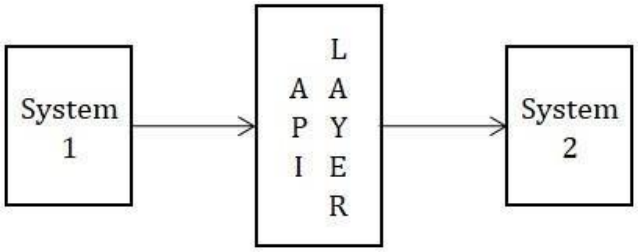
Learn API testing with Mr. Kapil Sir

API- Application Programming Interface.

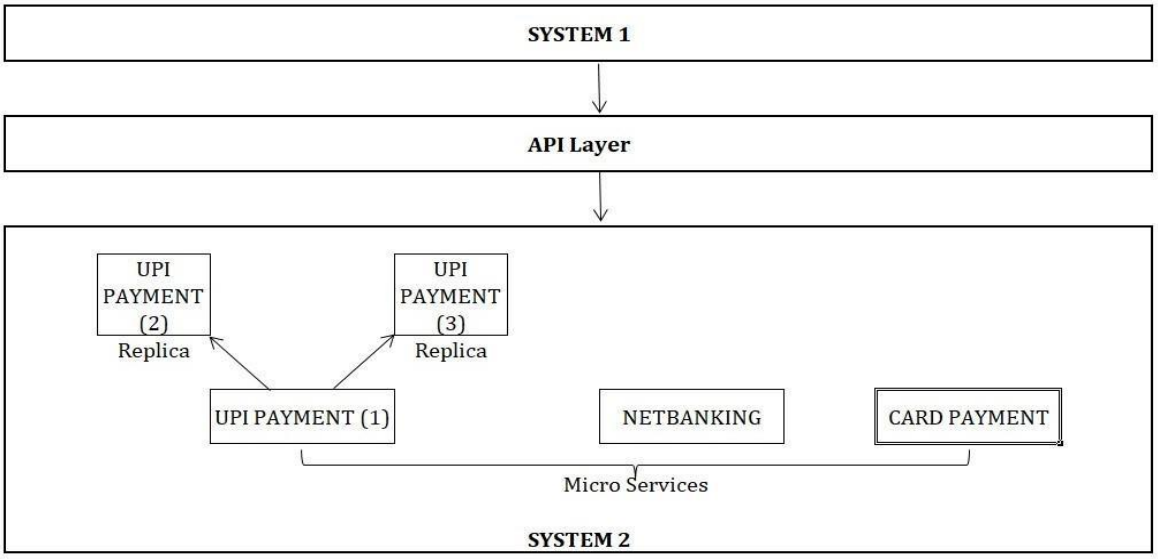
- API is use to communicate between two systems.
- It is simply know as sending the request from one system to another system and getting the required response.
- For Ex. Communication between IRCTC and OTP. Here we send request from IRCTC App to get the response as OTP.

Advantages of API-

- **API provides the security.**
  While communicating between two systems, API will provide API layer in between them which helps to secure out data.



- **To avoid the data breaching.**
  As we know API provides the API layer for security purpose, So it also avoids Data hacking or breaching.
- **To increase the performance and Balances the load.**



In API, Developer already created a Replica's of Micro cervices which helps to increase the performance of the system by sharing the request with replicas in case of load on system 2 increases. Here API will decide how much load to be shared with each micro service and replica. As this balances the load, the performance will get automatically increases.

- **API helps in Data hiding.**
  API helps to hide the data also.
- **API helps for Proper communication between two systems.**
  When first system is sending the request to second system, then the request should go for second system only not the third one. For ex. User wants to do a payment for Amazon order by using Gpay app. Then the payment request should go for Gpay only. This is nothing but the proper communication. In between this communication process, API will provide API layer for Security purpose.
- **API also checks and authenticates the data which we are passing.**
- **API tests core functionality.**
- **API is time effective.**
  We can hit lot of API's at a time with less time.
- **Language Independent.**
  API is Language independent i. e API reads multiple languages like XML, JSON, HTML, TEXT etc.
- **Easy interaction with GUI.**

TYPES OF API-

1) **REST API/SERVCES-** Uses POSTMAN tool(Representational state transfer)
2) **SOAP API/SERVCES -** Uses SOAPUI tool(simple object access protocol)

SOAP	REST
SOAP is Protocol.	REST is Architecture.
Uses XML only.	Uses XML, JSON, HTML, TEXT, JavaScript etc
SOAP reads WSDL file.	REST reads API only.
Type of security provided by SOAP is SOAP Enviornment.	Type of security provided by REST is AUTH Token, HEADER, PARAMS etc.
Heavy in weight API.	Light in weight API.
Response time is more.	Response time is less.
Not best for CRUD operations.	Best for CRUD operations.

CONCEPTS UNDER REST:

REST- REST is an architecture used to create rest API *(To create Rest API we need a REST).*
REST Assured- To automate rest API we need rest assured libraries. *(For Ex. to automate web browser UI we need External JAR files in Selenium)*

RESTFUL- when we automate rest API successfully it called as restful services.

Webservices:

Whenever we are calling any API over http (internet) protocol it called as **Webservice.**

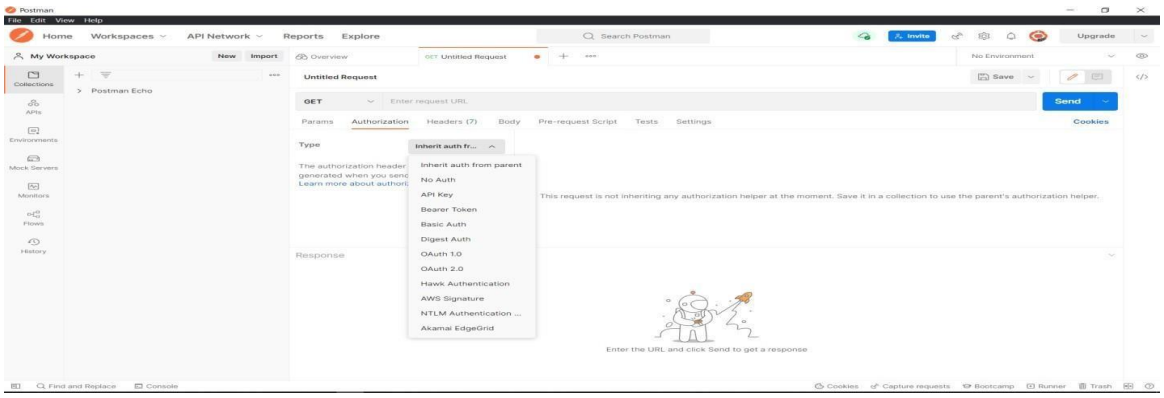
DIFF. BETWEEN REST & WEBSERVICES-

- API call without internet or over the internet and webservices call over the internet.

- The only **difference** is that a **Web service** facilitates interaction **between** two machines over a network. An **API** acts as an interface **between** two **different** applications so that they can communicate with each other. **Web service** also uses SOAP, REST, and XML-RPC as a means of communication.

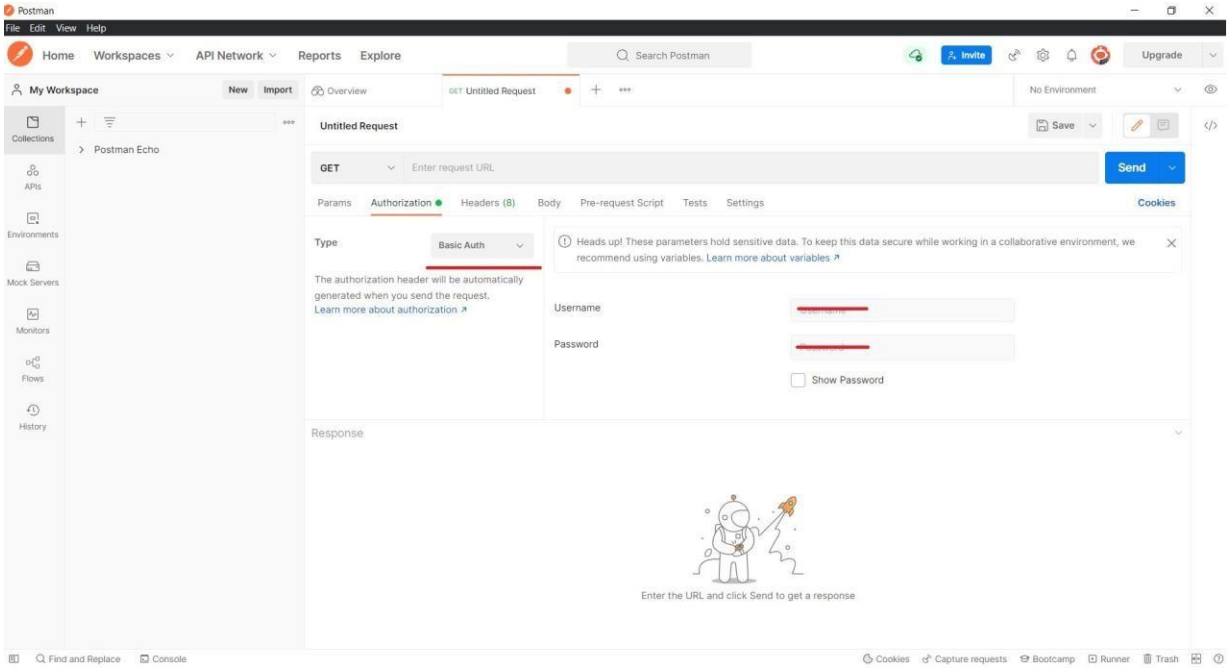
DIFF. TYPES OF AUTHORIZATIONS-

- Basic Auth
- Digest
- Token
- OAuth1
- OAuth2
- No auth
- AWS signature



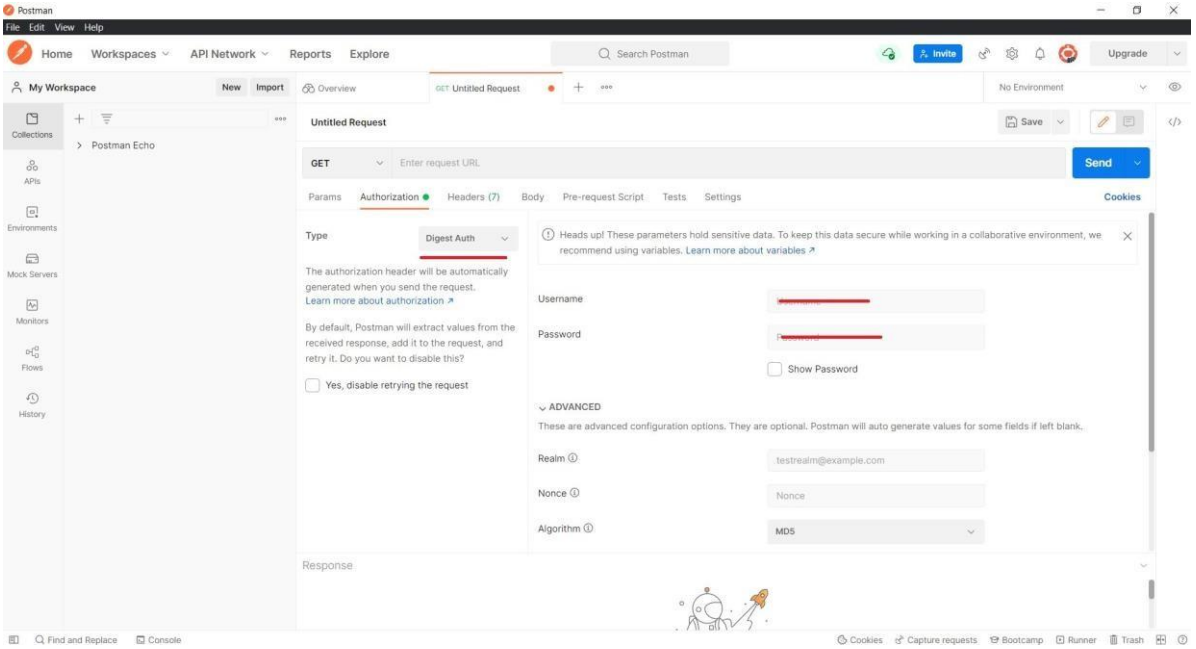
1) Basic Auth-

In Basic we have to pass **only Username and Password.**



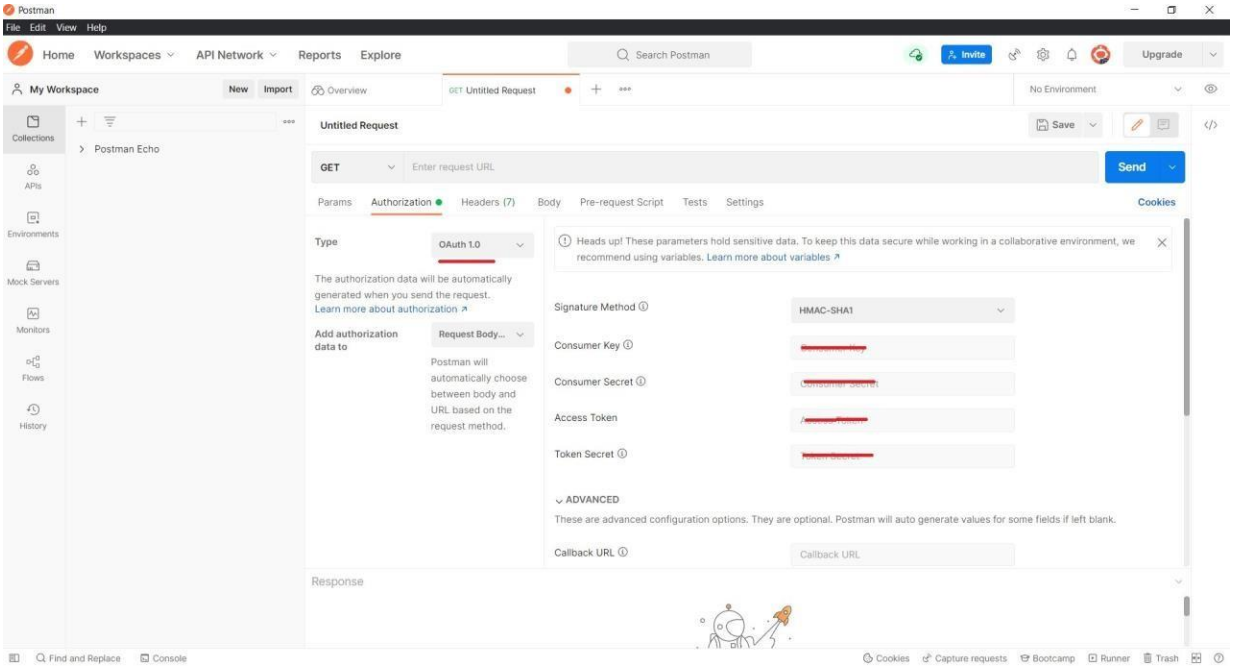
2) Digest Auth

In Digest auth we have to pass **Only Username and Password same** as that of Basic Auth but Digest Auth is more secure than Basic auth.



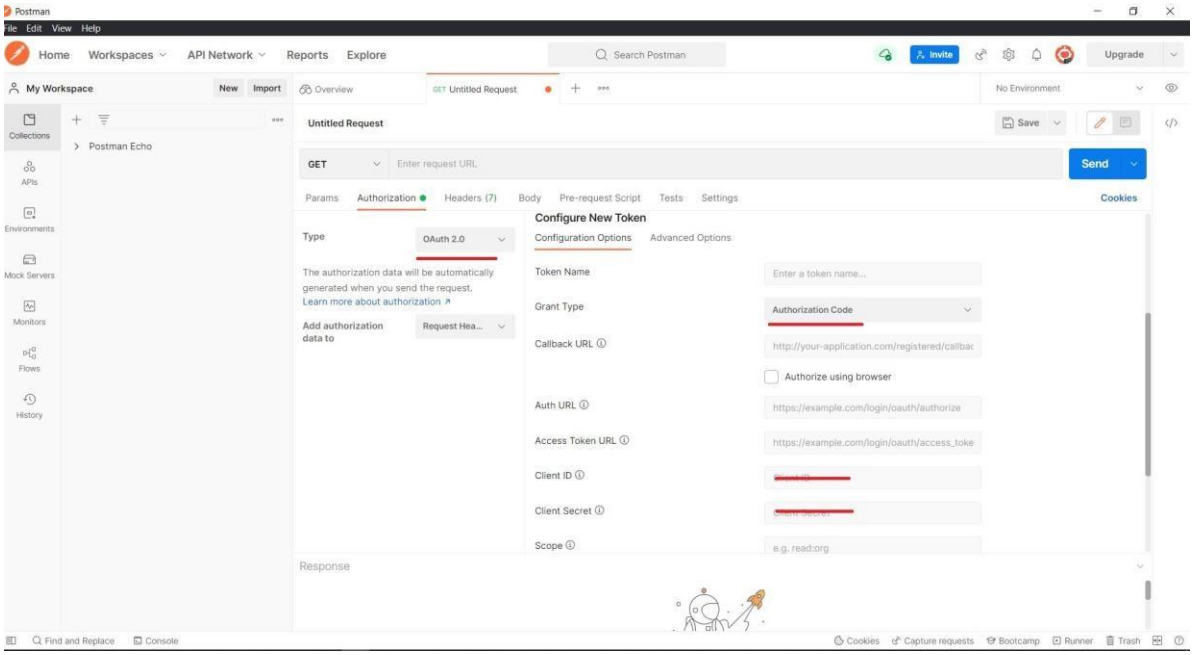
3) OAuth 1.0

In OAuth 1.0 we have to pass have to pass Consumer Key, Consumers secrete, Access Token, Token Secrete. All this details will be provided by developer side.



