' with a value of '2'. The 'Body' tab is selected. The status bar at the bottom indicates a 404 Not Found error with a response size of 819 B. A blue arrow points from the text 'Status: 404 Not found' to the status code in the status bar.

8.5 Verify by Passing wrong Authorization/Bearer Token/API Key in GET Method=>EC=Status Code-401-Un-Authorization & GET Method will not show the response.

8.6 Verify by changing GET Method into another Method=>EC=Status Code-405-Method not Allowed & GET Method will not show the response=Fail-Defect

This screenshot shows the same Postman environment and collection setup as the previous one. However, the request method is now set to 'PURGE' with the same URL 'https://reqres.in/api/users?page=2'. The 'Authorization' tab is selected, indicating that no token or key was provided. The status bar at the bottom shows a 403 Forbidden error with a response size of 603 B. A blue arrow points from the text 'Status: 403 Forbidden' to the status code in the status bar.

POST Method/Request:

- POST Method Service-URL/URI=https://reqres.in/api/register
- Body/Payload-

{

```

        "email": "eve.holt@reqres.in",
        "password": "pistol"
    }

```

- Username& Password(Basic Auth)/API Key/Barrier token-tcaabfj bjf lekf lklfl
- Time taken for the response=60sec
- Unit Testing Documents

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'My Workspace' containing various collections, APIs, environments, and history. The main area shows a collection named 'Viman_Jan 15_C' with a 'Register_User' endpoint selected. The request type is 'POST' to 'https://reqres.in/api/register'. The 'Body' tab is active, showing a JSON payload:

```

1 {
2   ...
3     "email": "eve.holt@reqres.in",
4     "password": "pistol"
5 }

```

Test Cases:

1. Validating the REST Response=Pass

The screenshot shows the same Postman interface after the request has been sent. The response tab is now active, displaying the JSON response received from the server:

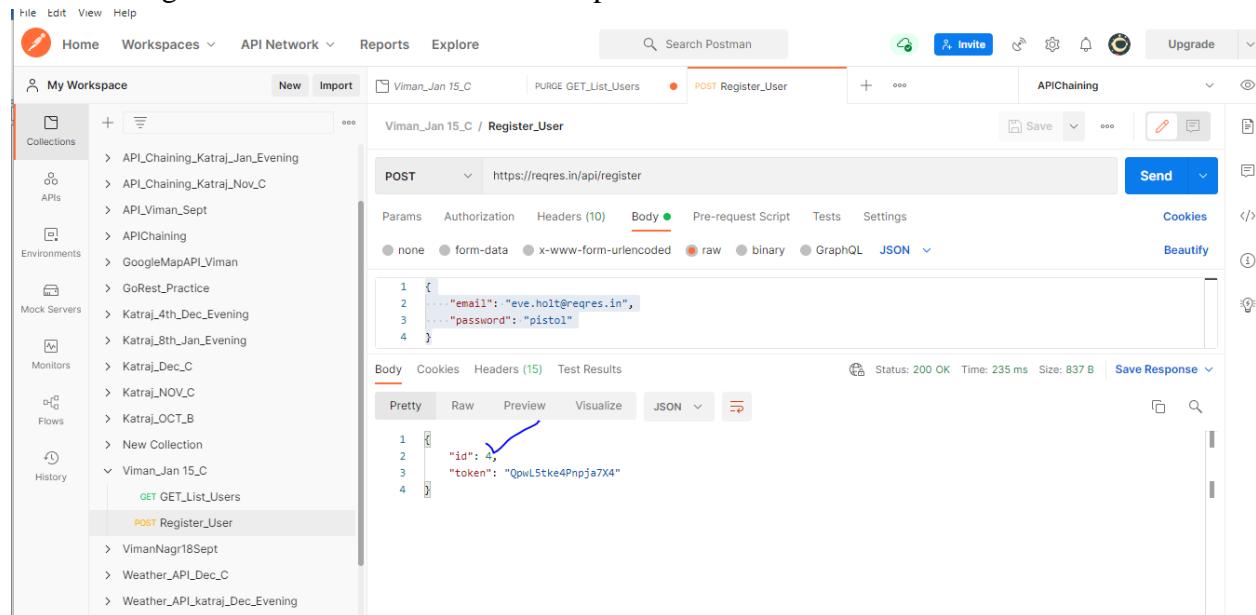
```

1 {
2   ...
3     "id": 4,
4     "token": "QpwL5tke4Pnpja7X4"
5 }

```

The status bar at the bottom indicates a successful response: Status: 200 OK, Time: 235 ms, Size: 837 B.

2. Validating data and count of data in the response=Pass



The screenshot shows the Postman interface with a collection named 'Viman_Jan 15_C'. A POST request is made to `https://reqres.in/api/register`. The response status is 200 OK, and the response body is:

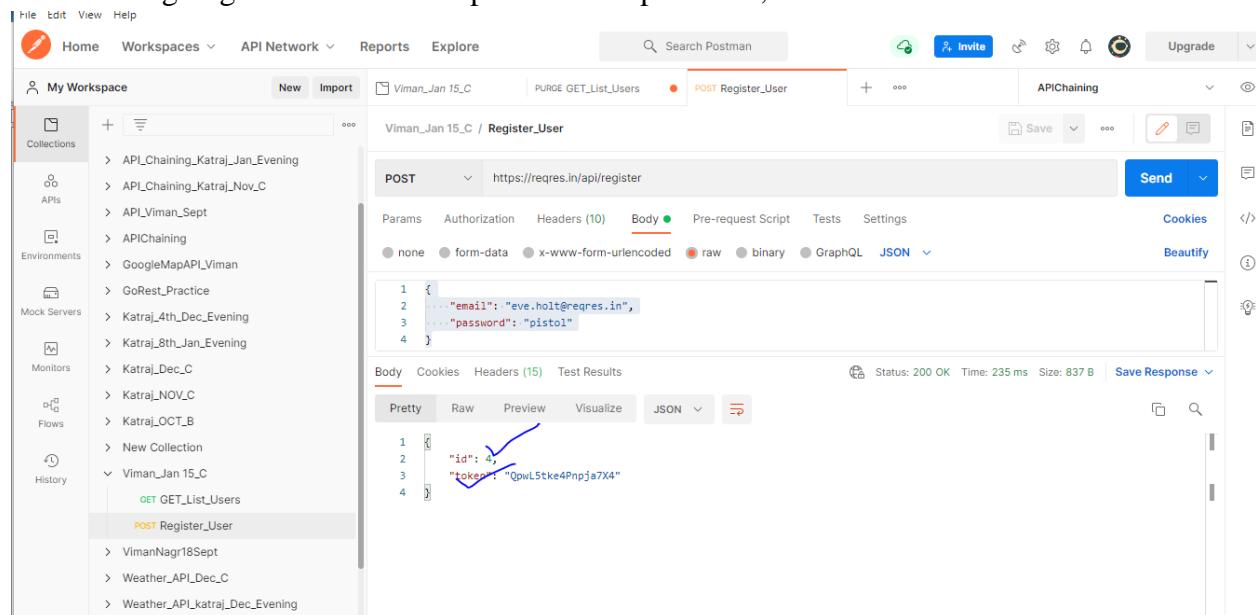
```
1 {
2   ... "email": "eve.holt@reqres.in",
3   ... "password": "pistol"
4 }
```

Below the response, the JSON tab is selected, showing the pretty-printed response:

```
1 [
2   "id": 4,
3   "token": "QpwL5tke4Pnpja7X4"
4 ]
```

A blue arrow points to the 'id' field in the response body.

3. Validating Tag names/Attributes present in response=Id, Token=Pass



The screenshot shows the Postman interface with a collection named 'Viman_Jan 15_C'. A POST request is made to `https://reqres.in/api/register`. The response status is 200 OK, and the response body is:

```
1 {
2   ... "email": "eve.holt@reqres.in",
3   ... "password": "pistol"
4 }
```

Below the response, the JSON tab is selected, showing the pretty-printed response:

```
1 [
2   "id": 4,
3   "token": "QpwL5tke4Pnpja7X4"
4 ]
```

Two blue arrows point to the 'id' and 'token' fields in the response body.

