

## **Agenda:**

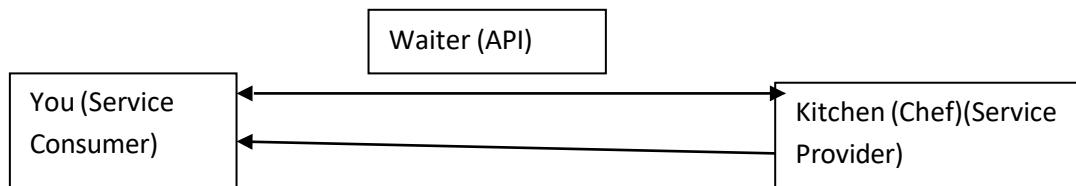
- What is Web Service?
- Basic Concept behind web service
- Why we use it?
- Why to test it?

## **Project Application:**

- Front End(UI)
- Backend(Database)
- **Service(API's)**

## **Web Service:**

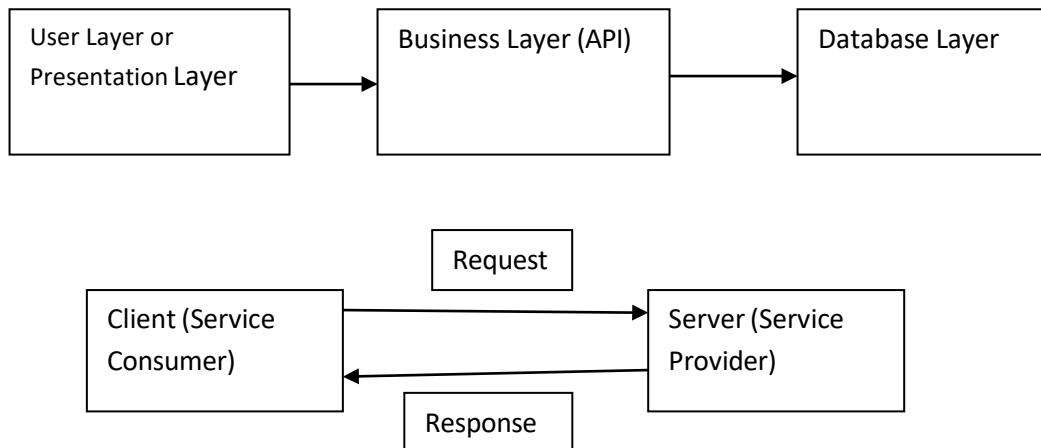
- Service available over the web(Internet)
- Waiter: Who is doing communication between you and kitchen
- Waiter: Waiter is performing the role of web service through an API
- We can say he(Waiter) is communicating between two applications and making sure that the communication between two applications is proper.



## **Web Service:**

- Service available over the web
- Enables communication between applications over the web
- Platform independent communication
- Provides a standard protocol/format for communication
- Using web services two different applications can talk to each other and exchange data/information
- An API Stands 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 instant message to check weather on your phone, each time you are using an API.
- Web Service/API is an Application or business Logic
- Web Service allows you to expose the functionality of your existing code over the network. Once it is exposed on the network, other applications can use the functionality of your program.

## What is API?



Medium: HTTP

Format/language: XML/JSON

**Agenda:** Types of Web Services, Components/Specifications/Standards of SOAP Web Service

### Types of Web Services:

- SOAP Service(Simple Object Access Protocol): XML Language
- REST Service(Representational State Transfer): JSON Language

### SOAP Webservice:

**Components/Specifications/Standards of SOAP Web Service:** WSDL, UDDI, SOAP

#### WSDL: Web Service Description Language:

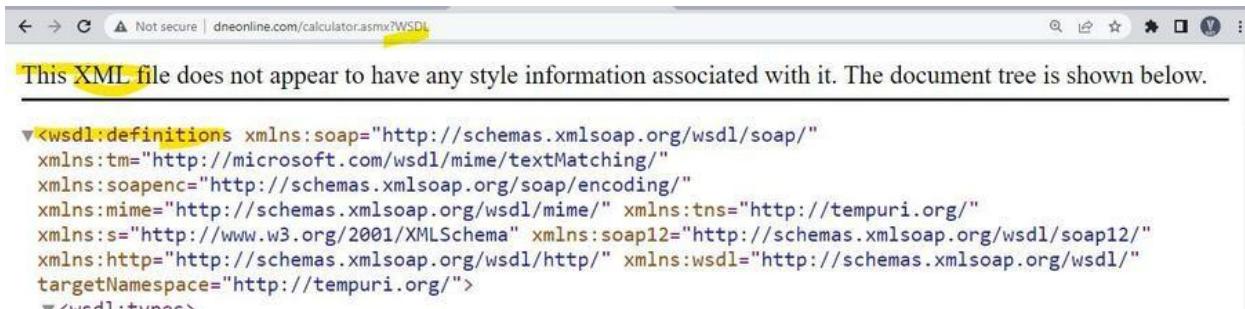
<http://www.dneonline.com/calculator.asmx?WSDL>

- WSDL stands for Web service description Language
- WSDL is the standard format for describing a web service
- WSDL is pronounced as „wiz-dull“ and spelled out as „W-S-D-L“
- WSDL definition describes how to access a web service and what operations it will perform
- A client program connecting to a web service can read WSDL to determine what functions are available on the server.
- WSDL is an XML document with a <definitions> element at the root and the child elements like <types>,

- <message>,  
<portType>,  
<binding>
- Web service description language is an XML based interface that is used to describe the functionalities of the web service

<definitions>:

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



```

<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://tempuri.org/"
  xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  targetNamespace="http://tempuri.org/">
  <wsdl:types>

```

<types>:

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

```

<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>
  </s:schema>

```

<message>:

- The <message> element describes the data being exchanged between the web service provider and the consumer
- Each Web Service has two message: 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>
    ▼<wsdl:message name="SubtractSoapIn">
        <wsdl:part name="parameters" element="tns:Subtract"/>
    </wsdl:message>
    ▼<wsdl:message name="SubtractSoapOut">
        <wsdl:part name="parameters" element="tns:SubtractResponse"/>
    </wsdl:message>
    ▼<wsdl:message name="MultiplySoapIn">
        <wsdl:part name="parameters" element="tns:Multiply"/>
    </wsdl:message>
    ▼<wsdl:message name="MultiplySoapOut">
        <wsdl:part name="parameters" element="tns:MultiplyResponse"/>
    </wsdl:message>
    ▼<wsdl:message name="DivideSoapIn">
        <wsdl:part name="parameters" element="tns:Divide"/>
    </wsdl:message>
    ▼<wsdl:message name="DivideSoapOut">
        <wsdl:part name="parameters" element="tns:DivideResponse"/>
    </wsdl:message>

```

<portType>:

- <portType> can combine one request and one response message into single request response operation
- A port Type can define multiple operations

```

    ▼<wsdl:portType name="CalculatorSoap">
        ▼<wsdl:operation name="Add">
            <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Adds two integers. This is a test
            WebService. ©DNE 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>:

- To combine all operations together

- The <binding> element provides specific details on how a port Type operation will actually be transmitted over the web

```

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

```

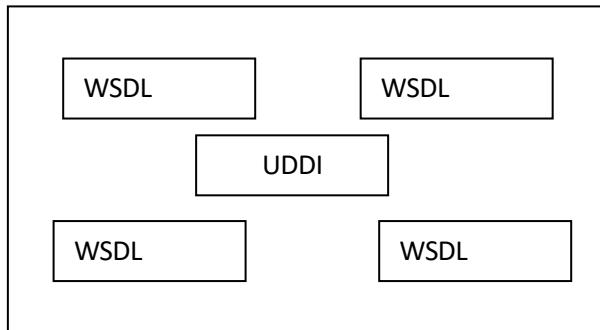
**WSDL File: <definitions>,<types>,<message>,<portType>,<binding>**

### UDDI: Universal Description Discovery Integration

- Universal description discovery Integration is an XML based standard for publishing and finding web services
- A web service provider publishes his web service (through WSDL) on an online directory from where consumers can query and search the web service.
- This is online registry/directory called UDDI.
- Its online directory used to store WSDL links

<http://www.dneonline.com/calculator.asmx?WSDL>

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



### SOAP: Simple Object Access Protocol

- SOAP is an XML based protocol for exchanging information between web services

- SOAP is Communication Protocol
- SOAP provides data transport for web services
- All information exchange happens over a common format: **XML**
- XML message has defined structure: **SOAP Message**
- SOAP Message Consists of: **Envelope, Header, Body, Fault**

### **Envelope :( Mandatory)**

- The SOAP envelope indicates the start and end of the message so that the receiver knows when an entire message has transmitted
- The SOAP Envelope solves the problem of knowing when you are done receiving a message and are ready to process it.
- Every SOAP Message has a root Envelope Element
- Envelope element is mandatory part of SOAP Message.

### **Header :( Optional)**

- Contains any optional attributes of the message used in processing the message, either at an intermediary point or at the ultimate end point

### **Body: (Mandatory)**

- Contains XML data comprising the message being sent
- The SOAP Body is a mandatory element which contains the application defined XML data being exchanged in a 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 a SOAP fault element in the body of the message and the fault is returned to the sender of the SOAP Message.

### **SOAP Message:**

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:tem="http://tempuri.org/">
```

```
    <soapenv:Header>
```

```
    <soapenv:Body>
```

```
<tem:Add>
  <tem:intA>A</tem:intA>
  <tem:intB>15</tem:intB>
</tem:Add>
</soapenv:Body>
</soapenv:Envelope>
```

**SOAP Fault:**

```
<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(XmlSerial
izationReader 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>

```

### **SOAP Web Service:**

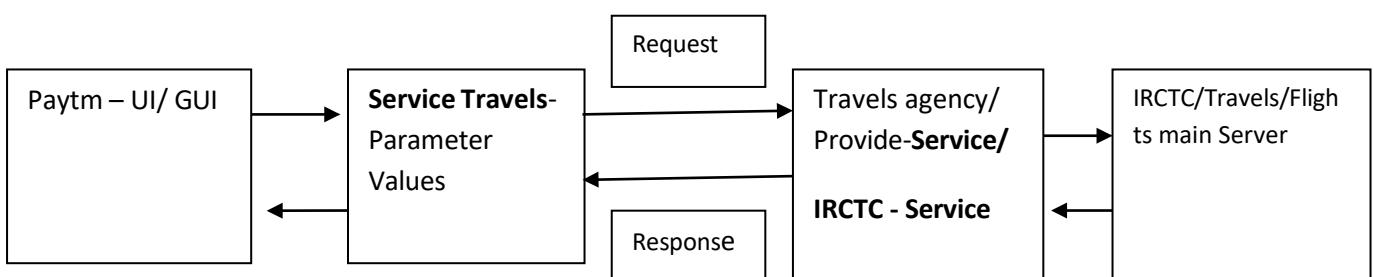
- A web service that follows soap web service specifications is a SOAP service.

### **What are these Specifications/Standards/Components?**

Basic Standards: WSDL,UDDI,SOAP

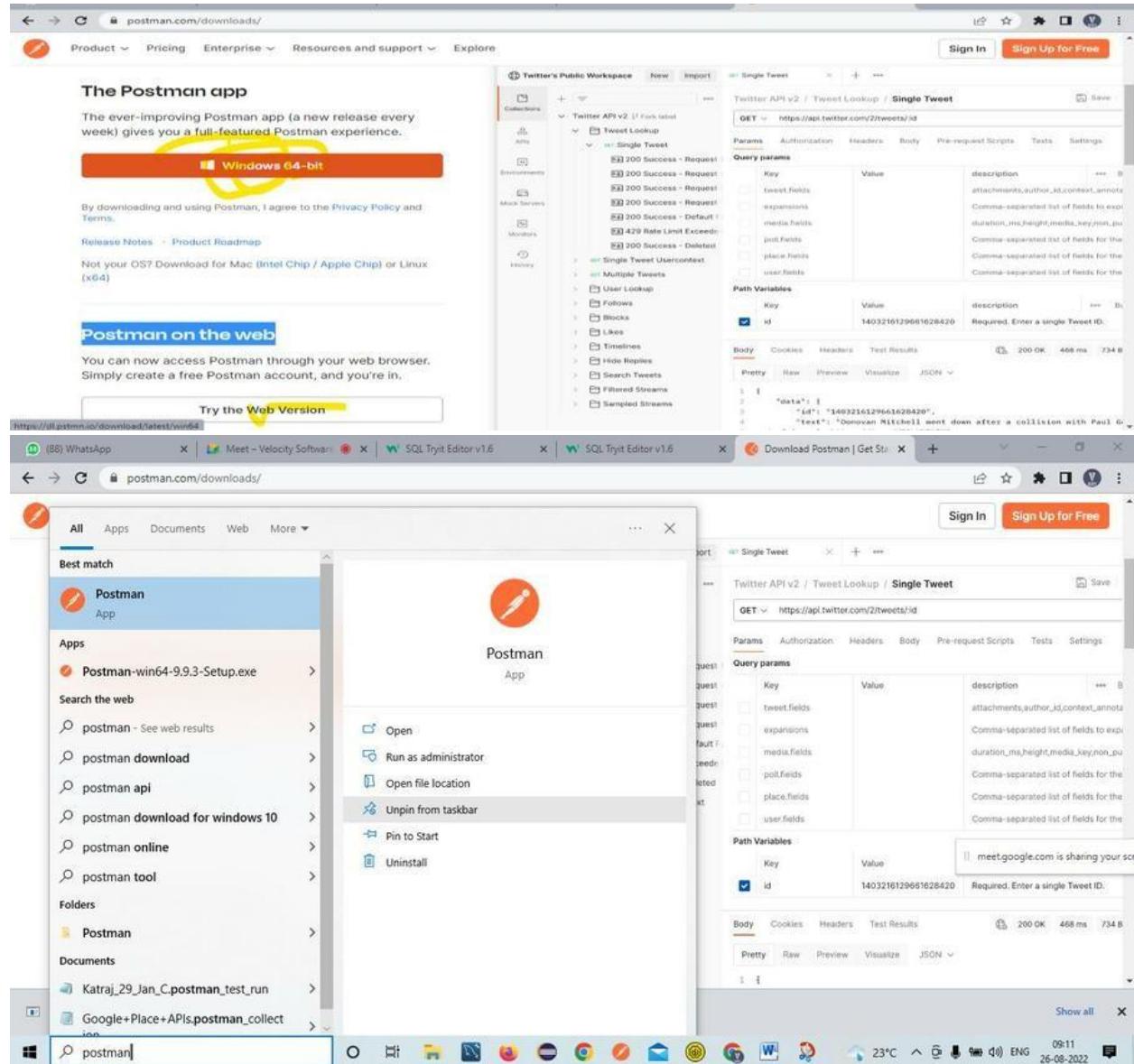
### **API Testing-**

- 2 Types Service
  1. SOAP service(**SOAP UI tool**)
  2. REST service(**SOAP UI,POSTMAN**)



## Agenda: Testing REST Service using POSTMAN Tool: GET Method/Service

Testing REST Service using POSTMAN Tool: <https://www.postman.com/downloads/>



The screenshot shows the Windows Start Menu search results for "postman". The top result is "Postman" (App), which is highlighted and has a context menu open. The menu options include "Open", "Run as administrator", "Open file location", "Unpin from taskbar" (which is selected), "Pin to Start", and "Uninstall". Below the search bar, there are sections for "All", "Apps", "Documents", "Web", and "More". To the right of the search results, a separate window titled "Twitter API v2 / Tweet Lookup / Single Tweet" is visible. This window shows a GET request to "https://api.twitter.com/2/tweets/:id" with various query parameters and path variables defined. The status bar at the bottom indicates "23°C" and "09:11 26-08-2022".

☰ Home Workspaces API Network Explore

Search Postman

Invite ⚙️ 🔍 ⏪ Upgrade

My Workspace

New Import + ...

Collections

- + API Chaining\_Katraj\_26Feb\_D
- + API Chaining\_Viman\_7th May-B
- + API Chaining\_12th\_March\_B
- + API Chaining\_21st May 2022
- + API Chaining\_Katraj\_12Feb\_C
- + API Chaining\_Katraj\_29\_Jan-B
- + API Chaining\_Katraj\_9th\_April\_Evening
- + API VCTC
- + API\_Chaining\_2nd\_April\_Evening
- + API\_Chaining\_16th\_April-Evening
- + API\_Chaining\_5th\_Feb\_Evening
- + API\_chaining\_5Th\_March\_B
- + API\_Chaining\_DEC\_EVE
- + API\_Chaining\_Katraj\_19thMarch\_Eve...
- + API\_Chaining\_katraj\_29\_Jan\_C
- + API\_Chaining\_katraj\_4th\_Dec\_Evening
- + API\_Chaining\_Katraj\_Jan\_Evening
- + API\_Chaining\_Katraj\_Nov\_C
- + API\_Chaining\_Viman15JanC
- + API\_Chaining\_Viman\_8th\_Jan
- + API\_Viman\_Sept

Environments

Mock Servers

Monitors

Flows

History

Online Find and Replace Console

Type here to search

Create a new request:

Cookies Capture requests Bootcamp Runner Trash

09:15 26-08-2022

This screenshot shows the Postman application interface. The left sidebar is titled 'My Workspace' and contains sections for Collections, APIs, Environments, Mock Servers, Monitors, Flows, and History. Under 'Collections', there is a large list of API chaining configurations with names like 'API Chaining\_Katraj\_26Feb\_D', 'API Chaining\_Viman\_7th May-B', etc. The 'Environments' section is currently expanded, showing various environment configurations. The top navigation bar includes links for Home, Workspaces, API Network, and Explore, along with a search bar and user account options. The bottom taskbar shows various application icons.

☰ Home Workspaces API Network Explore

Search Postman

Invite ⚙️ 🔍 ⏪ Upgrade

My Workspace

Search workspaces Create Workspace

Collections

- + Recently visited
- > + My Workspace
- > + Team Workspace
- > More workspaces
- > No workspaces found
- >
- >
- >
- >
- > View all workspaces →
- > APIChaining\_Katraj\_Nov\_C
- > API\_Chaining\_Viman15JanC
- > API\_Chaining\_Viman\_8th\_Jan
- > API\_Viman\_Sept

Environments

Mock Servers

Monitors

Flows

History

Online Find and Replace Console

Type here to search

Create a new request:

Cookies Capture requests Bootcamp Runner Trash

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

This screenshot shows the same Postman interface as the first one, but with a different view of the 'My Workspace' section. The 'Search workspaces' input field is active. The list of collections and environments is partially visible at the bottom. The rest of the interface, including the top navigation, sidebar, and bottom taskbar, is identical to the first screenshot.

## What is Collection?

Ans: Group of requests grouped into folder

The screenshot displays the Postman application interface. At the top, the navigation bar includes 'Home', 'Workspaces', 'API Network', 'Explore', 'Search Postman', 'Invite', 'Upgrade', and window controls. The main workspace is titled 'My Workspace'.

**Left Sidebar:** Shows a tree view of collections, APIs, environments, mock servers, monitors, flows, and history. A 'Create new collection' button is visible.

**Central Area:** Shows a collection named 'Katraj\_11th\_June\_Morning'. The collection details panel includes tabs for 'Authorization', 'Pre-request Script', 'Tests', and 'Variables'. It notes that no authorization is required. A 'Documentation' panel on the right provides instructions for team members.

**Bottom:** The taskbar shows various browser icons and system status (23°C, ENG, 09:20, 26-08-2022).

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections like 'Collections', 'APIs', 'Environments', 'Mock Servers', 'Monitors', 'Flows', and 'History'. The 'Collections' section has a dropdown menu open over 'Katraj\_11th\_June\_Morning', with options like 'Share', 'Move', 'Run collection', 'Edit', 'Add request', etc. The main workspace shows a collection named 'Katraj\_11th\_June\_Morning'. The 'Authorization' tab is active, showing 'No Auth' selected. A note says 'This collection does not use any authorization.' There's also a 'Documentation' panel on the right with a brief description.

This screenshot shows a specific request configuration in Postman. The URL is 'GET List\_Users'. The method dropdown shows 'GET' is selected. Below it, other methods like POST, PUT, PATCH, DELETE, COPY, HEAD, OPTIONS, LINK, UNLINK, PURGE, LOCK, UNLOCK, PROPFIND, and VIEW are listed. The 'Headers (6)' tab is selected in the request details panel. At the bottom, there's a note: 'Enter the URL and click Send to get a response.'

## GET List of Users:

- Returns list of users based on the query parameters
- The Endpoint is secured by authentication Policy and requires authentication token

- **Endpoint:** Base URL/Domain/Endpoint – <https://reqres.in>  
URI- <https://reqres.in/api/users?page=2>

Request Headers:

<u>Header Name</u>	<u>Required/Optional</u>	<u>Description</u>
Accept	Required	Application/json: it will accept request and response in JSON
Content-Type	Required	Application/json: content of request and response in JSON
Authorization	Required	Required basic auth/API Key/Bearer token

Query Parameters:

Parameter Name	Data Type	Required/Optional	Description
Page	Int	Required	It will fetch the specific page records

Invoking the Operations:

Operation URLs and Environments:

<u>Environment</u>	<u>Method</u>	<u>URL</u>
DEV	GET	<a href="https://reqresdev.in/api/users?page=2">https://reqresdev.in/api/users?page=2</a>
INT(Integration or SIT)	GET	<a href="https://reqresint.in/api/users?page=2">https://reqresint.in/api/users?page=2</a>
UAT	GET	<a href="https://reqresuat.in/api/users?page=2">https://reqresuat.in/api/users?page=2</a>
PRE-PROD	GET	<a href="https://reqrespreprod.in/api/users?page=2">https://reqrespreprod.in/api/users?page=2</a>
PROD	GET	<a href="https://reqres.in/api/users?page=2">https://reqres.in/api/users?page=2</a>

Request Body:

Not required for this operation

Response Attributes/tagnames:

<u>Attribute Name</u>	<u>Data Types</u>	<u>Comment</u>

page	int	
per_page	int	
total	int	
total_pages	int	
Data	String	
Id	Int	
email	String	
first_name	String	
last_name	String	
avatar	String	

### Response Examle:

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