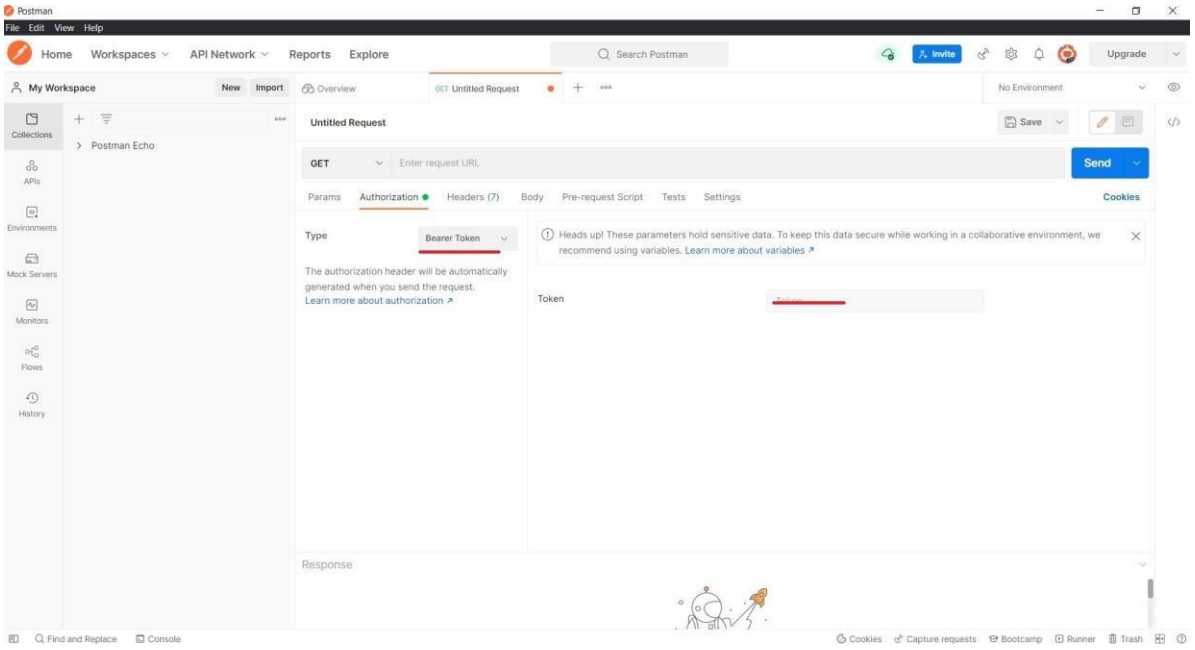
4) OAuth 2.0

In OAuth 2.0 we have to pass have to pass Grant type, Client Id, Client Secrete. All this details will be provided by developer side.



5. Token-

Here we have to just put the value of **token**. Token value may be the combination of integer and character values. While entering the token value, we need to mention **Bearer** keyword. Bearer means the identification for that token. **Token is mostly used type of authorizations.**

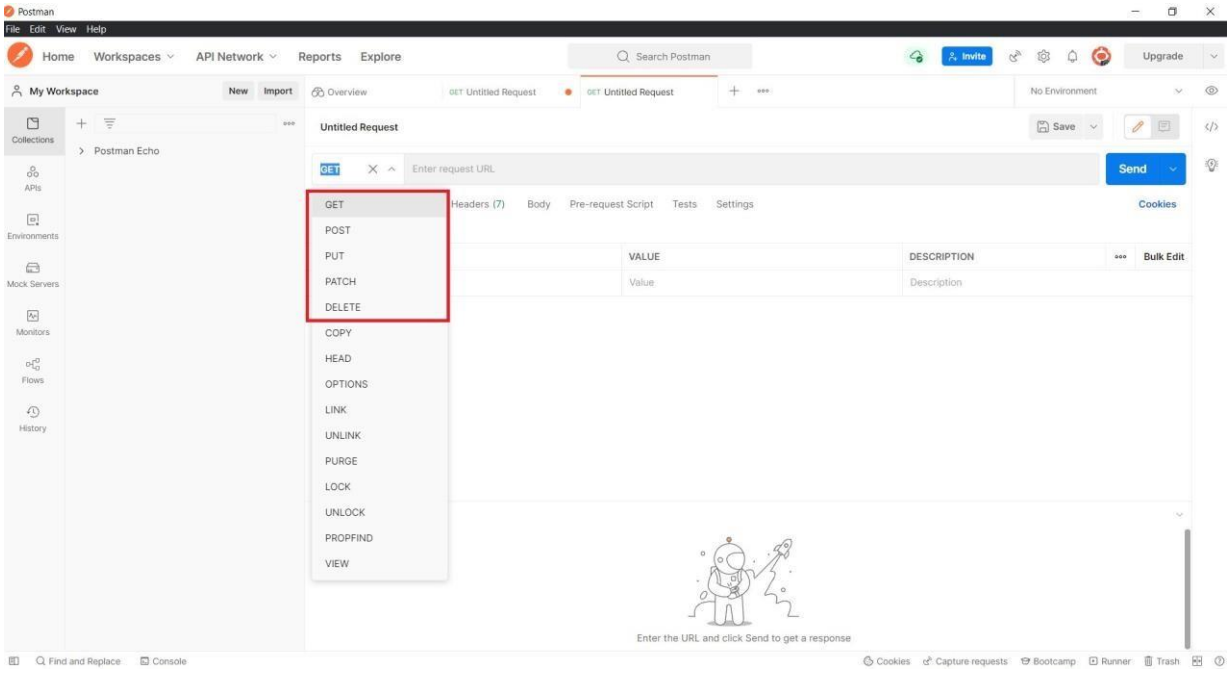


CRUD OPERATIONS-

C-Create the Data- **POST**
R- Retrieve/ Fetch the Data-**GET**
U- Update the data- **PUT**
D- Delete the data- **DELETE**

There are four different methods present in API which are-

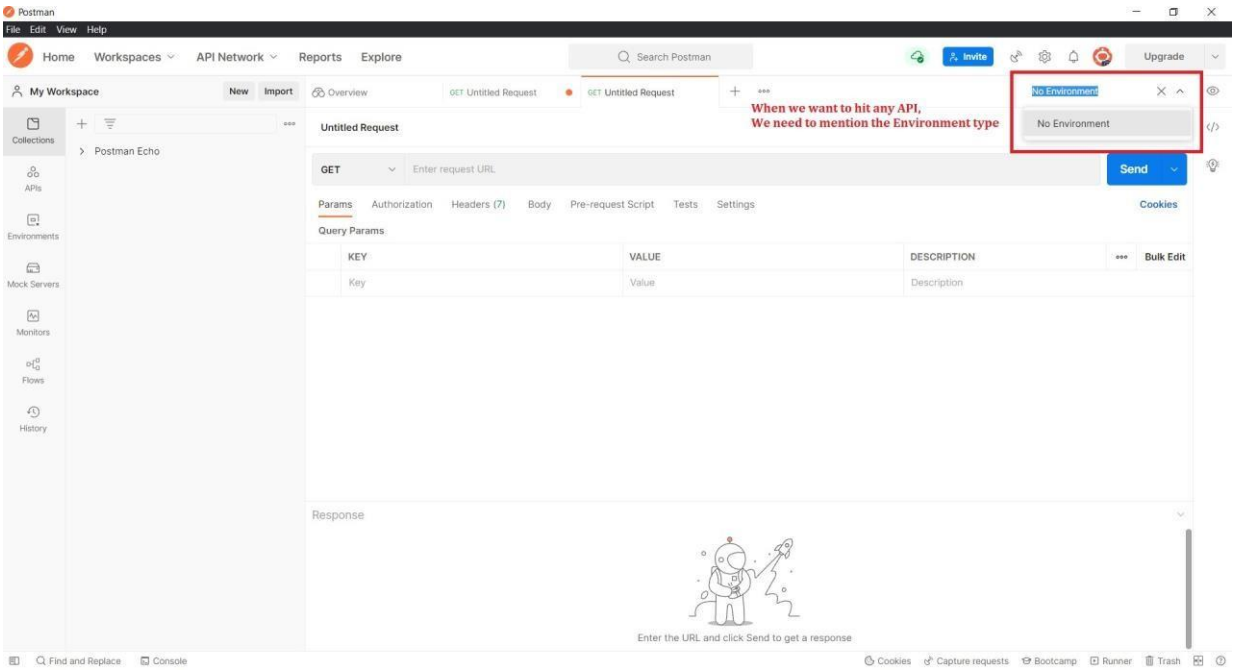
1) **GET-** Used to fetch the data.
2) **POST-** Used to create the data.
3) **PUT-** Used to update the data.
4) **DELETE-** Used to delete the data.



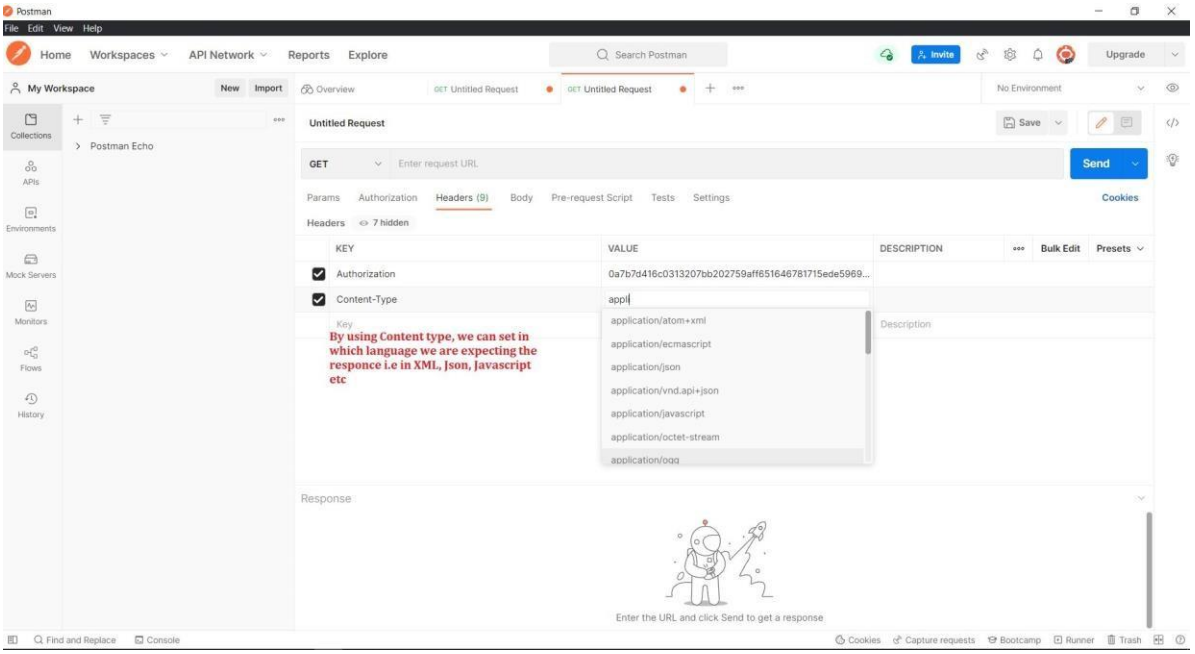
ERROR/STATUS codes-

1) 201- Created || when we create data
2) 200- OK || when we get Successful data.
3) 400-Bad Request || when URL wrong or end point missing.
4) 401- Unauthorised ||when session got expired, passing invalid token/ username/pass.
5) 403- Forbidden
6) 404-Page not found. || when we are trying to access the URL but URL not present
7) 500- Internal server Error || when any server down or network issue.
8) 503-Service not available

Test Environment-



Content-type-



Which type of Authorization is mostly used in organization? Why?

Token Authorization is mostly used in any organization because if it expires then we can easily regenerate the new token by simple clicking on regenerate the token. Token expiry limit will be decided by Developer. Token also provides more security.

WEBSERVICES-

Whenever we are hitting any service over the internet it known as **webservice**. Webservice is any piece of software that makes it available over the internet and uses a standardized XML messaging system.

WSDL-

WSDL stands for Web Service Descriptions Language. WSDL is basically an XML document contains all the details about web services & all API requests.

UDDI-

UDDI stands for Universal Description Discovery integration. UDDI is an XML based standardfor describing, publishing and finding the web services.

SOAP ELEMENTS-

- Envelop- It is beginning and end of message.
- Header- Header elements contain header information.
- Body- Body element contains call & response information.
- Fault- Fault contains error and status information.

WSDL ELEMENTS:

- Type- Define the data types used by the web services
- Message – Define the data element for each operation
- Port Type- Describe the operation that can be performed and message involve
- Binding- Fault contains error and status information.

WSDL ELEMENTS:

- Type- Define the data types used by the web services.
- Message – Define the data element for each operation.
- Port Type- Describe the operation that can be performed and message involve.
- Binding- Defines the protocol and data format for each port type.

Diff assertion

content- to verify string
not content- same as above but opposite
valid http status code- to verify status code
invalid http status code- same as above but opposite
response SLA- to verify response time

see the below page

SOME INTERVIEW Q&A

1. **What are the different methods present in API?**

   There are different methods present in API:

   GET

   POST

   PUT

   DELETE

   PATCH

2. **What are different operations performed in API?**

   Below are the operations performed in API:

   GET- used to fetch the data

   POST- Used to create data

   PUT- Used to update data

   DELETE- used to delete data

3. **What is difference between PUT and PATCH?**

   PUT- we can update all the fields as well as single field

   PATCH- we can update single/ partial fields

4. **What are main differences between API and Web Service?**

   Api call internally and webservices call over the internet

   The only **difference** is that a **Web service** facilitates interaction **between** two machines over network. An **API** acts as an interface **between** two **different** applications so that they can communicate with each other. **Web service** also uses SOAP, REST, and XML- RPC as a means of communication.

5. **What are the advantages of API Testing?**

   a) Api provides the security.

   b) API checks the authentication and the data that we are passing.

   c) Can transfer the load to diff micro services.

d) API helps to avoid data breaching.

e) Test for Core Functionality.

f) Time Effective- we can hit lots of APIs within less time.

g) Language-Independent- like Json, XML, html, text.

h) Easy Integration with GUI.

6. What is different Test Environment in project?

- Generally we will have below four test Environments:

DEV- where developers works.

SIT/QA- where Testers works.

UAT- where Testers and Client works.

PROD- It's a live environment.

7. What are the test environments of API?

Global- Global has large scope (used to pass variables between diff collections)

Local – Local has small scope (Used to pass variable from one request to another)

-we are using QA/UAT environment in which we are using Global and Local environment for API methods.

8. What must be checked when performing API testing?

Error codes, data which are coming (Retrieval data), Time.

9. What are tools could be used for API testing?

Postman

Swagger

SoapUI

Etc.

10. What are differences between API Testing and UI Testing?

API doesn't provide the GUI (Graphical User interface) but UI provides.

11. What are common API errors that often founded?

These are the common error getting during API testing

201-created

200-ok

400-Bad request

401-Unauthorised

403- Forbidden

404- Page not found

500- Internal server error

503-service not available

12. Any examples why error code generates?

200- When we get successful data.

201- When we create data into database.

400- URL wrong or end point missing.

401- When session got expired, passing invalid token/ username/pass.

404- When we are trying to access the URL but URL not present.

405- Method not allowed.

500- Any server down or network issue.

13. What are the collections?

Collections are used to store the services (API methods)

By using collection we can run all the methods at the same time.

We can Import/Export Collection.

14. What is mean bearer token?

Bearer token is one of the Authentication pass in headers

Bearer means identification for the token.

15. Where we pass the data in post?

We pass the data in Body-> Raw-> in the form of Json, XML. Html, text

16. Can we run collection?

Yes, we can run the collection and collection methods at the same time, but before we run the previous or old collection we have to update the authentication.

17. What is mean by end points/service URL?

End points are the different service URLs which are used to hit the URL with domain URL.

18. What is mean API?-Application programming interface

API stands for Application programming interface.

- Used to communicate between two systems.

- It simply knows as sending the request and getting the response.

19. What are headers?

Headers is nothing but the what kind of request it is

{content-type= application json/

application xml/

application text }

20. What is bearer?

Bearer is the identifier for particular token used for the Authentication.

21. Difference between SOAP and REST

SOAP	REST
SOAP is Protocol.	REST is Architecture.
Uses XML only.	Uses XML, JSON, HTML, TEXT, JavaScript etc
SOAP reads WSDL file.	REST reads API only.
Type os security provided by SOAP is SOAP Enviornment.	Type of security provided by REST is AUTH Token, HEADER, PARAMS etc.
Heavy in weight API.	Light in weight API.
Response time is more.	Response time is less.
Not best for CRUD operations.	Best for CRUD operations.

22. Types of API

REST API- Uses Postman tool (Representational state transfer)

SOAP API- Uses SOAPUI tool (simple object access protocol)

23. Concept under REST:

REST- REST is an architecture used to create rest API.

REST Assured- To automate rest API we need rest assured libraries.

RESTFUL- when we automate rest API it called as restful services.

24. What is the difference between '/' and '?'

/- Path parameter

?- Query parameter

25. What is producer and consumer?

Producer- who produce the data

Consumer- who consumes the data

26. What is URI?

URI- Unique resource identifier

URI= URL+ENDPOINT

Eg. https://www.amazon.com+/login/home

27. What are diff ways to pass the data/ scripting languages?

a) JSON:

```
{
 "name": "Suraj ",
 "email": "Suraj123@gmail.com",
 "gender": "Male",
 "status": "Active"
}
```

b) XML:

<name>suraj</name>

<email>suraj@gmail.com</email>

c) String

d) Text

e) Html

f) Javascript

Etc.

28. What are headers?

Headers mean what kind of data we are passing.

a. Authorization

b. Content Type

c. Language

d. Etc.

29. What we pass in http request?

a. URI

b. Headers

c. Payload

30. What are different authorizations?

a. Basic Auth

pass the username and pass.

b. Digest

Whenever we are passing username and pass it will get convert in # keys.

It means your username/pass will secured get server side too.

c. Oauth1

Oauth1 required below things:

1. Consumer Key
2. Consumer Secret
3. Access Token
4. Secret Token

Above info will get from developers.

e. Oauth2

Oauth2 required below things:

1. Client Id
2. Client Secret
3. Grant type

Above info will get from developers.

f. Bearer Token

g. NoAuth

31. What are Oauth1 and Oauth2?

Oauth1- this auth uses when we need third party logins.

