NestedFlow Automation Case Study 1: API Testing

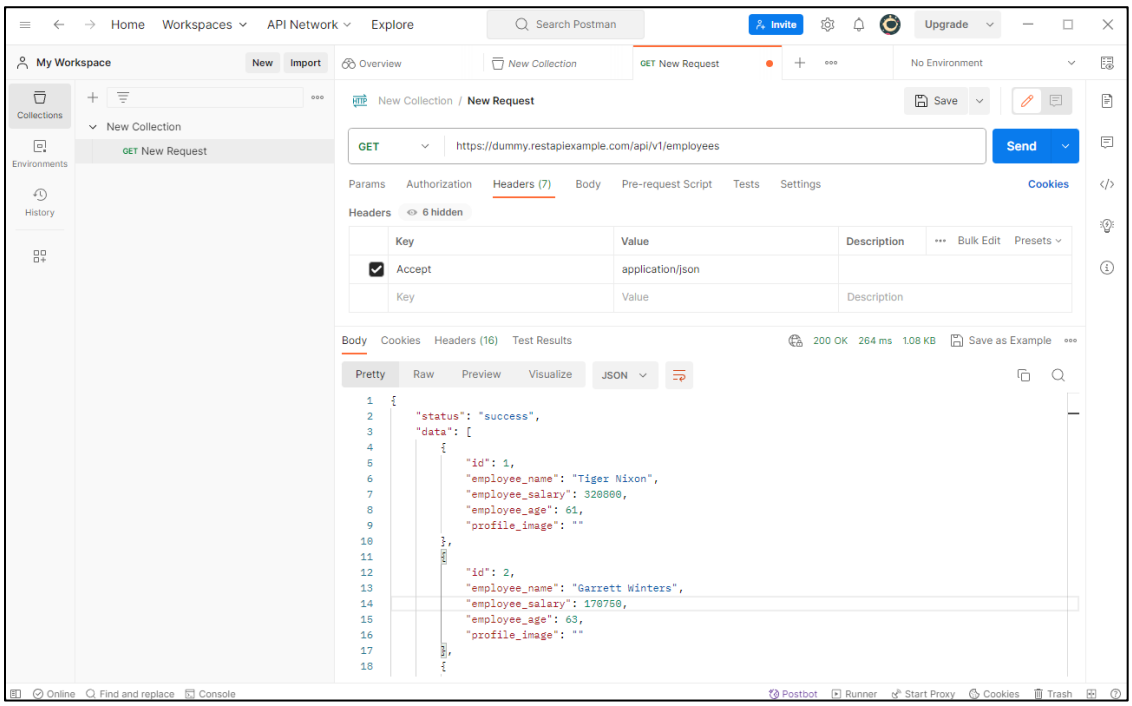
Creation Date: 10/30/2023

Table Of Contents

Introduction

This document describes the integration of API testing blocks with coding blocks to process the output of API tests. This will help build a proper API framework.