```
        "last_name": "Ferguson",
        "avatar": "https://reqres.in/img/faces/8-image.jpg"
    },
    {
        "id": 9,
        "email": "tobias.funke@reqres.in",
        "first_name": "Tobias",
        "last_name": "Funke",
        "avatar": "https://reqres.in/img/faces/9-image.jpg"
    },
    {
        "id": 10,
        "email": "byron.fields@reqres.in",
        "first_name": "Byron",
        "last_name": "Fields",
        "avatar": "https://reqres.in/img/faces/10-image.jpg"
    },
    {
        "id": 11,
        "email": "george.edwards@reqres.in",
        "first_name": "George",
        "last_name": "Edwards",
        "avatar": "https://reqres.in/img/faces/11-image.jpg"
    },
    {
        "id": 12,
        "email": "rachel.howell@reqres.in",
        "first_name": "Rachel",
        "last_name": "Howell",
        "avatar": "https://reqres.in/img/faces/12-image.jpg"
    }
],
"support": {
```

```

        "url": "https://reqres.in/#support-heading",
        "text": "To keep ReqRes free, contributions towards server costs are appreciated!"
    }
}

```

### **GET Method Service:**

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

### **Sample Response Status Code: 200**

#### **Sample Response:**

```
{
    "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"
        },
        {

```

```
        "id": 9,
        "email": "tobias.funke@reqres.in",
        "first_name": "Tobias",
        "last_name": "Funke",
        "avatar": "https://reqres.in/img/faces/9-image.jpg"
    },
    {
        "id": 10,
        "email": "byron.fields@reqres.in",
        "first_name": "Byron",
        "last_name": "Fields",
        "avatar": "https://reqres.in/img/faces/10-image.jpg"
    },
    {
        "id": 11,
        "email": "george.edwards@reqres.in",
        "first_name": "George",
        "last_name": "Edwards",
        "avatar": "https://reqres.in/img/faces/11-image.jpg"
    },
    {
        "id": 12,
        "email": "rachel.howell@reqres.in",
        "first_name": "Rachel",
        "last_name": "Howell",
        "avatar": "https://reqres.in/img/faces/12-image.jpg"
    }
],
"support": {
    "url": "https://reqres.in/#support-heading",
    "text": "To keep ReqRes free, contributions towards server costs are appreciated!"
}
}
19/06/2023:
```

```
{  
  "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"  
    },  
    {  
      "id": 9,  
      "email": "tobias.funke@reqres.in",  
      "first_name": "Tobias",  
      "last_name": "Funke",  
    }]
```

```
        "avatar": "https://reqres.in/img/faces/9-
image.jpg"
    },
{
    "id": 10,
    "email": "byron.fields@reqres.in",
    "first_name": "Byron",
    "last_name": "Fields",
    "avatar": "https://reqres.in/img/faces/10-
image.jpg"
},
{
    "id": 11,
    "email": "george.edwards@reqres.in",
    "first_name": "George",
    "last_name": "Edwards",
    "avatar": "https://reqres.in/img/faces/11-
image.jpg"
},
{
    "id": 12,
    "email": "rachel.howell@reqres.in",
    "first_name": "Rachel",
    "last_name": "Howell",
    "avatar": "https://reqres.in/img/faces/12-
image.jpg"
}
],

```

```

"support": {
    "url": "https://reqres.in/#support-heading",
    "text": "To keep ReqRes free, contributions towards server costs are appreciated!"
}

```

## 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 application interface. On the left, there's a sidebar with 'My Workspace' containing various collections like 'API\_Chaining\_Katraj\_Jan\_Evening', 'API\_Chaining\_Katraj\_Nov\_C', etc. The main area shows a request for 'GET List\_Users' with the URL 'https://reqres.in/api/users?page=1'. Under 'Query Params', there is a table with a single row for 'page' set to '1'. The 'Body' tab shows 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",

```

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

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 collection named "My Workspace". A specific test case named "GET\_List\_Users" is selected. The request method is set to "GET" and the URL is "https://reqres.in/api/users?page=2". In the "Params" tab, there is a single parameter "page" with the value "2". The response body is displayed in JSON format:

```

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",

```

The status bar at the bottom indicates a status of 200 OK, time 54 ms, and size 1.83 KB.

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 the same setup as the previous one, but with the parameter value changed to "3". The response body is:

```

1 {
2   "page": 3,
3   "per_page": 6,
4   "total": 12,
5   "total_pages": 2,
6   "data": [],
7   "support": {
8     "url": "https://reqres.in/#support-heading",
9     "text": "To keep ReqRes free, contributions towards server costs

```

A modal window titled "200 OK" provides standard documentation for a successful HTTP response.

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 a Postman collection named "Katraj\_11th\_June\_Morning" with a "List\_Users" request. The "Params" tab shows a "page" parameter set to "2.5". The "Body" tab displays a JSON response with a large yellow "X" over it. The status bar at the bottom right shows "Status: 200 OK".

```
1 {
  "page": 2,
  "per_page": 6,
  "total": 12,
  "total_pages": 2,
  "data": [
    {
      "id": 7,
      "email": "michael.lawson@reqres.in",
```

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

The screenshot shows a Postman collection with a "List\_Users" request. The "Params" tab shows a "page" parameter with an empty "Value" field. The "Body" tab displays a JSON response with a large yellow "X" over it. The status bar at the bottom right shows "Status: 200 OK".

```
1 {
  "page": 1,
  "per_page": 6,
  "total": 12,
  "total_pages": 2,
  "data": [
    {
      "id": 1,
      "email": "george.bluth@reqres.in",
```

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

The screenshot shows a POSTMAN interface with the following details:

- Method:** GET
- URL:** https://reqres.in/api/users?page=two
- Params:** page (Value: two)
- Body:** (Empty)
- Test Results:** Status: 200 OK, Time: 442 ms, Size: 1.79 KB
- Response Preview (Pretty JSON):**

```
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",
```

A large yellow 'X' is drawn over the JSON response body, indicating it is incorrect or unexpected.

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 a POSTMAN interface with the following details:

- Method:** GET
- URL:** https://reqres.in/vctc/api/users?page=2 (Note: 'vctc' is highlighted in yellow)
- Params:** page (Value: 2)
- Body:** (Empty)
- Test Results:** Status: 404 Not Found, Time: 437 ms, Size: 767 B
- Response Preview (Text):**

```
1 Not found
```

A large yellow checkmark is drawn over the 'Not found' message in the response body, indicating the test passed.

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

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'My Workspace' selected, displaying various collections like 'API\_Chaining\_Katraj\_Jan\_Evening' and 'Katraj\_11th\_June\_Morning'. The main workspace shows a 'Purge' request for 'GET\_List\_Users' with the URL <https://reqres.in/api/users?page=2>. The response status is 403 Forbidden, with the message: 'The request was a legal request, but the server is refusing to respond to it. Unlike a 401 Unauthorized response, authenticating will make no difference.'

### POST Register Users:

- This method will register existing users and set password for them
- Endpoint is secured by authentication Policy and requires authentication token
- **Endpoint:** Base URL/Domain/Endpoint – <https://reqres.in>  
URI- <https://reqres.in/api/register>

### Request Headers:

<u>Header Name</u>	<u>Required/Optional</u>	<u>Description</u>
<u>Accept</u>	<u>Required</u>	<u>Application/json: it will accept request and response in JSON</u>
<u>Content-Type</u>	<u>Required</u>	<u>Application/json: content of request and response in JSON</u>
<u>Authorization</u>	<u>Required</u>	<u>Required basic auth/API Key/Bearer token</u>

### Query Parameters:

This operation do not require query parameters.

Invoking the Operations:

Operation URLs and Environments:

<u>Environment</u>	<u>Method</u>	<u>URL</u>
<u>DEV</u>	<u>POST</u>	<a href="https://reqresdev.in/api/register">https://reqresdev.in/api/register</a>
<u>INT(Integration or SIT)</u>	<u>POST</u>	<a href="https://regressit.in/api/register">https://regressit.in/api/register</a>
<u>UAT</u>	<u>POST</u>	<a href="https://reqresuat.in/api/register">https://reqresuat.in/api/register</a>
<u>PRE-PROD</u>	<u>POST</u>	<a href="https://reqrespreprod.in/api/register">https://reqrespreprod.in/api/register</a>
<u>PROD</u>	<u>POST</u>	<a href="https://reqres.in/api/register">https://reqres.in/api/register</a>

Request Body:

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

Response Attributes/tagnames:

<u>Attribute Name</u>	<u>Data Types</u>	<u>Comment</u>
<u>Id</u>	<u>int</u>	
<u>token</u>	<u>String</u>	

Response Examle:

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

**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/Bearer token-tcaabfj bjf lekf lkfl  
• Time taken for the response=5sec  
• Unit Testing Documents
```

**Sample Response:**

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

## Response Status Code: 200-OK

The screenshot shows two instances of the Postman application interface. Both instances are focused on a POST request to <https://reqres.in/api/register>. The top instance has yellow arrows pointing to the 'Body' tab and the 'x-www-form-urlencoded' radio button, indicating the request type. The bottom instance shows the response body, which contains the number '1'. The interface includes standard Postman features like 'Params', 'Authorization', 'Headers (9)', 'Pre-request Script', 'Tests', 'Settings', 'Cookies', and a 'Send' button.

## Test Cases:

**Precondition: Only existing Users**

## 1. Validating the REST Response=Pass

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists various API collections and environments. In the center, a specific collection named 'Katraj\_31th\_June\_Morning' is selected, and within it, a 'POST Register\_User' request is highlighted. The request details show a POST method to the URL <https://reqres.in/api/register>. The 'Body' tab is selected, showing a JSON payload:

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

The response pane shows a successful status of 200 OK with a response time of 467 ms and a response size of 789 B. The response body is also displayed in JSON format:

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

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

This screenshot is identical to the one above, showing the same workspace, collection, and request details. The response pane again displays a status of 200 OK with a response time of 467 ms and a response size of 789 B. The response body is identical to the previous one:

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

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

The screenshot shows the Postman application interface. At the top, there are two tabs: 'port' (selected) and 'SIT'. Below them, a header bar displays 'Katraj\_11th\_June\_Morning / Register\_User' with a 'Save' button and other options.

The main workspace shows a 'POST' request to 'https://reqres.in/api/register'. The 'Body' tab is selected, showing the following JSON payload:

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

Below the request, the 'Test Results' section shows the response status: Status: 200 OK, Time: 467 ms, Size: 789 B. The response body is displayed as:

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

Yellow annotations highlight the 'email' field in the request body and the 'id' and 'token' fields in the response body. A large yellow checkmark is placed under the 'token' field in the response body.

#### 4. Validating different Status Code in the responses=>EC=200=Pass

The screenshot shows the Postman interface. A POST request is made to `https://reqres.in/api/register`. The request body is a JSON object:

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

The response status is **200 OK**, time taken is 467 ms, and size is 789 B. The response body is:

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

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

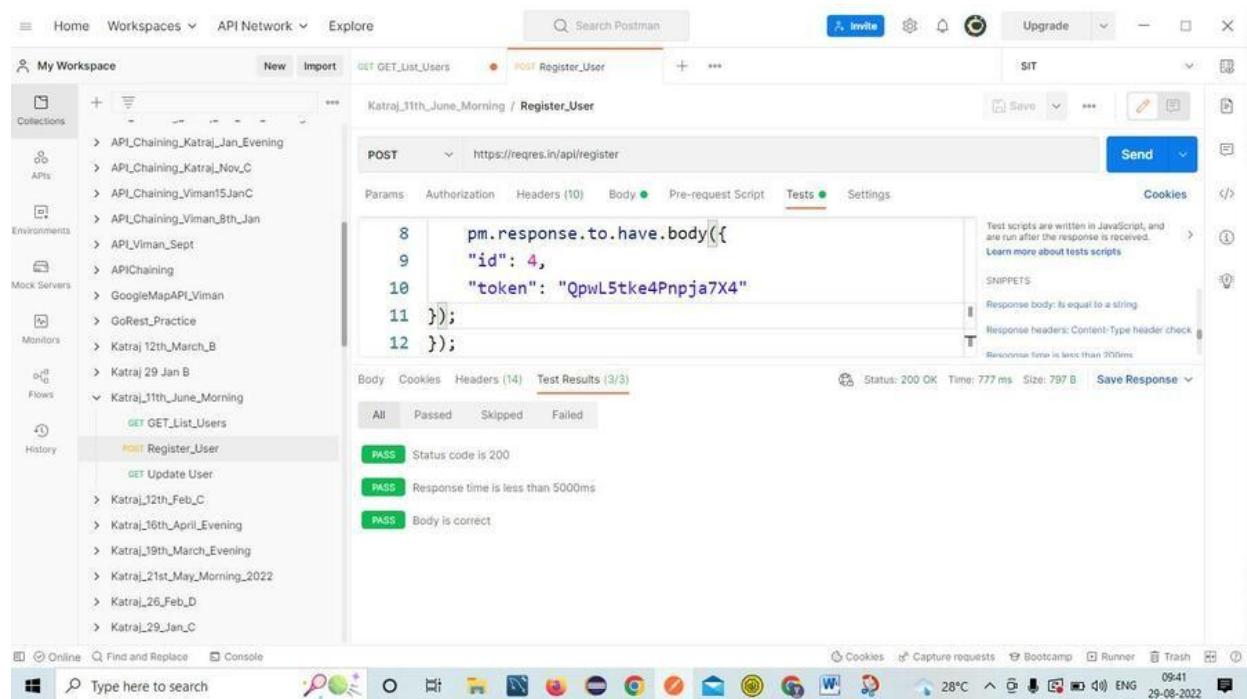
The screenshot shows the Postman interface with the same POST request to `https://reqres.in/api/register`. The request body is identical to the previous screenshot.

The response status is **200 OK**, time taken is 467 ms, and size is 789 B. The response body is identical to the previous screenshot.

A detailed timeline table is shown on the right, listing the following events and their times:

EVENT	TIME
Prepare	31.88 ms
Socket Initialization	2.37 ms
DNS Lookup	Cache
TCP Handshake	Cache
SSL Handshake	Cache
Transfer Start	454.83 ms
Download	9.47 ms
Process	0.37 ms
Total	498.91 ms

## 6. Applying Assertions for Verification (Validation)