4. Validating different Status Code in the responses=>EC=201-Created=Fail-Defect

The screenshot shows the Postman interface with a collection named 'Viman_Jan 15_C'. A POST request is made to 'https://reqres.in/api/register' with the following body:

```
1 {
2   ... "email": "eve.holt@reqres.in",
3   ... "password": "pistol"
4 }
```

The response status is 200 OK, time taken is 235 ms, and size is 837 B. The response body is:

```
1 {
2   "id": 4,
3   "token": "QpwL5tke4Pnpja7X4"
4 }
```

5. Validating time taken for the response=>EC<5secs=Pass

The screenshot shows the Postman interface with a collection named 'Viman_Jan 15_C'. A POST request is made to 'https://reqres.in/api/register' with the same body as the previous screenshot. The response status is 200 OK, time taken is 235 ms, and size is 837 B. The response body is identical to the previous one.

6. Applying Assertions for Verification (Validation)

The screenshot shows the Postman interface with a successful API call. The 'Tests' tab is selected, containing the following JavaScript code:

```
1 pm.test("POST Method Body is correct", function () {
2     pm.response.to.have.body({"id":4,"token":"QpwL5tke4Pnpja7X4"});
3 });
4 
```

The 'Test Results' section shows a single green 'PASS' status with the message "POST Method Body is correct".

7. Verifying the Functionality by passing Test Data(Re-Testing)

7.1 Verify POST method/ request by passing “george.bluth@reqres.in” with password → EC= For these email id password will be inserted in serve=Pass

The screenshot shows the Postman interface with a successful API call. The 'Body' tab is selected, showing the following JSON payload:

```
1 {
2     "email": "george.bluth@reqres.in",
3     "password": "pistol"
4 }
```

The 'Test Results' section shows a single green 'PASS' status.

7.2 Verify POST method/ request by passing “janet.weaver@reqres.in” with password → EC= For these email id password will be inserted in serve=Pass

Postman screenshot showing a successful POST request to <https://reqres.in/api/register>. The request body is JSON with fields `email: "janet.weaver@reqres.in"` and `password: "pistol"`. The response status is 200 OK.

```

1 {
2   ...
3   "email": "janet.weaver@reqres.in",
4   ...
5   "password": "pistol"
6 }
    
```

Status: 200 OK Time: 235 ms Size: 849 B Save Response

7.3 Verify POST method/ request by passing “emma.wong@reqres.in” with password → EC= For these email id password will be inserted in serve=Pass

Postman screenshot showing a successful POST request to <https://reqres.in/api/register>. The request body is JSON with fields `email: "emma.wong@reqres.in"` and `password: "pistol"`. The response status is 200 OK. Handwritten blue checkmarks are present on the `id` and `token` fields of the response JSON.

```

1 {
2   ...
3   "email": "emma.wong@reqres.in",
4   ...
5   "password": "pistol"
6 }
    
```

Status: 200 OK Time: 231 ms Size: 849 B Save Response

7.4 Verify POST method/ request by passing “charles.morris@reqres.in” with password→ EC= For these email id password will be inserted in serve=Pass

Postman interface showing a successful POST request to <https://reqres.in/api/register>. The request body is set to JSON and contains:

```

1 ...
2 ... "email": "charles.morris@reqres.in",
3 ... "password": "pistol"
4 ...
5

```

The response status is 200 OK, with a response body containing:

```

1 ...
2 ... "id": 5,
3 ... "token": "QpwL5tke4Pnpja7X5"
4 ...

```

7.5 Verify POST method/ request by passing “lindsay.ferguson@reqres.in” with password→ EC= For these email id password will be inserted in serve=Pass

Postman interface showing a successful POST request to <https://reqres.in/api/register>. The request body is set to JSON and contains:

```

1 ...
2 ... "email": "lindsay.ferguson@reqres.in",
3 ... "password": "pistol"
4 ...
5

```

The response status is 200 OK, with a response body containing:

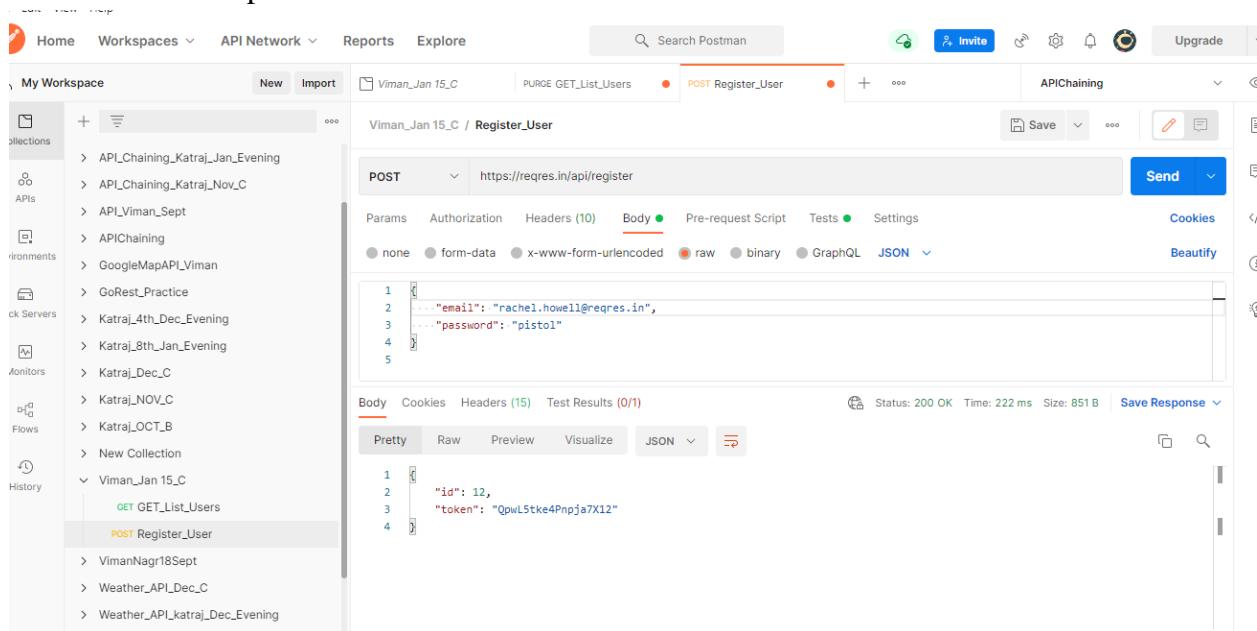
```

1 ...
2 ... "id": 8,
3 ... "token": "QpwL5tke4Pnpja7X6"
4 ...

```

Handwritten blue checkmarks are present on the JSON response body.

7.3 Verify POST method/ request by passing “rachel.howell@reqres.in” with password → EC= 200
 For these email id password will be inserted in serve=Pass



The screenshot shows the Postman interface with the following details:

- Collection:** Vimani_Jan 15_C
- Request:** Register_User (POST)
- URL:** https://reqres.in/api/register
- Body (JSON):**

```

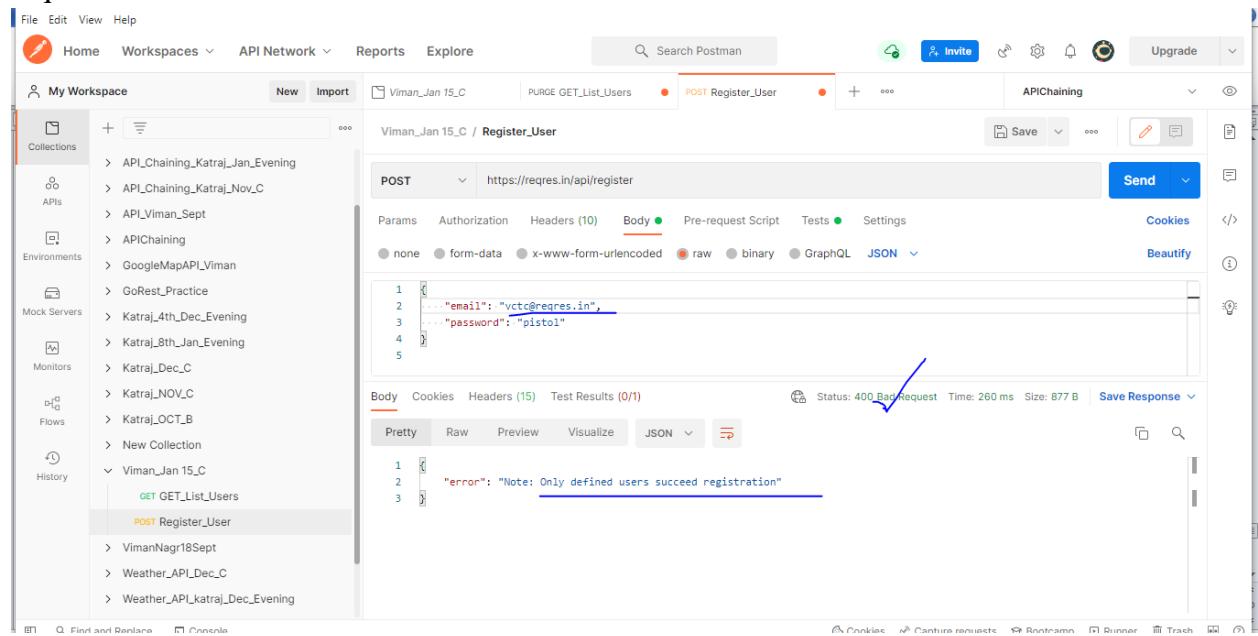
1 {
2   "email": "rachel.howell@reqres.in",
3   "password": "pistol"
4 }
```
- Response Status:** 200 OK
- Response Body:**

```

1 {
2   "id": 12,
3   "token": "QpwL5tke4Pnpja7X12"
4 }
```