Oauth2- this auth uses when we have single url and different enpoints

32. What is WSDL file

WSDL basically an XML document contains all the details about web service and all API request

33. What is Web service?

Whenever we are hitting any service over the internet it known as webservice.

Webservice is any piece of software that makes it available over the internet and uses a standardized XML messaging system.

34. What is UDDI?

- Universal description discovery integration

-UDDI is an XML based standard for describing, publishing and finding the webservices.

35. What are diff soap elements/components?

a. Envelop – It is beginning and end of message.

b. Header – Header elements contain header information.

c. Body – body element contains call and response information.

d. Fault – Fault contain error and status information.

36. What is diff WSDL element/component?

a. Type- Define the data types used by the webservices.

b. Message – Define the data element for each operation.

c. Port Type- Describe the operation that can be performed and message involve.

d. Binding- Defines the protocol and data format for each port type.

37. What are different assertions present in SOAPUI?

Below are the different assertions present in SOAPUI:

a. Contains - checks for the existence of a specified string

b. Not Contains - checks for the non-existence of a specified string

c. Response SLA - validates that the last received response time was within the defined limit. Applicable to Script TestSteps and TestSteps that send requests and receive responses.

d. Invalid HTTP Status Codes - checks that the target TestStep received an HTTP result with a status code not in the list of defined codes. Applicable to any TestStep that receives HTTP messages

e. Valid HTTP Status Codes - checks that the target TestStep received an HTTP result with a status code in the list of defined codes. Applicable to any TestStep that receives HTTP messages.Etc.

38. What are different API gatways.

a. SSL certificate

b. Routing

c. Adapter

d. Cache

e. Load balancer

39. Difference between monolithic and miscroservice?

Monolethic - all api available under one service

Microservice- for api have different microservice.

40. What is means URI(API)

URI= URL+endpoints(resource)

url: www.facebook.com

endpoints: /login/home

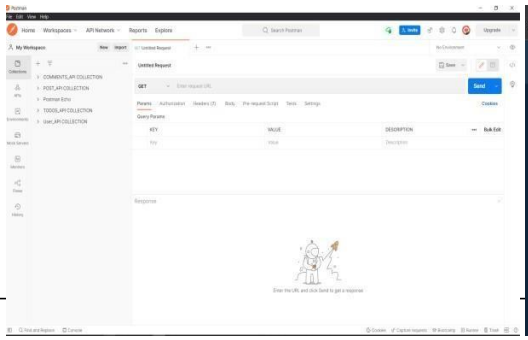
URI- unified resource identifier

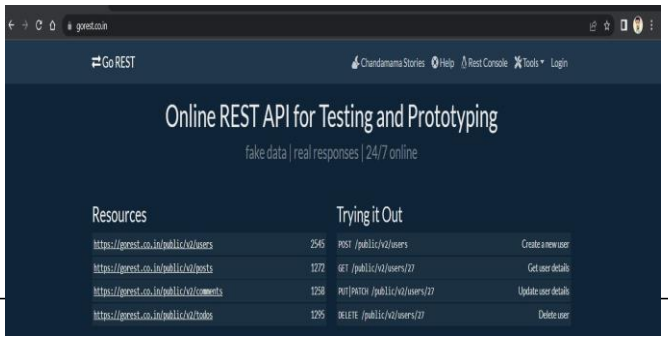
url- unified resource locator

HOW TO hit services IN POSTMAN-

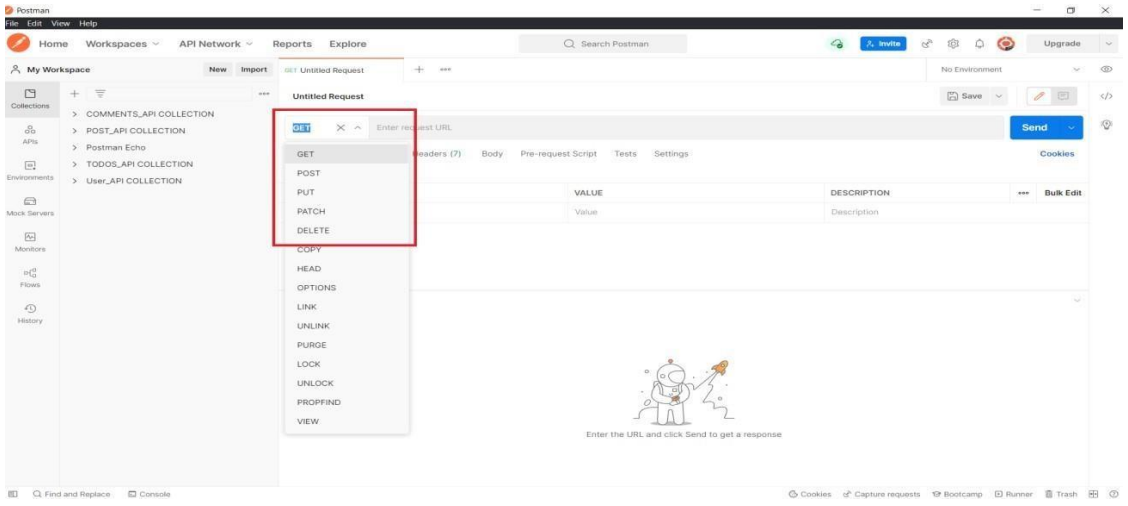
Steps:

1. Open the postman as well as Gorest site.





2. Select the method (i.e GET, PUT, POST, and DELETE).



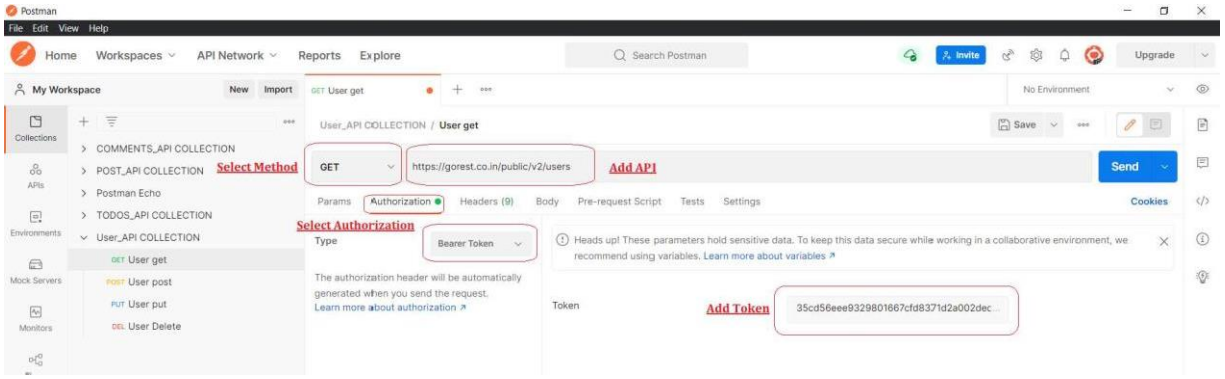
3. Add the URL/API.
4. Add the headers
   a. Authorization
   b. Content-type

5. Only in case of POST and PUT add the body not for the GET.

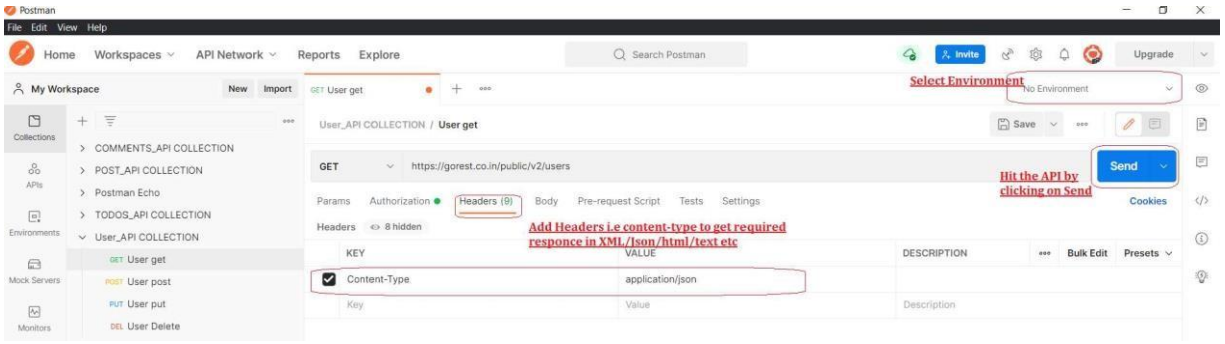
6. Click in send.