```
pm.response.to.have.body({
  "id": 4,
  "token": "QpwL5tke4Pnpja7X4"
});
```

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

SNIPPETS

Response body: Is equal to a string  
Response headers: Content-Type header check  
Response time is less than 200ms

Body Cookies Headers (14) Test Results (3/3)

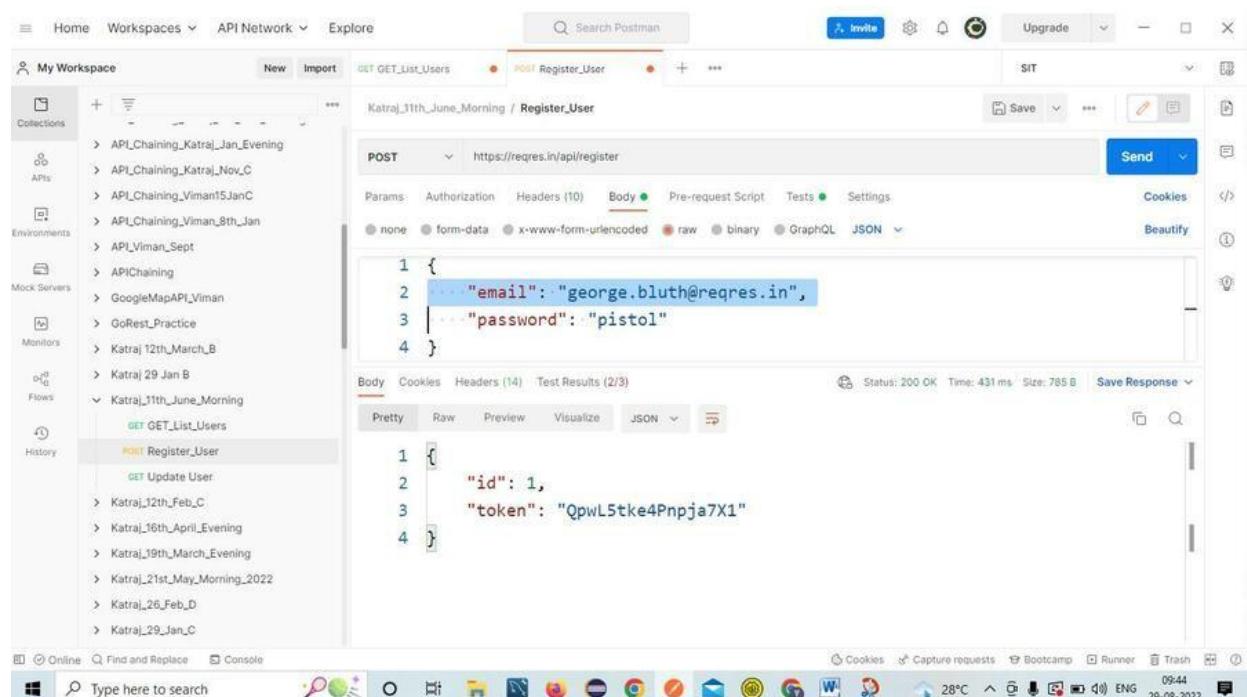
All Passed Skipped Failed

PASS Status code is 200  
PASS Response time is less than 5000ms  
PASS Body is correct

Status: 200 OK Time: 777 ms Size: 797 B Save Response

## 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



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

```
1 {
  "id": 1,
  "token": "QpwL5tke4Pnpja7X1"
}
```

Body Cookies Headers (14) Test Results (2/2)

Pretty Raw Preview Visualize JSON

Status: 200 OK Time: 431 ms Size: 785 B Save Response

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

The screenshot shows the Postman interface with a successful POST request to https://reqres.in/api/register. The request body is a JSON object with "email": "janet.weaver@reqres.in" and "password": "pistol". The response status is 200 OK, and the response body contains "id": 2 and "token": "QpwL5tke4Pnpja7X2".

```
POST https://reqres.in/api/register
{
  "email": "janet.weaver@reqres.in",
  "password": "pistol"
}
{
  "id": 2,
  "token": "QpwL5tke4Pnpja7X2"
}
```

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

The screenshot shows the Postman interface with a successful POST request to https://reqres.in/api/register. The request body is a JSON object with "email": "emma.wong@reqres.in" and "password": "pistol". The response status is 200 OK, and the response body contains "id": 3 and "token": "QpwL5tke4Pnpja7X3".

```
POST https://reqres.in/api/register
{
  "email": "emma.wong@reqres.in",
  "password": "pistol"
}
{
  "id": 3,
  "token": "QpwL5tke4Pnpja7X3"
}
```

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

The screenshot shows the Postman interface with a successful POST request to <https://reqres.in/api/register>. The request body is a JSON object with fields `email` and `password`. The response status is 200 OK, and the response body contains `id: 5` and `token: "OpwL5tke4Pnpja7X5"`.

```

POST https://reqres.in/api/register
{
  "email": "charles.morris@reqres.in",
  "password": "pistol"
}
{
  "id": 5,
  "token": "OpwL5tke4Pnpja7X5"
}
    
```

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

7.6 Verify POST method/ request by passing “rachel.howell@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 POST request to <https://reqres.in/api/register>. The request body is a JSON object with fields `email` and `password`. The response status is 200 OK, and the response body contains `id: 12` and `token: "OpwL5tke4Pnpja7X12"`.

```

POST https://reqres.in/api/register
{
  "email": "rachel.howell@reqres.in",
  "password": "Indian"
}
{
  "id": 12,
  "token": "OpwL5tke4Pnpja7X12"
}
    
```

## 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 a collection named "My Workspace". A POST request titled "Register\_User" is selected. The request URL is "https://reqres.in/api/register". The request body is set to "x-www-form-urlencoded" and contains the following JSON:

```
1 {
2   "email": "vctc@reqres.in",
3   "password": "Indian"
4 }
```

The response status is "400 Bad Request" with the message "The request cannot be fulfilled due to bad syntax." and the error "error": "Note: Only defined users succeed registration".

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

The screenshot shows the Postman interface with a collection named "My Workspace". A POST request titled "Register\_User" is selected. The request URL is "https://reqres.in/api/register". The request body is set to "x-www-form-urlencoded" and contains the following JSON:

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

The response status is "400 Bad Request" with the message "The request cannot be fulfilled due to bad syntax." and the error "error": "Missing email or username".

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

The screenshot shows the Postman interface with a POST request to `https://reqres.in/api/register`. The request body is a JSON object with a single field `"email": "rachel.howell@reqres.in"`. The response status is 400 Bad Request, and the error message is `"error": "Missing password"`.

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

The screenshot shows the Postman interface with a POST request to `https://reqres.in/api/register`. The request body is a JSON object with fields `"email": "rachel.howell@gmail.com", "password": "Indian"`. The response status is 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

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

The screenshot shows the Postman interface with a collection named 'My Workspace'. A POST request for 'Register\_User' is selected. The URL is set to `https://reqres.in/v2/api/register`. The request body is a JSON object:

```
1 {
2   "email": "rachel.howell@reqres.in",
3   "password": "Indian"
4 }
```

The response status is 404 Not Found, with the message: "The requested resource could not be found but may be available again in the future. Subsequent requests by the client are permissible."

## 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 with a collection named 'My Workspace'. A PURGE request for 'Register\_User' is selected. The URL is set to `https://reqres.in/api/register`. The request body is a JSON object:

```
1 {
2   "email": "rachel.howell@reqres.in",
3   "password": "Indian"
4 }
```

The response status is 403 Forbidden, with the message: "The request was a legal request, but the server is refusing to respond to it. Unlike a 401 Unauthorized response, authenticating will make no difference."

## **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",  
  "type": "Shoe Park"  
}
```

The screenshot shows the Postman interface. In the left sidebar, under 'My Workspace', there is a collection named 'GoogleMapAPI\_Viman / AddPlace'. A POST request is selected for this collection. The URL is set to <https://rahulshettyacademy.com/maps/api/place/add/json?key=qaclick123>. The 'Body' tab is active, showing the JSON payload:

```

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

```

Below the body, the response status is 200 OK with a time of 506 ms and a size of 610 B. The response body is displayed as:

```

1 {
2   "status": "OK",
3   "place_id": "6339c7207d35b30e1c2ff000b5d78132",
4   "scope": "APP",
5   "reference": "21ccccf078f71e5e10bcfd3218f9993a21ccc078f71e5e10bcfd3218f9993a",
6   "id": "21ccccf078f71e5e10bcfd3218f9993a"
7 }

```

## 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](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(POST) 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. The top navigation bar includes File, Edit, View, Help, Home, Workspaces, API Network, Reports, Explore, and a search bar. Below the navigation is a toolbar with various icons for file operations like Save, Import, Export, and a Send button.

The main workspace is titled "GoogleMapAPI\_Viman / New Request". A GET request is being prepared to the URL [http://rahulshettyacademy.com/maps/api/place/get/json?place\\_id=6339c7207d35b30e1c2ff000b5d78132&key=qadclick123](http://rahulshettyacademy.com/maps/api/place/get/json?place_id=6339c7207d35b30e1c2ff000b5d78132&key=qadclick123). The "Params" tab is selected, showing two parameters: "place\_id" with value "6339c7207d35b30e1c2ff000b5d78132" and "key" with value "qadclick123".

The "Body" tab shows the JSON response received from the API. The response is a single object with the following properties:

```

{
  "location": {
    "latitude": "-38.383494",
    "longitude": "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 status bar at the bottom indicates a 200 OK response with a time of 2.16 s and a size of 673 B. A tooltip from a browser window says "meet.google.com is sharing your screen." The bottom of the screen shows the Windows taskbar with various pinned icons and system status indicators.

## 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](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"  
}
```

## **Google Maps Delete API (DELETE):**

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"  
}
```

### Real API:

- GET method Service –URL/ URI= api.openweathermap.org/data/2.5/weather?q={city name}&appid={API key} → Weather API
- **API Key/ Bearer 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 => 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 flows. The main workspace shows a 'GET GET\_List\_Users' request under the 'Katraj\_11th\_June\_Morning' collection. The request URL is 'api.openweathermap.org/data/2.5/weather?q=Pune&appid=abe5bdb0a984910eb4b5a6a82a0a245d'. The 'Params' tab shows 'q' set to 'Pune' and 'appid' set to 'abe5bdb0a984910eb4b5a6a82a0a245d'. The 'Body' tab shows a JSON response with a single weather object:

```
"weather": [  
  {  
    "id": 804,  
    "main": "Clouds",  
    "description": "overcast clouds",  
    "icon": "04d"  
  }]
```

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

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections like 'My Workspace', 'Collections', 'APIs', 'Environments', 'Mock Servers', 'Monitors', 'Flows', and 'History'. In the main area, a 'GET GET\_List\_Users' request is selected. A new 'GET Weather Info' request is being configured with the URL 'api.openweathermap.org/data/2.5/weather?q=Mumbai&appid=abe5bdb0a984910eb4b5a6a82a0a245d'. The 'Params' tab shows 'q' set to 'Mumbai' and 'appid' set to 'abe5bdb0a984910eb4b5a6a82a0a245d'. The 'Body' tab is empty. The 'Test Results' section shows a status of '200 OK' with a response time of 76 ms and a size of 793 B. The response body is displayed in JSON format:

```
6 "weather": [
7   {
8     "id": 803,
9     "main": "Clouds",
10    "description": "broken clouds",
11    "icon": "04d"
12 }
```

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

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

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

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 a collection named 'Katraj\_31th\_June\_Morning'. A 'GET Weather Info' request is selected. The URL is set to `api.openweathermap.org/data/2.5/weather?q=Deoni&appid=abe5bdb0a984910eb4b5a6a82a0a245d`. The 'Params' tab shows 'q' with value 'Deoni' and 'appid' with value 'abe5bdb0a984910eb4b5a6a82a0a245d'. The response status is 404 Not Found, with the message: "cod": "404", "message": "city not found".

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 the same collection and request setup as the previous screenshot. The 'Params' tab shows 'q' as an empty string 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 a collection named "Katraj\_11th\_June\_Morning". A GET request is made to `api.openweathermap.org/data/2.5/weather?q=Canada&appid=abe5bdb0a984910eb4b5a6a82a0a245d`. The response status is 200 OK, and the JSON body contains weather information for Canada.

```

{
  "weather": [
    {
      "id": 800,
      "main": "Clear",
      "description": "clear sky",
      "icon": "01n"
    }
  ]
}
  
```

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

The screenshot shows the Postman interface with a collection named "Katraj\_11th\_June\_Morning". A GET request is made to `api.openweathermap.org/data/2.5/weather?q=Pinepone&appid=abe5bdb0a984910eb4b5a6a82a0a245d`. The response status is 404 Not Found, indicating the city was not found.

```

{
  "cod": "404",
  "message": "city not found"
}
  
```

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 a failed API request. The request is a GET to `api.openweathermap.org/v2.5/weather?q=Pune&appid=abe5bdb0a984910eb4b5a6a82a0a245d`. The status code is 404 Not Found. The response body is an HTML error page:

```
1 <html>
2
3 <head>
4   <title>404 Not Found</title>
5 </head>
6
7 <body>
```

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

The screenshot shows the Postman interface with a failed API request. The request is a GET to `api.openweathermap.org/data/2.5/weather?q=Pune&appid=abe5bdb0a984910eb4b5a6a82a0a245`. The status code is 401 Unauthorized. The response body is a JSON object:

```
1 {
2   "cod": 401,
3   "message": "Invalid API key. Please see http://openweathermap.org/faq#error401 for more info."
4 }
```



## How to Run Collection?

The screenshot shows the Postman application interface. On the left, there is a sidebar titled "My Workspace" containing sections for Collections, APIs, Environments, Mock Servers, Monitors, Flows, and History. Under "Collections", a context menu is open over a collection named "Katraj\_11th\_June\_Morning". The menu options include "Share", "Move", and "Run collection". The "Run collection" option is highlighted. The main workspace area features a large circular icon with a pencil and a rocket ship. Below it is a button labeled "Open Workspace Overview". At the bottom of the screen, there is a taskbar with various icons and a search bar.

Home Workspaces API Network Explore Search Postman Invite Upgrade SIT

My Workspace New Import Runner + ...

Collections APIs Environments Mock Servers Monitors History

API\_Chaining\_Viman15JanC API\_Chaining\_Viman\_8th\_Jan API\_Viman\_Sept APIChaining GoogleMapAPI\_Viman GoRest\_Practice Katraj\_12th\_March\_B Katraj\_29\_Jan\_B Katraj\_11th\_June\_Morning

GET List\_Users POST Register\_User GET Update User GET Weather Info

Iterations: 1 Delay: 0 ms Data: Select File Advanced settings: Save responses, Keep variable values, Run collection without using stored cookies, Save cookies after collection run

Run Katraj\_11th\_June\_Morning Run Katraj\_11th\_June\_Morning

Online Find and Replace Console Cookies Capture requests Bootcamp Runner Trash

Type here to search

10:08 30-08-2022

Home Workspaces API Network Explore Search Postman Invite Upgrade SIT

My Workspace New Import Katraj\_11th\_June\_Morning + ...

Collections APIs Environments Mock Servers Monitors History

API\_Chaining\_Viman15JanC API\_Chaining\_Viman\_8th\_Jan API\_Viman\_Sept APIChaining GoogleMapAPI\_Viman GoRest\_Practice Katraj\_12th\_March\_B Katraj\_29\_Jan\_B Katraj\_11th\_June\_Morning

GET List\_Users POST Register\_User GET Update User GET Weather Info

All Tests Passed (8) Failed (2) Skipped (0) View Summary

Iteration 1

GET List\_Users https://reqres.in/api/users?page=2 / GET\_List\_Users 200 OK 51 ms 1.876 KB

Pass Validating the JSON Response

Pass Expected Status code is 200

Pass Expected Response time is less than 5000ms

Pass Your test name

POST Register\_User https://reqres.in/api/register / Register\_User 200 OK 454 ms 795 B

Pass Status code is 200

Pass Response time is less than 5000ms

Fail Body is correct | AssertionError: expected response body json to equal { id: 4, token: 'QpwL5tkc4Pnpja7X4' } but got [ id: 12, token: 'QpwL5tk...'

Pass Response time is less than 5000ms

Fail Body is correct | AssertionError: expected response body json to equal { id: 4, token: 'QpwL5tkc4Pnpja7X4' } but got [ id: 12, token: 'QpwL5tk...'

Pass Status code is 200

Online Find and Replace Console Cookies Capture requests Bootcamp Runner Trash

Type here to search

10:08 30-08-2022

How will you skip any methods in collection?

The screenshot shows the Postman Runner interface. On the left, the 'My Workspace' sidebar lists various collections, environments, mock servers, and monitors. The main area displays the 'Katraj\_11th\_June\_Morning' collection. Under 'RUN ORDER', the 'GET\_List\_Users' request is selected and checked. In the 'Run Settings' panel, 'Iterations' is set to 1, and there is a 'Run Katraj\_11th\_June\_Morning' button.

The screenshot shows the Postman Test Results interface after running the 'Katraj\_11th\_June\_Morning' collection. The results summary shows 4 Passed tests, 0 Failed, and 0 Skipped. The detailed results for 'Iteration 1' show four test cases: 'Expected Response time is less than 5000ms' (Pass), 'Expected Status code is 200' (Pass), 'Validating the JSON Response' (Pass), and 'Your test name' (Pass). A red 'Run Again' button is visible at the top right.

## Miscellaneous:

### 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.

Postman

File Edit View Help

Home Workspaces API Network Reports Explore Search Postman

VimanNagr18Sept / GET Emp

Set as variable

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

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