8. Validating Negative Test Cases

8.1 Verify POST method/ request by passing invalid email id “vctc@reqres.in” → EC= 400-Bad request=Pass



The screenshot shows the Postman interface with the following details:

- Collection:** Vimani_Jan 15_C
- Request:** Register_User (POST)
- URL:** https://reqres.in/api/register
- Body (JSON):**

```

1 {
2   "email": "vctc@reqres.in",
3   "password": "pistol"
4 }
```
- Response Status:** 400 Bad Request
- Response Body:**

```

1 {
2   "error": "Note: Only defined users succeed registration"
3 }
```

8.2 Verify POST method/ request by passing null/blank email id → EC= 400-Bad request=Pass

The screenshot shows the Postman interface. On the left, the sidebar lists collections, APIs, environments, mock servers, monitors, flows, and history. A collection named 'Viman_Jan 15_C' is expanded, showing a 'GET GET_List_Users' request and a 'POST Register_User' request, which is currently selected. The main panel displays a POST request to 'https://reqres.in/api/register'. The 'Body' tab is selected, showing a JSON payload with a single field 'email' set to an empty string. The 'Test Results' section shows a status of 400 Bad Request with the error message: "error": "Note: Only defined users succeed registration".

8.3 Verify POST method/ request by passing only email id “ rachel.howell@reqres.in” without password → EC= 400-Bad request=Pass

This screenshot is identical to the one above, showing the same setup in Postman. The 'POST Register_User' request is selected. The 'Body' tab shows a JSON payload with a single field 'email' set to 'rachel.howell@reqres.in'. The 'Test Results' section again shows a 400 Bad Request status with the error message: "error": "Missing password".

8.4 Verify POST method/ request by passing invalid email id “ rachel.howell@gamil.com” → EC= 400-Bad request=Pass

The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace
- Request:** POST Register_User
- URL:** https://reqres.in/api/register
- Body (JSON):**

```

1 {
2   "email": "rachel.howell@gamil.com",
3   "password": "pistol"
4 }
5
    
```
- Status:** 400 Bad Request
- Response Body:**

```

1 "error": "Note: Only defined users succeed registration"
2
3
    
```

8.5 Verify POST method/ request by passing valid id (id= 1) with password→ EC= 400-Bad request=Pass

The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace
- Request:** POST Register_User
- URL:** https://reqres.in/api/register
- Body (JSON):**

```

1 {
2   "email": 1,
3   "password": "pistol"
4 }
5
    
```
- Status:** 400 Bad Request
- Response Body:**

```

1 "error": "Note: Only defined users succeed registration"
2
3
    
```

8.6 Verify POST method/ request by passing wrong URL/URI → EC= 404-Not found=Pass

The screenshot shows the Postman interface. On the left, the 'My Workspace' sidebar lists various collections and environments. In the center, a request for 'Viman_Jan 15_C / Register_User' is displayed. The method is set to 'POST' and the URL is 'https://reqres.in/v2/api/register'. The 'Body' tab is selected, showing a JSON payload:

```
1 {
2   "email": 1,
3   "password": "pistol"
4 }
```

The 'Test Results' section shows a status of '404 Not Found' with a checkmark, indicating the test passed.

8.7 Verify POST method/ request by passing wrong authorization/ API Key/ barrier token→ EC= 401-Un- authorization

8.8 Verify by changing POST method/ request into another method/ request → EC= 405-Not method=Fail-Defect

The screenshot shows the Postman interface. The setup is identical to the previous one, but the method is now set to 'PURGE' and the URL is 'https://reqres.in/api/register'. The 'Body' tab is selected, showing the same JSON payload as before:

```
1 {
2   "email": "lindsay.ferguson@reqres.in",
3   "password": "pistol"
4 }
```

The 'Test Results' section shows a status of '403 Forbidden' with a blue underline, indicating the test failed.

Practice API: Assignment 1

Google Maps Add API (POST):

This API Will add new place into Server

Complete URL: <https://rahulshettyacademy.com>/maps/api/place/add/json?key=qaclick123

Base URL/End Point: <https://rahulshettyacademy.com>

Resource: /maps/api/place/add/json

Query Parameters: key =qaclick123

Http Method: POST

Sample Body :

```
{  
  "location": {  
    "lat": -38.383494,  
    "lng": 33.427362  
  },  
  "accuracy": 50,  
  "name": "Frontline house",  
  "phone_number": "(+91) 983 893 3937",  
  "address": "29, side layout, cohen 09",  
  "types": [  
    "shoe park",  
    "shop"  
  ],  
  "website": "http://google.com",  
  "language": "French-IN"
```

}

Sample Response:

{

```
"status": "OK",
"place_id": "928b51f64aed18713b0d164d9be8d67f",
"scope": "APP",
"reference": "736f3c9bec384af62a184a1936d42bb0736f3c9bec384af62a184a1936d42bb0",
"id": "736f3c9bec384af62a184a1936d42bb0"
```

}

The screenshot shows the Postman application interface. The left sidebar displays 'My Workspace' with collections like 'API VCTC', 'API_Viman_Sept', and 'GoogleMapAPI_Viman'. The 'GoogleMapAPI_Viman' collection is expanded, showing a 'POST AddPlace' request. The main panel shows a POST request to <https://rahulshettyacademy.com/maps/api/place/add/json?key=qaclick123>. The request body is JSON:

```
1 {
2   "location": {
3     "lat": -38.383494,
4     "lng": 33.427362
5   },
6   "accuracy": 50,
7   "name": "Fernhill Mantra"
```

The response status is 200 OK, with a response body identical to the one shown above.

Google Maps Delete API (POST):

This API Will delete existing place from Server

Complete URL:

<https://rahulshettyacademy.com/maps/api/place/delete/json?key=qaclick123>

Base URL: <https://rahulshettyacademy.com>

Resource: /maps/api/place/delete/json

Query Parameters: key

Http request : DELETE

Sample Body :

```
{  
    "place_id": "928b51f64aed18713b0d164d9be8d67f"  
}
```

Sample Response

```
{  
    "status": "OK"  
}
```

Google Maps get Place API (GET):

This API Will get existing place details from Server

Complete URL :

http://rahulshettyacademy.com/maps/api/place/get/json?place_id=xxxx&key=qaclick123

Base URL: <https://rahulshettyacademy.com>

Resource: /maps/api/place/get/json

Query Parameters: key, place_id // (place_id value comes from Add place response)

Http request: GET

Note: Key value is hardcoded and it is always qaclick123

Sample Response for the Provided Place_Id

{

 "location":{

 "lat" : -38.383494,

 "lng" : 33.427362

 },

 "accuracy":50,

 "name":"Frontline house",

 "phone_number":"(+91) 983 893 3937",

 "address" : "29, side layout, cohen 09",

 "types": ["shoe park","shop"],

```
"website" : "http://google.com",
```

```
"language" : "French-IN"
```

```
}
```

The screenshot shows the Postman application interface. On the left, the sidebar displays 'My Workspace' with collections like 'API_VIMAN' and 'GoogleMapAPI_Viman'. Under 'GoogleMapAPI_Viman', there is a 'POST AddPlace' request and a 'GET New Request' which is currently selected. The main workspace shows a 'GET' request to 'http://rahulshettyacademy.com/maps/api/place/get/json?place_id=6339c7207d35b30e1c2ff000b5d78132&key=qaclick123'. The 'Params' tab shows 'place_id' with value '6339c7207d35b30e1c2ff000b5d78132' and 'key' with value 'qaclick123'. The 'Body' tab contains the JSON payload:

```
1 {  
2   "location": {  
3     "latitude": "-38.383494",  
4     "longitude": "33.427362"  
5   },  
6   "accuracy": "50",  
7   "name": "Frontline house",  
8   "phone_number": "(+91) 983 893 3937",  
9   "address": "29, side layout, cohen 09",  
10  "types": "shoe park,shop",  
11  "website": "http://google.com",  
12  "language": "French-IN"  
13}
```

The status bar at the bottom indicates 'Status: 200 OK'.