HOW TO GET*(FETCH)* THE RESPONSE BY HITTING API IN POSTMAN-

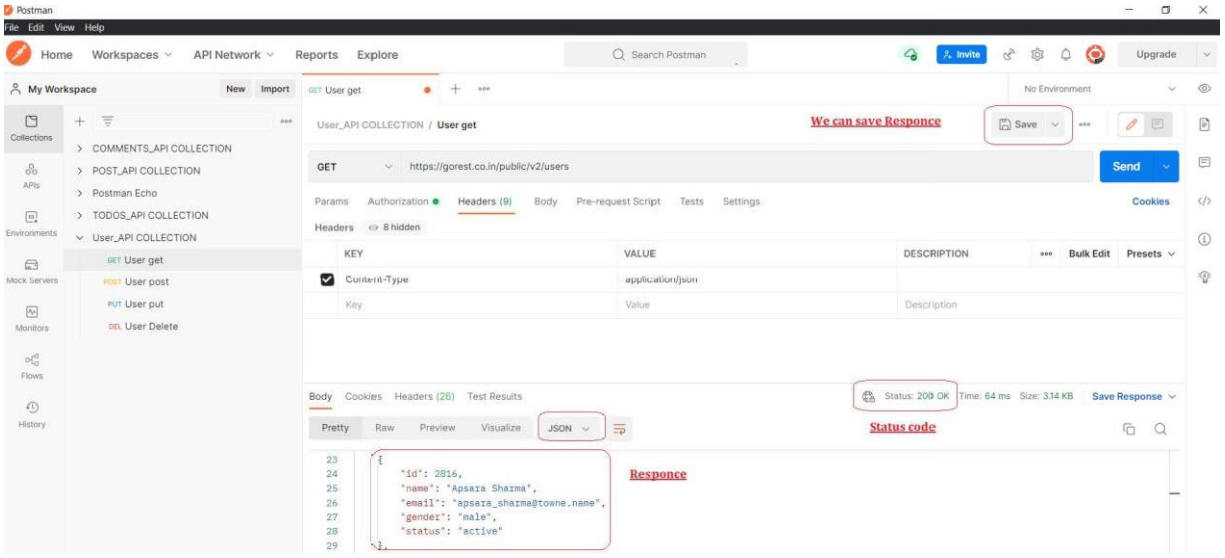
Step1



Step2

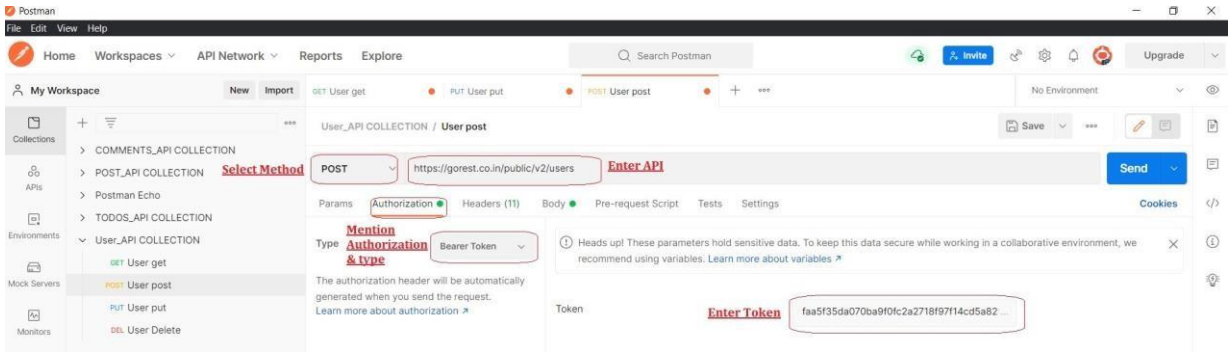


Step3

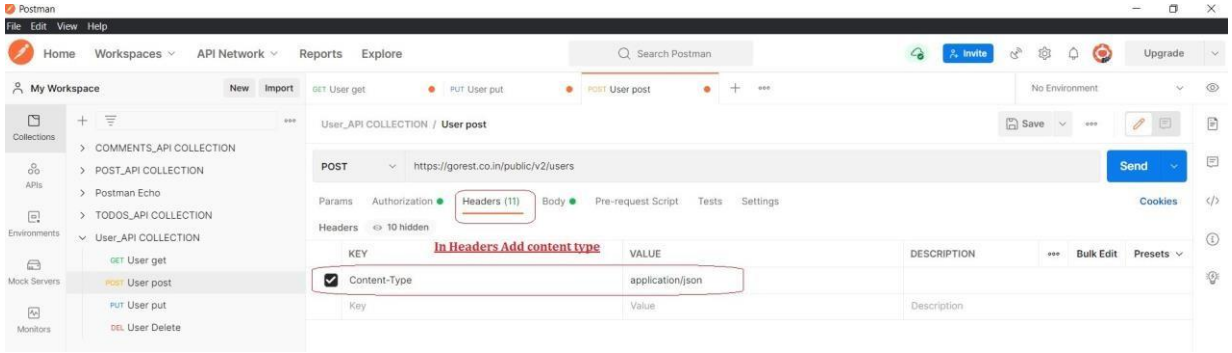


HOW TO POST*(CREATE)* THE RESPONSE BY HITTING API IN POSTMAN-

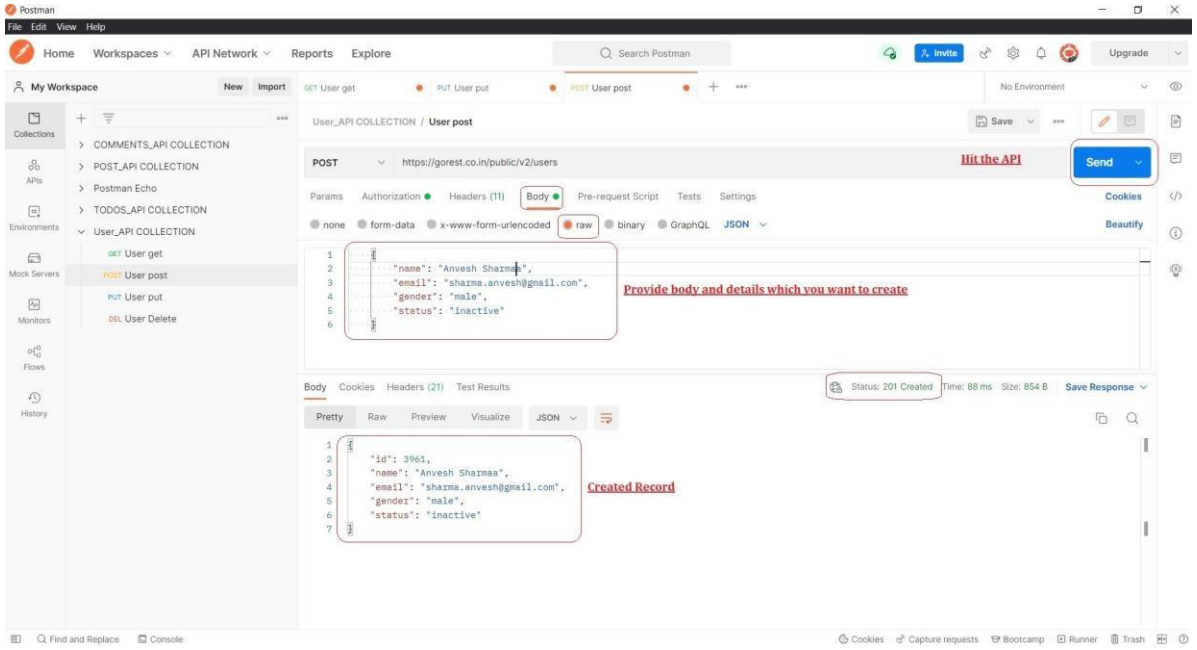
Step1



Step2

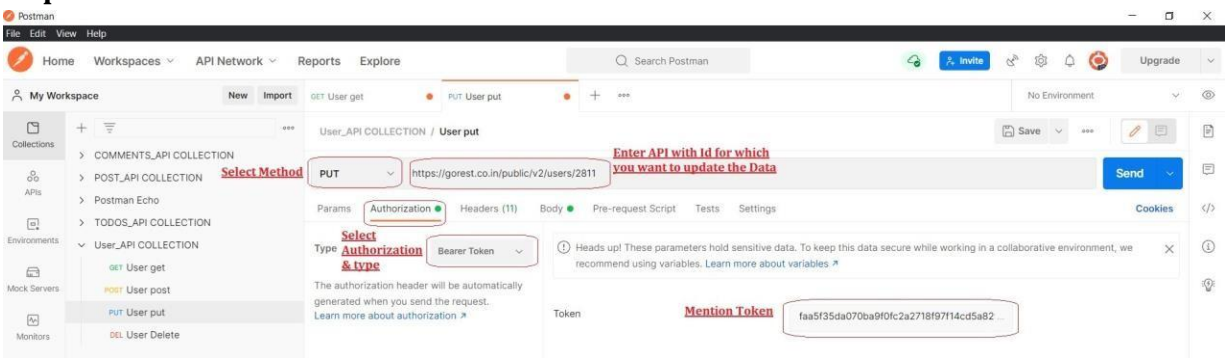


Step3

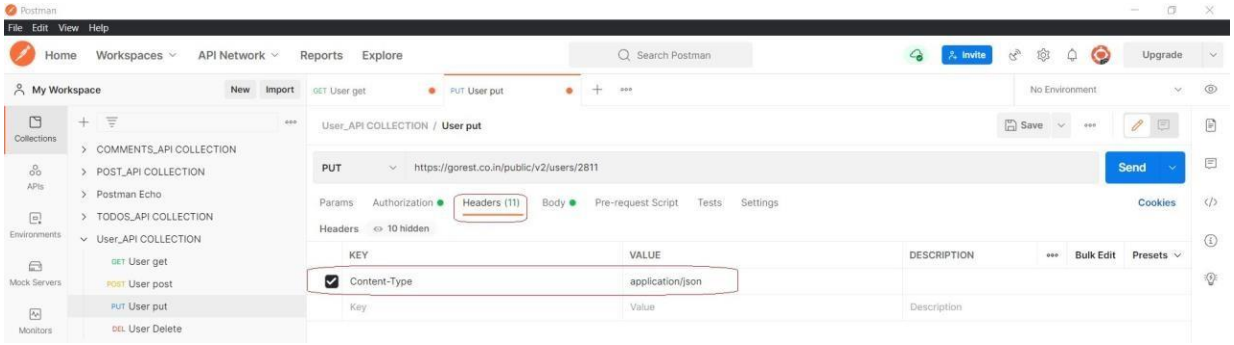


HOW TO PUT*(UPDATE)* THE RESPONSE BY HITTING API IN POSTMAN-

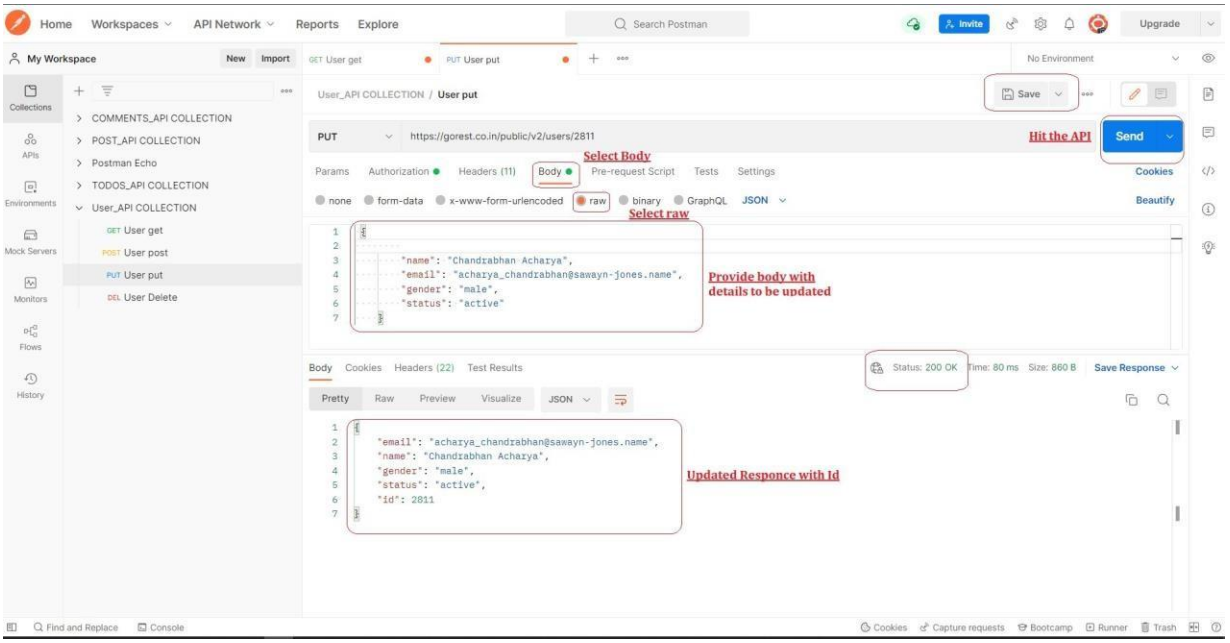
Step1



Step2



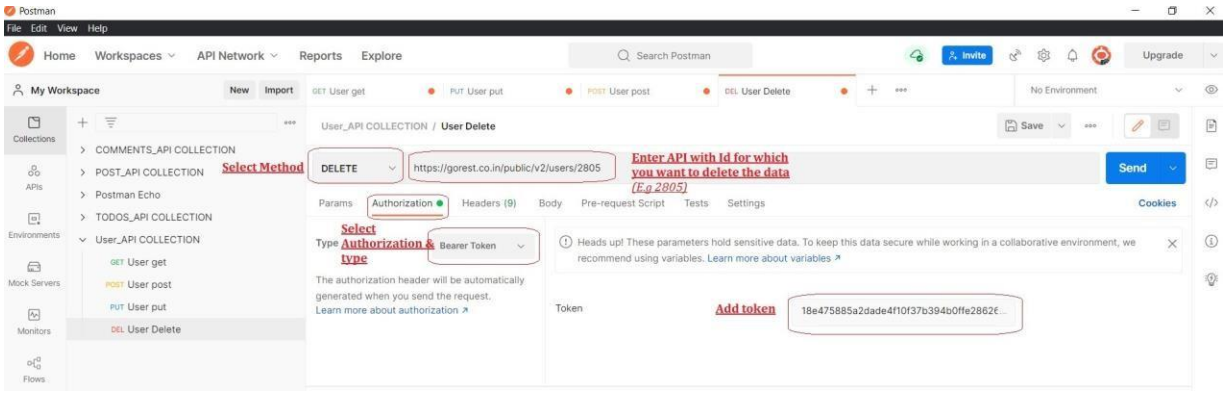
Step3



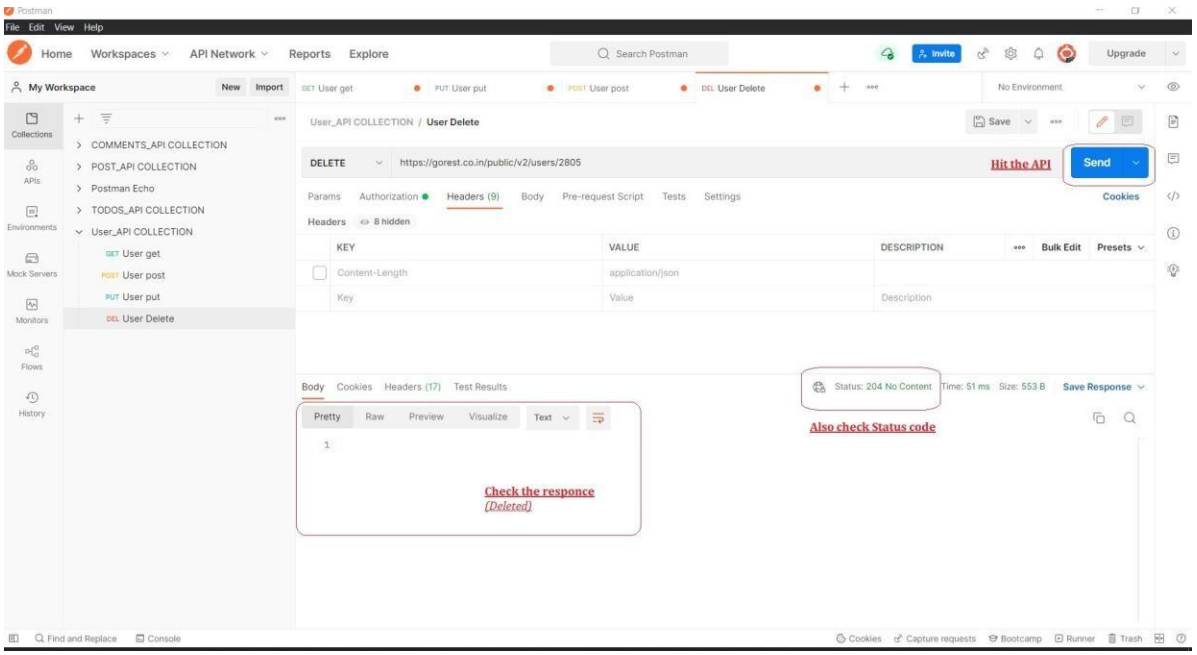
- √ Get token from gorest site(login->in howdy-Api token)
- √ Ex: AUTHORIZATION- Bearer kdhfkgshfk745874697456740bhchjfvx4889

HOW TO DELETE THE RESPONSE BY HITTING API IN POSTMAN-

Step1



Ste

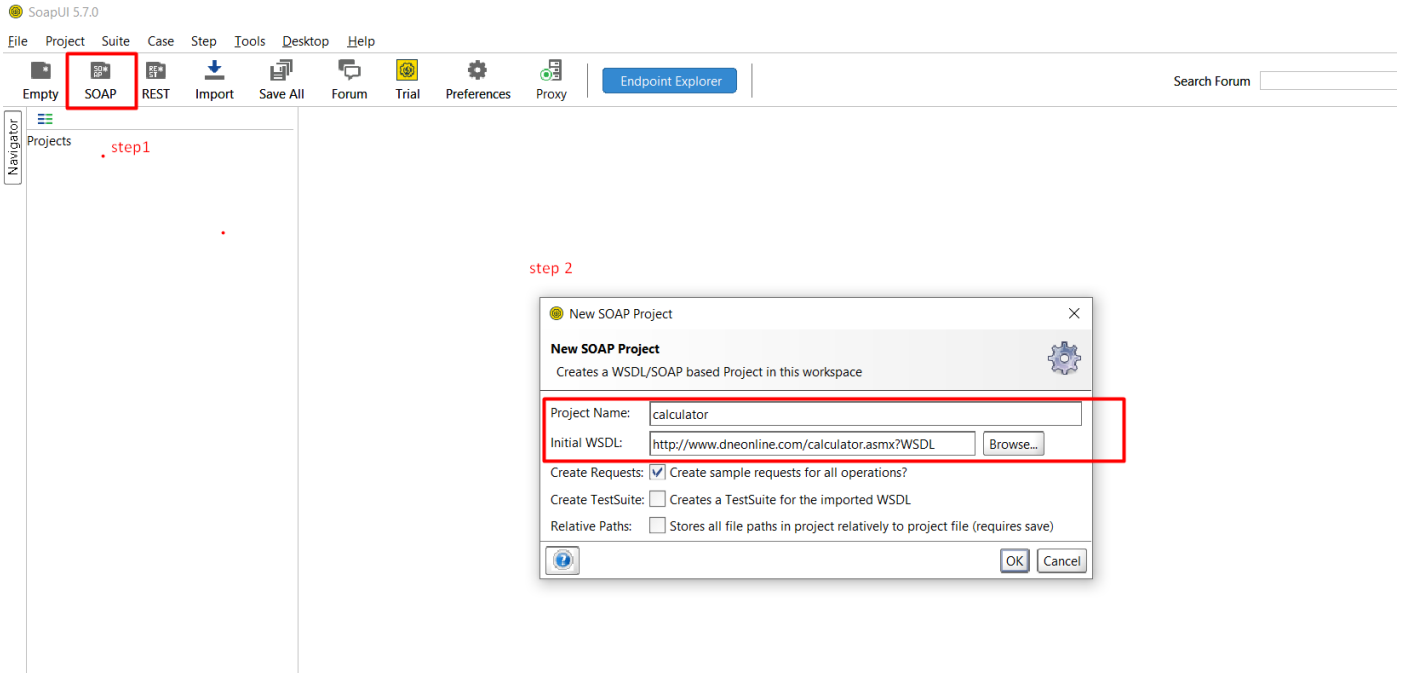


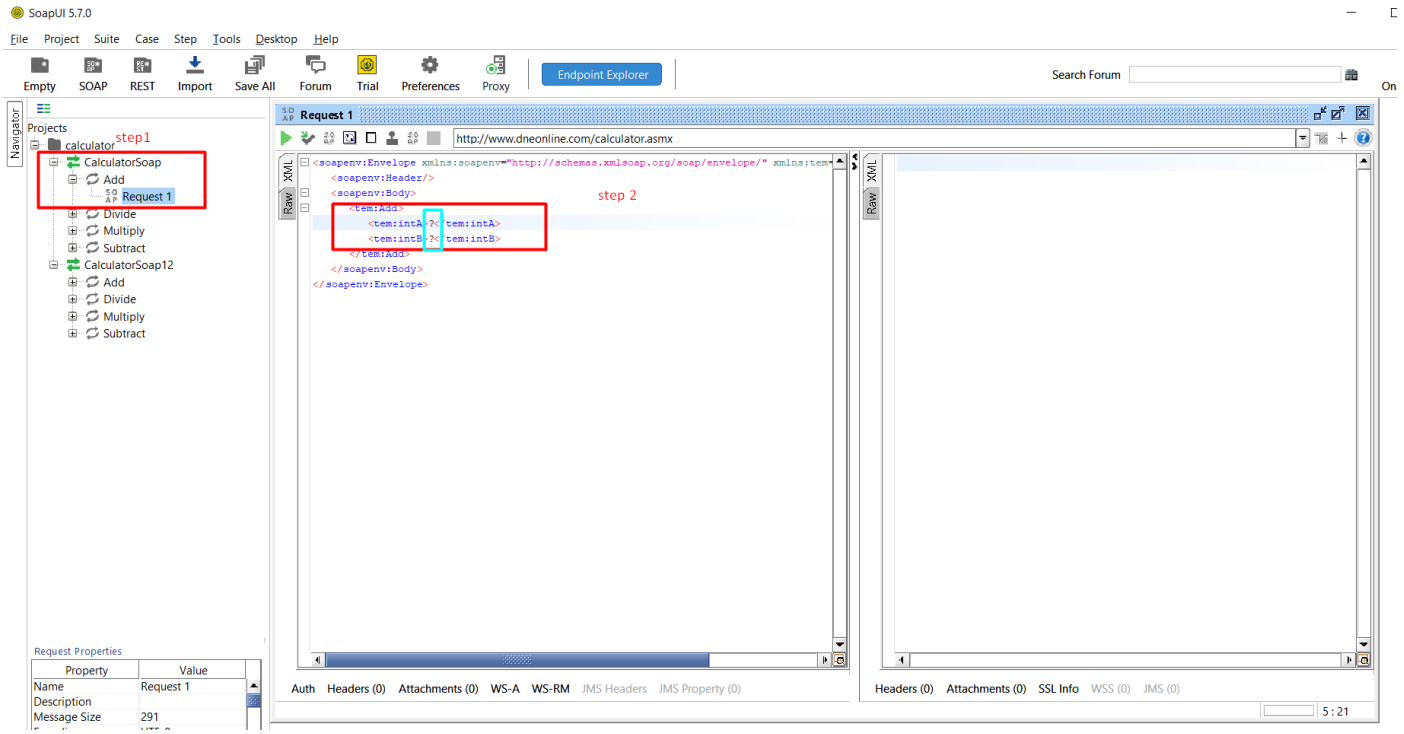
How to hit services in SOAPUI

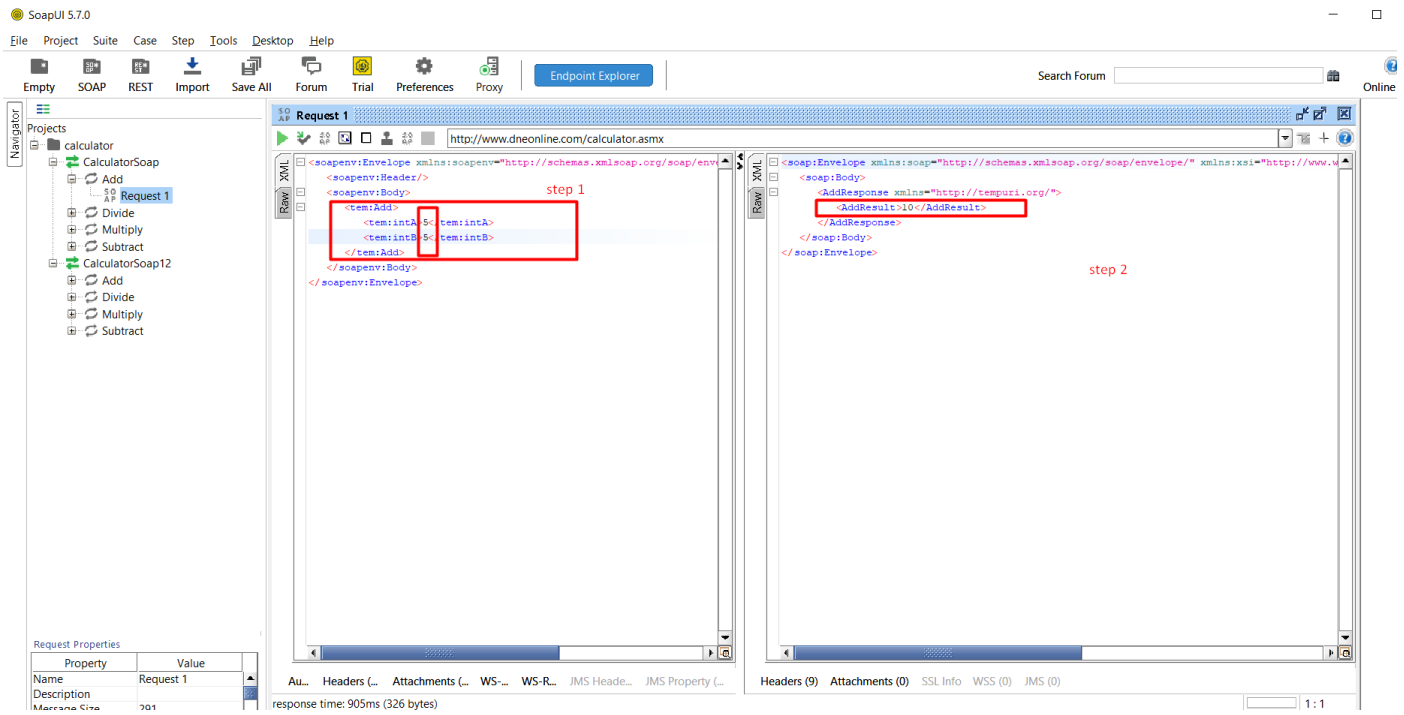
1. How to hit WSDL services

Steps in Soap service:

1. click on soap project(wsdl file:
   http://www.dneonline.com/calculator.asmx?WSDL)
2. ADD wsdl file/url and give project name, make all
   checkboxes as bydefault
   eg. http://www.dneonline.com/calculator.asmx?WSDL
3. click on any api (+) and then click on request
4. in request pass then values
   eg. in add pass 10 and 10 as int a and int b by removing
   question mark
5. click on green run button.
   6.now you will get the response.







