Read JSON File

Read an array of JSON objects into a SPSS Modeler Table

Product: IBM® SPSS® Modeler

Extension type: Utility

Table of Contents

Description:

This SPSS Modeler 'source' node allows you to read data from a JSON array. This node parses the data from a JSON file and puts it in a table that can be used in SPSS Modeler.

The array is expanded to fit in a SPSS Modeler table. See Example for more information.

Requirements:

- SPSS Modeler v16.0 or later
- R: http://www.r-project.org/
- 'R Essentials for SPSS Modeler' plugin: https://developer.ibm.com/predictiveanalytics/downloads/

Installation:

Close SPSS Modeler. Save the .cfe file in the CDB folder of the IBM SPSS Modeler installation directory for Windows and Linux. The copy should reside in that same folder and not in a sub-folder.

For example, for Windows 7 the default location is "C:\ProgramData\IBM\SPSS\Modeler\16\CDB". If the ProgramData folder is hidden type the path manually.

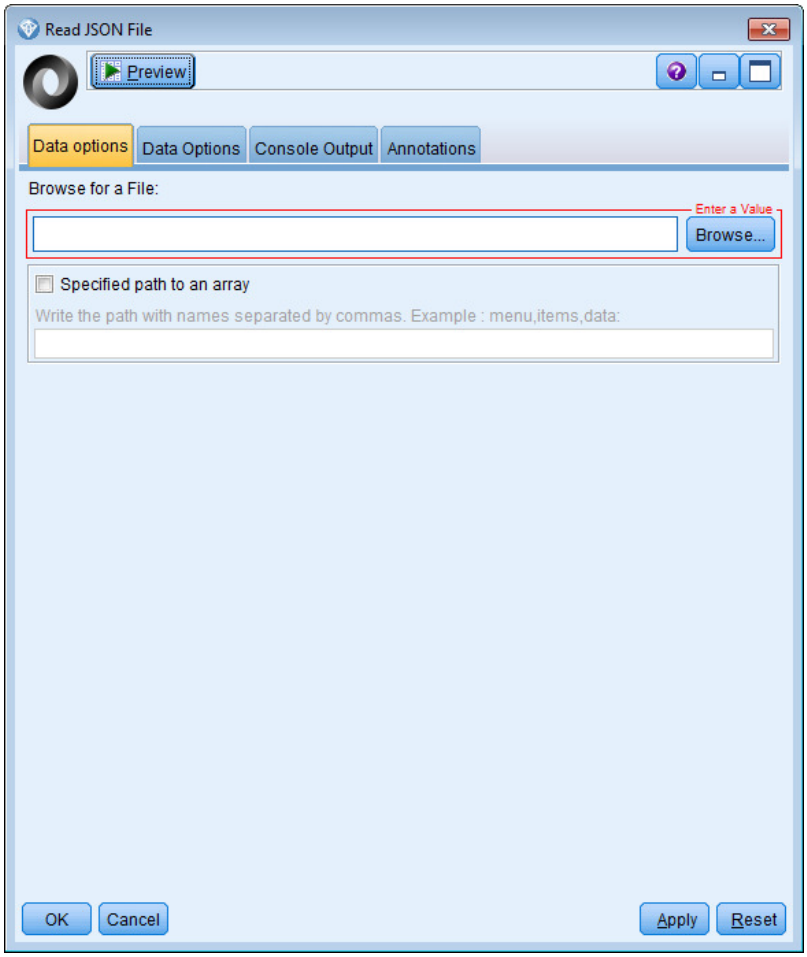
Restart SPSS Modeler: the node will now appear in the Field Ops palette.

R Packages used:

- 'RJSONIO' package created by Duncan Temple Lang http://cran.r-project.org/web/packages/RJSONIO
- 'plyr' package created by Hadley Wickham http://cran.r-project.org/web/packages/plyr

User Interface

- Double click on the node to get to the options. There are the following fields:
  - Browse for a File: click Browse… to find a JSON file to read
  - Specified path to array: Check the box if your JSON file isn't an array. Then, specify the path to the array you want to display in your table. An example of how this should be used can be found in the example starting on page 5.