Google Maps Put Place API (PUT):

This API Will update existing place in Server with new values

Complete URL :

http://rahulshettyacademy.com/maps/api/place/get/json?place_id=xxxx&key=qaclick123

Base URL : <https://rahulshettyacademy.com>

Resource: /maps/api/place/update/json

Query Parameters: key
Http Method: PUT -
Note: Key value is hardcoded and it is always qaclick123

Sample Request:

```
{  
  "place_id": "8d2573bdf6ceec0e474c5f388fa917fb",  
  "address": "70 Summer walk, USA",  
  "key": "qaclick123"  
}
```

Sample Response for the Provided Place_Id

```
{  
  
  "location": {  
    "lat": -38.383494,  
    "lng": 33.427362  
  },  
  "accuracy": 50,  
  "name": "Frontline house",  
  "phone_number": "(+91) 983 893 3937",  
  "address": "29, side layout, cohen 09",  
  "types": ["shoe park", "shop"],  
  "website": "http://google.com",  
  "language": "French-IN"  
}
```

Real API:

- GET method Service –**URL/ URI**= api.openweathermap.org/data/2.5/weather?q={city name}&appid={API key}
- **API Key/ Barrier token** - abe5bdb0a984910eb4b5a6a82a0a245d
- **Time take to response** = 5 sec
- Unit Testing documents-

Info Site: <https://openweathermap.org/current>

1. Verify the Weather API for Pune City of Maharashtra=>EC=Weather Information should return and Status Code=200-OK= Pass

The screenshot shows the Postman interface with a successful API call. The URL is `api.openweathermap.org/data/2.5/weather?q=Pune&appid=abe5bdb0a984910eb4b5a6a82a0a245d`. The response status is 200 OK, and the body contains weather information for Pune.

```

4
5   "lat": 18.5196
6 },
7   "weather": [
8     {
9       "id": 800,
10      "main": "Clear",
11      "description": "clear sky",
12      "icon": "01d"
13    }
  
```

2. Verify the Weather API for Mumbai City of Maharashtra=>EC=Weather Information should return and Status Code=200-OK= Pass

The screenshot shows the Postman interface with a successful API call. The URL is `api.openweathermap.org/data/2.5/weather?q=Mumbai&appid=abe5bdb0a984910eb4b5a6a82a0a245d`. The response status is 200 OK, and the body contains weather information for Mumbai.

```

5
6 },
7   "weather": [
8     {
9       "id": 721,
10      "main": "Haze",
11      "description": "haze",
12      "icon": "50d"
13    }
  
```

3. Verify the Weather API for Nasik City of Maharashtra=>EC=Weather Information should return and Status Code=200-OK=Pass

The screenshot shows the Postman application interface. The left sidebar displays 'My Workspace' with various collections, environments, and mock servers. The main workspace shows a collection named 'Weather_API_Real_Viman 15 Jan C / Get_weather_information'. A GET request is selected with the URL `api.openweathermap.org/data/2.5/weather?q=Nasik&appid=abe5bdb0a984910eb4b5a6a82a0a245d`. The 'Query Params' section contains parameters `q` (value: Nasik) and `appid` (value: abe5bdb0a984910eb4b5a6a82a0a245d). The 'Body' tab shows the JSON response, which includes a 'weather' array with one item: {id: 800, main: "Clear", description: "clear sky", icon: "01d"}, and a 'base' field set to "stations". The status bar at the bottom indicates a 200 OK status with a time of 96 ms and a size of 812 B.

4. Verify the Weather API for Delhi City of Delhi=>EC=Weather Information should return and Status Code=200-OK= Pass

This screenshot is identical to the previous one, showing the Postman interface for the same collection. The difference is in the 'Query Params' section where the parameter `q` is now set to 'Delhi'. The rest of the setup and the JSON response in the 'Body' tab remain the same, indicating a successful API call for Delhi.

5. Verify the Weather API for Goa City of Goa=>EC=Weather Information should return and Status Code=200-OK=Pass

The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace > Vimana_Jan 15_C
- Request:** GET /Get_weather_information
- URL:** `api.openweathermap.org/data/2.5/weather?q=Goa&appid=abe5bdb0a984910eb4b5a6a82a0a245d`
- Query Params:**
 - q:** Goa
 - appid:** abe5bdb0a984910eb4b5a6a82a0a245d
- Status:** 200 OK
- Body (Pretty):**

```

5 },
6 "weather": [
7   {
8     "id": 801,
9     "main": "Clouds",
10    "description": "few clouds",
11    "icon": "02d"
12  },
13  "status": "ok"
        
```

6. Verify the Weather API for Deoni Village of Maharashtra=>EC=Weather Information should not return and Status Code=404-Not Found= Pass

The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace > Vimana_Jan 15_C
- Request:** GET /Get_weather_information
- URL:** `api.openweathermap.org/data/2.5/weather?q=Deoni&appid=abe5bdb0a984910eb4b5a6a82a0a245d`
- Query Params:**
 - q:** Deoni
 - appid:** abe5bdb0a984910eb4b5a6a82a0a245d
- Status:** 404 Not Found
- Body (Pretty):**

```

1 {
2   "cod": "404",
3   "message": "city not found"
4 }
        
```

8. Verify the Weather API for Null/Blank city name=>EC=Weather Information should not return and Status Code=400-Bad Request= Pass

The screenshot shows the Postman interface with a collection named "Weather_API_Real_Viman 15 Jan C". A GET request is made to `api.openweathermap.org/data/2.5/weather?q=&appid=abe5bdb0a984910eb4b5a6a82a0a245d`. The "Query Params" section contains two entries: "q" with value "" and "appid" with value "abe5bdb0a984910eb4b5a6a82a0a245d". The response status is 400 Bad Request, with the message "cod": "400", "message": "Nothing to geocode".

9. Verify the Weather API by passing Country name=>EC=Weather Information should not return and Status Code=400-Bad Request= Fail-Defect

The screenshot shows the Postman interface with the same collection. A GET request is made to the same URL. The "Query Params" section now contains "q" with value "India" and "appid" with value "abe5bdb0a984910eb4b5a6a82a0a245d". The response status is 200 OK, indicating a successful request for India.

10. Verify the Weather API by passing Invalid City/State Name=>EC=Weather Information should not return and Status Code=404= Pass

The screenshot shows the Postman interface with a collection named "Weather_API_Real_Viman 15 Jan C". A specific test case, "Get_weather_information", is selected. The request method is GET, and the URL is `api.openweathermap.org/data/2.5/weather?q=Pinepone&appid=abe5bdb0a984910eb4b5a6a82a0a245d`. The response status is 404 Not Found, and the body contains the JSON object:

```
1: {<html>
2: <head>
3: <title>404 Not Found</title>
4: </head>
5: <body>
6: <center>
7: <h1>404 Not Found</h1>
8: </center>
9: </body>
10: </html>
```

11. Verify the Weather API by passing Invalid URL/URI=>EC=Weather Information should not return and Status Code=404= Pass

The screenshot shows the Postman interface with the same collection and test case as the previous screenshot. The URL has been modified to include an invalid component: `api.openweathermap.org/vctc/data/2.5/weather?q=Pinepone&appid=abe5bdb0a984910eb4b5a6a82a0a245d`. The response status is 404 Not Found, and the body contains the same HTML error page as in the previous screenshot.

12. Verify the Weather API by passing Invalid API Key=>EC=Weather Information should not return and Status Code=401=Pass

The screenshot shows the Postman interface with a collection named "My Workspace". A GET request titled "Get_weather_information" is selected. The URL is `api.openweathermap.org/data/2.5/weather?q=Pune&appid=abe5bdb0a984910eb4b5a6a82a0a245d`. The response status is 401 Unauthorized, and the JSON body contains the message: "cod": 401, "message": "Invalid API key. Please see <http://openweathermap.org FAQ#error401> for more info."