```
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 jsonData=JSON.parse(responseBody)
12 value=jsonData.data[1].first_name
13 console.log(value)
14 pm.environment.set("Username1", value);
```

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: 717 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

Type here to search

Postman

File Edit View Help

Home Workspaces API Network Reports Explore Search Postman

Katraj\_4th\_Dec\_Evening / GET Get\_User\_details

Set as variable

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

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

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
page	2		
Key	Value	Description	

Response

Click Send to get a response

Find and Replace Console

Type here to search

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

Katraj\_8th\_Jan\_Evening GET Get\_User\_Details

APIChaining

New Import

My Workspace

Collections

+ Add Collection

API VCTC

API\_Chaining\_DEC\_EVE

API\_Chaining\_Katraj\_Nov\_C

API\_Viman\_Sept

APIChaining

GoogleMapAPI\_Viman

GoRest\_Practice

Katraj\_8th\_Jan\_Evening

GET Get\_User\_Details

POST Register\_User

Katraj\_Dec\_C

Katraj\_NOV\_C

Katraj\_OCT\_B

New Collection

VimanNagr18Sept

Weather\_API\_Dec\_C

Weather\_API\_Katraj\_Nov\_C

WeatherAPI

Set as variable

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

Send

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

Query Params

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

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 Reques 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 19:06 21-03-2022

```
49,
50, "support": {
51, "url": "https://reqres.in/#support-heading",
52, "text": "To keep Reques 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

APIChaining

My Workspace

New Import WeatherAPI\_Jan\_Evening GET Get\_User\_Details

Save Send

API VCTC Katraj\_8th\_Jan\_Evening / Get\_User\_Details

APIChaining DEC\_EVE APIChaining\_Katraj\_Nov\_C API\_Viman\_Sept GoRest\_Practice Katraj\_8th\_Jan\_Evening

GET Params Authori Query Params KEY page Value

baseUrl global\_base\_url URL Username vctc Username\_1

Body Cookies Headers

Pretty Raw Preview Visualize JSON

Description

Status: 200 OK Time: 243 ms Size: 188 KB Save Response

Find and Replace Console

Type here to search

23°C ENG 19:07 21-03-2022

Home Workspaces API Network Reports Explore

Search Postman

Katraj\_8th\_Jan\_Evening / Get\_User\_Details

APIChaining

My Workspace

New Import WeatherAPI\_Jan\_Evening GET Get\_User\_Details

Save Send

API VCTC Katraj\_8th\_Jan\_Evening / Get\_User\_Details

APIChaining DEC\_EVE APIChaining\_Katraj\_Nov\_C API\_Viman\_Sept GoRest\_Practice Katraj\_8th\_Jan\_Evening

GET Params Authori Query Params KEY page Value

Name domain Value https://reqres.in Scope Active Environment: APIChaining

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

Pretty Raw Preview Visualize JSON

Description

Status: 200 OK Time: 243 ms Size: 188 KB Save Response

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

My Workspace

New Import WeatherAPI\_Jan\_Evening GET Get\_User\_Details

DEV

Katraj\_8th\_Jan\_Evening / Get\_User\_Details

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

Params Author

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

Key Value

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

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

Pretty Raw Preview Visualize JSON

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

Cookies Capture requests Bootcamp Runner Trash

Find and Replace Console

Type here to search

Home Workspaces API Network Reports Explore

Search Postman

My Workspace

New Import WeatherAPI\_Jan\_Evening GET Get\_User\_Details

DEV

Katraj\_8th\_Jan\_Evening / Get\_User\_Details

GET

Set as new variable

Name global\_domain

Value https://reqres.in

Scope Global

Set variable

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

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

Pretty Raw Preview Visualize JSON

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

My Workspace

New Import WeatherAPI\_Jan\_Evening GET Get\_User\_Details

DEV

Katraj\_8th\_Jan\_Evening / Get\_User\_Details

GET

Set as new variable

Name global\_domain

Value https://reqres.in

Scope Global

Set variable

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

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

Pretty Raw Preview Visualize JSON

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

WeatherAPI\_Jan\_Evening / GET Get\_User\_Details

DEV

Katraj\_8th\_Jan\_Evening / Get\_User\_Details

GET `{[global_domain]}/api/users?page=2`

Params Authorization Headers Body Pre-request Script Tests Settings Cookies

Query Params

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

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 interface with a successful API call. The collection 'Katraj\_8th\_Jan\_Evening' contains a 'Get\_User\_Details' request. The request is a GET to `{[global_domain]}/api/users?page=2`. The 'Query Params' section has a 'page' parameter set to 2. The response status is 200 OK, time is 243 ms, and size is 1.88 KB. The response body is a JSON object with a 'support' field containing a URL and a message about server costs.

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

WeatherAPI\_Jan\_Evening / GET Get\_User\_Details

APIChaining

Katraj\_8th\_Jan\_Evening / Get\_User\_Details

GET `{(domain)}/api/users?page=2`

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

Query Params

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

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 interface with a successful API call. The collection 'Katraj\_8th\_Jan\_Evening' contains a 'Get\_User\_Details' request. The request is a GET to `{(domain)}/api/users?page=2`. The 'Query Params' section has a 'page' parameter set to 2. The response status is 200 OK, time is 1008 ms, and size is 1.88 KB. The response body is identical to the previous screenshot, containing a 'support' field with a URL and a message about server costs.

The screenshot shows the Postman application window. On the left, the sidebar includes 'My Workspace' (Collections, APIs, Environments, Mock Servers, Monitors, Flows, History), 'File' menu, and 'Upgrade' button. The main area displays a collection named 'Katraj\_8th\_Jan\_Evening' with a sub-request 'Get\_User\_Details'. The request details show a GET method to '({domain})/api/users?page=2'. The 'Params' tab lists 'username' with value 'xyz' and 'vctc' with value 'https://reqres.in'. The 'Query Params' tab lists 'page' with value '2' and 'key' with value 'global\_domain'. The 'Body' tab shows a JSON response with a 'support' object containing a URL and text. The 'Globals' section shows 'baseURL' and 'global\_base\_url' both set to 'https://reqres.in'. The bottom status bar shows 'meet.google.com is sharing your screen.' and the date/time '21-03-2022 19:12'.

## How to Add Environments?

The screenshot shows the Postman application window. The sidebar includes 'My Workspace' (Collections, APIs, Environments, Mock Servers, Monitors, Flows, History), 'File' menu, and 'Upgrade' button. The main area displays a collection named 'Katraj\_8th\_Jan\_Evening' with a sub-request 'Get\_User\_Details'. The request details show a GET method to '({domain})/api/users?page=2'. The 'Params' tab lists 'page' with value '2'. The 'Body' tab shows a JSON response with a 'support' object containing a URL and text. The bottom status bar shows 'meet.google.com is sharing your screen.' and the date/time '21-03-2022 19:18'.

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

VimanNagr18Sept / GET\_Emp

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

Params Author `url_new` Body Cookies Headers (19) Test Results (1/3)

INITIAL https://reqres.in

Tests `pm.expect(pm.response.statusCode).to.be.below(200);`

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

Status: 200 OK Time: 717 ms Size: 1.69 KB Save Response

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

All Passed Skipped Failed

PASS Status code is 200.

FAIL Response time is less than 200ms | AssertionError: expected 717 to be below 200

FAIL Verify Response | AssertionError: 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

VimanNagr18Sept / GET\_Emp

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

Params Authorization Header

INITIAL https://reqres.in

QA

VARIABLE INITIAL VALUE CURRENT VALUE

url https://reqres.in

Username1 Lindsay

URL123 http://dummy.restapieexample.com http://dummy.restapieexample.com

URL123 http://dummy.restapieexample.com http://dummy.restapieexample.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.

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 two instances of the Postman application side-by-side. Both instances have the same workspace structure and requests defined.

**Left Instance (Initial State):**

- Request URL:** `https://reqres.in/api/users?page=2`
- Tests Tab (Visible):**

```

5 pm.expect(response);
6 });
7 pm.test("Ve
8 var jsonData = response.json();
9 pm.expect(jsonData.data[1].first_name).to.eql("Michael");
10 });
11 bodyData=JSON.stringify(response);
12 value=bodyData.data[1].first_name
13 console.log(value)
14 pm.environment.set("Username1", value);
    
```
- Test Results (Visible):**
  - 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'

**Right Instance (After Chaining):**

- Request URL:** `((baseUrl))/api/users?page=2`
- Tests Tab (Visible):**

```

5 pm.expect(response);
6 });
7 pm.test("Ve
8 var jsonData = pm.response.json();
9 pm.expect(jsonData.data[1].first_name).to.eql("Michael");
10 });
11 bodyData=JSON.stringify(response);
12 value=bodyData.data[1].first_name
13 console.log(value)
14 pm.environment.set("Username1", value);
    
```
- Test Results (Visible):**
  - 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'

22/06/2023:

### 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 two instances of the Postman application interface.

**Top Window:**

- Header:** 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).
- Central Area:**
  - Collection: APIChaining / UpdateRequest
  - Method: PUT, URL: https://reqres.in/api/users/2
  - Body tab (selected): JSON content type, Body content:

```

1 {
2   "name": "{{Username1}}",
3   "job": "zion resident"
4 }

```
  - Test Results: Status: 200 OK, Time: 456 ms, Size: 938 B, Save Response.
- Bottom:** Find and Replace, Console, Capture requests and cookies, Bootcamp, Runner, Trash.

**Bottom Window:**

- Header:** 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).
- Central Area:** A modal dialog titled "Create New" is open, showing "Building Blocks" and "Advanced" sections.
  - 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.
    - 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.
- Bottom:** Find and Replace, Console, Capture requests and cookies, Bootcamp, Runner, Trash.

The screenshot shows the Postman application interface. On the left, the sidebar includes sections for Collections, APIs, Environments, Mock Servers, Monitors, Flows, and History. The main workspace is titled 'APIChaining' and contains a table for defining variables. A variable named 'Username' is listed with a type of 'default', initial value 'xyz', and current value 'xyz'. Below the table, there is 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 information.

VARIABLE	TYPE ⓘ	INITIAL VALUE ⓘ	CURRENT VALUE ⓘ
Username	default	xyz	xyz

① 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

## Go to 2<sup>nd</sup> Method (Put) body & Give Parameter name as Username

{

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

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists various collections, APIs, environments, and flows. The 'APIChaining / UpdateRequest' collection is selected. In the main workspace, a PUT request is configured to 'https://reqres.in/api/users/2'. The 'Body' tab shows a JSON payload:

```

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

```

The response status is 200 OK, with a response body showing a user object with updated name and job details.

Send request and Make Sure you are using Mentioned Environment above

Step 3: Add Script to fetch value from response from 1<sup>st</sup> API

Go to JSON Path Finder and get the JSON Path:

The screenshot shows the JSON Path Finder tool in a browser window. It displays a sample JSON object representing a list of users. A search bar at the top contains the path 'x.data[0].first\_name'. The results pane on the right shows the path and the extracted value 'Michael' under the first user object.

```

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    },
14    {
15      "id": 8,
16      "email": "lindsay.ferguson@reqres.in",
17      "first_name": "Lindsay",
18      "last_name": "Ferguson",
19      "avatar": "https://reqres.in/img/faces/8-image.jpg"
20    },
21    {
22      "id": 9,
23      "email": "tobias.funke@reqres.in",
24      "first_name": "Tobias",
25      "last_name": "Funke",
26      "avatar": "https://reqres.in/img/faces/9-image.jpg"
27    },
28    {
29      "id": 10,
30      "email": "byron.fields@reqres.in",
31      "first_name": "Byron",
32      "last_name": "Fields",
33    }
  ]
}

```

Go to Tests=>>>

```

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

```

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

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

The screenshot shows the Postman application interface. On the left, the sidebar displays 'My Workspace' with various collections and environments. The main area shows an API chaining setup for a 'PUT UpdateRequest' to 'https://reqres.in/api/users/2'. The 'Params' tab shows a single parameter 'name' with the value '`name": "{{Username2}}"`'. The 'Headers' tab is empty. The 'Body' tab shows a JSON payload with fields 'name', 'job', and 'updatedAt'. The 'Globals' tab shows a global variable 'baseUrl' set to 'https://reqres.in'. The status bar at the bottom indicates the current temperature is 28°C and the date is 25-01-2022.

Step 4: Value coming from response of 1<sup>st</sup> 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

Setting Global Variable in API Chaining:

Home Workspaces API Network explore

Search Postman

New Import POST Update\_USER GET GET\_LIST\_USER PUT Update\_User + \*\*\* QA

My Workspace Personal

Collections API Chaining\_Katraj\_25th\_JuneB\_2022 / QA

PUT https://reqres.in/api/

No Environment variables

An environment is a set of variables that allow you to switch the context of your requests.

Params Authorization Headers (10)

none form-data x-www-form

Globals

VARIABLE	INITIAL VALUE	CURRENT VALUE
Global_BaseURL	https://reqres.in	https://reqres.in
Global_URL	https://reqres.in	https://reqres.in
Username1	xyz	Michael

Body Cookies Headers (15) Test Result

Pretty Raw Preview Visual

```
1 {  
2   "name": "Mic  
3   "job": "QA L  
4   "updatedAt":  
5 }
```

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.

Online Find and Replace Console

Cookies Capture requests Bootcamp Runner Trash

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** My Workspace (Personal), Collections, APIs, Environments, Mock Servers, Monitors, Flows, History.
- Top Bar:** Home, Workspaces, API Network, explore, Search Postman, New, Import, POST Update\_USER, GET GET\_LIST\_USER, PUT Update\_User, +, \*\*\*, QA.
- Current Collection:** API Chaining\_Katraj\_25th\_JuneB\_2022 / QA
- Request Details:** PUT https://reqres.in/api/
- Environment:** No Environment variables
- Globals:** Global\_BaseURL: https://reqres.in, Global\_URL: https://reqres.in, Username1: xyz (current value Michael).
- Body:** JSON payload with variables:

```
1 {  
2   "name": "Mic  
3   "job": "QA L  
4   "updatedAt":  
5 }
```

A tooltip for the variable "name" explains: "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."
- Bottom:** Cookies, Capture requests, Bootcamp, Runner, Trash.