Same for the divide, multiply,substract

You can apply the assertion for rest as well as for soap(wsdl)

service

Note: assertions are shown in rest service in below

2.How to hit rest services in soapui tool

Steps in rest service:

1. click on rest project

2.pass the url from service

eg. https://gorest.co.in/public/v2/users

3. add the method(GET)

4.by clicking on + button add headors in header

key:Authorization

key:Content-type

note: make sure all spells are correct

5. you will get response here but you cant apply any assertions.

6. to apply any assertion/validations right click on project and select-> new test suit

- give any name for that test suit

7. now right click on testsuit and create a new test case-> new test case

8. then right click on testcase and add new test request->add step-> rest request

9. select the request(automatically populate the request)

10. now you will get the assertion tab below side

11. click on assertion and then clickon + icon

12. you will get the diff assertion

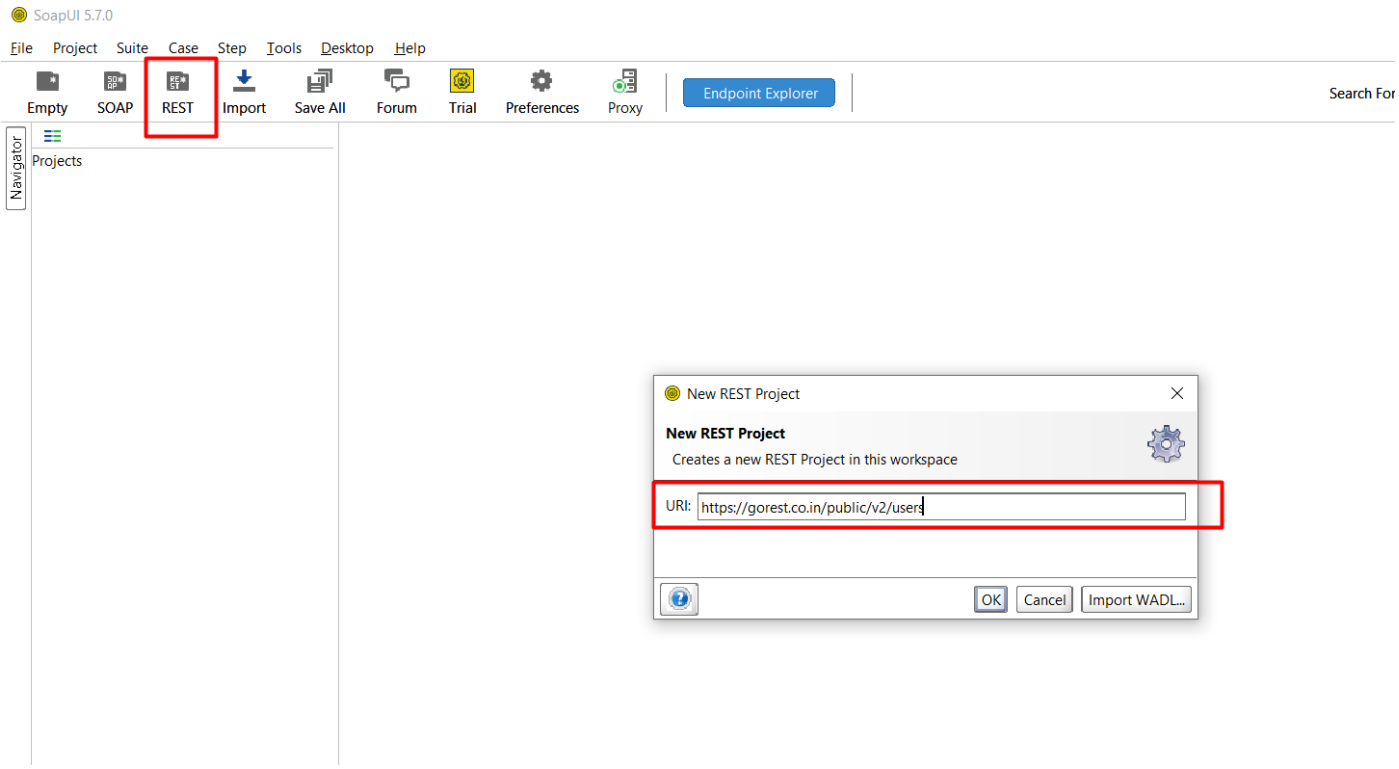
eg. content- to verify string

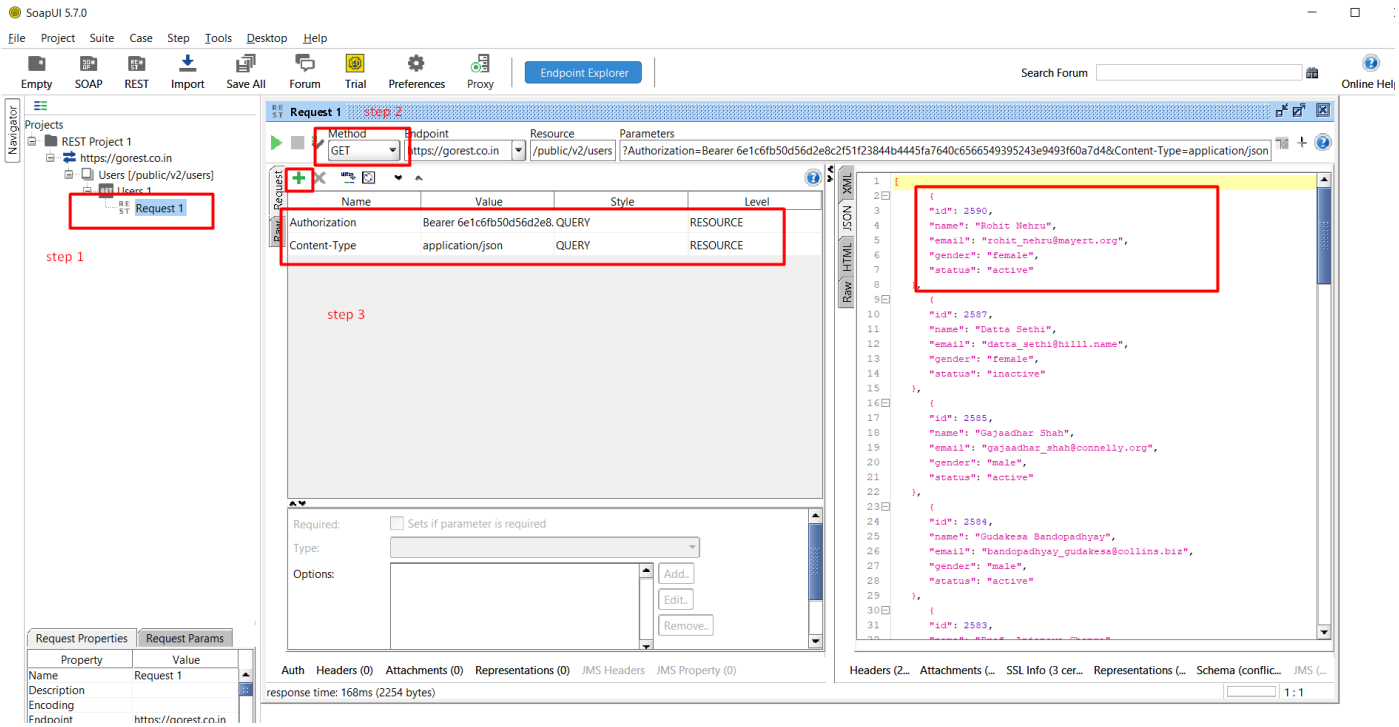
not content- same as above but opposite

valid http status code- to verify status code

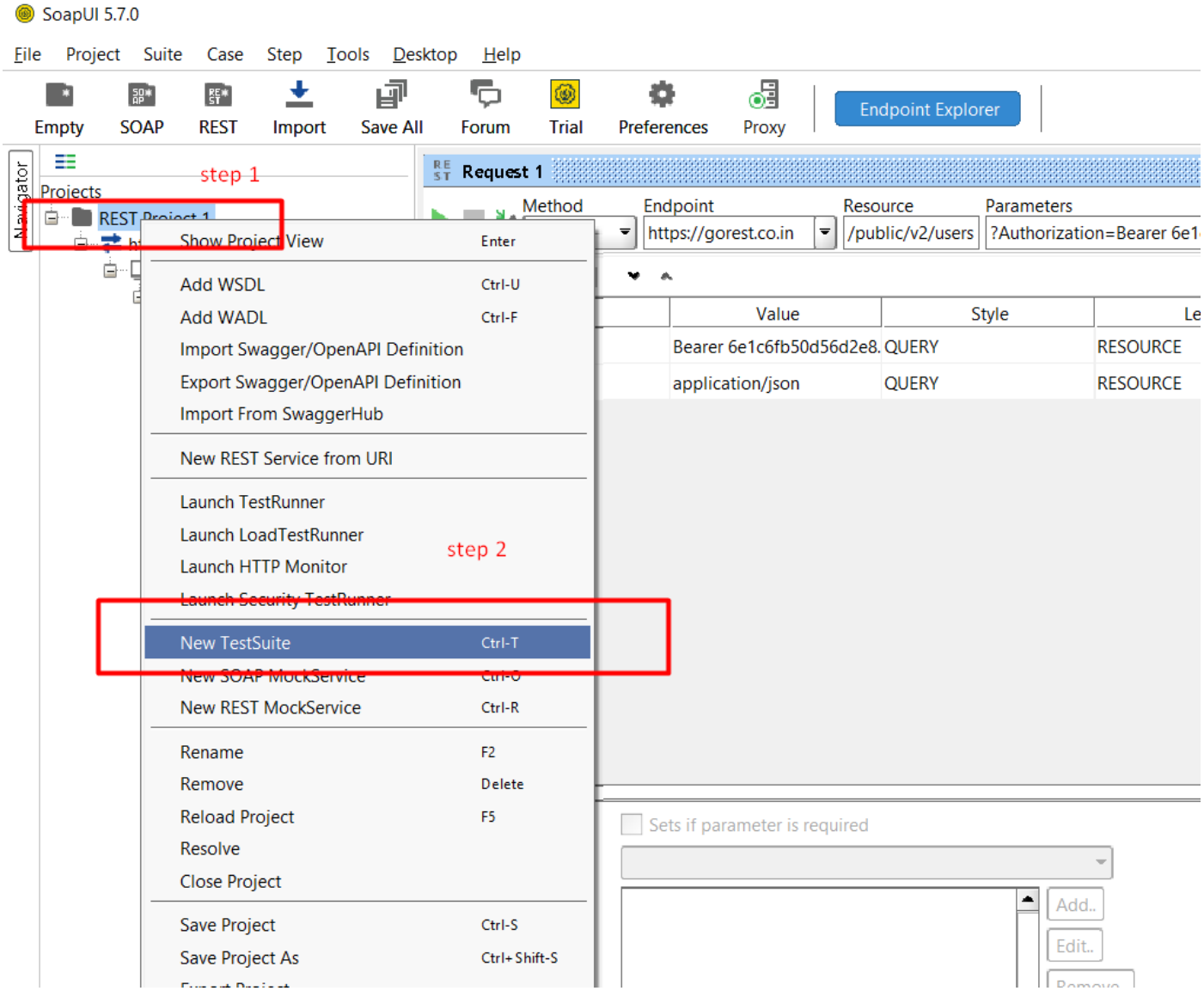
invalid http status code- same as above but opposite