The screenshot shows the Postman interface with a collection named "My Workspace". A PURGE request titled "Get_weather_information" is selected. The URL is `api.openweathermap.org/data/2.5/weather?q=Pune&appid=abe5bdb0a984910eb4b5a6a82a0a245d`. The response status is 405 Method Not Allowed, and the JSON body contains the message: "cod": 405, "message": "Internal error"

Variables:(POSTMAN)

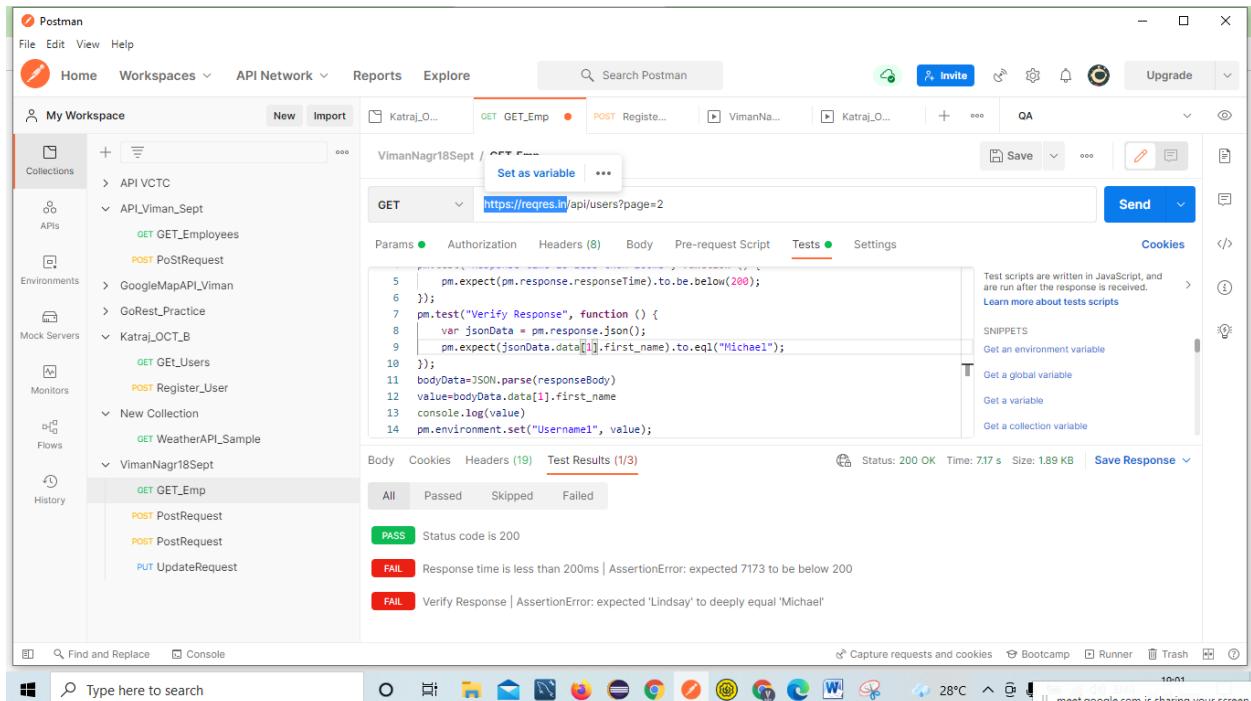
2 Types: Global Variables & Environment Variable (Local Variable)

Variables enable you to store and reuse values in your requests and scripts. By storing a value in a variable, you can reference it throughout your collections, environments, and requests. If you need to update the value, you only have to change it in one place.

Global variables enable you to access data between collections, requests, test scripts, and environments. Global variables are available throughout a [workspace](#).

Environment variables enable you to tailor your processing to different environments, for example local development versus testing or production. Only one environment can be active at a time.

1. Select the **Environment quick look**  in the top right of Postman and select **Edit** next to **Globals**.
2. Add a variable named `my_variable` and give it an initial value of `Hello`.
3. Select **Save** and close the environment dialog.
4. Open a new request tab and enter `https://postman-echo.com/get?var={{my_variable}}` as the URL. Hover over the variable name and you'll see the value.
5. Select **Send** and send the request. In the response, you'll see that Postman sent the variable value to the API.
6. Try changing the value in the Environment quick look and sending the request again.



The screenshot shows the Postman application interface. On the left, the sidebar displays 'My Workspace' with various collections, APIs, environments, and other tools. The main workspace shows a collection named 'VimanNagr18Sept' containing a 'GET GET_Emp' request. The 'Tests' tab for this request contains the following JavaScript code:

```
5 |     pm.expect(pm.response.responseTime).to.be.below(200);
6 | });
7 | pm.test("Verify Response", function () {
8 |     var jsonData = pm.response.json();
9 |     pm.expect(jsonData.data[1].first_name).to.eql("Michael");
10 | });
11 | bodyData=JSON.parse(responseBody)
12 | value=bodyData.data[1].first_name
13 | console.log(value)
14 | pm.environment.set("username1", value);
```

The 'Body' tab shows the response status as 'Status: 200 OK' and the execution time as 'Time: 7.17 s'. There are two test results: a 'PASS' result for 'Status code is 200' and a 'FAIL' result for 'Verify Response | Assertion: expected 'Lindsay' to deeply equal 'Michael''. The bottom status bar indicates 'Capture requests and cookies' and 'meet.google.com is sharing your screen'.

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

My Workspace

GET Get_User_Details

Katraj_4th_Dec_Evening

Set as variable

Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
page	2		

Response

Click Send to get a response



Find and Replace Console

Cookies Capture requests Bootcamp Runner Trash

Type here to search

Windows Taskbar: Type here to search, Mail, File Explorer, Firefox, Chrome, Edge, etc.

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

My Workspace

GET Get_User_Details

Katraj_8th_Jan_Evening

Set as variable

Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
page	2		

Body Cookies Headers (19) Test Results (5/6)

Pretty Raw Preview Visualize JSON

```
49 ],
50 "support": [
51   "url": "https://reqres.in/#support-heading",
52   "text": "To keep ReqRes free, contributions towards server costs are appreciated!"
53 }
54 }
```

Status: 200 OK Time: 243 ms Size: 1.88 KB Save Response

Find and Replace Console

Cookies Capture requests Bootcamp Runner Trash

Type here to search

Windows Taskbar: Type here to search, Mail, File Explorer, Firefox, Chrome, Edge, etc.

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

Katraj_8th_Jan_Evening / Get_User_Details

GET <https://reqres.in/api/users/2>

Params Authorize

Query Params

KEY Value

page

Key

baseUrl
global_base_url
URL
Username
vctc
Username_1

Body Cookies Headers

Pretty Raw Preview Visualize JSON

Status: 200 OK Time: 243 ms Size: 1.88 KB Save Response

Send

Save Tests Settings Cookies

Description

APIChaining

Find and Replace Console

Type here to search

Stop sharing Hide

Cookies Capture requests Bootcamp Runner Trash

19:07 21-03-2022

```
49 ],  
50 "support": {  
51 "url": "https://reqres.in/#support-heading",  
52 "text": "To keep ReqRes free, contributions towards server costs are appreciated!"  
53 }  
54 }
```

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

Katraj_8th_Jan_Evening / Get_User_Details

GET <https://reqres.in/api/users/2>

Params Authorize

Query Params

Name domain

Value https://reqres.in

Scope Active Environment: APIChaining

Set as new variable

Save variable

Body Cookies Headers (19) Test Results (5/6)

Pretty Raw Preview Visualize JSON

Status: 200 OK Time: 243 ms Size: 1.88 KB Save Response

Send

Save Tests Settings Cookies

Description

APIChaining

Find and Replace Console

Type here to search

Stop sharing Hide

Cookies Capture requests Bootcamp Runner Trash

19:07 21-03-2022

```
49 ],  
50 "support": {  
51 "url": "https://reqres.in/#support-heading",  
52 "text": "To keep ReqRes free, contributions towards server costs are appreciated!"  
53 }  
54 }
```

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

My Workspace

New Import WeatherAPI_Jan_Evening GET Get_User_Details

Katraj_8th_Jan_Evening / Get_User_Details

DEV

Save Send

Unresolved Variable

Make sure the variable is defined and enabled in the active environment, collection or globals.