The screenshot shows the SoapUI interface with two API requests:

- GET\_LIST\_USERS Request:**
  - Method: GET
  - URL: <https://reqres.in/api/users?page=2>
  - Pre-request Script (Tests tab):

```
1 bodyData=JSON.parse(responseBody)
2 Value=bodyData.data[0].first_name
3 console.log(Value)
4 pm.globals.set("Username1",Value);
```
  - Response Body (Pretty):

```
1 "data": [
```
- PUT\_Update\_User Request:**
  - Method: PUT
  - URL: <https://reqres.in/api/users/2>
  - Body (Pretty):

```
1 {
2   "name": "{{Username1}}",
3   "job": "QA Lead"
4 }
```
  - Response Body (Pretty):

```
1 {
2   "name": "Michael",
3   "job": "QA Lead",
4   "updatedAt": "2022-10-12T04:41:42.695Z"
5 }
```

## Agenda: GET Method (Testing REST service using SOAP UI Tool)

### Testing REST Service using SOAP UI Tool:

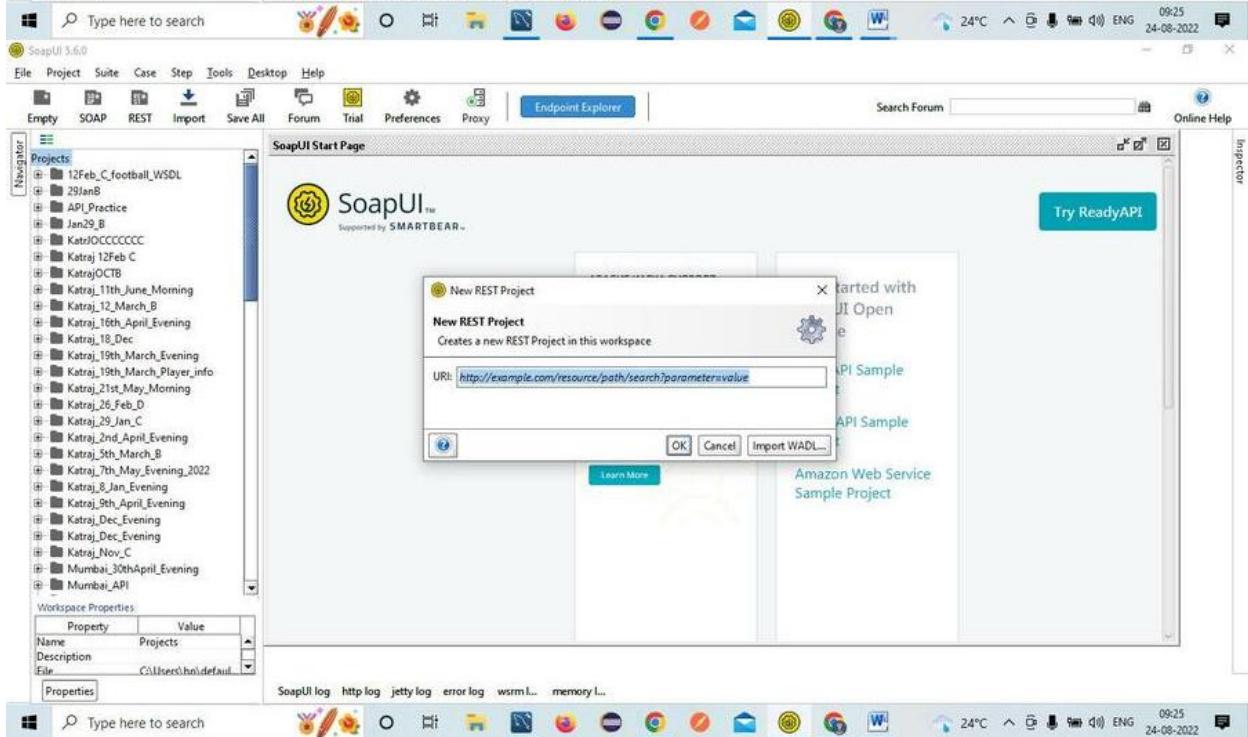
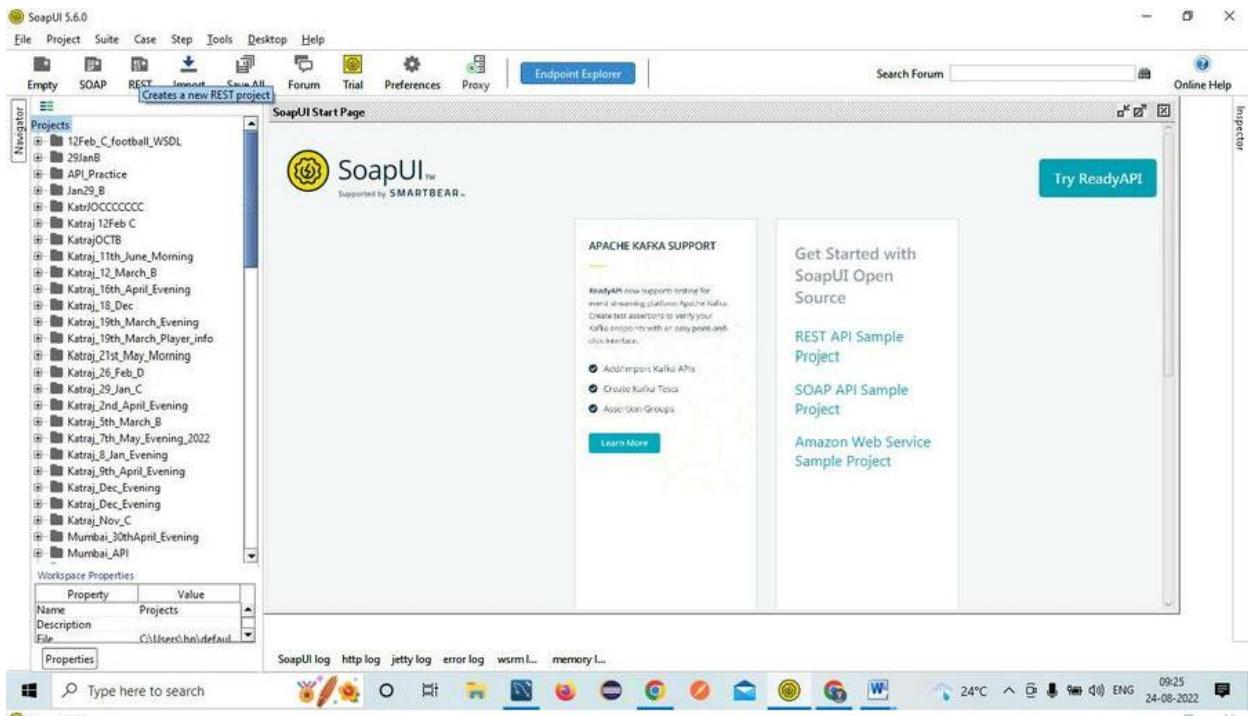
GET Method Service: <https://reqres.in/>

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

Response: 200

## Sample Response:

```
{  
    "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"  
        },  
        {  
            "id": 9,  
            "email": "tobias.funke@reqres.in",  
            "first_name": "Tobias",  
            "last_name": "Funke",  
            "avatar": "https://reqres.in/img/faces/9-image.jpg"  
        },  
        {  
            "id": 10,  
            "email": "byron.fields@reqres.in",  
            "first_name": "Byron",  
            "last_name": "Fields",  
            "avatar": "https://reqres.in/img/faces/10-image.jpg"  
        },  
        {  
            "id": 11,  
            "email": "george.edwards@reqres.in",  
            "first_name": "George",  
            "last_name": "Edwards",  
            "avatar": "https://reqres.in/img/faces/11-image.jpg"  
        },  
        {  
            "id": 12,  
            "email": "rachel.howell@reqres.in",  
            "first_name": "Rachel",  
            "last_name": "Howell",  
            "avatar": "https://reqres.in/img/faces/12-image.jpg"  
        }  
https://reqres.in/#support-heading",  
}
```



**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

**Projects**

- 12Feb\_C\_football\_WSDL
- 29JanB
- API\_Practice
- Jan29\_B
- KatrajOCCCCCCC
- Katraj12Feb C
- KatrajOCTB
- Katraj\_11th\_June\_Morning
- Katraj\_12\_March\_B
- Katraj\_16th\_April\_Evening
- Katraj\_18\_Dec
- Katraj\_19th\_March\_Evening
- Katraj\_19th\_March\_Player\_info
- Katraj\_21st\_May\_Morning
- Katraj\_26\_Feb\_D
- Katraj\_29\_Jan\_C
- Katraj\_2nd\_April\_Evening
- Katraj\_5th\_March\_B
- Katraj\_7th\_May\_Evening\_2022
- Katraj\_8\_Jan\_Evening
- Katraj\_9th\_April\_Evening
- Katraj\_Dec\_Evening
- Katraj\_Dec\_Evening
- Katraj\_Nov\_C
- Mumbai\_30thApril\_Evening
- Mumbai\_API

Workspace Properties

Property	Value
Name	Projects
Description	C:\Users\chen\default...
File	C:\Users\chen\default...

Type here to search

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

Try ReadyAPI

SoapUI Start Page

**New REST Project**

New REST Project  
Creates a new REST Project in this workspace

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

OK Cancel Import WADL... Learn More API Sample API Sample

Amazon Web Service Sample Project

24°C 09:26 24-08-2022 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

**Projects**

- 11th\_June\_REST\_Project
- 12Feb\_C\_football\_WSDL
- 29JanB
- API\_Practice
- Jan29\_B
- KatrajOCCCCCCC
- Katraj12Feb C
- KatrajOCTB
- Katraj\_11th\_June\_Morning
- Katraj\_12\_March\_B
- Katraj\_16th\_April\_Evening
- Katraj\_18\_Dec
- Katraj\_19th\_March\_Evening
- Katraj\_19th\_March\_Player\_info
- Katraj\_21st\_May\_Morning
- Katraj\_26\_Feb\_D
- Katraj\_29\_Jan\_C
- Katraj\_2nd\_April\_Evening
- Katraj\_5th\_March\_B
- Katraj\_7th\_May\_Evening\_2022
- Katraj\_8\_Jan\_Evening
- Katraj\_9th\_April\_Evening
- Katraj\_Dec\_Evening
- Katraj\_Dec\_Evening

Project Properties | Custom Properties

Property	Value
Name	11th_June_REST_P...
Description	

Type here to search

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

Request 1

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

Value Style Level

Required:  Sets if parameter is required

Type:

Options:  Add  Edit

Raw | HTML | JSON | XML

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

24°C 09:27 24-08-2022 ENG

Soap Save All Forum Trial Preferences Proxy | Online Help

SoapUI Start Page

**Request 1**

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

Name	Value	Style	Level
page	2	QUERY	RESOURCE

Raw XML JSON

Required:  Sets if parameter is required

Type:

Options:

A... Headers... Attachment... Representation... JMS Headers... JMS Properties...

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

Properties REST Project

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

Projects

- 11th\_June\_REST Project
  - https://reqres.in
    - Users [/api/users]
  - 12Feb\_C\_football\_WSDL
  - 29JanB
  - API\_Practice
  - Jan29\_B
  - Katraj0CCCCCCC
  - Katraj12Feb\_C
  - KatrajOCTB
  - Katraj\_11th\_June\_Morning
  - Katraj\_12\_March\_B
  - Katraj\_16th\_April\_Evening
  - Katraj\_18\_Dec
  - Katraj\_19th\_March\_Evening
  - Katraj\_19th\_March\_Player\_info
  - Katraj\_21st\_May\_Morning
  - Katraj\_26\_Feb\_D
  - Katraj\_29\_Jan\_C
  - Katraj\_2nd\_April\_Evening
  - Katraj\_5th\_March\_B
  - Katraj\_7th\_May\_Evening\_2022
  - Katraj\_8\_Jan\_Evening
  - Katraj\_9th\_April\_Evening
  - Katraj\_Dec\_Evening
  - Katraj\_Dec\_Evening

Project Properties Custom Properties

Property	Value
Name	11th_June_REST Pr...
Description	

Properties

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

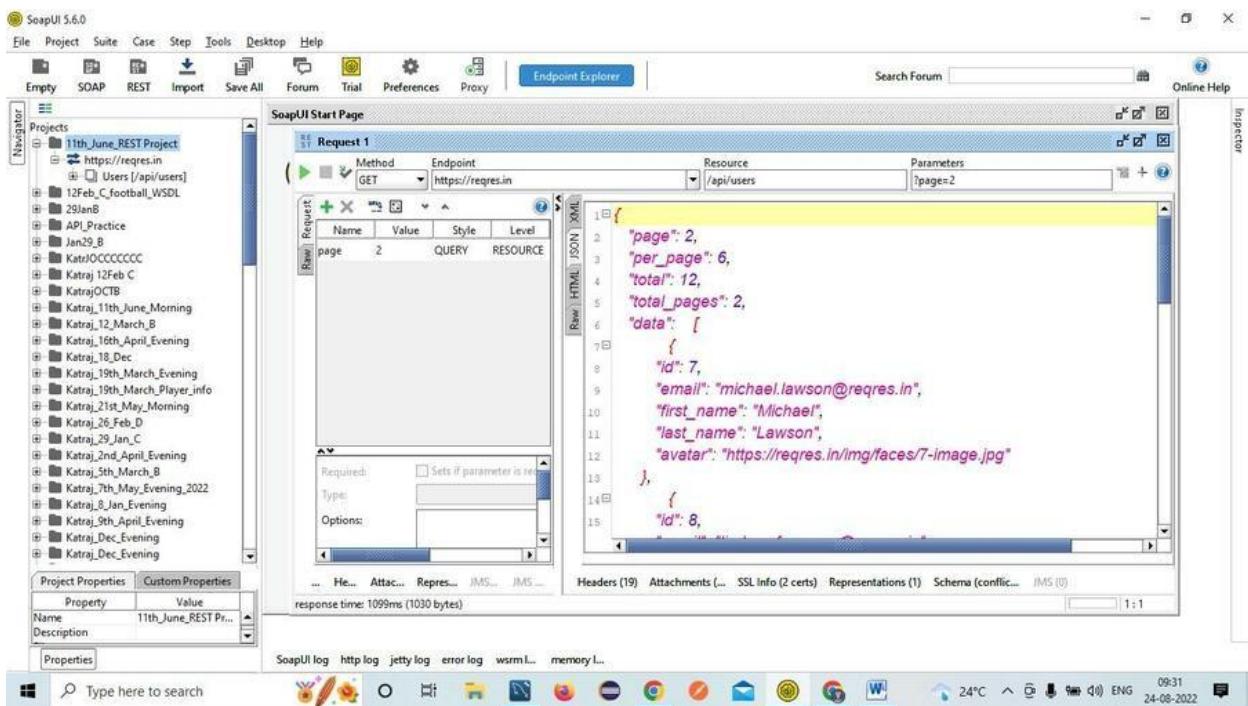
Raw JSON

Headers (19) Attachments (0) SSL Info (2 certs) Representations (1) Schema (conflict...) JMS (0)

response time: 1099ms (1030 bytes)

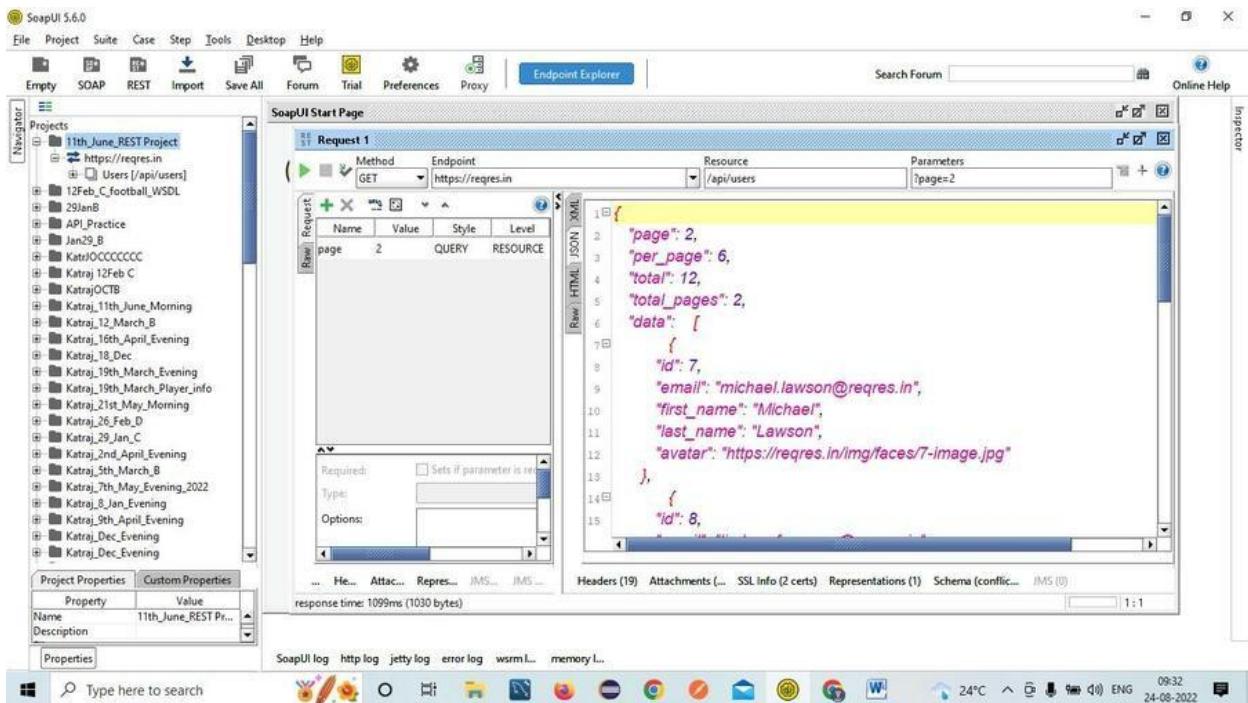
SoapUI log http log jetty log error log wsrm log memory log

24°C 09:30 24-08-2022



## Test Cases:

## 1. Validating the REST Response=Pass



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

The screenshot shows the SoapUI interface with a project named "11th\_June\_REST Project". In the Request tab, a GET request is made to the endpoint `https://reqres.in/api/users?page=2`. The response is displayed in the Response tab as a JSON object:

```

{
  "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=Tag names should contain id,email(first\_name,last\_name,avtar=Pass

The screenshot shows the SoapUI interface with the same project and request setup as the previous screenshot. The response is identical, displaying the JSON object with the specified fields.

#### 4. Validating different status codes present in the response → EC=Status Code=200-Ok=Pass

The screenshot shows the SoapUI interface with a project named "11th\_June\_REST Project". A request is being made to the endpoint `https://reqres.in/api/users?page=2` using a GET method. The response is displayed in the Raw tab, showing a JSON structure:

```

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

The Headers tab shows standard HTTP headers like Content-Type, Accept-Encoding, and Status 200 OK.

