

AIM : Extract data from JSON file and display output in message box

OBJECTIVES :

1. To create a bot that can extract data from a JSON file and display the output inside message box

TOOLS USED : Automation Anywhere

Problem Statement :

1. Automate the task of extracting data from JSON file and display output in message box.

Controls used :

1. REST Web Services: Post Method

REST Web Services: Post Method action is used to send requests to and receive responses from an API.

2. String: Extract Text

String: Extract Text action is used to extract text from the specific location in the statement, by giving before or after string value.

3. Message Box

Message Box action is used to display the output to the user.

Solution:

Steps to create and execute bot to achieve above task:

Step 1: Create a new bot by clicking on Home page of Automation Anywhere, Click on Create bot option. Give the bot name and click on "Create and edit".

Step 2: Add a **REST Web Services: Post method** control, enter the URL, in content type add the custom parameters and assign the output to a variable **JSON**

REST Web Services: Post method

Creates a new resource in the URI. Parameters are passed in the request body, and there is no limit on the length for a request body.

Required bot agent version: 20.11 or above

URI

☒ Enter the URI

(x)

e.g https://domainname.com/resource/search?param1=value1¶m2=value2

☐ Select credential as URI

Content type

JSON (application/json) ▼

Custom parameters

☒ Enter the parameters

(x)

Enclose values in {{double curly braces}} and add your substitution below

Substitutions (optional)

Name in request body	Substitute with
----------------------	-----------------

REST Web Services: Post method

Assign the output to a variable

(x)

Use response header name as key to fetch header value and 'Body' as key to fetch body of response

Step 3: Add a **Message Box** control to display the output variable created

Message box

Inserts a message box to show a message when the task runs
Required bot agent version: 20.11 or above

Enter the message box window title

Automation Anywhere Enterprise Client (x)

Enter the message to display

JSON{Body}\$ (x)

Scrollbar after lines

30 (x)

☐ Close message box after

Seconds

#

Step 4 : Add a **String: Extract text** control and configure according to the below screenshots

String: Extract text

Extracts a range of text using logical operators from the source string.
Required bot agent version: 20.11 or above

Source string (optional)

JSON{Body}\$ (x)

Get characters

☐ After

Start after text (optional)

Occurrence

#

☒ Before and/or after

Start after text (optional)

JSON{Body}\$ (x)

Occurrence

1 (x)

AND


End before text (optional)

JSON{Body}\$ (x)

Occurrence

1 (x)

E.g. to extract 5250 from "Price is 5250 today" specify "Start after text" as "Price is" and "End before text" as "today" Tip: Press F2 to enter variables

String: Extract text 

If no match found, return

☒ Source String

☐ Empty (null) String

Number of characters to get

☒ All

☐ Only

#

☐ Up to

#

When extracting



☐ Match case


☒ Do not match case

☒ Trim the extracted text (remove blank spaces)