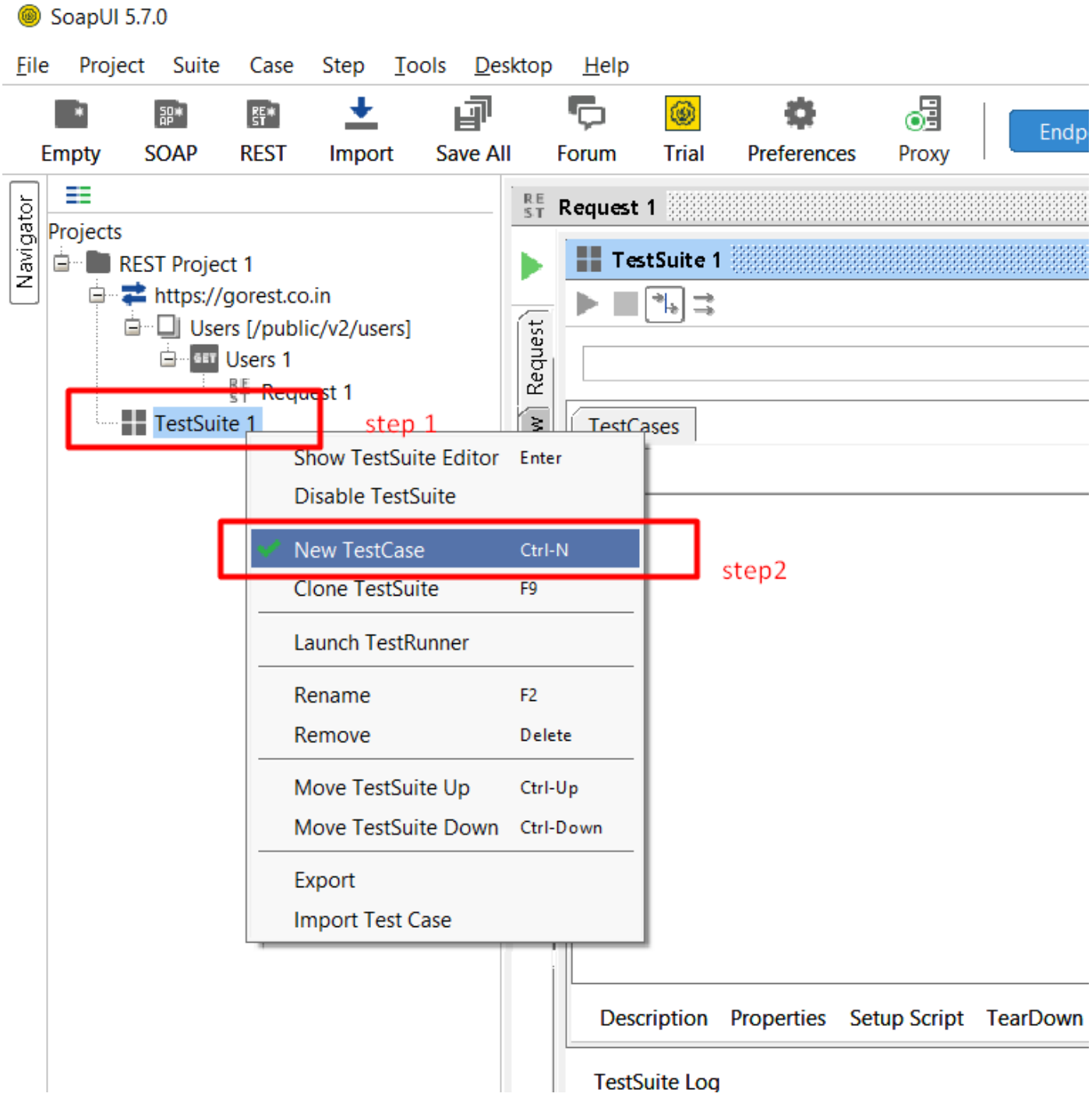
response SLA- to verify response time

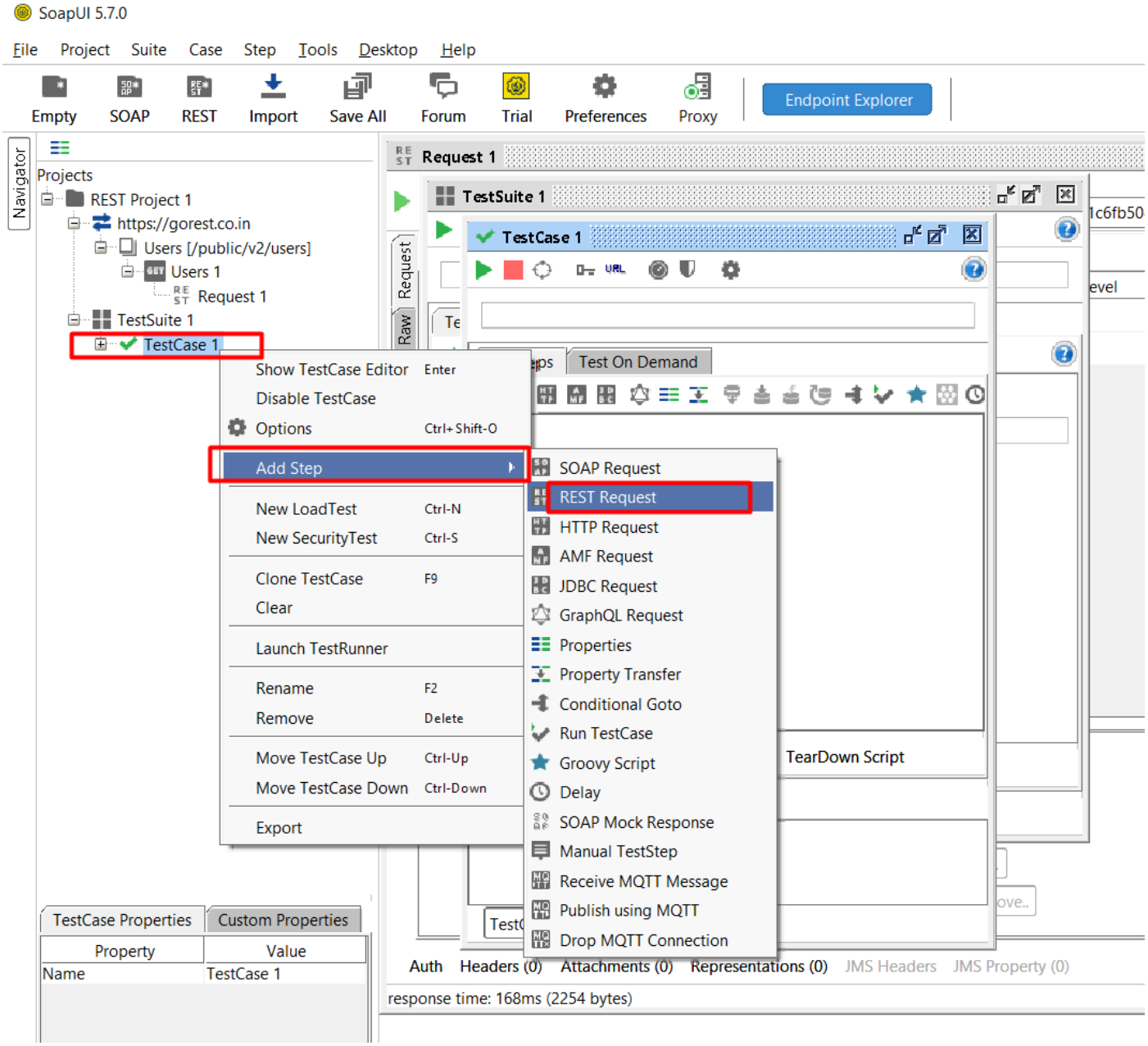


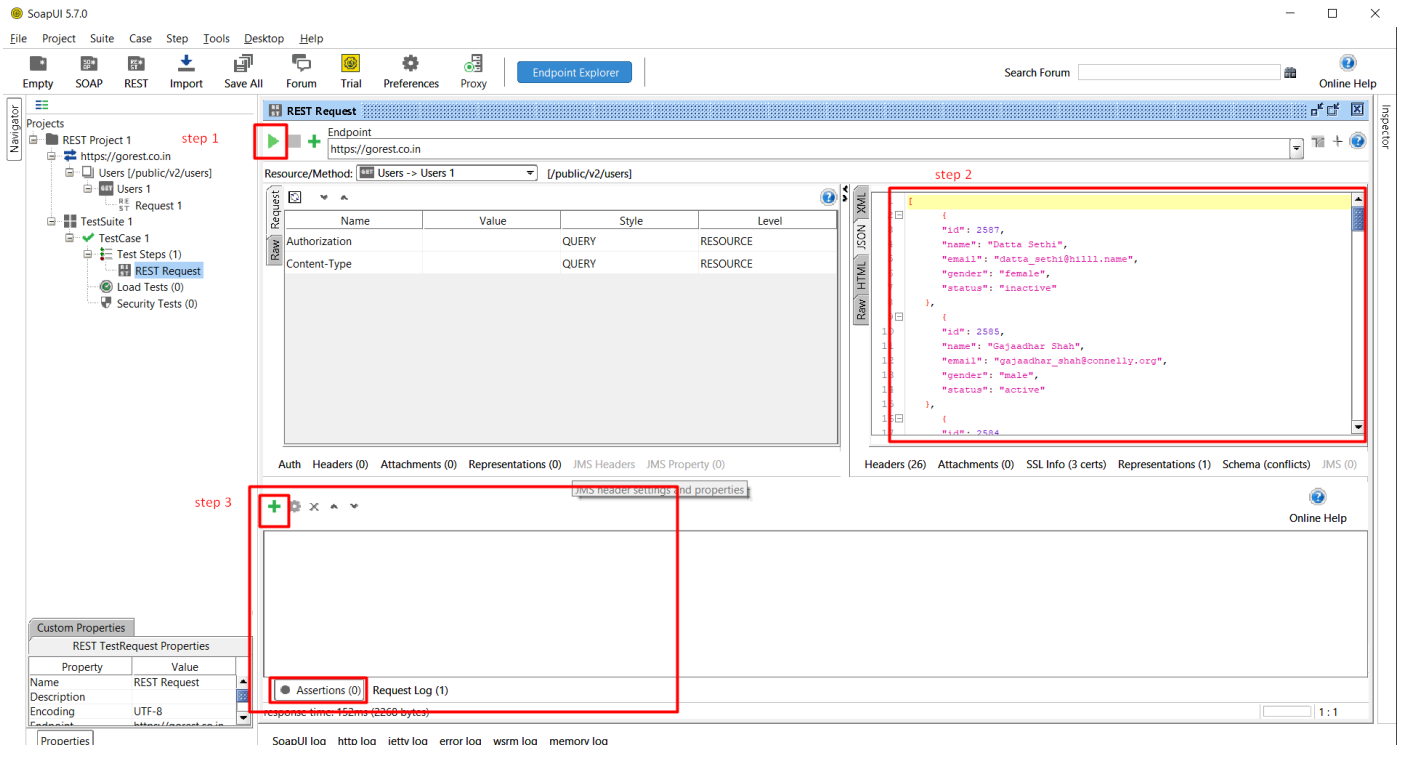


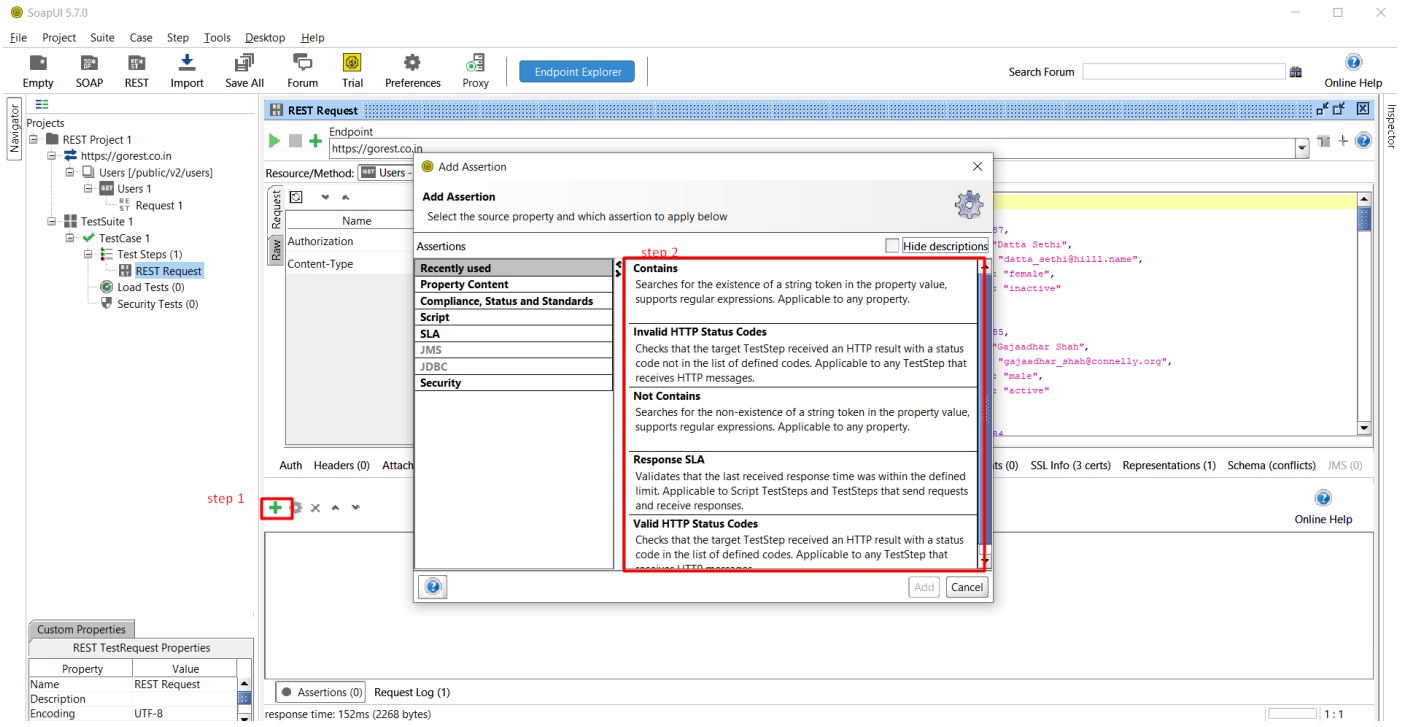
To apply the assertions need to convert the project into test cases


SoapUI 5.7.0
File Project Suite Case Step Tools Desktop Help
Empty SOAP REST Import Save All Forum Trial Preferences Proxy
Endpoint Explorer
Navigator
Projects
step 1
REST Project 1
Request 1
Method
Endpoint
https://gorest.co.in
Resource
/public/v2/users
Parameters
?Authorization=Bearer 6e1
Value
Style
Bearer 6e1c6fb50d56d2e8.
QUERY
RESOURCE
application/json
QUERY
RESOURCE
Show Project View Enter
Add WSDL Ctrl-U
Add WADL Ctrl-F
Import Swagger/OpenAPI Definition
Export Swagger/OpenAPI Definition
Import From SwaggerHub
New REST Service from URI
Launch TestRunner
Launch LoadTestRunner
Launch HTTP Monitor
step 2
Launch Security TestRunner
New TestSuite Ctrl-T
New SOAP MockService Ctrl-O
New REST MockService Ctrl-R
Rename F2
Remove Delete
Reload Project F5
Resolve
Close Project
Save Project Ctrl-S
Save Project As Ctrl+Shift-S
Sets if parameter is required
Add..
Edit..

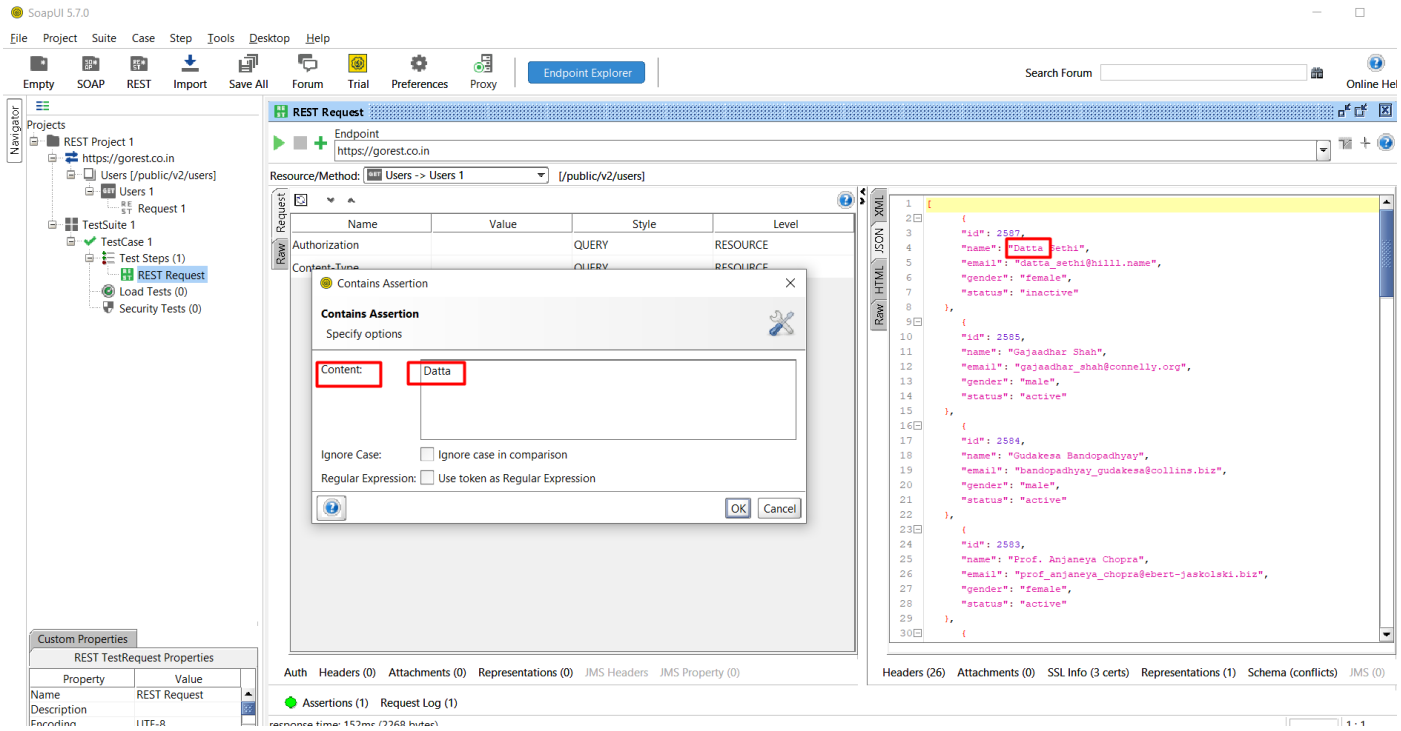

SoapUI 5.7.0
File Project Suite Case Step Tools Desktop Help
Empty SOAP REST Import Save All Forum Trial Preferences Proxy
Endp
Navigator
Projects
REST Project 1
https://gorest.co.in
Users [/public/v2/users]
Users 1
Request 1
TestSuite 1
step 1
Request 1
TestSuite 1
Request
TestCases
Show TestSuite Editor Enter
Disable TestSuite
New TestCase Ctrl-N
Clone TestSuite F9
Launch TestRunner
Rename F2
Remove Delete
Move TestSuite Up Ctrl-Up
Move TestSuite Down Ctrl-Down
Export
Import Test Case
step2
Description Properties Setup Script TearDown
TestSuite Log

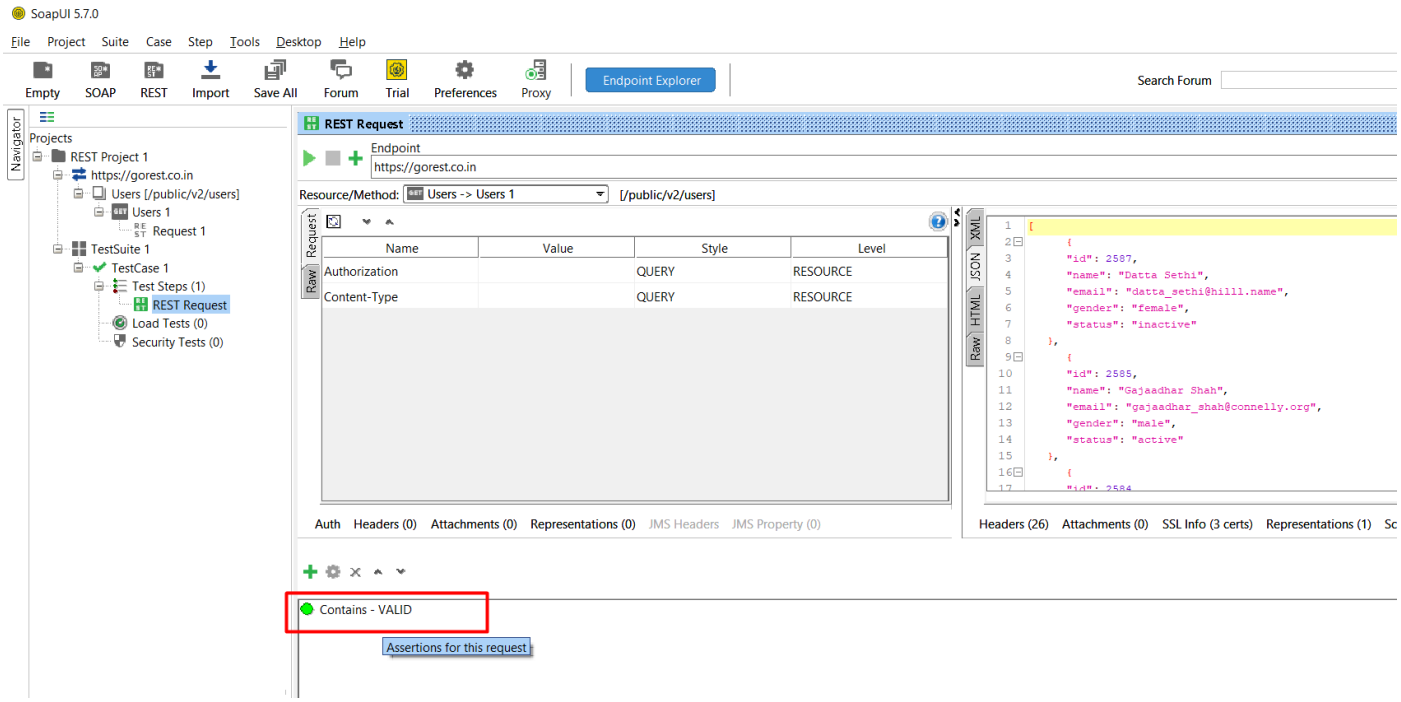

SoapUI 5.7.0
File Project Suite Case Step Tools Desktop Help
Empty SOAP REST Import Save All Forum Trial Preferences Proxy
Endpoint Explorer
Navigator
Projects
REST Project 1
https://gorest.co.in
Users [/public/v2/users]
Users 1
Request 1
TestSuite 1
TestCase 1
Request 1
TestSuite 1
TestCase 1
Test On Demand
Show TestCase Editor Enter
Disable TestCase
Options Ctrl+Shift-O
Add Step
New LoadTest Ctrl-N
New SecurityTest Ctrl-S
Clone TestCase F9
Clear
Launch TestRunner
Rename F2
Remove Delete
Move TestCase Up Ctrl-Up
Move TestCase Down Ctrl-Down
Export
SOAP Request
REST Request
HTTP Request
AMF Request
JDBC Request
GraphQL Request
Properties
Property Transfer
Conditional Goto
Run TestCase
Groovy Script
Delay
SOAP Mock Response
Manual TestStep
Receive MQTT Message
Publish using MQTT
Drop MQTT Connection
TearDown Script
TestCase Properties
Custom Properties
Property
Value
Name
TestCase 1
Auth Headers (0) Attachments (0) Representations (0) JMS Headers JMS Property (0)
response time: 168ms (2254 bytes)