Query Params

KEY page

Add new variable

Tests Settings Cookies

DESCRIPTION Bulk Edit

Body Cookies Headers (19) Test Results (5/6)

Pretty Raw Preview Visualize JSON

Status: 200 OK Time: 243 ms Size: 1.88 KB Save Response

meet.google.com is sharing your screen. Stop sharing Hide

Find and Replace Console

Type here to search

Windows Taskbar: Type here to search, Start, Task View, Help, File Explorer, Mail, Photos, Firefox, Chrome, Edge, Internet Explorer, File, Task Manager, 23°C, ENG, 19:09, 21-03-2022

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

My Workspace

New Import WeatherAPI_Jan_Evening GET Get_User_Details

Katraj_8th_Jan_Evening / Get_User_Details

DEV

Save Send

Set as new variable

Name global_domain

Value https://reqres.in

Scope Global

Set variable

Tests Settings Cookies

DESCRIPTION Bulk Edit

Body Cookies Headers (19) Test Results (5/6)

Pretty Raw Preview Visualize JSON

Status: 200 OK Time: 243 ms Size: 1.88 KB Save Response

meet.google.com is sharing your screen. Stop sharing Hide

Find and Replace Console

Type here to search

Windows Taskbar: Type here to search, Start, Task View, Help, File Explorer, Mail, Photos, Firefox, Chrome, Edge, Internet Explorer, File, Task Manager, 23°C, ENG, 19:09, 21-03-2022

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

My Workspace

New Import WeatherAPI_Jan_Evening GET Get_User_Details

Katraj_8th_Jan_Evening / Get_User_Details

DEV

Save Send

GET {{global_domain}}/api/users?page=2

Params Authorization Headers Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
page	2		

Body Cookies Headers (19) Test Results (5/6)

Pretty Raw Preview Visualize JSON

```
49 ],
50 "support": [
51   "url": "https://reqres.in/#support-heading",
52   "text": "To keep ReqRes free, contributions towards server costs are appreciated!"
53 }
54 }
```

Status: 200 OK Time: 243 ms Size: 1.88 KB Save Response

Find and Replace Console

Type here to search

Stop sharing Hide

Cookies Capture requests Bootcamp Runner Trash

23°C ENG 19:10 21-03-2022

This screenshot shows the Postman application interface. On the left, there's a sidebar with 'My Workspace' containing collections like 'API VCTC', 'API_Chaining_DEC_EVE', etc. The main area displays a 'GET' request to '{{global_domain}}/api/users?page=2'. The 'Query Params' section has a row for 'page' with value '2'. The 'Test Results' section shows a partial JSON response with a 'support' object containing a URL and a message about server costs.

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

My Workspace

New Import WeatherAPI_Jan_Evening GET Get_User_Details

Katraj_8th_Jan_Evening / Get_User_Details

APIChaining

Environment quick look

Save Send

GET {{domain}}/api/users?page=2

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
page	2		

Body Cookies Headers (19) Test Results (4/6)

Pretty Raw Preview Visualize JSON

```
49 ],
50 "support": [
51   "url": "https://reqres.in/#support-heading",
52   "text": "To keep ReqRes free, contributions towards server costs are appreciated!"
53 }
54 }
```

Status: 200 OK Time: 1008 ms Size: 1.88 KB Save Response

Find and Replace Console

Type here to search

Stop sharing Hide

Cookies Capture requests Bootcamp Runner Trash

23°C ENG 19:11 21-03-2022

This screenshot shows the Postman application interface. It's similar to the first one but with a different environment selected, labeled 'APIChaining' at the top right. The request URL is now '{{domain}}/api/users?page=2'. The 'Test Results' section shows a partial JSON response with a 'support' object containing a URL and a message about server costs.

The screenshot shows the Postman application window. On the left, the sidebar includes sections for Collections, APIs, Environments, Mock Servers, Monitors, Flows, and History. The main workspace displays a collection named 'Katraj_8th_Jan_Evening' with a sub-collection 'Get_User_Details'. A specific GET request for 'Get_User_Details' is selected, showing its configuration. The 'Params' tab is active, with a 'page' parameter set to '2'. The 'Body' tab shows a JSON response with a 'support' key. To the right of the request details, there are two tables: 'APIChaining' and 'Globals', both listing variables with their initial and current values.

How to Add Environments?

The screenshot shows the Postman application window with the 'Environments' section highlighted in the sidebar. It lists environments: 'API_Chaining_Katraj_Nov_C', 'APIChaining', 'DEV', 'New Environment', and 'QA'. The 'New Environment' option is currently selected. In the main workspace, a GET request for 'Get_User_Details' is shown with a 'page' parameter set to '2'. The 'Body' tab displays a JSON response. Below the request, a status bar indicates 'Status: 200 OK Time: 1008 ms Size: 1.88 KB'. The bottom of the screen shows the Windows taskbar with various pinned icons.

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

VimanNagr18Sept / GET_Emp

GET `(url_new)/api/users?page=2`

Params Authorization Headers

```
pm.expect(pm.response.status).to.be.below(200);  
var jsonData = pm.response.json();  
pm.expect(jsonData.data[1].first_name).to.eql("Michael");  
bodyData=JSON.parse(responseBody)  
value=bodyData.data[1].first_name  
console.log(value)  
pm.environment.set("Username1", value);
```

INITIAL https://reqres.in

Tests QA

Send

Cookies

Test scripts are written in JavaScript, and are run after the response is received. Learn more about tests scripts

SNIPPETS

Get an environment variable

Get a global variable

Get a variable

Get a collection variable

Body Cookies Headers (19) Test Results (1/3)

Status: 200 OK Time: 7.17 s Size: 1.89 KB Save Response

All Passed Skipped Failed

PASS Status code is 200

FAIL Response time is less than 200ms | Assertion: expected 7173 to be below 200

FAIL Verify Response | Assertion: expected 'Lindsay' to deeply equal 'Michael'

Find and Replace Console

Capture requests and cookies Bootcamp Runner Trash

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

VimanNagr18Sept / GET_Emp

GET `(url_new)/api/users?page=2`

Params Authorization Headers

```
pm.expect(pm.response.status).to.be.below(200);  
var jsonData = pm.response.json();  
pm.expect(jsonData.data[1].first_name).to.eql("Michael");  
bodyData=JSON.parse(responseBody)  
value=bodyData.data[1].first_name  
console.log(value)  
pm.environment.set("Username1", value);
```

INITIAL https://reqres.in

QA

VARIABLE	INITIAL VALUE	CURRENT VALUE
url	https://reqres.in	
Username1	Lindsay	
URL123	http://dummy.restapitexample.com	http://dummy.restapitexample.com
URL123	http://dummy.restapitexample.com	http://dummy.restapitexample.com
url_new	https://reqres.in	https://reqres.in

Globals Add

No global variables

Global variables are a set of variables that are always available in a workspace.

④ Use variables to reuse values and protect sensitive data

Store sensitive data in variable type secret to keep its values masked on the screen. Learn more about variable type

Work with the current value of a variable to prevent sharing sensitive values with your team. Learn more about variable values

Find and Replace Console

Capture requests and cookies Bootcamp Runner Trash

The screenshot shows the Postman application interface with two requests:

- Request 1:** GET /api/users?page=2
 - Params: baseUrl
 - Body (JSON):


```
5 pm.expect(response);
6 });
7 pm.test("Ve
8 var js...
9 pm.expe
10 );
11 bodyData=JSON
12 value=bodyData.data[1].first_name
13 console.log(value)
14 pm.environment.set("Username1", value);
```
 - Tests (JavaScript):


```
5 pm.expect(response);
6 });
7 pm.test("Ve
8 var js...
9 pm.expe
10 );
11 bodyData=JSON
12 value=bodyData.data[1].first_name
13 console.log(value)
14 pm.environment.set("Username1", value);
```
- Request 2:** GET /api/users/1
 - Params: baseUrl
 - Body (JSON):


```
5 pm.expect(response);
6 });
7 pm.test("Ve
8 var js...
9 pm.expe
10 );
11 bodyData=JSON
12 value=bodyData.data[1].first_name
13 console.log(value)
14 pm.environment.set("Username1", value);
```
 - Tests (JavaScript):


```
5 pm.expect(response);
6 });
7 pm.test("Ve
8 var js...
9 pm.expe
10 );
11 bodyData=JSON
12 value=bodyData.data[1].first_name
13 console.log(value)
14 pm.environment.set("Username1", value);
```

How to Access response from one method into another Method?

Answer: With the help of chaining API Concept (Environment Variables)

How to chain API Requests?