Example: https://dummy.restapiexample.com

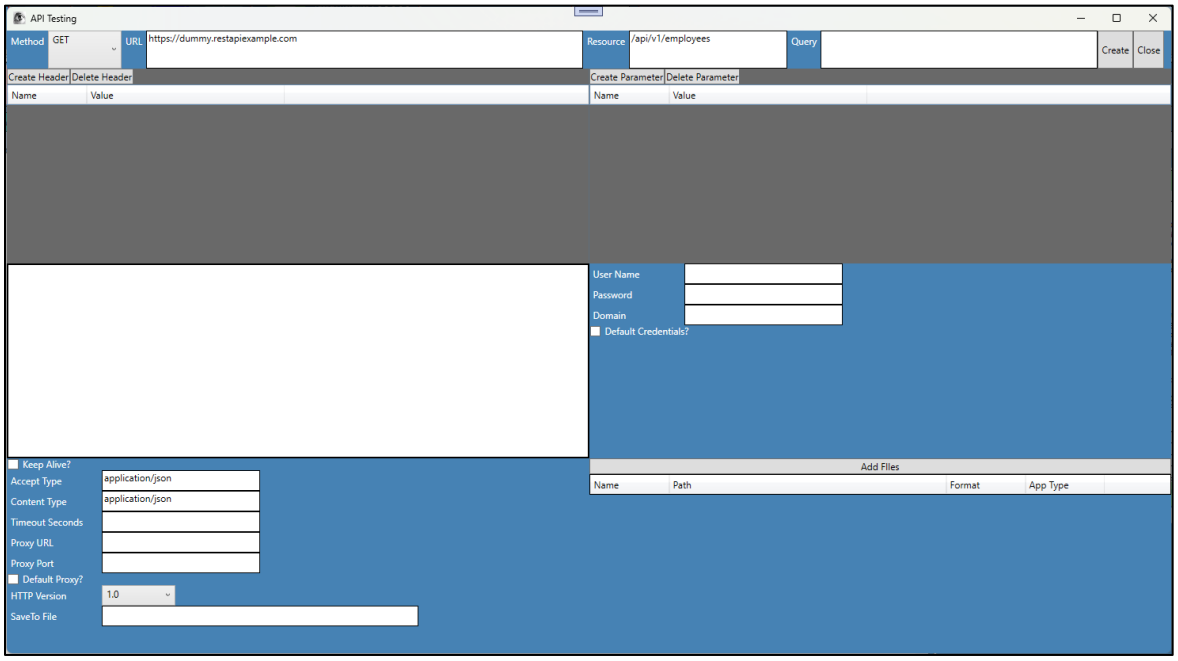


The API response is a JSON which has data about multiple fake members.

The target here is to parse the JSON output and get all Names, Salary and age from the output.