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

The screenshot shows the SoapUI interface with a project named "11th\_June\_REST Project". A request is being made to the endpoint `https://reqres.in/api/users?page=2` using a GET method. The response is displayed in the Raw tab, showing a JSON 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"
        },
        {
          "id": 8,
          ...
        }
      ]
    }
  
```

The Headers tab shows standard HTTP headers like Content-Type, Accept-Encoding, and Status 200 OK. The response time is noted as 1099ms (1030 bytes).

#### 6. Applying assertion for verification

The screenshot shows the SoapUI 5.6.0 interface. The left sidebar displays a project named "11th\_June\_REST Project" with various items like "Publish to SwaggerHub", "Generate TestSuite", and "Generate REST Mock Service". The main area is titled "SoapUI Start Page" and shows a "Request 1" configuration for a GET request to "https://reqres.in". The "Resource" is set to "/api/users" and the "Parameters" include "?page=2". The response pane shows a JSON object with two entries, each containing an "id", "email", "first\_name", "last\_name", and "avatar". The "Headers" tab at the bottom is selected, showing the following header details:

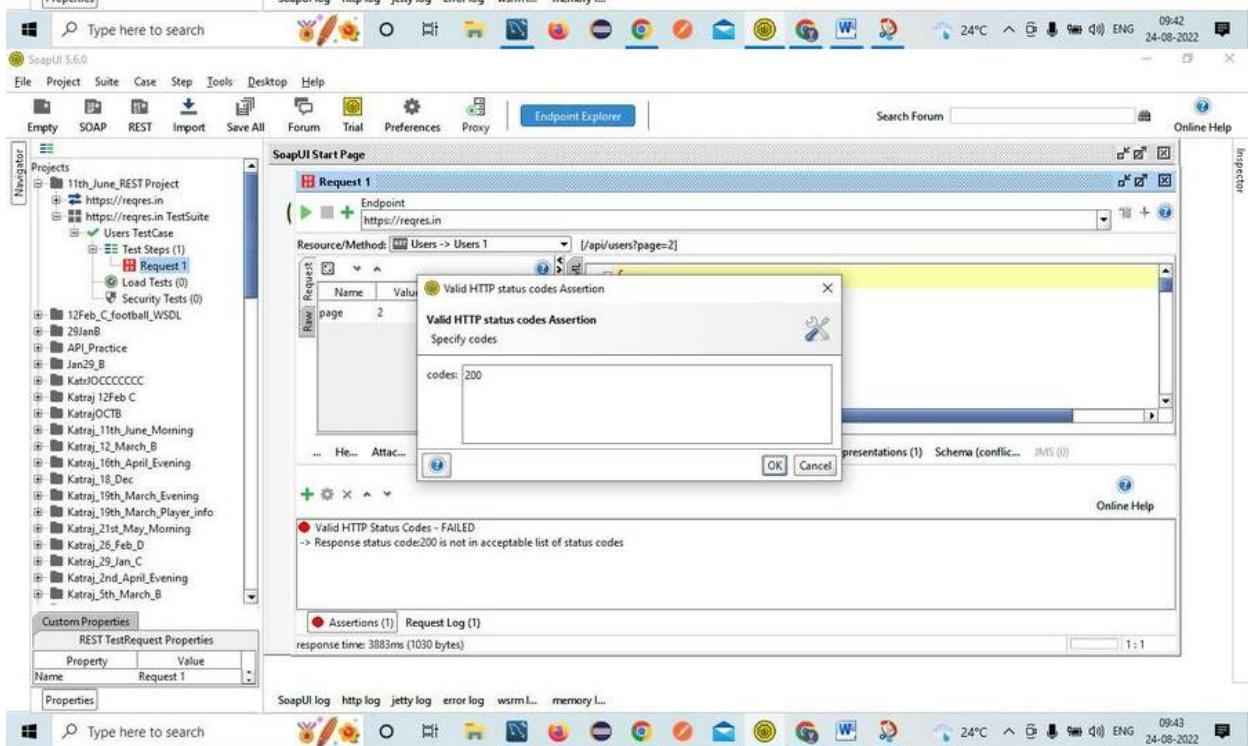
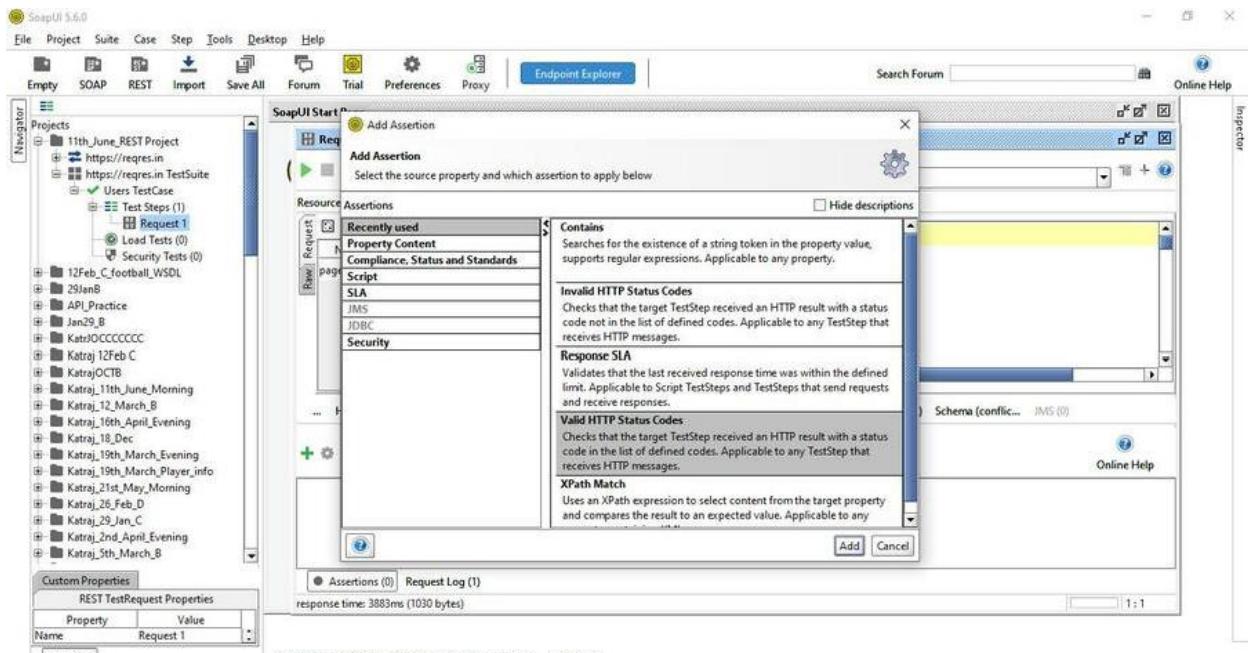
Header	Value
Report-To	[redacted]
Content-Encoding	gzip
Vary	Accept-Encoding
#status#	HTTP/1.1 200 OK
Age	3953
Content-Type	application/json; charset=utf-8
X-Powered-By	Express
Expect-CT	max-age=504800, report-uri="https://report-uri.cloudflare...

At the bottom, the status bar indicates "response time: 1099ms (1030 bytes)" and "1:1".

The screenshot shows the SoapUI 5.6.0 interface. The left sidebar displays a project tree with several test suites and tests. The main workspace shows a successful API request to `/api/users?page=2`. The response body is displayed as JSON:

```
1 {  
2   "page": 2,  
3   "per_page": 6,  
4   "total": 12,  
5   "total_pages": 2,  
6   "data": [  
7     {}  
8   ]  
}
```

Below the response, there are tabs for Assertions (0), Request Log (1), and Headers (19). The status bar at the bottom indicates a response time of 3883ms (1030 bytes).



The screenshot shows the SoapUI interface with the following details:

- Navigator:** Projects - 11th\_June\_REST Project, https://reqres.in TestSuite, Users TestCase, Test Steps (1), Request 1.
- Request 1:** Endpoint https://reqres.in, Resource/Users, Method GET, URL /api/users?page=2.
- Response:** Status code 200 OK, Response body (JSON):
 

```

      {
        "page": 2,
        "per_page": 6,
        "total": 12,
        "total_pages": 2,
        "data": [
          ...
        ]
      }
      
```
- Assertions:** Valid HTTP Status Codes - VALID, Response SLA - VALID.
- Properties:** REST TestRequest Properties, Request 1.

## 7. Validating the 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 response=Pass

The screenshot shows the SoapUI interface with the following details:

- Navigator:** Projects - 11th\_June\_REST Project, https://reqres.in TestSuite, Users TestCase, Test Steps (1), Request 1.
- Request 1:** Endpoint https://reqres.in, Method GET, URL /api/users, Parameters {?page=1}.
- Response:** Status code 200 OK, Response body (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"
          }
        ]
      }
      
```
- Properties:** REST TestRequest Properties, Request 1.

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

The screenshot shows the SoapUI interface with the following details:

- Project:** 11th\_June\_REST Project
- Request:** Request 1 (Method: GET, Endpoint: https://reqres.in/api/users?page=2)
- Response:**

```

    {
        "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"
            }
        ]
    }
  
```
- Properties:** Request 1

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

The screenshot shows the SoapUI interface with the following details:

- Project:** 11th\_June\_REST Project
- Request:** Request 1 (Method: GET, Endpoint: https://reqres.in/api/users?page=3)
- Response:**

```

    {
        "page": 3,
        "per_page": 6,
        "total": 12,
        "total_pages": 2,
        "data": [
            {
                "id": 8,
                "email": "lucy.mitchell@reqres.in",
                "first_name": "Lucy",
                "last_name": "Mitchell",
                "avatar": "https://reqres.in/img/faces/8-image.jpg"
            }
        ]
    }

    support: {
        "url": "https://reqres.in/#support-heading",
        "text": "To keep ReqRes free, contributions towards server costs are appreciated."
    }
  
```
- Properties:** Request 1

8. Validating negative test cases!

8.1 Verify by passing 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 SoapUI interface with a failed API request. The request details are as follows:

- Method:** GET
- Endpoint:** https://reqres.in/api/users
- Parameters:** ?page=1.5

The request body (Raw tab) contains the parameter `page` with the value `1.5`. The response body (Raw tab) shows a JSON object with a yellow cross over it:

```

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",
10:      "first_name": "George",
11:      "last_name": "Bluth",
12:      "avatar": "https://reqres.in/img/faces/1-l.jpg"
13:    }
14:  ]
15: }
  
```

The response headers include:

- Cache-Control: max-age=14400
- NEL
- Report-To
- Content-Encoding: gzip
- Vary: Accept-Encoding
- #status#
- Content-Type: application/json; charset=utf-8
- X-Powered-By: Express

A large yellow 'X' is drawn over the status code area in the header table.

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

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

- Method:** GET
- Endpoint:** https://reqres.in/api/users
- Parameters:** None

The request body (Raw tab) contains the parameter `page` with the value  (empty). The response body (Raw tab) shows a JSON object with a yellow cross over it:

```

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",
10:      "first_name": "George",
11:      "last_name": "Bluth",
12:      "avatar": "https://reqres.in/img/faces/1-l.jpg"
13:    }
14:  ]
15: }
  
```

The response headers include:

- Cache-Control: max-age=14400
- NEL
- Report-To
- Content-Encoding: gzip
- Vary: Accept-Encoding
- #status#
- Age: 2007
- Content-Type: application/json; charset=utf-8

A large yellow 'X' is drawn over the status code area in the header table.

8.3 Verify by passing parameter as Character/String 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 successful JSON response. The request details are as follows:

- Method:** GET
- Endpoint:** https://regres.in/api/users
- Resource:** /api/users
- Parameters:** page=two

The response body is a JSON object:

```

1: {
  "page": 1,
  "per_page": 6,
  "total": 12,
  "total_pages": 2,
  "data": [
    {
      "id": 1,
      "email": "george.bluth@reqres.in",
      "name": "George Bluth Sr."
    }
  ]
}

```

The response headers include:

- Cache-Control: max-age=14400
- NEL: {"success\_fraction":0,"report\_to":"cf-nel","max\_age":604800}
- Report-To: {"endpoints":[{"url":"https://a.nel.cloudflare.com/report?"}]}
- Content-Encoding: gzip
- Accept-Encoding: gzip
- #status#: HTTP/1.1 200 OK
- Content-Type: application/json; charset=utf-8
- X-Powered-By: Express

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 a 404 Not Found error. The request details are as follows:

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

The response message is: "The content you are trying to view cannot be viewed as JSON".

The response headers include:

- Cache-Control: max-age=14400
- 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
- Content-Type: text/html; charset=utf-8
- X-Powered-By: Express
- Expect-CT: max-age=604800, report-uri="https://report-uri.cloudflare...

8.5 Verify by Passing wrong Authorization/API Key/Bearer Token/Basic Auth(Username & Password) in GET Method → EC=Status Code-401-UnAuthorized & GET Method will not show the response=

8.6 Verify by changing GET Method into another method → EC=Status Code-405-Not Allowed & We will not get Response=Pass

The screenshot shows the SoapUI interface with the following details:

- Projects:** 11th\_June\_REST Project, https://regres.in TestSuite, Users TestCase.
- Request 1:** Method: TRACE, Endpoint: https://regres.in/api/users?page=2.
- Content:** The content you are trying to view cannot be viewed as JSON.
- Headers:** Server: cloudflare, CF-RAY: ~, Connection: close, #status#: HTTP/1.1 405 Not Allowed, Content-Length: 155, Date: Thu, 25 Aug 2022 03:43:41 GMT, Content-Type: text/html.
- Request Properties:** Request 1.
- Request Params:** Name: page, Value: 2.
- Log:** response time: 296ms (155 bytes).
- System Tray:** Stop sharing, Hide, 23°C, ENG, 09:13, 25-08-2022.

## REST Terminologies:

**End Point/Base URL/Domain:** 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/PUT/PATCH Method?

## Ans: Using Body Payload

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 Request 1 Request 1 Method POST Endpoint https://reqres.in Resource /api/users Parameters

Request Name Value Style Level

Required Sets if parameter

Media Type application/json

Raw JSON XML

```
1 "name": "Pramod",
2 "job": "Senior QA",
3 "id": "677",
4 "createdAt": "2022-08-25T03:57:49.201Z"
5
6 }
```

Headers (15) Attachments (...) SSL Info (2 certs) Representations (1) Schema (conflict...) JMS (0)

Response time: 439ms (85 bytes)

Properties Request Properties Request Params

Property	Value
Name	Request 1
Description	

Type here to search

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

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

23°C ENG 09:27 25-08-2022

**Request 1**

Method: POST | Endpoint: https://reqres.in | Resource: /api/users | Parameters: [empty]

**Raw Request**

Name	Value	Style	Level
Raw	[redacted]		

Required: [checkbox]

Media Type: application/json

```
{
  "name": "Pramod",
  "job": "Senior QA"
}
```

**Response**

```

1 "name": "Pramod",
2 "job": "Senior QA",
3 "id": "677",
4 "createdAt": "2022-08-25T03:57:49.201Z"
5
6
    
```

Headers (15) Attachments (...) SSL Info (2 certs) Representations (1) Schema (conflict...) JMS (0)

response time: 439ms (85 bytes)

**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**

Navigator: REST Project 22, REST Project 23, REST Project 24, REST Project 25, REST Project 26, REST Project 27, REST Project 28, REST Project 29, REST Project 3, REST Project 30, REST Project 31, REST Project 32, REST Project 33, REST Project 34, REST Project 35, REST Project 36, REST Project 37, REST Project 38, REST Project 39, REST Project 4, REST Project 40, REST Project 41, REST Project 42, REST Project 43, https://reqres.in, Users [/api/users], Users 1, Request 1

**Request 1**

Method: POST | Endpoint: https://reqres.in | Resource: /api/users | Parameters: [empty]

**Raw Request**

Name	Value	Style	Level
Raw	[redacted]		

Required: [checkbox]

Media Type: application/json

```
{
  "name": "Pratiksha",
  "job": "Associate QA"
}
```

**Response**

```

1 "name": "Pratiksha",
2 "job": "Associate QA",
3 "id": "476",
4 "createdAt": "2022-08-25T03:59:01.372Z"
5
6
    
```

Headers (15) Attachments (...) SSL Info (2 certs) Representations (1) Schema (conflict...) JMS (0)

response time: 885ms (91 bytes)

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

SoapUI log http log jetty log error log wsrm... memory L... 23°C 09:29 25-08-2022

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

**DELETE:** Remove the contents from the target resource

23/06/2023:

## Resources:

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

**Google.com/maps**

**Google.com/images**

**Google.com/search**

**www.amazon.in/orders**

### **Parameters: Path & Query**

**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 identified by ID

[\*\*https://www.google.com/images/1123343\*\*](https://www.google.com/images/1123343)

[\*\*https://www.google.com/docs/1123343\*\*](https://www.google.com/docs/1123343)

[\*\*https://www.amazon.com/orders/112\*\*](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\*\*](https://www.amazon.com/orders?sort_by=2/20/2022)

End Point Request URL can be constructed as below:

**URI**=Base URL/Resource/(Query/Path)Parameters

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

### **URL & URI:**

URL – Uniform resource locator

URI- Uniform resource identifier

EX.

[\*\*https://reqres.in/api/users?page=2\*\*](https://reqres.in/api/users?page=2)

where

Base URL/Domain/Endpoint – [\*\*https://reqres.in\*\*](https://reqres.in)

URI- [\*\*https://reqres.in/api/users?page=2\*\*](https://reqres.in/api/users?page=2)

URI contains: URL & other part

contains

1. **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"  
}
```

- Username& Password/API Key/Barrier Token-dasdakfkfbfsas
- Time taken for the response=60secs
- Unit testing documents

## JSON Object:

JSON object literals are surrounded by curly braces {}.

JSON object literals contains key/value pairs.

Keys and values are separated by a colon.