SoapUI 5.7.0
File Project Suite Case Step Tools Desktop Help
Empty SOAP REST Import Save All Forum Trial Preferences Proxy
Endpoint Explorer
Search Forum
Online Help
Navigator
Projects
REST Project 1
https://gorest.co.in
Users [/public/v2/users]
Users 1
Request 1
TestSuite 1
TestCase 1
Test Steps (1)
REST Request
Load Tests (0)
Security Tests (0)
step 1
REST Request
Endpoint
https://gorest.co.in
Resource/Method:
Users -> Users 1
[/public/v2/users]
step 2
Request
Raw
Name
Value
Style
Level
Authorization
QUERY
RESOURCE
Content-Type
QUERY
RESOURCE
XML
JSON
HTML
Raw
{
"id": 2587,
"name": "Datta Sethi",
"email": "datta_sethi@hilll.name",
"gender": "female",
"status": "inactive"
},
{
"id": 2585,
"name": "Gajaadhar Shah",
"email": "gajaadhar_shah@connelly.org",
"gender": "male",
"status": "active"
},
Auth Headers (0) Attachments (0) Representations (0) JMS Headers JMS Property (0)
Headers (26) Attachments (0) SSL Info (3 certs) Representations (1) Schema (conflicts) JMS (0)
step 3
JMS header settings and properties
Online Help
Custom Properties
REST TestRequest Properties
Property
Value
Name
REST Request
Description
Encoding
UTF-8
Assertions (0)
Request Log (1)
1 : 1
SoapUI log http log jetty log error log wsrm log memory log

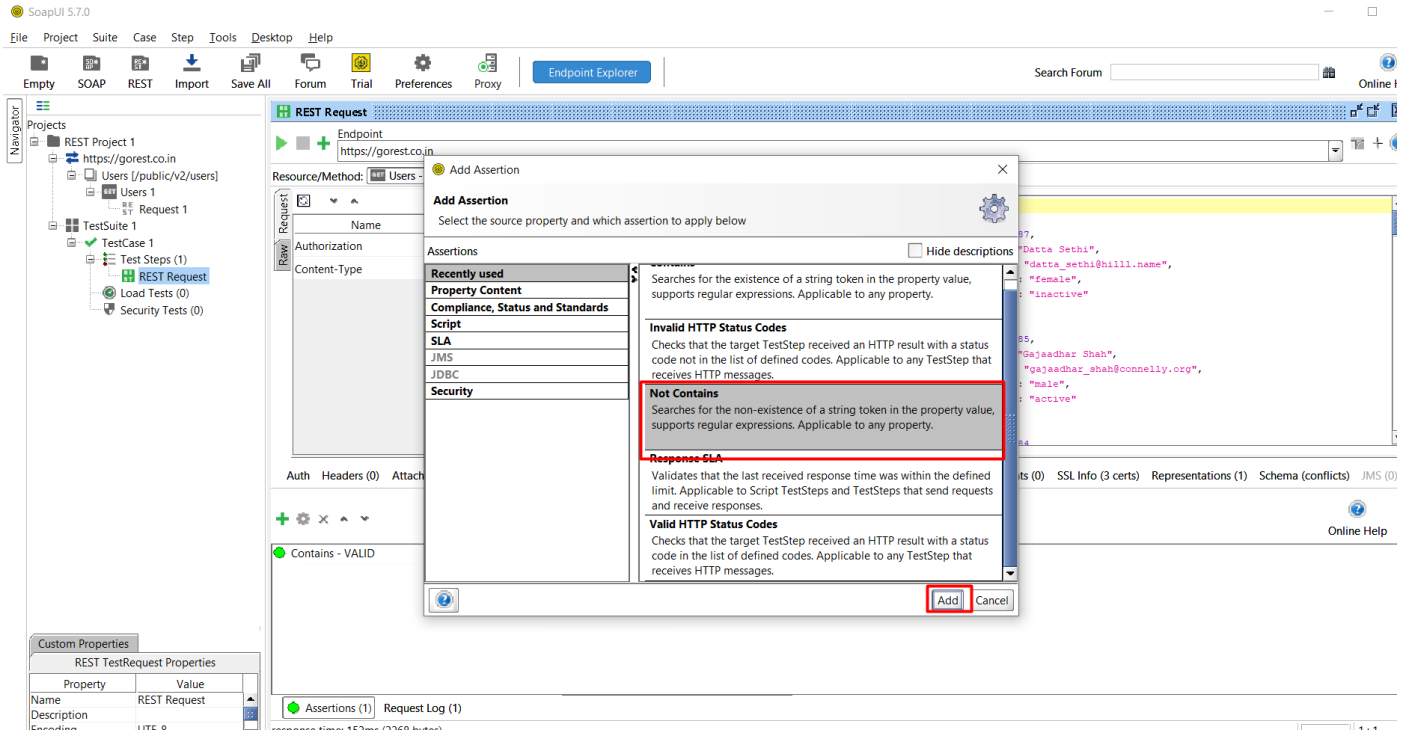


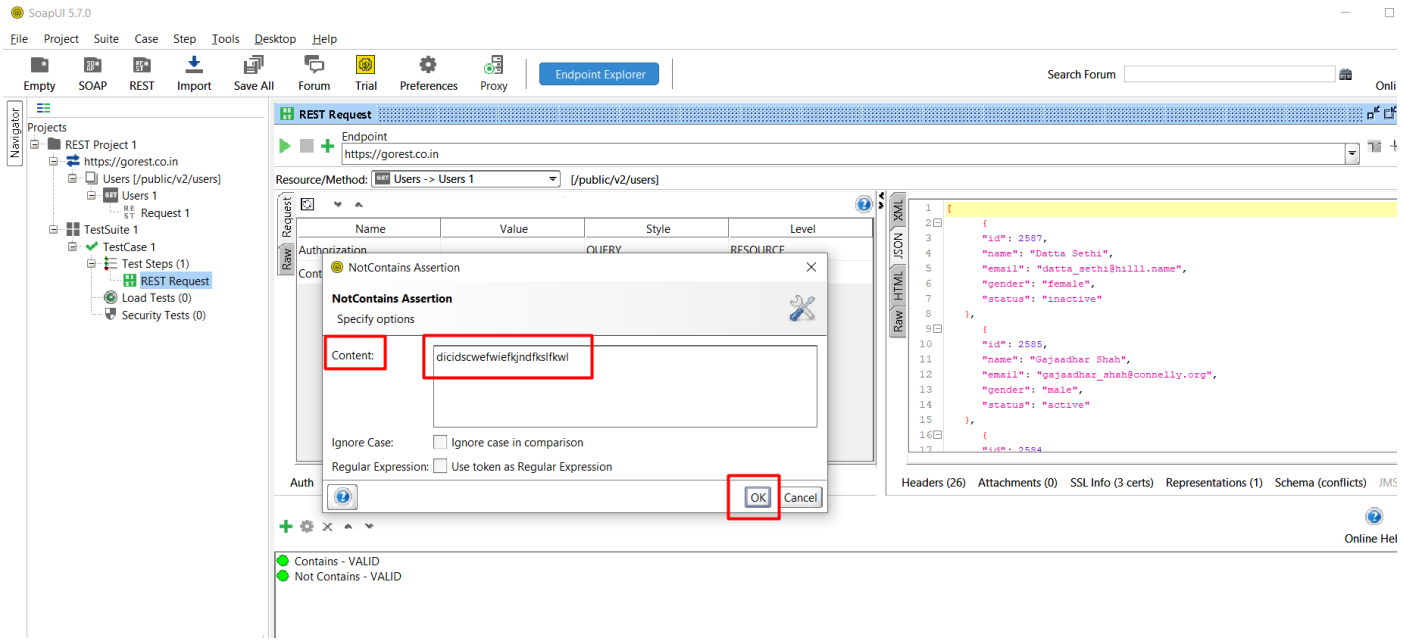
You will get the assertion anywhere in assertion popup.(just find the highlighted assertions)