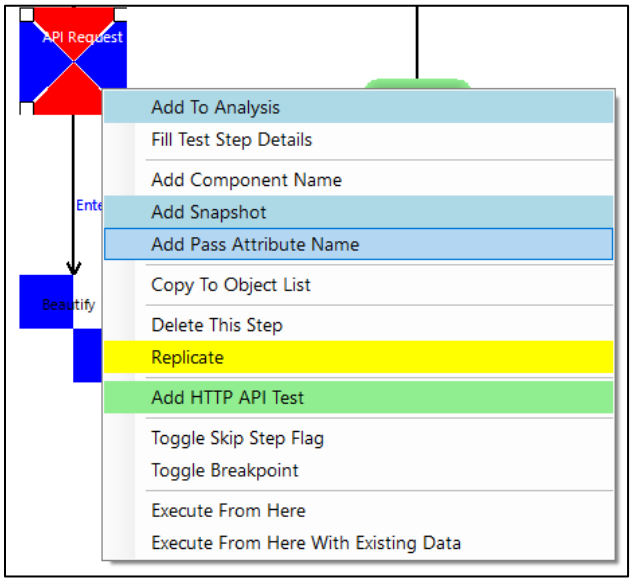
STEP 1: API Call

Right click on the component and click **Add HTTP API Test** option

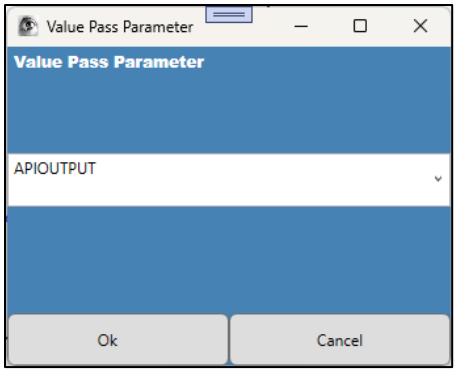


Enter Method, URL, Accept Type etc. as shown above.Click on **Create** button

Now right click on the component and click on **Add Pass Parameter Name**

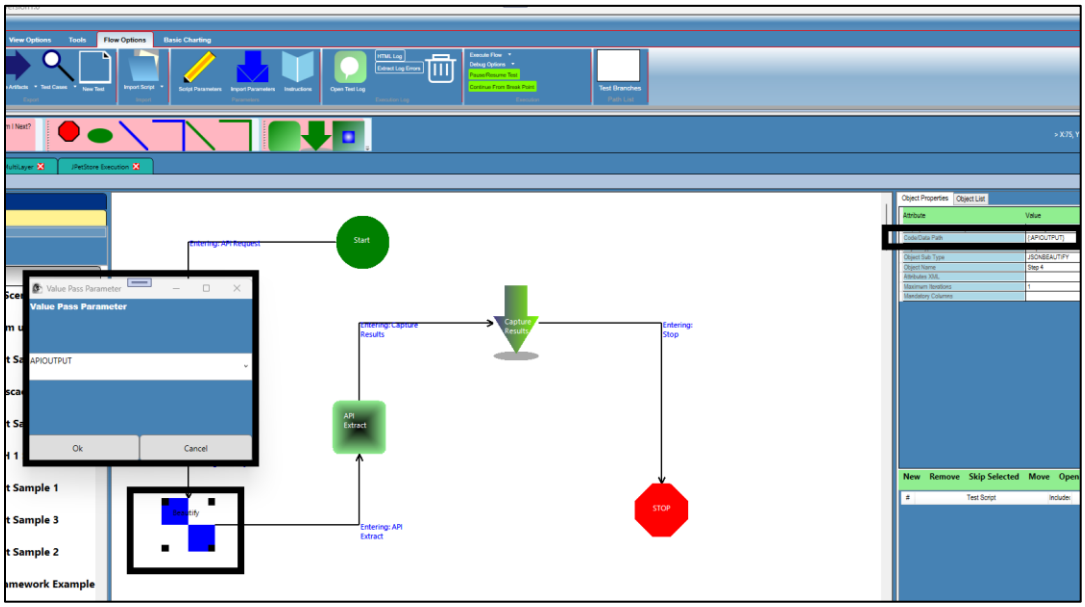


Enter the parameter name and click on **Ok** button.


Value Pass Parameter
Value Pass Parameter
APIOUTPUT
Ok
Cancel

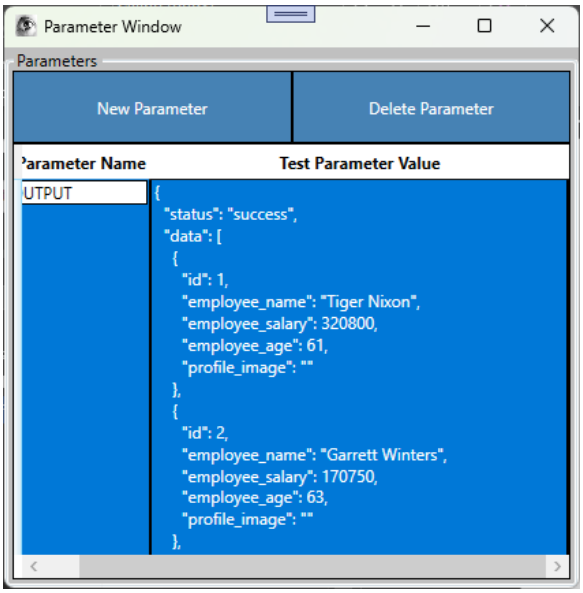
STEP 2: Beautify JSON

The API output will always be unformatted and it is difficult to read it. Hence, it becomes necessary to format the data.



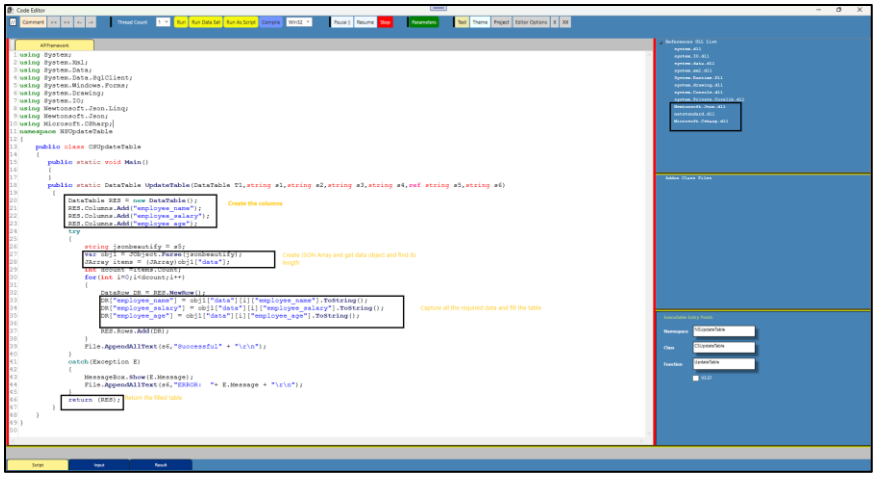
Choose the Beautify JSON component from the **Other Controls** component section and drag to draw panel. Click on the component and double click on Code/Data Path property and enter {:APIOUTPUT} which is the pass parameter created in last step. Also, right click on the component and add a pass parameter and assign the same parameter.