```
{  
    "name": "morpheus",  
  
    "job": "leader"  
}
```

## PUT method/ request-

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

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

- Username & Password/ **API Key/ Barrier token** - tdccbshdnjhgfvbhdhcnd
- **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** - tdccbshdnjhgfvbhdhcnd

- **Time take to response** = 60 sec
- Unit Testing documents

#### **JSON Path Assertion: It is used to validate JSON response**

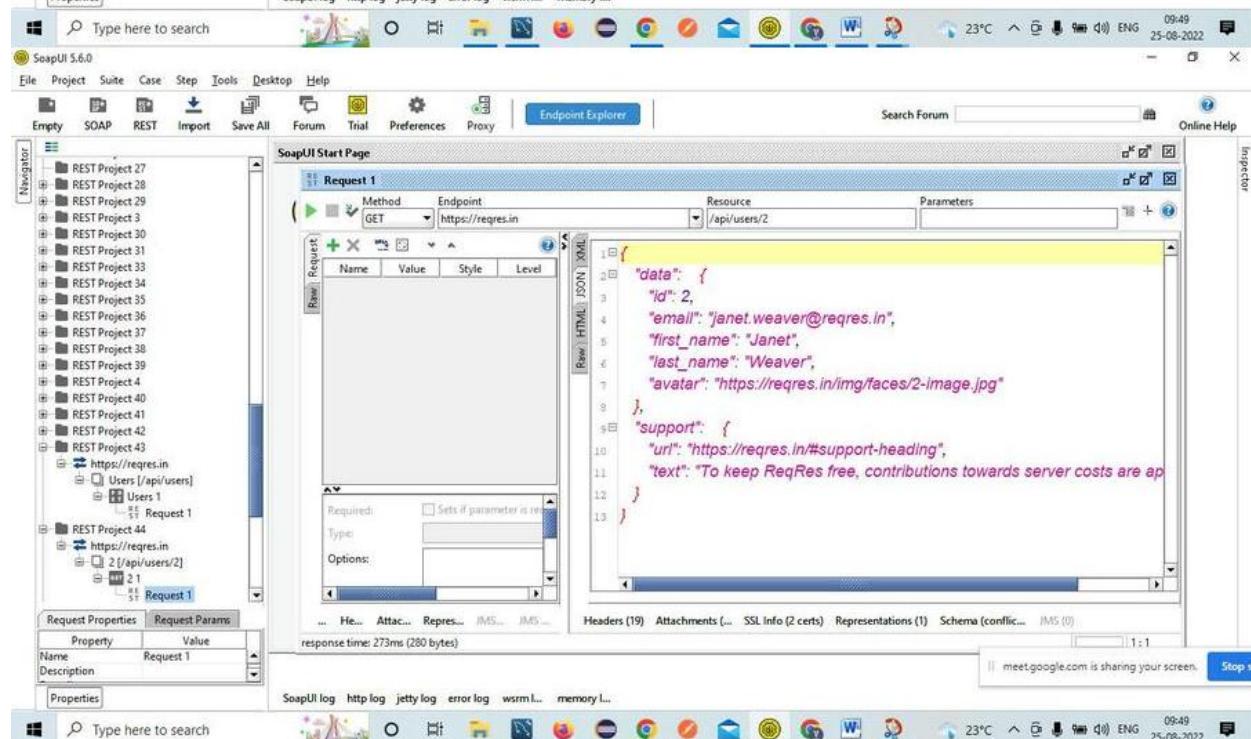
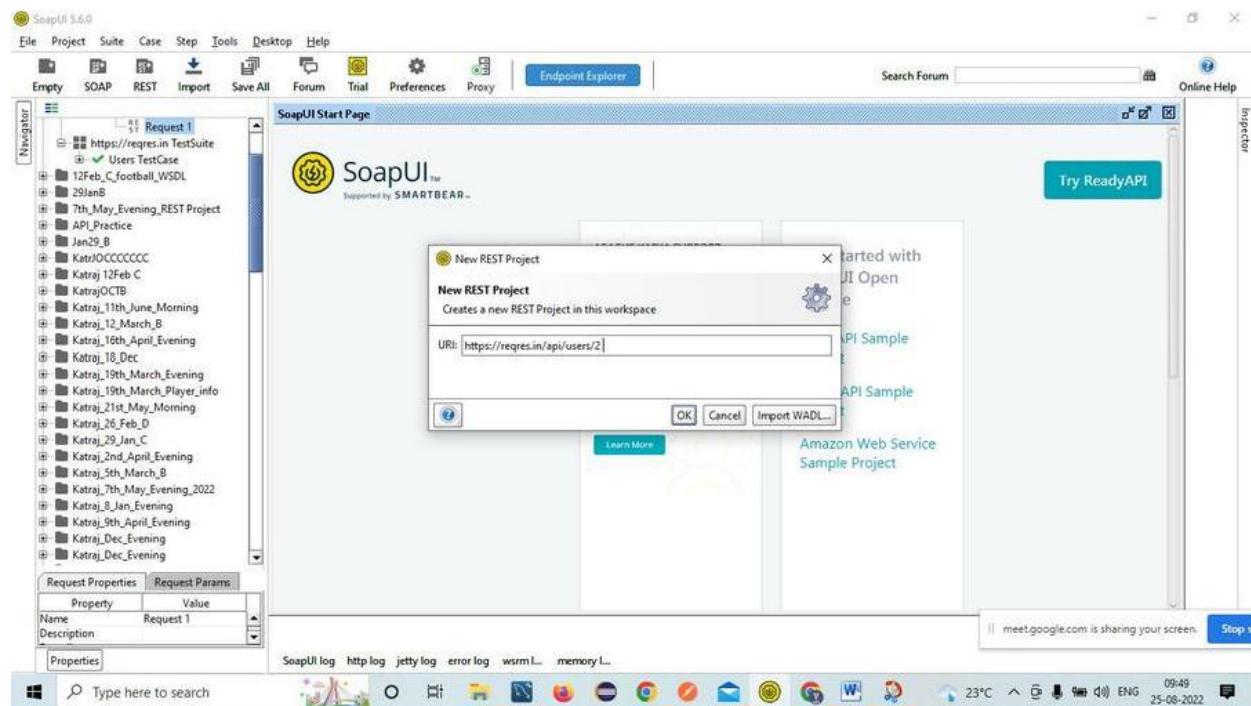
- JSON Path Count
- JSON Path Existance Match
- JSON Path Match

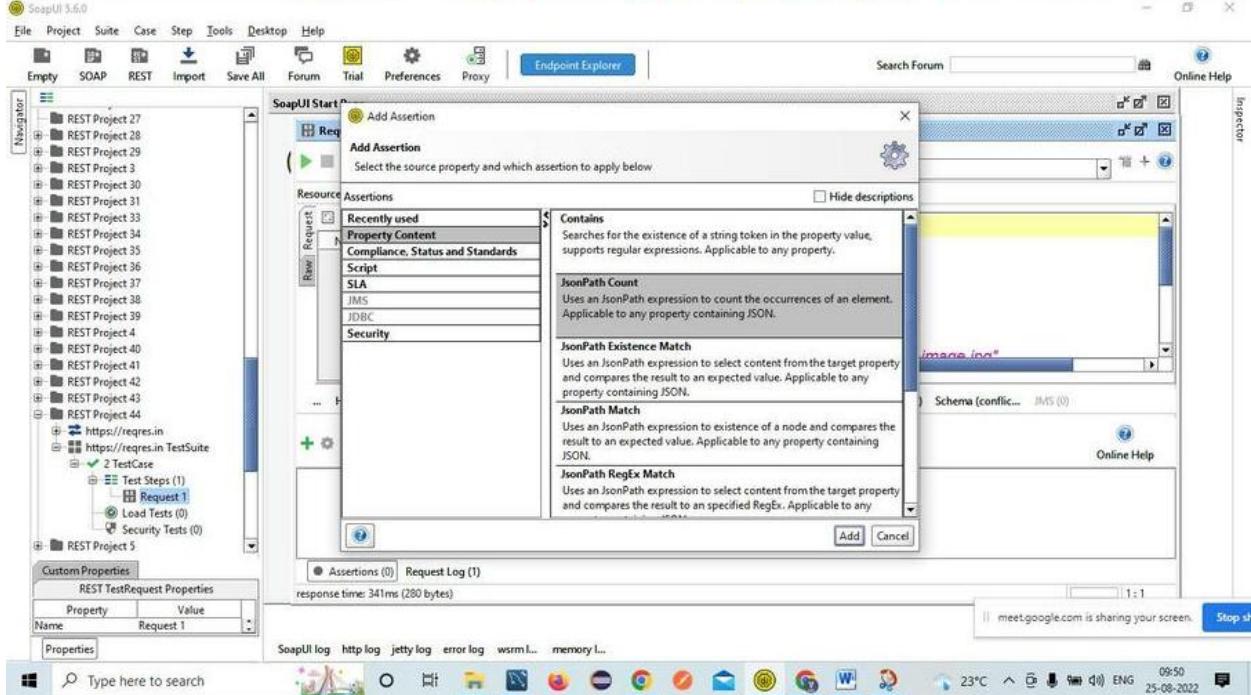
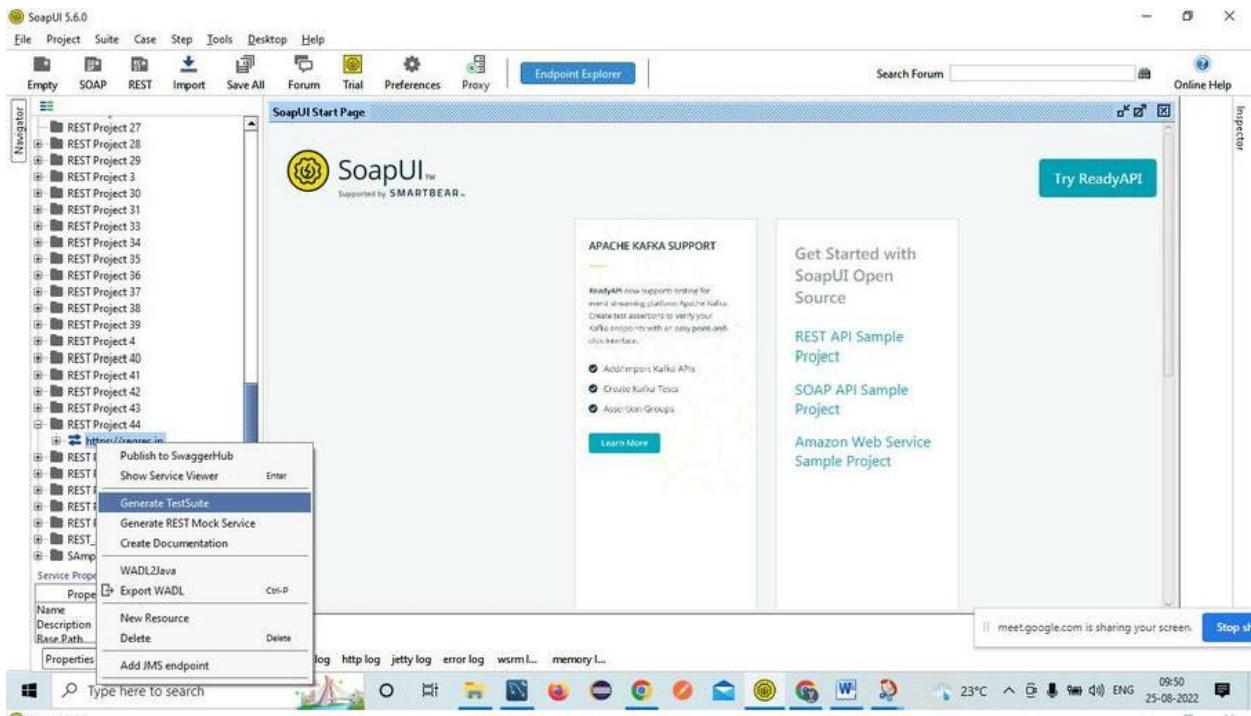
Ex: <https://reqres.in/api/users/2>

Sample Response:

```
{  
    "data": {  
        "id": 2,  
        "email": "janet.weaver@reqres.in",  
        "first_name": "Janet",  
        "last_name": "Weaver",  
        "avatar": "https://reqres.in/img/faces/2-image.jpg"  
    },  
    "support": {  
        "url": "https://reqres.in/#support-heading",  
        "text": "To keep ReqRes free, contributions towards server costs are appreciated!"  
    }  
}
```

## Status Code: 200-OK





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

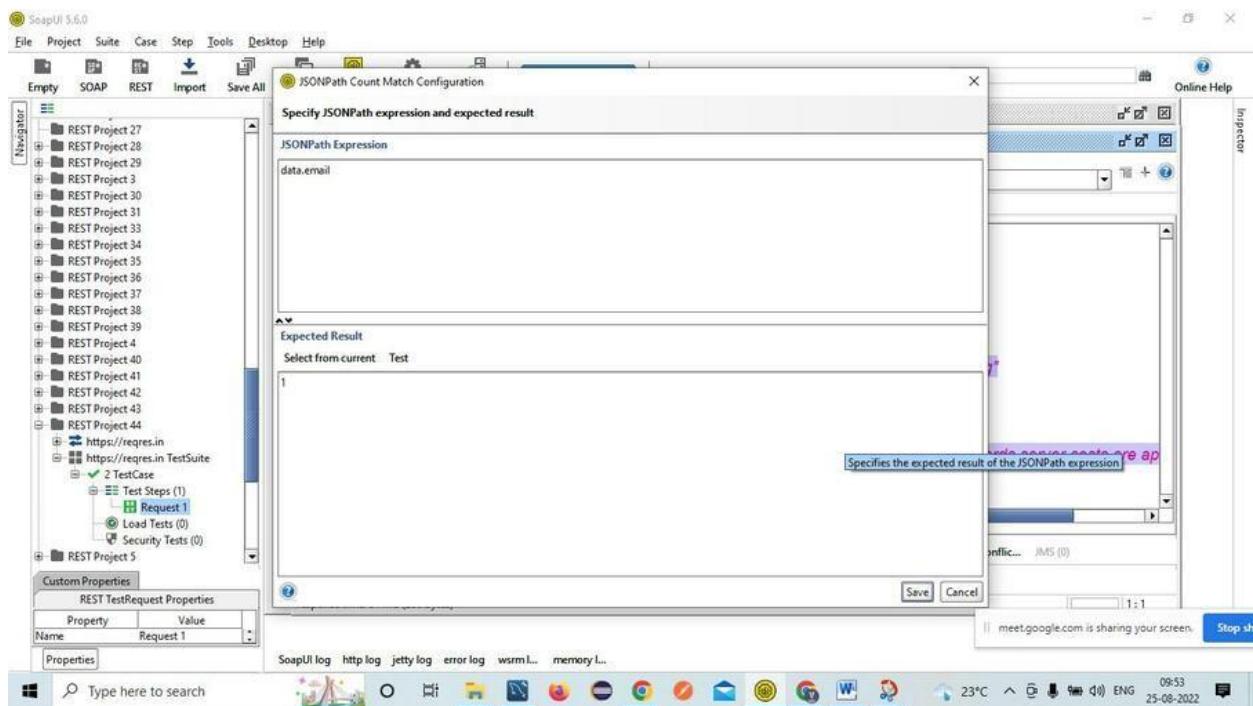
Created by Joe Beach

The screenshot shows a browser window with the JSON Path Finder tool. The left panel is a code editor with the following JSON:

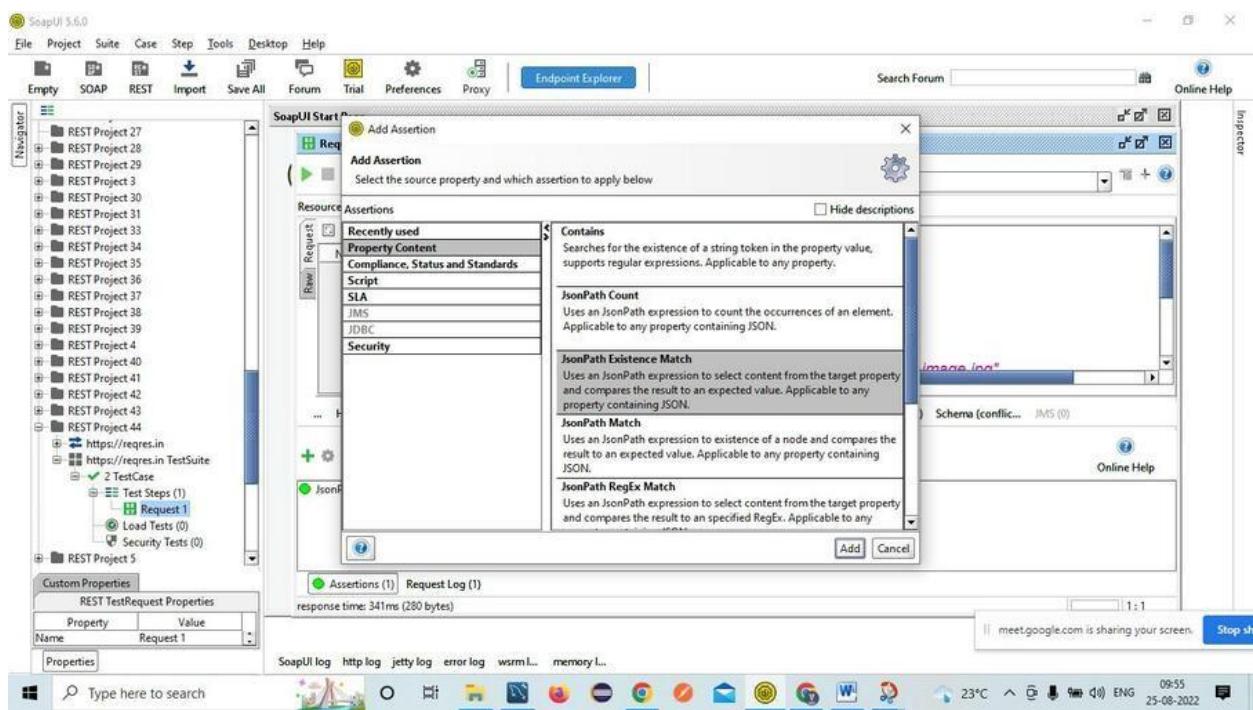
```
1: {
2:   "instructions": [
3:     "Enter your JSON in the editor.",
4:     "Select an item to view its path.",
5:     "Replace 'x' with the name of your variable."
6:   ]
7: }
```

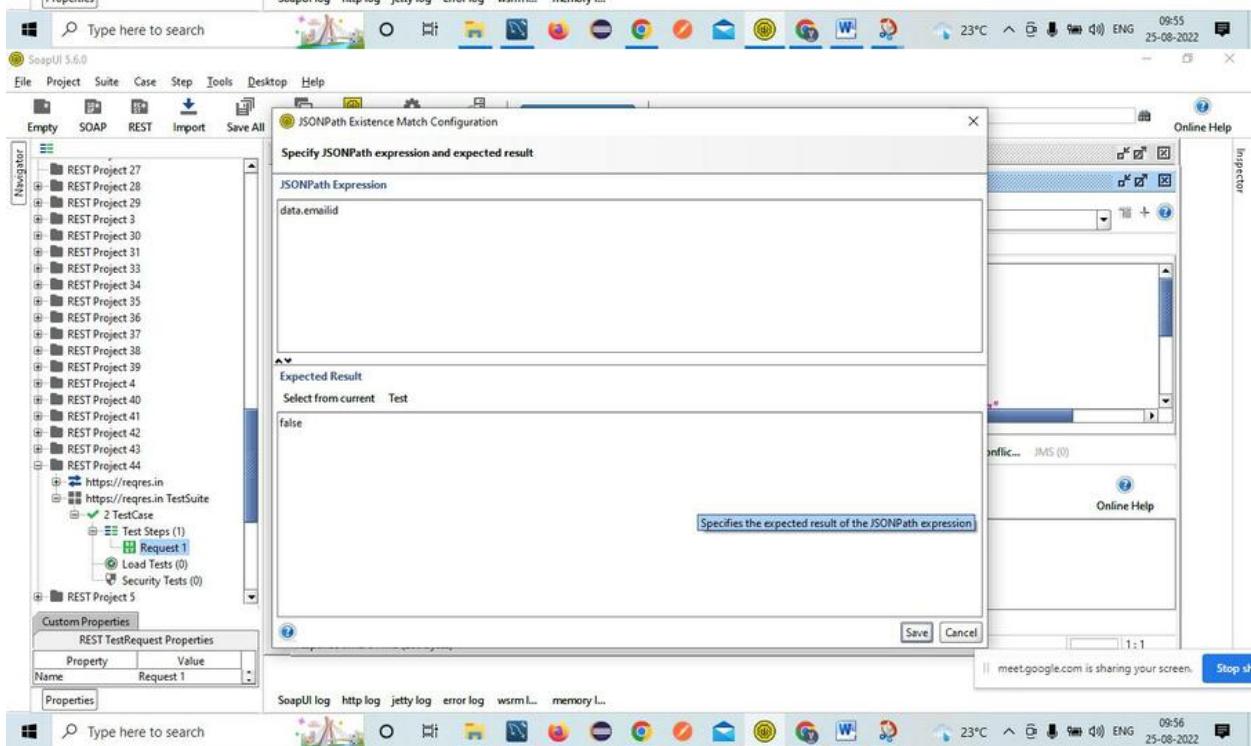
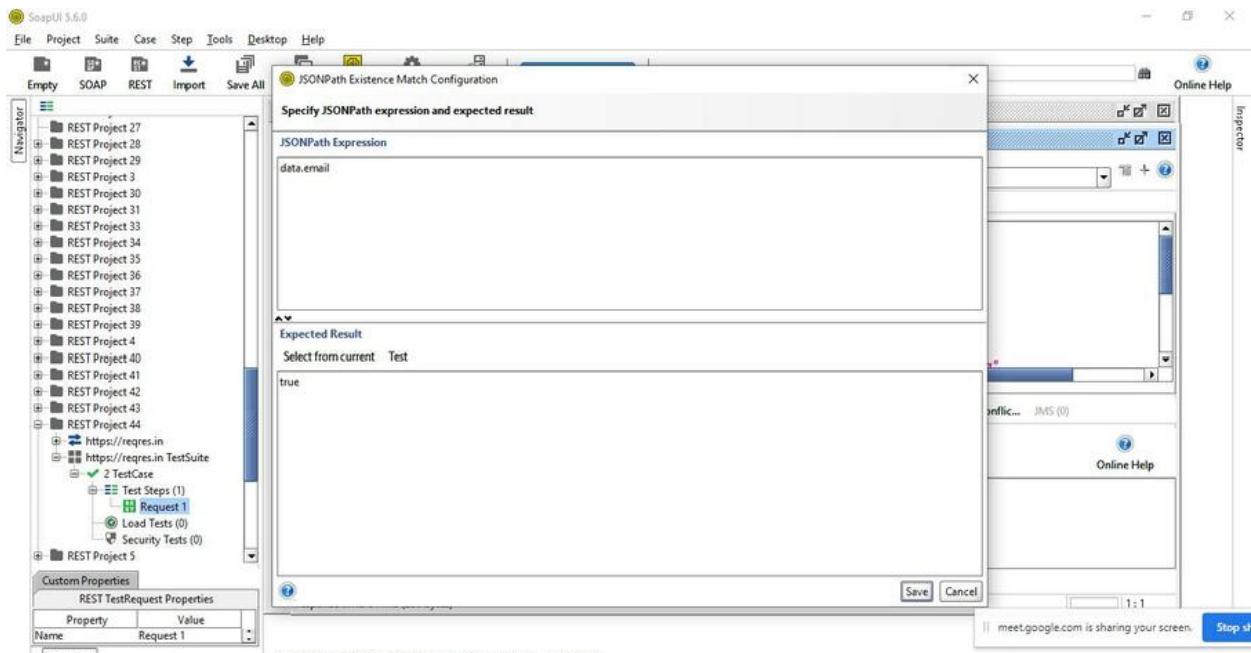
The right panel displays the results of a search for the path `x.data.email`. The results are:

- Path: `x.data.email`
- data:
  - id: 2
    - email: janet.weaver@reqres.in
    - first\_name: Janet
    - last\_name: Weaver
    - avatar: https://reqres.in/img/faces/2-image.jpg
- support:
  - url: https://reqres.in/#support-heading
  - text: To keep ReqRes free, contributions towards server cost

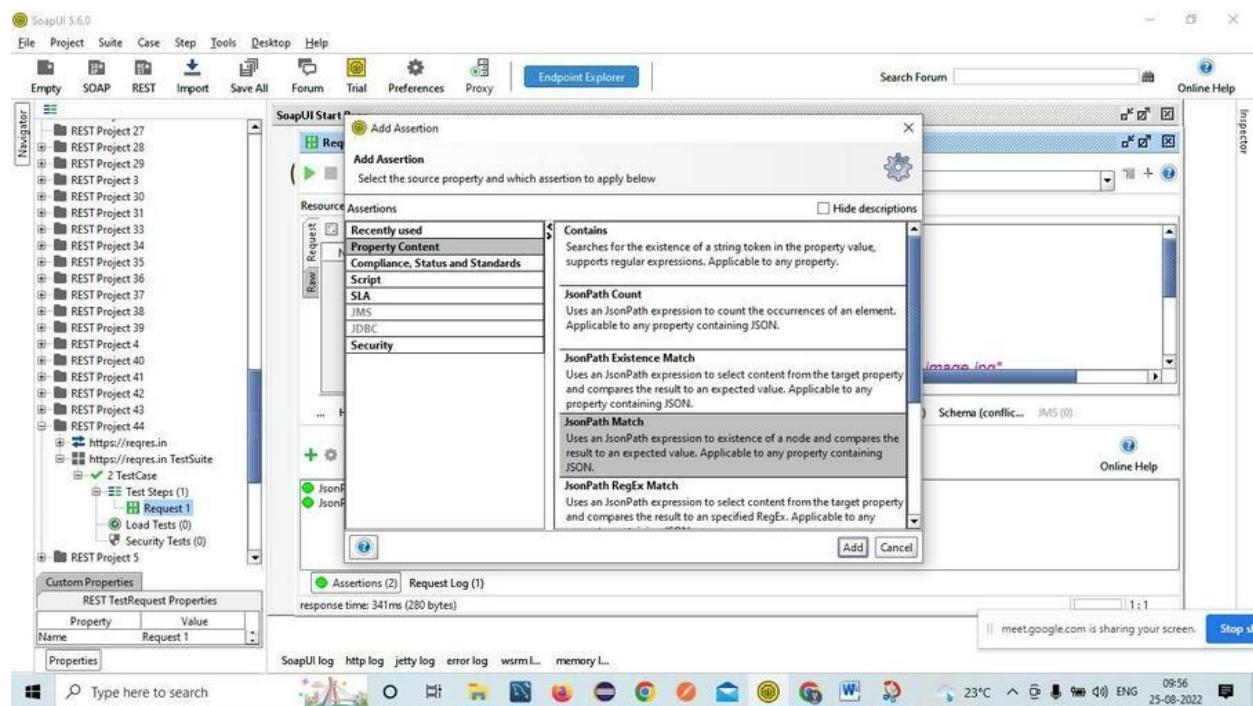


## JSON Path Existence Match:





## JSON Path Match:



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 27 REST Project 28 REST Project 29 REST Project 3 REST Project 30 REST Project 31 REST Project 32 REST Project 33 REST Project 34 REST Project 35 REST Project 36 REST Project 37 REST Project 38 REST Project 39 REST Project 40 REST Project 41 REST Project 42 REST Project 43 REST Project 44 https://reqres.in https://reqres.in TestSuite 2 TestCase Test Steps (1) Request 1 Load Tests (0) Security Tests (0) REST Project 5

Custom Properties REST TestRequest Properties

Name Value

Request 1

Properties

Type here to search

SoapUI Start Page Request 1 Endpoint https://reqres.in Resource/Method: /api/users/2 Raw JSON XML

1 "data": {  
2 "id": 2,  
3 "email": "janet.weaver@reqres.in",  
4 "first\_name": "Janet",  
5 }

JSONPath Match Configuration Specify JSONPath expression and expected result

JSONPath Expression

data.email

Expected Result Select from current Test

janet.weaver@reqres.in

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

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 Inspector

Navigator

REST Project 27 REST Project 28 REST Project 29 REST Project 3 REST Project 30 REST Project 31 REST Project 32 REST Project 33 REST Project 34 REST Project 35 REST Project 36 REST Project 37 REST Project 38 REST Project 39 REST Project 40 REST Project 41 REST Project 42 REST Project 43 REST Project 44 https://reqres.in https://reqres.in TestSuite 2 TestCase Test Load Tests Security Tests REST Project 5

Custom Properties REST TestRequest Properties

Name Value

Request 1

Properties

Type here to search

SoapUI Start Page JSONPath Match Configuration Specify JSONPath expression and expected result

JSONPath Expression

data.email

Expected Result Select from current Test

janet.weaver@reqres.in

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

Online Help

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

# 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**

<b>PUT</b>	<b>Patch</b>
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 API 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. Client/server Error Codes
7. Success codes
8. Use of GET, PUT, POST, PATCH, DELETE
9. Difference between Put and Patch

10. Difference between Put and Post
11. How to handle authentication
12. Json/xml structure format
13. How to validate json response
14. Use of JsonPath
15. How to construct json body for Put, Post, Patch request.
16. Use of JsonObject.
17. What's is URI
18. What's a resource
19. What are the common headers used.
20. Common content types

**Websites:**

<https://reqres.in/>

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

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

**Project-Database Testing-**

**Tables Name in your Project-**

**Stk.Contract\_info** - end user/ investor details will store, CustID, Fname, PAN no., mobile, Email id, Address, etc

**Stk.BOB\_chagre**- Brokerage charge

**Stk.RDOS table**- Buy, Sell stock/share, holding information & Stock name

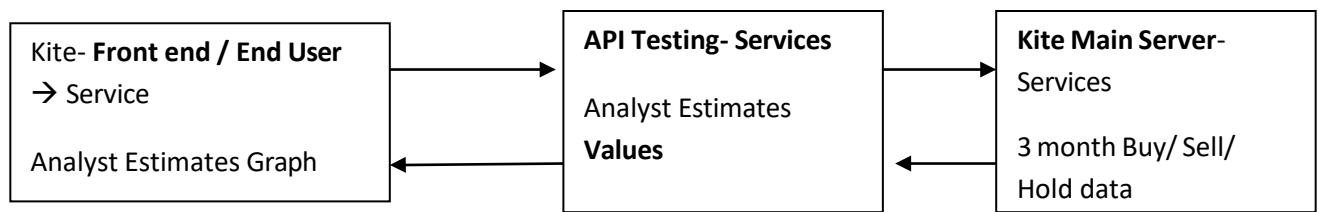
**Stk.ACCTRANS**- Transaction related/ order place information- TrasnctionID

**Stk.TransDate** – Transaction related to bank related date

**Q. Where you have used join quires?**

- **Answer-** In project, I have two table
- **Stk .Contract\_info** - end user/ investor details will store (Cust\_ID= PK)
- **Stk.ACCTRANS** - Customer Transaction related/ order place information- TrasnctionID (Cust\_No =FK)
- Join commands used for –
  - **Ex.**    Select T1.CN1, T1.CN2, T2.CN1, T2.CN2  
 From stk .Contract\_info T1  
 Inner join stk.ACCTRANS T2  
 ON T1.Cust\_ID = T2.Cust\_No;

## API Testing-



### 1. User story- Analyst Estimates values for Stock module form method

**Description-** Analyst estimation chart will show data for user. User will use for suggestion how it is stock moving. Take a holistic view of all the weekly/monthly options and their movement for indices and equities. Real time market movement to help you trade faster with ease. Extremely easy to use with real time tracking of Profit/Loss. Excellent features to help you trade easily in any segment you like.

#### Accept criteria-

1. Estimates **values** present for NITFY 100 (top 100 in share market)
2. Estimates **values** generated depending on Buyer & seller – analysis of last 3 month
3. Estimates **values** is not present for another stock (other than NITFY 100)
4. Estimates values will shows in **% format**, **Tagname- Buy, Sell & Hold**

## **Developer provide-**

1. URL/URI- **GET method/ Request-**

**172.10.21.130:8080/kite.com/api/estimate\_graph?StockName= TCS**

2. API Key- sgfdggdsdfgjhgfgbn
3. Unit Testing – Tables, Screenshot, etc.

```
{  
    "StockName" : "TCS",  
    "BUY" : "88%",  
    "SELL" : "10%",  
    "HOLD" : "2%"  
}
```

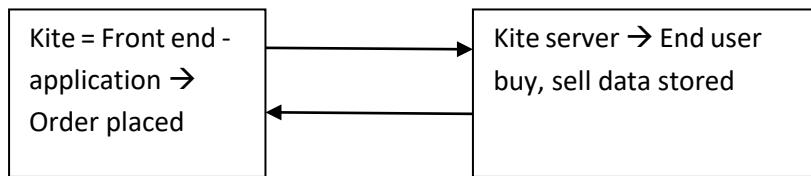
1. Verify Rest response →
2. Verify data and count → EC= Data should be comes in only one
3. Verify Tagname present in GET method → EC= Get method will shows Tagname as Buy, Sell and Hold
4. Verify different status code in GET method → EC= Get method will shows 200-ok
5. Verify time takes into response GET method → EC= Response < 5 sec
6. Verify GET method by passing different Stockname (TCS) → EC= Response should shows data for Buy, Sell & Hold for stockname & Status code = 200-ok
7. Verify GET method by passing different Stockname (HDFC) → EC= Response should shows data for Buy, Sell & Hold for stockname & Status code = 200-ok
8. Verify GET method by passing different Stockname (Coil india) → EC= Response should shows data for Buy, Sell & Hold for stockname & Status code = 200-ok
9. Verify GET method by passing different Stockname (Apllo Hospita) → EC= Response should shows data for Buy, Sell & Hold for stockname & Status code = 200-ok
10. Verify GET method by passing different Stockname (SBI) → EC= Response should shows data for Buy, Sell & Hold for stockname & Status code = 200-ok
11. Verify GET method by passing different Stockname other than NIFFT 100 (IRCTC) → EC= Response should not shows & Status code = 400-Bad request
12. Verify GET method by passing different Stockname other than NIFFT 100 (Trident) → EC= Response should not shows & Status code = 400-Bad request
13. Verify GET method by passing null/blank data → EC= Response should not shows & Status code = 400-Bad request

14. Verify GET method by passing wrong URL/URI → EC= Response should not shows & Status code = 404-Not found
15. Verify GET method by passing wrong API Key → EC= Response should not shows & Status code = 401-Un-authorized
16. Verify by changing method GET method into another method → EC= Response should not shows & Status code = 405-Mentod not allowed

## 2. Service - Portfolio updates notification

### Get Method/URI-

172.20.30.120:8080/account.kite.com/Portfolio/apl/notification/cust\_id=101356



1. Verify Portfolio updates notification API by sent valid cust\_id=101356
  - POSTMAN tool hit Get Method/URI
  - We got data in JSON in POSTMAN tool console/ body
  - Data return from API, for validation we throw database quires (Select \* from **Stk.Contract\_info** where cust\_id = 101356)
1. Verify GET metod by passing different cust\_id → EC = response should shows data for cust\_id and statues code = 200-ok