☒ Remove Enter from the extracted text

Assign the output to variable

 Answer 




Step 5 : Add a **Message Box** control to display the extracted text

Message box 


inserts a message box to show a message when the task runs

Required bot agent version: 20.11 or above


Enter the message box window title

☞ Automation Anywhere Enterprise Client 

Enter the message to display


☞ Extracted text : \$Answer\$ 

Scrollbar after lines

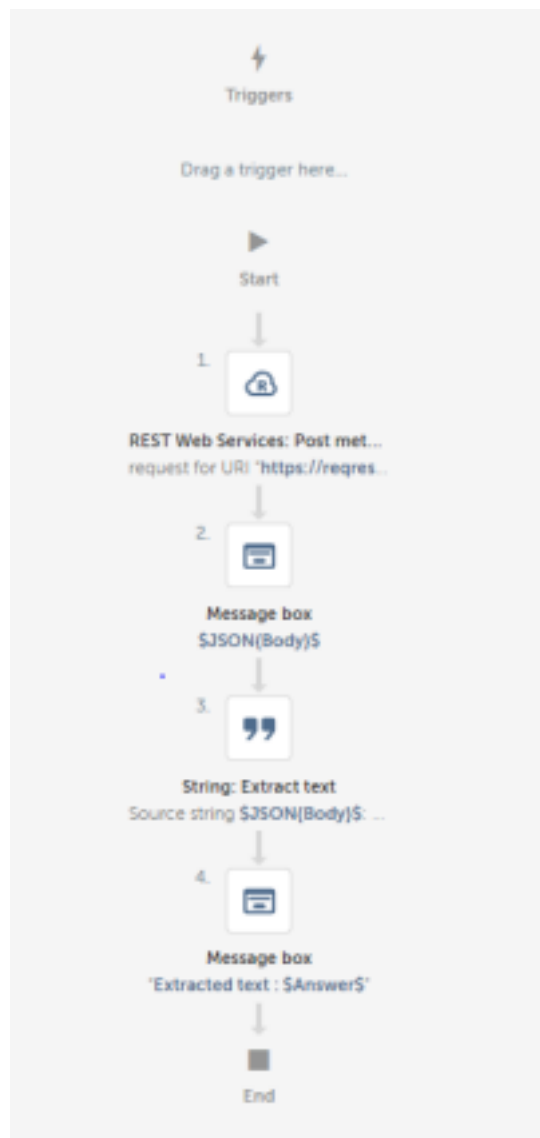
30 

☐ Close message box after

Seconds

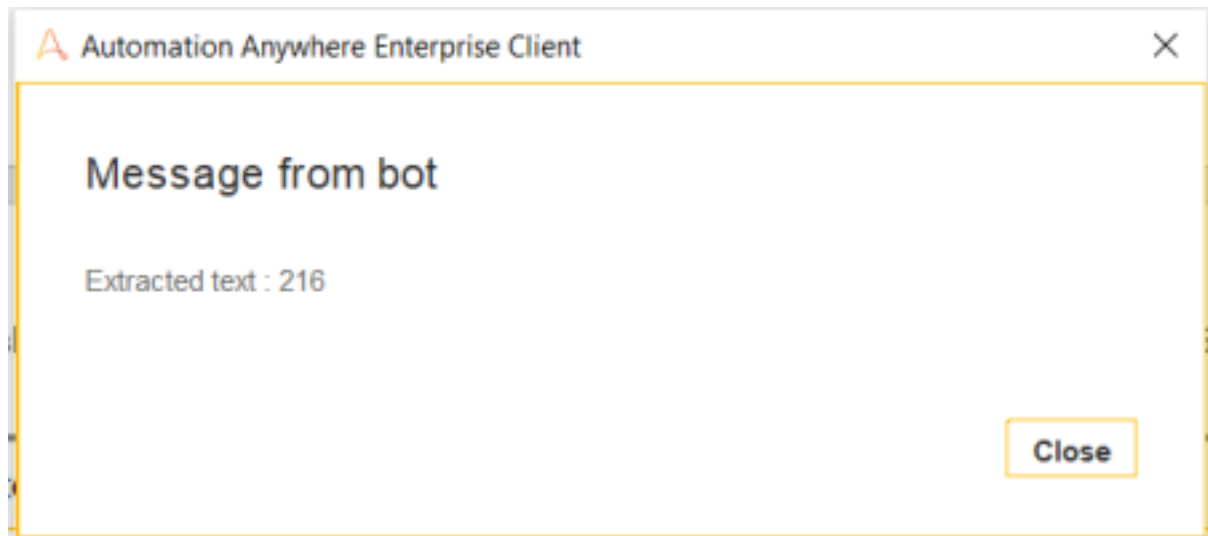


This is how the final flow should look like



Click on RUN

```
regresitsapi/users
[{"page":1,"per_page":1,"total":12,"total_pages":1,"data":[{"id":1,"email":"george.kirby@reqres.in","first_name":"George","last_name":"Kirby","avatar":"https://reqres.in/img/faces/1-image.jpg"}, {"id":2,"email":"janet.conner@reqres.in","first_name":"Janet","last_name":"Conner","avatar":"https://reqres.in/img/faces/2-image.jpg"}, {"id":3,"email":"jane.doe@reqres.in","first_name":"Jane","last_name":"Doe","avatar":"https://reqres.in/img/faces/3-image.jpg"}, {"id":4,"email":"morpheus@reqres.in","first_name":"Morpheus","last_name":"Doe","avatar":"https://reqres.in/img/faces/4-image.jpg"}, {"id":5,"email":"charles.morris@reqres.in","first_name":"Charles","last_name":"Morris","avatar":"https://reqres.in/img/faces/5-image.jpg"}, {"id":6,"email":"tracy.ramos@reqres.in","first_name":"Tracy","last_name":"Ramos","avatar":"https://reqres.in/img/faces/6-image.jpg"}], "support":["url":"https://reqres.in/support","text":"To keep ReqRes free, contributions towards server costs are appreciated!"}]
```

**Observation:**

Through this experiment, I learned how to use Automation Anywhere to create a bot that deals with JSON files. I used the REST Web Services action to communicate with an external service, extracted data from the response using the Extract Text action, and displayed it in a message box for the user. This exercise showed me how Automation Anywhere can interact with different data formats and external services, making automation tasks more versatile and efficient.