After the execution, the JSON will get converted to beautified string:



STEP 3: Code to Process Data

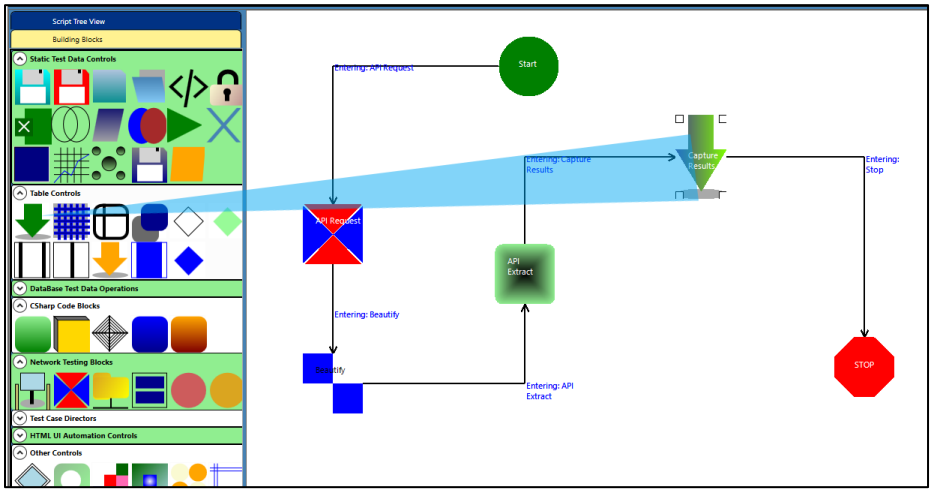
```
using System;
using System.Xml;
using System.Data;
using System.Data.SqlClient;
using System.Windows.Forms;
using System.Drawing;
using System.IO;
using Newtonsoft.Json.Linq;
using Newtonsoft.Json;
using Microsoft.CSharp;
namespace NSUpdateTable
{
  public class CSUpdateTable
  {
    public static void Main()
    {
    }
    public static DataTable UpdateTable(DataTable T1,string s1,string s2,string s3,string s4,ref string s5,string s6)
    {
      DataTable RES = new DataTable();
      RES.Columns.Add("employee_name");
      RES.Columns.Add("employee_salary");
      RES.Columns.Add("employee_age");
      try
      {
        string jsonbeautify = s5;
        var obj1 = JObject.Parse(jsonbeautify);
        JArray items = (JArray)obj1["data"];
        int dcount =items.Count;
        for(int i=0;i<dcount;i++)
        {
          DataRow DR = RES.NewRow();
          DR["employee_name"] = obj1["data"][i]["employee_name"].ToString();
          DR["employee_salary"] = obj1["data"][i]["employee_salary"].ToString();
          DR["employee_age"] = obj1["data"][i]["employee_age"].ToString();

          RES.Rows.Add(DR);
        }
        File.AppendAllText(s6,"Successful" + "\r\n");
      }
      catch(Exception E)
      {
        MessageBox.Show(E.Message);
        File.AppendAllText(s6,"ERROR:  "+ E.Message + "\r\n");
      }
      return (RES);
    }
  }
}
```