Get data from response of one API and refer it in Another API

Step1: Add requests in POSTMAN

Create new Collection and Add one GET Method: List Users-Gives list of users in the response

URL/URI: <https://reqres.in/api/users?page=2>

Save and Send

Next Add another Request-PUT Request->>Update USER

URL/URI: <https://reqres.in/api/users/2>

Body/Payload: Give the below details in body-> raw-Select JSON from dropdown

```
{  
    "name": "morpheus",  
    "job": "zion resident"  
}
```

Save and Send

Step 2: Use environment variables to Parameterize values to be referred

Fetch users from request 1 and provide it in request 2

Go to new-Add Environment---APIChaining

Variable	Initial Value	Current Value
Username	xyz	xyz

The screenshot shows the Postman interface with the following details:

- Header Bar:** File, Edit, View, Help, Home, Workspaces, API Network, Reports, Explore, Search Postman, Invite, Upgrade.
- Left Sidebar:** My Workspace, Collections, APIs, Environments, Mock Servers, Monitors, Flows, History.
- Request Panel:** APIChaining / UpdateRequest, PUT, https://reqres.in/api/users/2. Body tab selected, showing JSON content:

```
1 {
2   ...
3     "name": "{{Username}}",
4     ...
5       "job": "zion resident"
6 }
```
- Response Panel:** Status: 200 OK, Time: 456 ms, Size: 938 B, Save Response.
- Bottom:** Find and Replace, Console, Capture requests and cookies, Bootcamp, Runner, Trash.

The screenshot shows the Postman interface with the 'Create New' dialog open, displaying various building blocks:

- Building Blocks:**
 - HTTP Request (GET): Create a basic HTTP request.
 - WebSocket Request (BETA): Test and debug your WebSocket connections.
 - Collection: Save your requests in a collection for reuse and sharing.
 - Environment: Save values you frequently use in an environment.
 - gRPC Request (BETA): Test and debug your gRPC request.
 - Workspace: Create a workspace to build independently or in collaboration.
- Advanced:**
 - API Documentation: Create and publish beautiful documentation for your APIs.
 - Mock Server: Create a mock server for your in-development APIs.
 - Flows (BETA): Test real-world workflows by connecting series of requests logically.
 - Monitor: Schedule automated tests and check performance of your APIs.
 - API: Manage all aspects of API design, development, and testing.

At the bottom right of the dialog, there is a link: "Learn more on Postman Docs".

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections for Collections, APIs, Environments, Mock Servers, Monitors, Flows, and History. The main area displays an environment named 'APIChaining' with a table for variables. A variable named 'Username' is defined with a type of 'default' and an initial value of 'xyz', which is also shown as the current value. Below the table, there's a note about using variables to reuse values and protect sensitive data. The bottom of the screen shows the Windows taskbar with various pinned icons and system status.

Go to 2nd Method (Put) body & Give Parameter name as Username

{

```
"name": "{{Username}}",
"job": "zion resident"
}
```

The screenshot shows the Postman application interface. On the left, the sidebar displays 'My Workspace' with various collections like 'API VCTC', 'APL_Viman_Sept', 'APIChaining', and 'Katra_OCT_B'. The 'APIChaining' collection is expanded, showing a 'PUT UpdateRequest' endpoint. The main workspace shows a 'PUT' request to 'https://reqres.in/api/users/2'. The 'Body' tab is selected, showing the following JSON payload:

```

1 {
2   ...
3     "name": "{{Username}}",
4     "job": "zion resident"
5 }

```

Below the request, the response status is shown as 'Status: 200 OK Time: 456 ms Size: 938 B'. The response body is also displayed in JSON format:

```

1 {
2   ...
3     "name": "morpheus",
4     "job": "zion resident",
5     "updatedAt": "2022-01-25T13:55:12.203Z"
6 }

```

Send request and Make Sure you are using Mentioned Environment above

Step 3: Add Script to fetch value from response from 1st API

Go to JSON Path Finder and get the JSON Path:

The screenshot shows a browser window with multiple tabs open. The active tab is 'JSON Path Finder' at jsonpathfinder.com. The page displays a JSON response with several users. A search bar at the top contains the path expression 'x.data[0].first_name'. The results pane shows the path 'total_pages: 2' and a list of users under 'data':

- 0: id: 7, email: michael.lawson@reqres.in, first_name: Michael, last_name: Lawson, avatar: https://reqres.in/img/faces/7-image.jpg
- 1: id: 8, email: lindsay.ferguson@reqres.in, first_name: Lindsay, last_name: Ferguson, avatar: https://reqres.in/img/faces/8-image.jpg
- 2: id: 9, email: tobias.funke@reqres.in, first_name: Tobias, last_name: Funke, avatar: https://reqres.in/img/faces/9-image.jpg
- 3: id: 10, email: byron.fields@reqres.in, first_name: Byron, last_name: Fields,

Go to Tests=>>>

```

bodyData=JSON.parse(responseBody)
value=bodyData.data[0].first_name
console.log(value)
pm.environment.set("Username1", value);

```

Note: The value is nothing but the value which we are getting and we will be using in 2nd request.

`pm.environment.set("Username2", value);` → This statement is setting Environment Variable and the value of this environment variable is used in second method(PUT)

VARIABLE	INITIAL VALUE	CURRENT VALUE
url		https://reqres.in
Username1		Byron
URL123	http://dummy.restapiexample.com	http://dummy.restapiexample.com
URL123		http://dummy.restapiexample.com
url_new	https://reqres.in	https://reqres.in
url_new		https://reqres.in

VARIABLE	INITIAL VALUE	CURRENT VALUE
baseURL	https://reqres.in	https://reqres.in

Step 4: Value coming from response of 1st method will be fetched in 2nd method.

Step 5: Run the second request and Validate now go to Environment and check if the value is correct

HTTP	HTTPS
<ul style="list-style-type: none"> • HTTP stands for ‘HyperText Transfer Protocol’. 	<ul style="list-style-type: none"> • HTTPS stands for ‘HyperText Transfer Protocol Secure’.
<ul style="list-style-type: none"> • HTTP works at the application layer. 	<ul style="list-style-type: none"> • HTTPS works at the transport layer.
<ul style="list-style-type: none"> • The default port number is 80, for communication. 	<ul style="list-style-type: none"> • Here, the default port number is 443.
<ul style="list-style-type: none"> • No encryption is present in HTTP websites. 	<ul style="list-style-type: none"> • Both encryption and decryption exist on HTTPS websites.

How to Run Collection?

Postman

File Edit View Help

Home Workspaces API Network Reports Explore Search Postman

My Workspace

Collections + Run collection

APIs > API_Viman

Environments > Vimani_Jan 15_C

Mock Servers > Vimani_Jan 15_C

Monitors > Vimani_Jan 15_C

Flows > Vimani_Jan 15_C

History > Vimani_Nagr18Sept

GET GET_List_Users

POST Register_User

Viman_Jan 15_C / Register_User

POST https://reqres.in/api/register

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

Response

Click Send to get a response

Send

Save

Invite

Share

Move

Edit

Add request

Add folder

Monitor collection

Mock collection

Create a fork

Create pull request

Merge changes

New Collection

View documentation

Send

Find and Replace Console

meet.google.com is sharing your screen. Stop sharing Hide

Cookies Capture requests Bootcamp Runner Trash

DEV

Upgrade

API Network Reports Explore

Search Postman

File Edit View Help

Home Workspaces API Network Reports Explore Search Postman

My Workspace

New Import

GET GET_List_Users POST Register_User

RUN ORDER Deselect All | Select All | Reset

Iterations 1

Delay 0 ms

Data Select File

Save responses

Keep variable values

Run collection without using stored cookies

Save cookies after collection run

Run Viman_Jan 15_C

Run Viman_Jan 15_C

Find and Replace Console

meet.google.com is sharing your screen. Stop sharing Hide

Cookies Capture requests Bootcamp Runner Trash

DEV

Upgrade

Type here to search

File Edit View Help

Home Workspaces API Network Reports Explore Search Postman

My Workspace

New Import

GET GET_List_Users POST Register_User

RUN ORDER Deselect All | Select All | Reset

Iterations 1

Delay 0 ms

Data Select File

Save responses

Keep variable values

Run collection without using stored cookies

Save cookies after collection run

Run Viman_Jan 15_C

Run Viman_Jan 15_C

Type here to search

meet.google.com is sharing your screen. Stop sharing Hide

Cookies Capture requests Bootcamp Runner Trash

DEV

Upgrade

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

My Workspace

New Import

Viman_Jan 15_C DEV, just now

All Tests Passed (3) Failed (2)

Iteration 1

GET GET_List_Users https://reqres.in/api/users?page=2 / GET_List_Users 200 OK 1138 ms 1.93 KB

Pass Status code is 200

Fail Response time is less than 200ms | Assertion: expected 1138 to be below 200

Pass Test Expected Output

Pass Content-Type is present

POST Register_User https://reqres.in/api/register / Register_User 200 OK 230 ms 847 B

Fail POST Method Body is correct | Assertion: expected response body json to equal { id: 4, token: 'QpwL5tke4Pnpja7X4' } but got { id: 8, to...