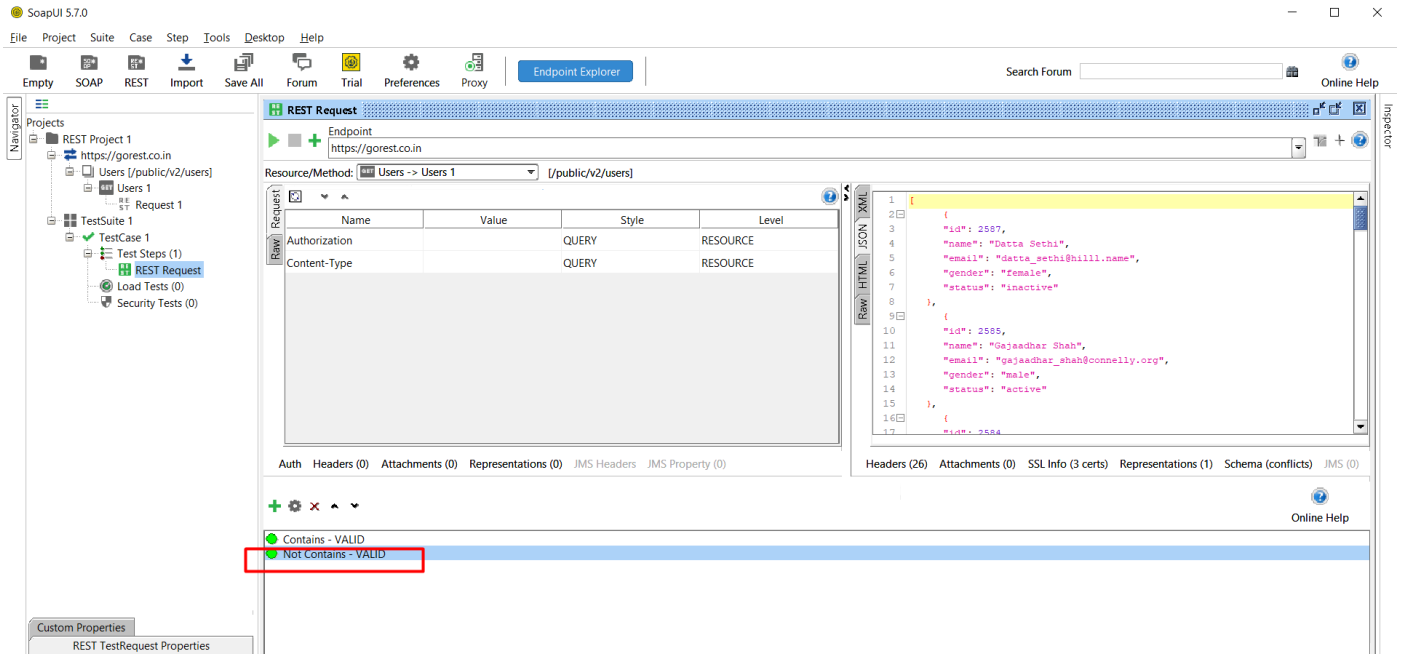


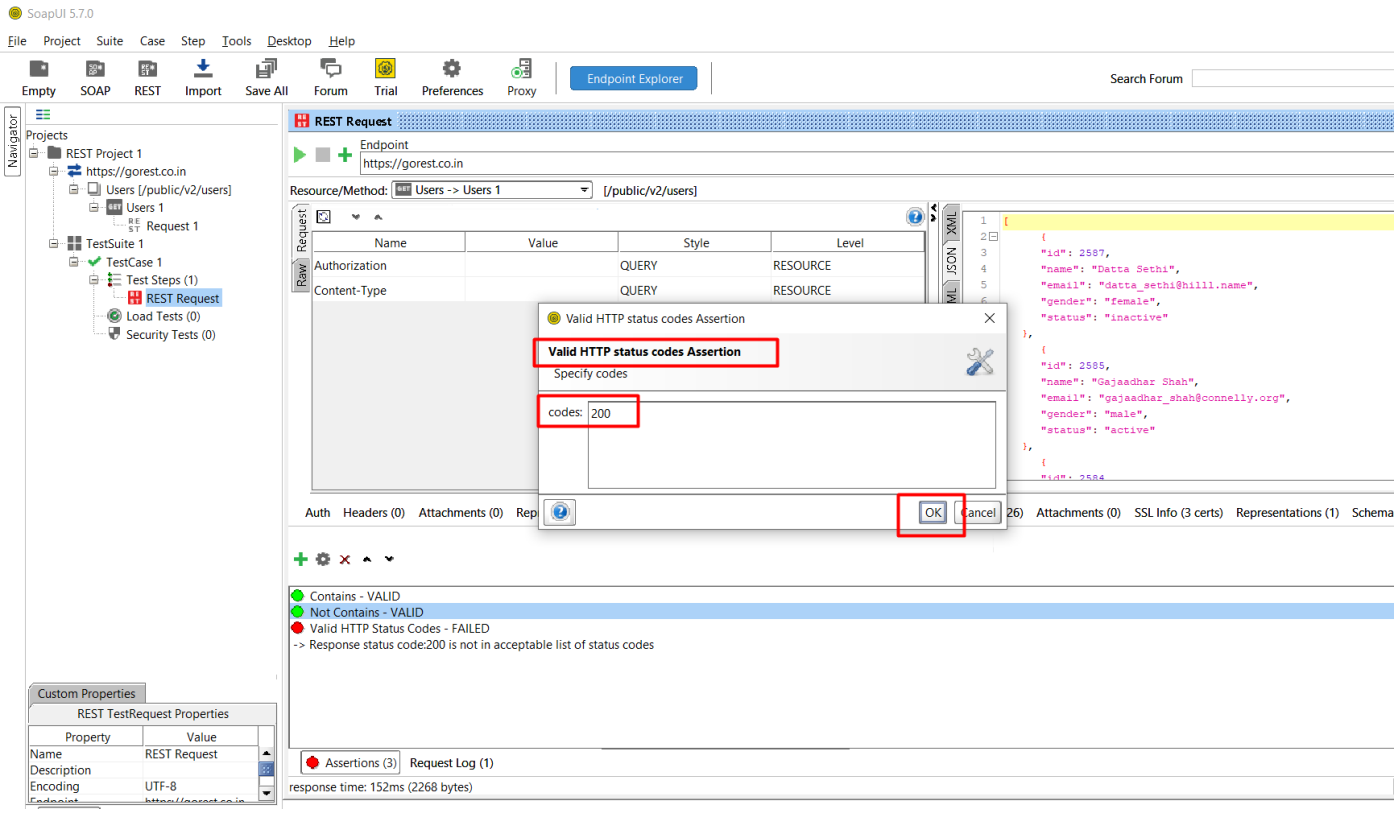


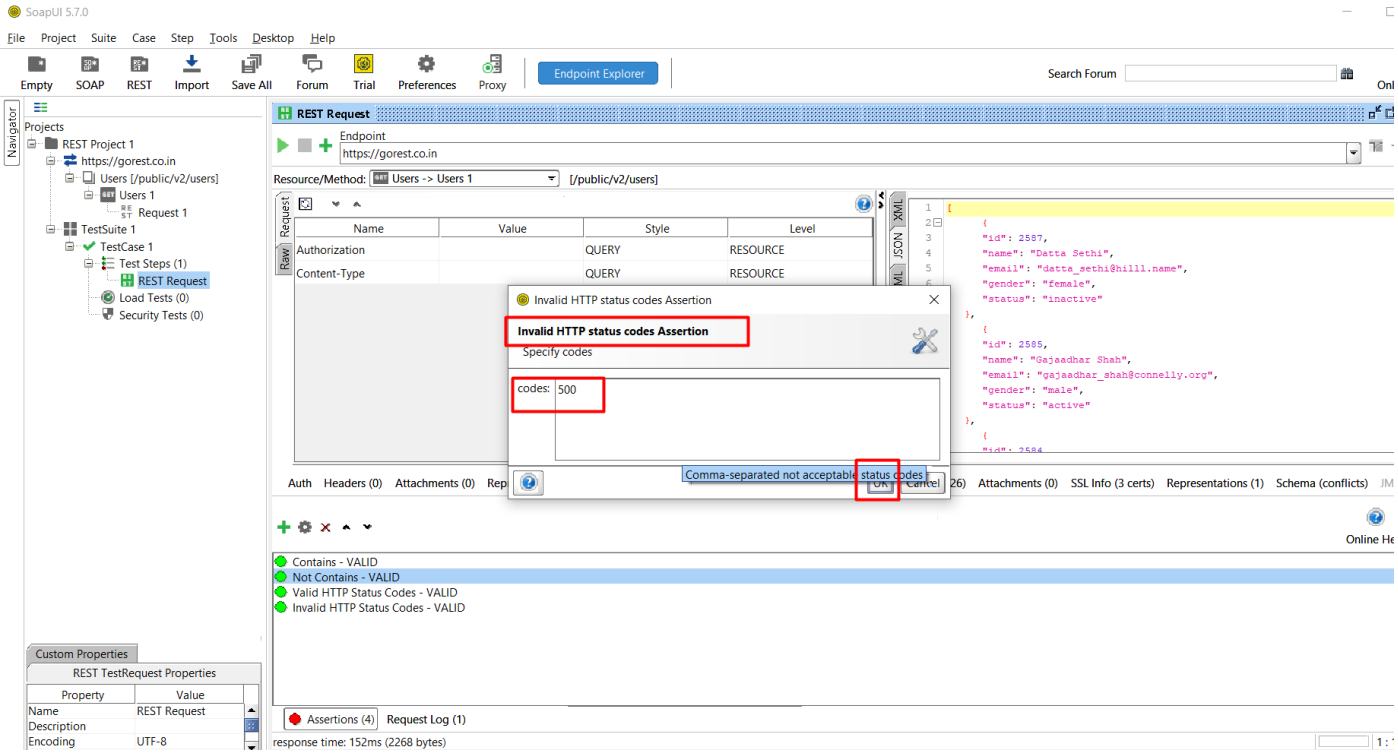
Same for the others

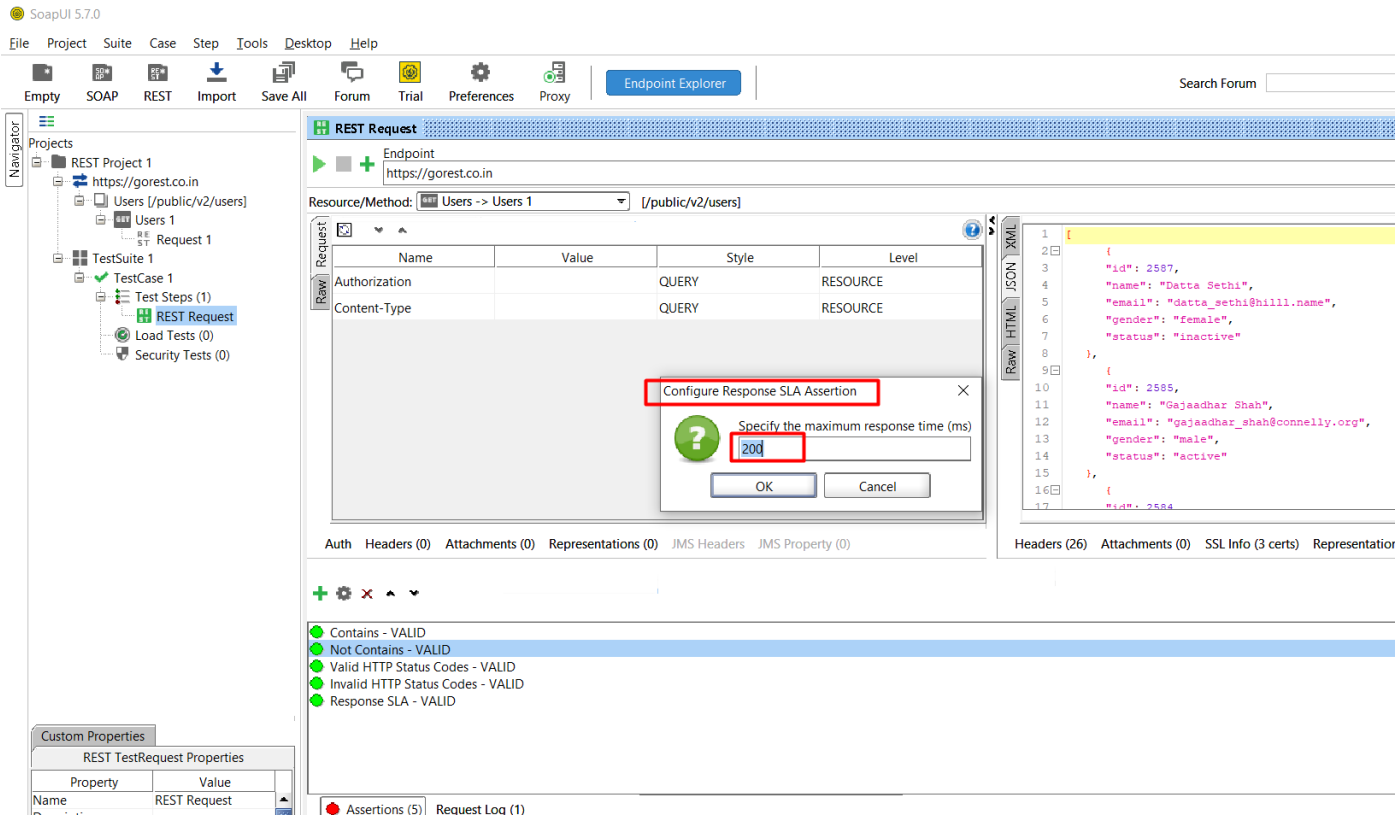



SoapUI 5.7.0
File Project Suite Case Step Tools Desktop Help
Empty SOAP REST Import Save All Forum Trial Preferences Proxy
Endpoint Explorer
Search Forum
Navigator
Projects
REST Project 1
https://gorest.co.in
Users [/public/v2/users]
Users 1
Request 1
TestSuite 1
TestCase 1
Test Steps (1)
REST Request
Load Tests (0)
Security Tests (0)
REST Request
Endpoint
https://gorest.co.in
Resource/Method: Users -> Users 1
[/public/v2/users]
Request
Raw
Name
Value
Style
Level
Authorization
QUERY
RESOURCE
NotContains Assertion
NotContains Assertion
Specify options
Content:
dicidscwefwiefkjndfkslfkwl
Ignore Case:
Ignore case in comparison
Regular Expression:
Use token as Regular Expression
OK
Cancel
XML
JSON
HTML
Raw
Headers (26)
Attachments (0)
SSL Info (3 certs)
Representations (1)
Schema (conflicts)
Online Hel
Contains - VALID
Not Contains - VALID

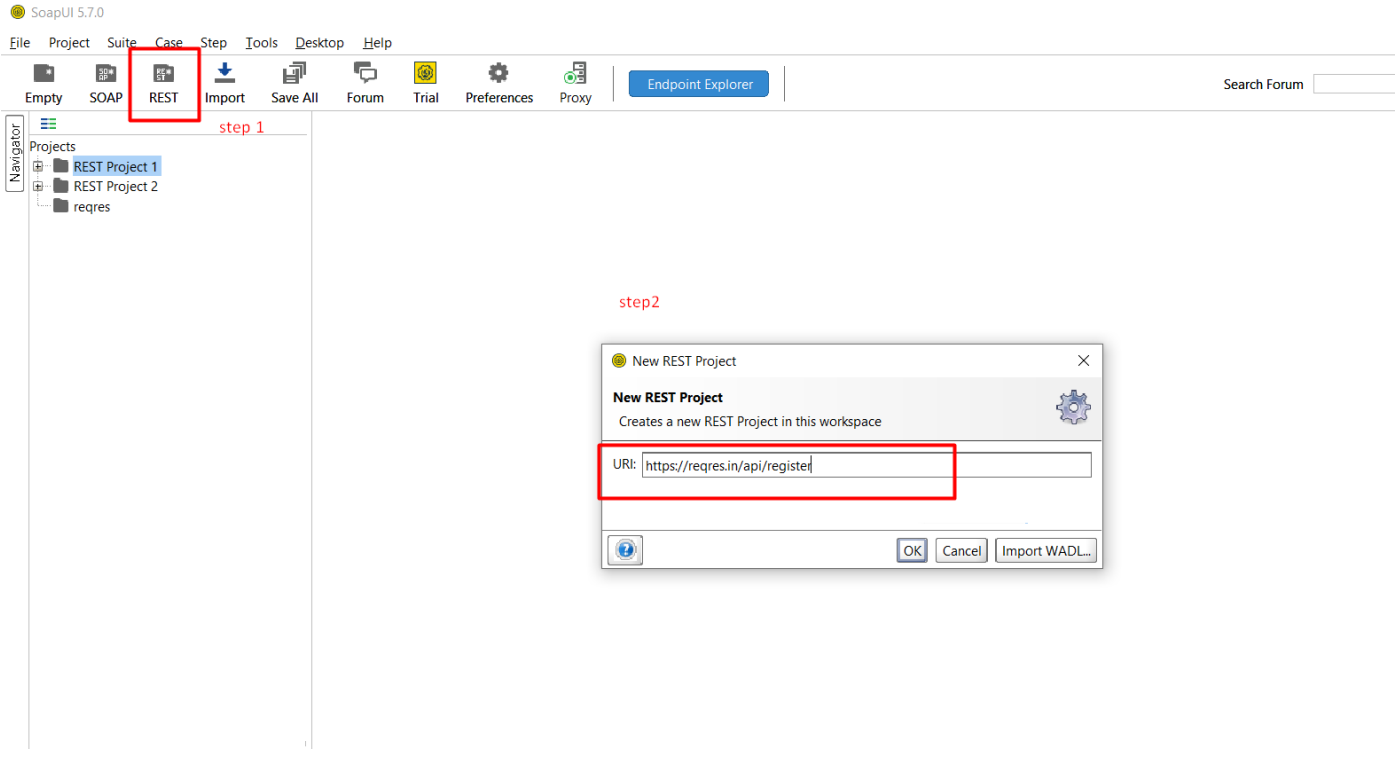

SoapUI 5.7.0
File Project Suite Case Step Tools Desktop Help
Empty SOAP REST Import Save All Forum Trial Preferences Proxy
Endpoint Explorer
Search Forum
Online Help
Navigator
Projects
REST Project 1
https://gorest.co.in
Users [/public/v2/users]
Users 1
Request 1
TestSuite 1
TestCase 1
Test Steps (1)
REST Request
Load Tests (0)
Security Tests (0)
REST Request
Endpoint
https://gorest.co.in
Resource/Method: Users -> Users 1
[/public/v2/users]
Request
Raw
Name
Value
Style
Level
Authorization
QUERY
RESOURCE
Content-Type
QUERY
RESOURCE
XML
JSON
HTML
Raw
Auth
Headers (0)
Attachments (0)
Representations (0)
JMS Headers
JMS Property (0)
Headers (26)
Attachments (0)
SSL Info (3 certs)
Representations (1)
Schema (conflicts)
JMS (0)
Online Help
Contains - VALID
Not Contains - VALID
Inspector
Custom Properties
REST TestRequest Properties

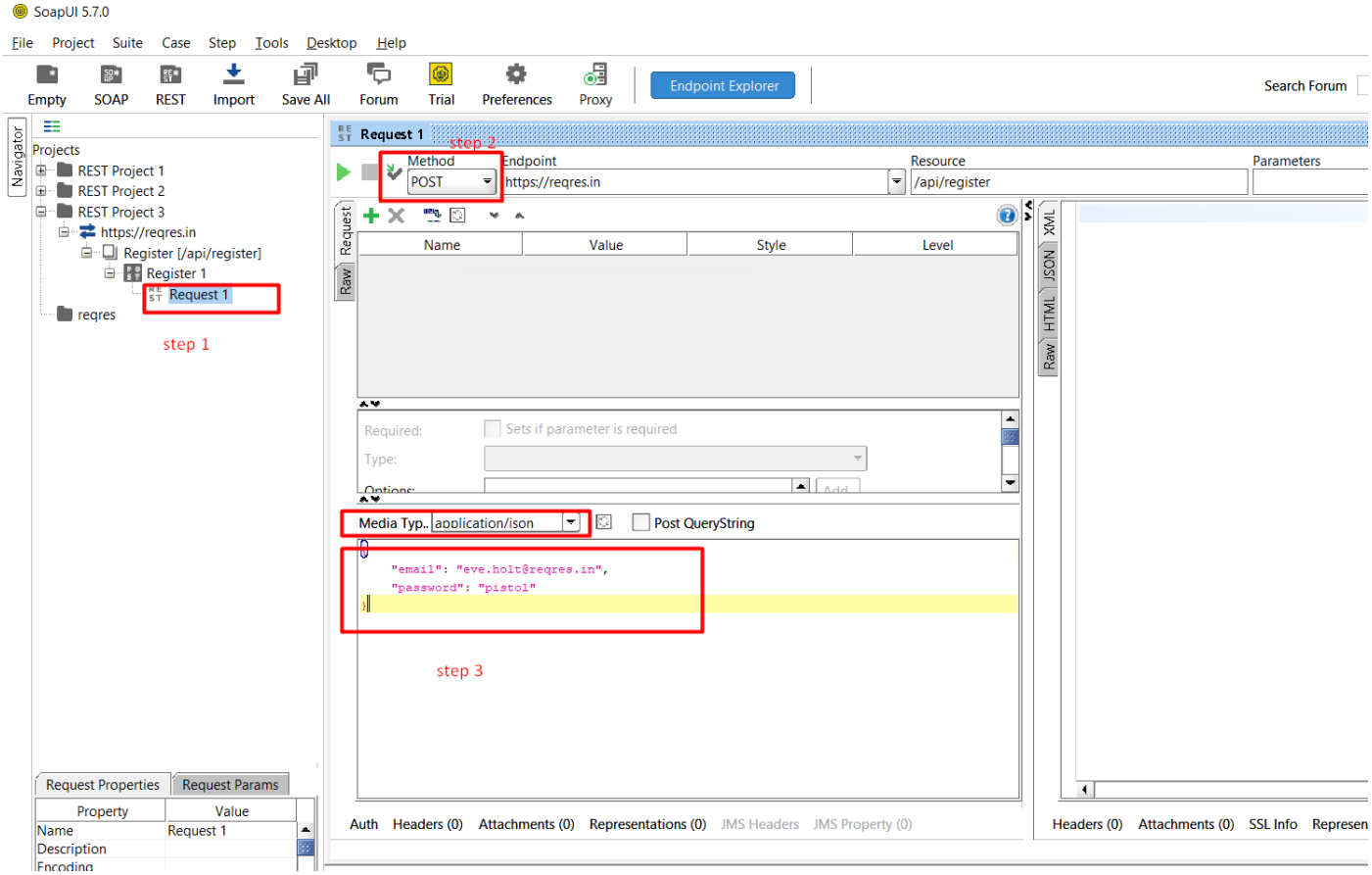

SoapUI 5.7.0
File Project Suite Case Step Tools Desktop Help
Empty SOAP REST Import Save All Forum Trial Preferences Proxy
Endpoint Explorer
Search Forum
Navigator
Projects
REST Project 1
https://gorest.co.in
Users [/public/v2/users]
Users 1
Request 1
TestSuite 1
TestCase 1
Test Steps (1)
REST Request
Load Tests (0)
Security Tests (0)
REST Request
Endpoint
https://gorest.co.in
Resource/Method: Users -> Users 1
[/public/v2/users]
Request
Raw
Name
Value
Style
Level
Authorization
QUERY
RESOURCE
Content-Type
QUERY
RESOURCE
Valid HTTP status codes Assertion
Valid HTTP status codes Assertion
Specify codes
codes: 200
OK
Cancel
Auth
Headers (0)
Attachments (0)
Attachments (0)
SSL Info (3 certs)
Representations (1)
Schema
Contains - VALID
Not Contains - VALID
Valid HTTP Status Codes - FAILED
-> Response status code:200 is not in acceptable list of status codes
Custom Properties
REST TestRequest Properties
Property
Value
Name
REST Request
Description
Encoding
UTF-8
Assertions (3)
Request Log (1)
response time: 152ms (2268 bytes)

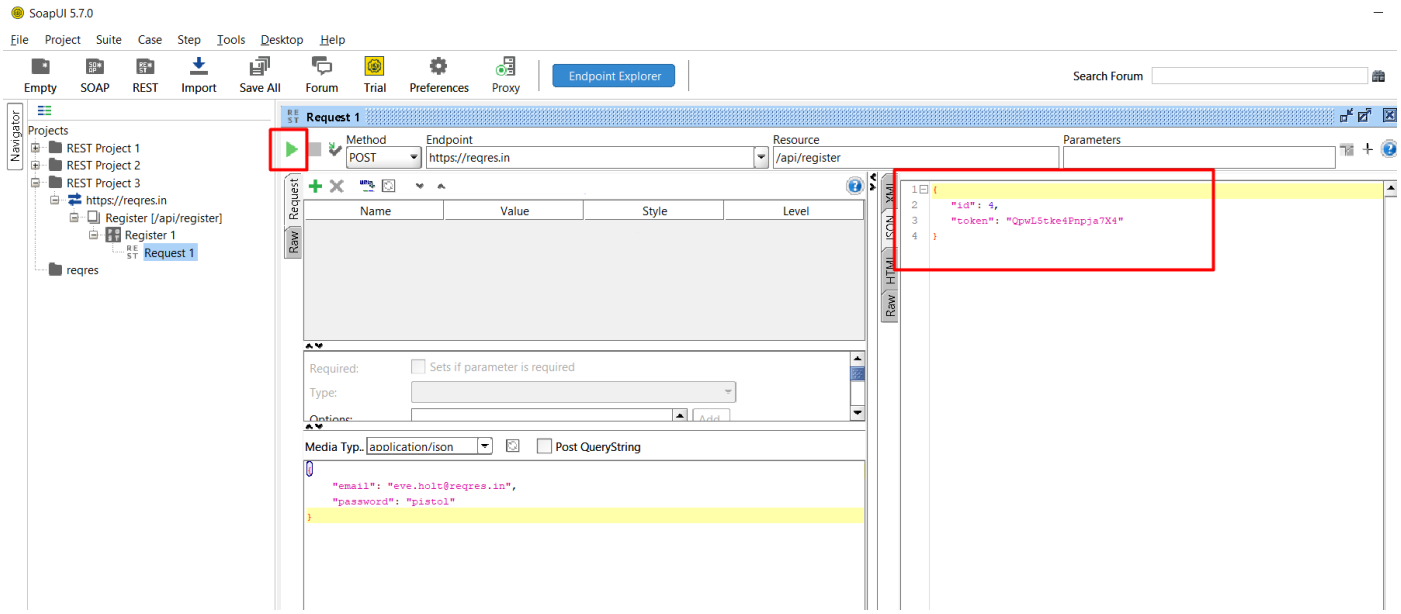

SoapUI 5.7.0
File Project Suite Case Step Tools Desktop Help
Empty SOAP REST Import Save All Forum Trial Preferences Proxy
Endpoint Explorer
Search Forum
Navigator
Projects
REST Project 1
https://gorest.co.in
Users [/public/v2/users]
Users 1
Request 1
TestSuite 1
TestCase 1
Test Steps (1)
REST Request
Load Tests (0)
Security Tests (0)
REST Request
Endpoint
https://gorest.co.in
Resource/Method: Users -> Users 1
[/public/v2/users]
Request
Raw
Name
Value
Style
Level
Authorization
QUERY
RESOURCE
Content-Type
QUERY
RESOURCE
Invalid HTTP status codes Assertion
Invalid HTTP status codes Assertion
Specify codes
codes: 500
Comma-separated not acceptable status codes
OK
Cancel
Auth
Headers (0)
Attachments (0)
Attachments (0)
SSL Info (3 certs)
Representations (1)
Schema (conflicts)
Contains - VALID
Not Contains - VALID
Valid HTTP Status Codes - VALID
Invalid HTTP Status Codes - VALID
Custom Properties
REST TestRequest Properties
Property
Value
Name
REST Request
Description
Encoding
UTF-8
Assertions (4)
Request Log (1)
response time: 152ms (2268 bytes)



How to hit post/put







Same for the PUT(just select put method)- we can apply assertions

here as well(post as well as put)