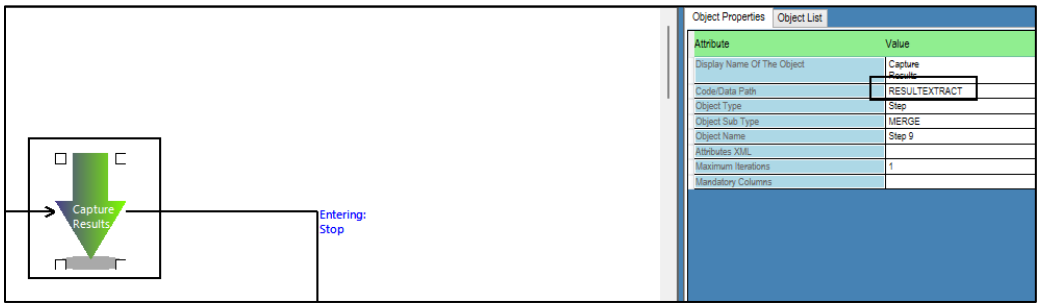


Need to add Micosoft.CSharp.dll, Newtonsoft.Json.Dll and netstandard.dll in references and if any of these dlls not present then add them to %APPDATA% /one/CoreApp folder

STEP 4: Save Data

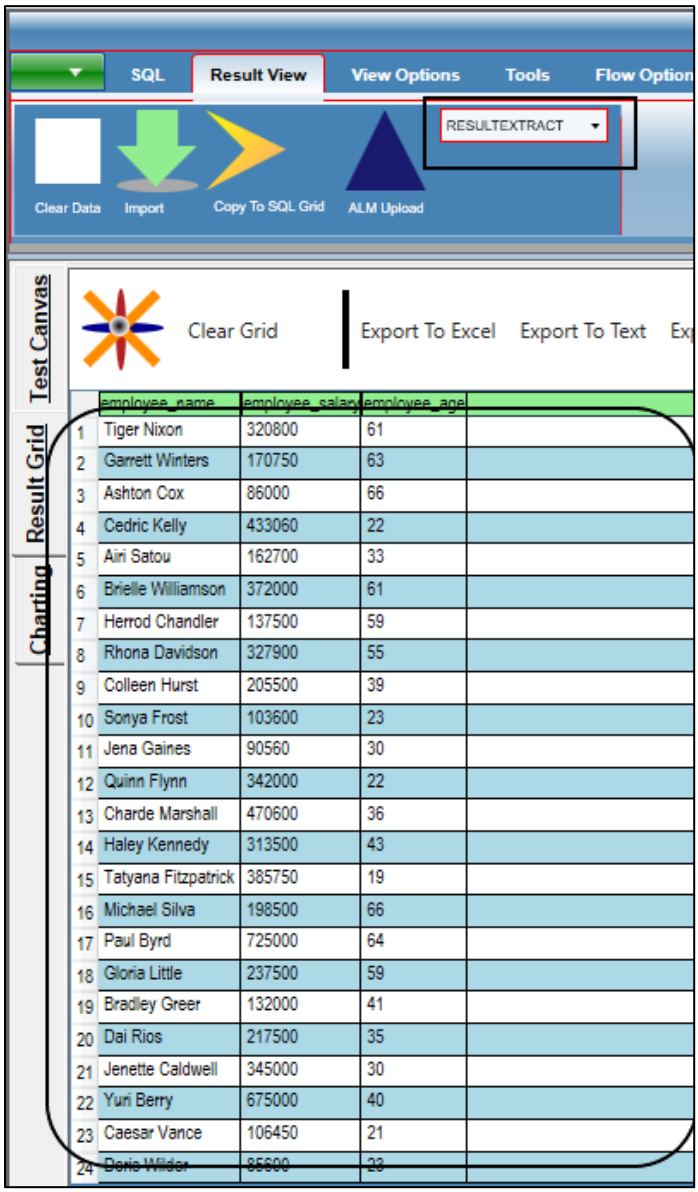
Save the data generated by the code to a data table