View Summary Run Again New Export Results

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

My Workspace

New Import

Viman_Jan 15_C DEV, just now

All Tests Passed (3) Failed (2)

Iteration 1

GET GET_List_Users https://reqres.in/api/users?page=2 / GET_List_Users 200 OK 1138 ms 1.93 KB

Pass Status code is 200

Pass Test Expected Output

Pass Content-Type is present

POST Register_User https://reqres.in/api/register / Register_User 200 OK 230 ms 847 B

Fail POST Method Body is correct | Assertion: expected response body json to equal { id: 4, token: 'QpwL5tke4Pnpja7X4' } but got { id: 8, to...

View Summary Run Again New Export Results

The screenshot shows two instances of the Postman application window side-by-side. Both windows have the following configuration:

- File**: Home, Workspaces, API Network, Reports, Explore.
- Search Bar**: Search Postman.
- Toolbar Buttons**: Invite, Settings, Notifications, Upgrade.
- Left Sidebar**: My Workspace, Collections, APIs, Environments, Mock Servers, Monitors, Flows, History.
- Current Collection**: Vimam_Jan 15_C (DEV, just now).
- Test Status**: All Tests (Passed 3), Failed (2).
- Iteration 1** (Failed Tests):
 - GET GET_List_Users**: Response time is less than 200ms | Assertion: expected 1138 to be below 200.
 - POST Register_User**: POST Method Body is correct | Assertion: expected response body json to equal { id: 4, token: 'QpwL5tke4Pnpja7X4' } but got { id: 8, to...
- Bottom Buttons**: View Summary, Run Again, New, Export Results.

The second window below it shows the "RUN SUMMARY" view for the same collection, displaying the results of the failed tests in a detailed table format.

Method	Endpoint	Status	Count
GET	GET_List_Users	Pass	3 1
POST	Register_User	Fail	0 1

Bottom Status Bar: meet.google.com is sharing your screen, Stop sharing, Hide, Cookies, Capture requests, Bootcamp, Runner, Trash, 28°C, ENG, 09:34, 19-04-2022.

How to Skip any Request?

File Edit View Help

Home Workspaces API Network Reports Explore Search Postman invite Upgrade

My Workspace New Import GET GET_List_Users Runner + ... DEV

Collections APIs Environments Mock Servers Monitors Flows History

APIs: API_Viman_Sept, APIChaining, GoogleMapAPI_Viman, GoRest_Practice, Katraj_4th_Dec_Evening, Katraj_8th_Jan_Evening, Katraj_Dec_C, Katraj_NOV_C, Katraj_OCT_B, New Collection, Vimani_Jan 15_C

Environments: Vimani_Jan 15_C

Mock Servers: VimaniNagr18Sept, Weather_API_Dec_C, Weather_API_katraj_Dec_Evening, Weather_API_Katraj_Nov_C

Monitors: Vimani_Jan 15_C

Flows: Vimani_Jan 15_C

History: VimaniNagr18Sept, Weather_API_Dec_C, Weather_API_katraj_Dec_Evening, Weather_API_Katraj_Nov_C

RUN ORDER: Deselect All Select All Reset

Iterations: 1

Delay: 0 ms

Data: Select File

Save responses:

Keep variable values:

Run collection without using stored cookies:

Save cookies after collection run:

Run Vimani_Jan 15_C

Postman File Edit View Help

Home Workspaces API Network Reports Explore Search Postman invite Upgrade

My Workspace New Import GET GET_List_Users Vimani_Jan 15_C + ... DEV

Collections APIs Environments Mock Servers Monitors Flows History

APIs: API_Viman_Sept, APIChaining, GoogleMapAPI_Viman, GoRest_Practice, Katraj_4th_Dec_Evening, Katraj_8th_Jan_Evening, Katraj_Dec_C, Katraj_NOV_C, Katraj_OCT_B, New Collection, Vimani_Jan 15_C

Environments: Vimani_Jan 15_C

Mock Servers: VimaniNagr18Sept, Weather_API_Dec_C, Weather_API_katraj_Dec_Evening, Weather_API_Katraj_Nov_C

Monitors: Vimani_Jan 15_C

Flows: Vimani_Jan 15_C

History: VimaniNagr18Sept, Weather_API_Dec_C, Weather_API_katraj_Dec_Evening, Weather_API_Katraj_Nov_C

All Tests Passed (6) Failed (2)

Iteration 1

GET GET_List_Users https://reqres.in/api/users?page=2 / GET_List_Users 200 OK 211 ms 1.93 KB

	Pass	Fail	Content-Type	Status
Pass	Status code is 200			
Fail		Response time is less than 200ms AssertionError: expected 211 to be below 200		
Pass	Test Expected Outpu:			
Pass	Content-Type is present			
Fail		Response time is less than 200ms AssertionError: expected 211 to be below 200		
Pass	Status code is 200			
Pass	Content-Type is present			
Pass	Test Expected Outpu:			

Find and Replace Console Stop sharing Hide Cookies Capture requests Bootcamp Runner Trash

Type here to search

meet.google.com is sharing your screen. Stop sharing Hide

Cookies Capture requests Bootcamp Runner Trash

28°C ENG 09:36 19-04-2022

AUTHENTICATION

VERSUS

AUTHORIZATION

AUTHENTICATION

Process of confirming the truth of an attribute of a single piece of data claimed true by an entity

Checks a person's details to identify him

Verifies user's credentials

Occurs before authorization

Ex: A student can authenticate himself before accessing the Learning Management System of a University

AUTHORIZATION

Process of specifying access rights/ privileges to resources related to information security

Checks a user's privileges to access resources

Validates user's permissions

Occurs after authentication

Ex: He can access lecture slides and other learning material of the courses based on the permissions given to him

API vs Web Service

1. All web service are APIs but all APIs are not web service
2. Web services might not perform all the operations that an API would perform
3. A web service always need a network for its operation whereas an API doesn't need a network to perform.
4. Web service uses only 3 types of style SOAP, REST and XML-RPC for communication, whereas API can be exposed in number of ways which include: COM Objects, DLL files, JAR files, XML over HTTP, JSON over HTTP etc

What is idempotent?

Idempotency means that multiple identical requests will have the same outcome. So it does not matter if a request is sent once or multiple times. The following HTTP methods are idempotent: **GET, HEAD, OPTIONS, TRACE, PUT and DELETE**

PUT	Patch
PUT is a method of modifying resource where the client sends data that updates the entire resource .	PATCH is a method of modifying resources where the client sends partial data that is to be updated without modifying the entire data.
HTTP PUT is said to be idempotent, So if you send retry a request multiple times, that should be equivalent to a single request modification	HTTP PATCH is basically said to be non-idempotent. So if you retry the request N times, you will end up having N resources with N different URIs created on the server.
It has High Bandwidth	Since Only data that need to be modified if send in the request body as a payload , It has

	Low Bandwidth
--	---------------

Questions-

1. How much experience you have in web service testing/API?
2. What is difference between SOAP & REST service?
3. How you have tested web service/API ? Tell the Process?
4. What are different status code & meaning?
5. What are different type of method as GET, POST, PUT, PATCH, DELETE
6. What is procedure for testing web services in POSTMAN Tool.
7. What is difference between URI & URL
8. What is difference between Authorization& authentication
9. What is RestAssured?-

Ans - API service then I will use RestAssured library for doing Automation.

1. What are the Problems you facing in API Testing?
 - A. Understanding the functionality of API
 - B. Status Code is not Standard (becoz developers are putting random status codes)
 - C. Expected result should know as a Tester
1. What is Collection? How to run Collection?
2. In Collection if one requests you want to skip .So how will you do it?
3. What is difference between WebService& API?
4. How to Access response from one method into another Method?
5. Difference between HTTP and HTTPS?
6. What is Collection? How to run Collection?
7. What is JSON?
8. What is JSON Object?

Websites:

<https://reqres.in/>

<https://gorest.co.in/>

-----**Thank You & ALL THE BEST for Preparations**-----