Json-FusionGBS

Readme

Version 1.01

06/04/2020

Table of Contents

1. Introduction

This document contains all essential information for the user to make full use of the Compare Files Metabot. This manual includes a description of the functions and capabilities and step-by-step procedures for setup & configuration of the Bot.

1.1 Overview

JSON Metabot offers following functionality: -

- Simply put this bot is used for converting input Json string to a flat structure.
- JSON is now a days widely used by vendors to exchange information and most often or not all the vendors have different JSON output. In order to process the output of JSON every time a logic is required. This becomes very challenging and at time very complicated and time consuming to achieve.
- This Metabot accepts any valid* json string as input and provide output in two formats
    - 2-D Array: The first row of the array consists of the column header (Key Name), subsequent rows contains the values corresponding to the key.
    - String (Flattened Json): This contains the output as a string, each line of the string is a combination of Key and value.
- This MetaBot can handle any level of nesting, including attributes of string or array type.

1.2 Common Use cases

The Metabot can be used together with other task and or metabots where JSON output requires processing. Hence it is most likely going to be used when bots are required to handle the output of the webservice in JSON format.

**Valid Json: The bot won't be able to correctly create a 2-D array for input Json string when the "object" in json contains different "key" name*

For example:

```
[
 {“id":1, “name”: “Johnson”, “amount":345.33, "Remark":"Ok", "Class”: “Masters”},
 {"id":2,"name":"Sam ","amount":993.44, "Remark":"","Class":"Ph.D."},
 {“id":3, "name":" Smith", "amount":993.44, "Remark”: “Not Ok"}
]
```

The last object does not contain the key named “Class”. This will result in failure to construct 2-D Array; however, you can still use the Flatten Json logic.

2. Requirements & Prerequisites

2.1 System Requirements

For the PC or server where the bot needs to run:

1. RAM: 8GB or higher
2. PROCESSOR: Intel Core i5 or higher.
3. Hard Disk: Up to 2GB of overall free space in the AA Client installation drive.

   Refer to Automation Anywhere recommended guidelines for setting up AAE infrastructure.

2.2 Prerequisites

The following prerequisites are mandatory

1. Automation Anywhere Enterprise Control room 11.3.3 (or above)
2. Automation Anywhere Enterprise Client 11.3.3 (or above)

2.3 Security Measures

N/A

2.4 Disclaimers

N/A

3. Getting Started

3.1 Skill Matrix

The functionality of the Bot has been divided into a set of skills.

Below is an overview of how the task bots and metabots map to these skills:

Skill	Task Files	MetaBot Files
Json Processing	Json.atmx	Json.mbot

3.2 Installation Hierarchy

This section describes the installation hierarchy and the folder structure generated by a bot post installation.

Directory	Files
• <$AAAplicationPath> ○ My Tasks ▪ Bot Store • Json-FusionGBS ○ My Tasks	○ Json.atmx
• <$AAAplicationPath> ○ My Tasks ▪ Bot Store • Json-FusionGBS • Error Folder	○ History-<DD>-<MM>-<YYYY>.Log
• <$AAAplicationPath> ○ My Tasks ▪ Bot Store • Json-FusionGBS • Output Folder	○ History-<DD>-<MM>-<YYYY>.Log
• <$AAAplicationPath> ○ My Tasks ▪ Bot Store • Json-FusionGBS • My Metabots	○ Json.mbot

3.3 Quick Start

3.3.1 Setup

Download the bot from Bot Store.

1. Double click on the .msi file.
2. On Welcome to Installation wizard, click Next to continue.
3. Click I agree to the terms in the license agreement radio button to accept the agreement.
4. Get/Copy the License key from Bot Store Downloads into License Key, click Next to continue.
5. Click Install to begin the installation.
6. Click Finish to complete the installation.
7. To view the installation, go to 'My Tasks' folder on AAE Client to see bot files.

3.3.2 Configuration

This section describes various input parameters required to work with Metabot logic. The sample task bots have been supplied in the package. These task bots are some example of how to use Metabot.

- For Metabot Logic – **JsonToArray (Logic)**

INPUT VARIABLES: Input Variables to be mentioned in this Table				
Variable Name	**Type**	**Mandatory**	**Purpose**	**Example Input**
vJsonString	Text	Yes	Valid JSON String	{ "race" : { "entries" : [{ "id":11,"name":"John" }, { "id":22,"name":"Sam" }] } }

OUTPUT VARIABLES: Output Variables to be mentioned in this Table.				
Variable Name	**Type**	**Mandatory**	**Purpose**	**Example Output**
vResponseCode	Number	Yes	It contains the return code of the logic; its value is set to -1 if error has occurred and 0 is the logic was successful. Use this value If condition	Exit Status, 0 (Success) -1 (Error)
vResponseDescription	Text	Yes	Description for return value, for -1 it has corresponding error description and for 0 it has the success message	Error Description or Output Array (without column header)
arrJsonOutput	Array	Yes	Json Converted to 2-D Array. First Row: Contains the column names (header) Second Row: Onwards contains the Data of Json String	**Id** | **Name** **11** | John **22** | Sam

- For Metabot Logic – **FlattenJson (Logic)**

INPUT VARIABLES: Input Variables to be mentioned in this Table				
Variable Name	**Type**	**Mandatory**	**Purpose**	**Example Input**
vJsonString	Text	Yes	Valid JSON String	{ "race" : { "entries" : [{ "id":11,"name":"John" }, { "id":22,"name":"Sam" }] } }

OUTPUT VARIABLES: Output Variables to be mentioned in this Table.				
Variable Name	**Type**	**Mandatory**	**Purpose**	**Example Output**

vResponseCode	Number	Yes	It contains the return code of the logic; its value is set to -1 if error has occurred and 0 is the logic was successful. Use this value If condition	Exit Status, 0 (Success) -1 (Error)
vResponseDescription	Text	Yes	Description for return value, for -1 it has corresponding error description and for 0 it has the success message	race.entries.0.id: 11 race.entries.0.name: John race.entries.1.id: 22 race.entries.1.name: Sam

4. Reports

N/A

5. Logs

Refer to section 3.2.

6. Troubleshooting & Support

6.1 Support

Please contact support@fusiongbs.com for support related issues.

Appendix A: Record of Changes

No.	Version Number	Date of Change	Author	Notes
1	1.0	09/03/2020	Dilpreet Sohanpal	First Release
2	1.01	06/04/2020	Dilpreet Sohanpal	Additional Error Handling