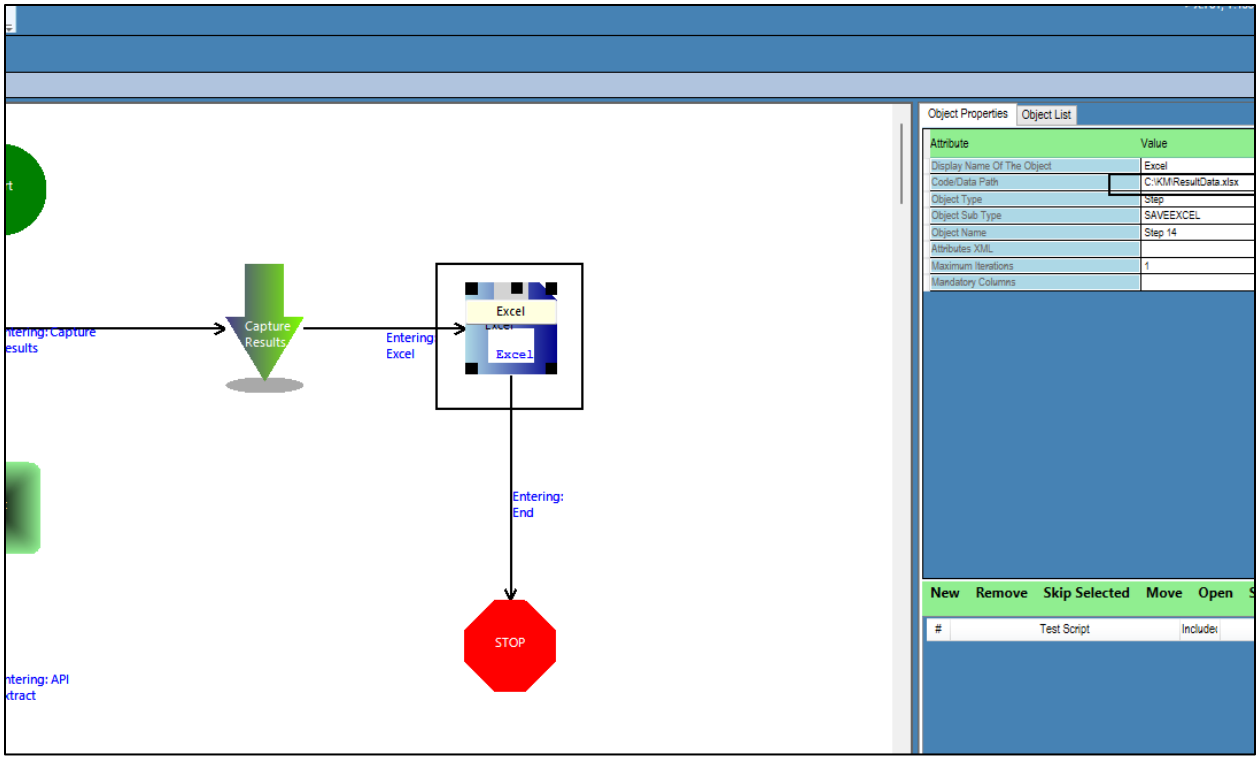
Execution And Results

When the flow execution completes, check the RESULTEXTRACT data table in the **Result Grid**



	employee_name	employee_salary	employee_age
1	Tiger Nixon	320800	61
2	Garrett Winters	170750	63
3	Ashton Cox	86000	66
4	Cedric Kelly	433060	22
5	Airi Satou	162700	33
6	Brielle Williamson	372000	61
7	Herrod Chandler	137500	59
8	Rhona Davidson	327900	55
9	Colleen Hurst	205500	39
10	Sonya Frost	103600	23
11	Jena Gaines	90560	30
12	Quinn Flynn	342000	22
13	Charde Marshall	470600	36
14	Haley Kennedy	313500	43
15	Tatyana Fitzpatrick	385750	19
16	Michael Silva	198500	66
17	Paul Byrd	725000	64
18	Gloria Little	237500	59
19	Bradley Greer	132000	41
20	Dai Rios	217500	35
21	Jenette Caldwell	345000	30
22	Yuri Berry	675000	40
23	Caesar Vance	106450	21
24	Doris Wilder	85600	23

We can also make the data to be saved in Excel/XML or Text files as well.


Capture Results
Entering Excel
Excel
Excel
Entering: End
STOP
Object Properties
Object List
Attribute
Value
Display Name Of The Object
Excel
Code/Data Path
C:\KM\ResultData.xlsx
Object Type
Step
Object Sub Type
SAVEEXCEL
Object Name
Step 14
Attributes XML
Maximum Iterations
1
Mandatory Columns
New
Remove
Skip Selected
Move
Open
#
Test Script