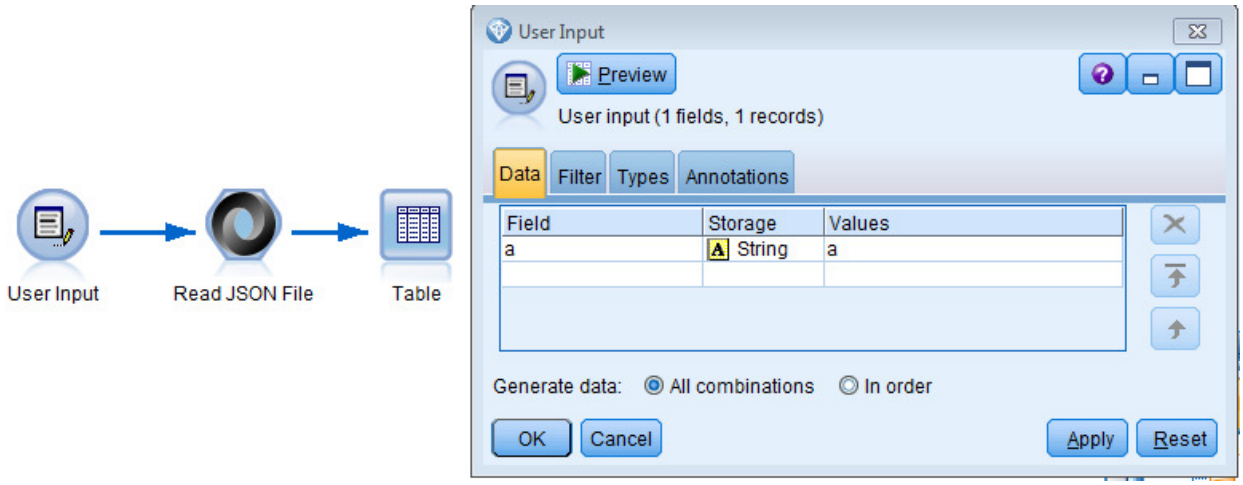


Example

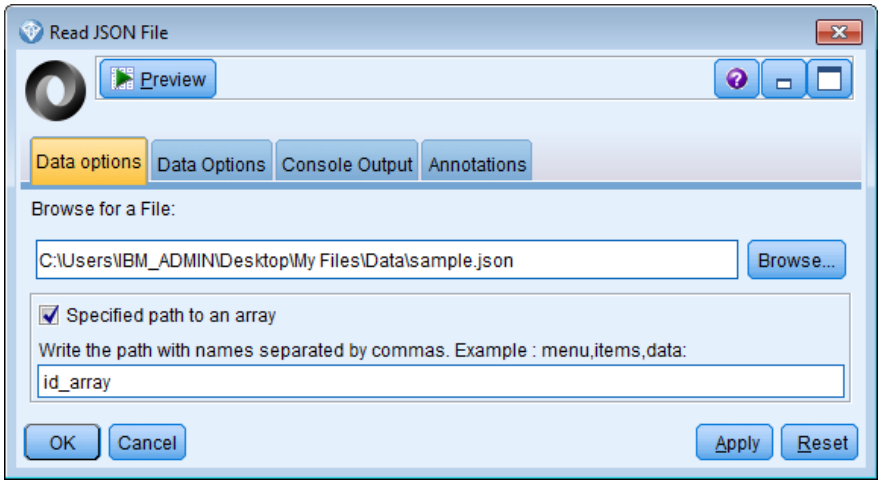
This example will demonstrate how to read a JSON file into SPSS Modeler and output it to a Table. The sample data is included as an appendix to this document as well an explanation of some limitations with this extension. To follow along with this example, copy the JSON data in the appendix and paste it into an empty .json file then save.

User Input

1. In SPSS Modeler v16.0 it is not possible to create proper 'source' nodes with R code. However it is made possible if you trick the software a bit: you just have to put a user input with fake fields before the JSON Import node in order to make the stream start. In this example we put a user input with: Field *a*; Storage *String*; Values *"a"* (see the screenshot below).



2. Double click on the 'Read JSON File' node.



3. Now fill the fields:

   Browse for a File: choose the JSON you want to import your data from

   Specified path to array: Check the box if your JSON file is not an array. Then, specify the path to the array you want to display in your table. In our example the path would be "id_array".

4. Add an output node after the 'Read JSON File' node, in order to visualize the data you gathered. For example you can start with a 'Table'.

5. Click on the table and run the stream

Results and Interpretation

If you used the sample JSON data in this file and followed steps above, your results should match the table below:

	X_id	age	name	likes1	likes2	friends.id	friends.name	friends.id.1	friends.name.1	isActive	weight	likes3
1	53ea1d4734e8189d0385473b	26	Karyn Craig	cats	sleeping	0	Lane Castaneda	1	Cline Mccoy	$null$	$null$	$null$
2	53ea19146bc5d2192ef92733	$n...	Connie Barron	dogs	jogging	0	Claire Phelps	$null$	$null$	FALSE	61.345	music

There are four things to notice here:

1. Data types are correctly transferred: the output table contains strings, integers, doubles... If a data type is not well recognized, the value is coerced to string.

2. All arrays and sub-objects are expanded and indexes are added as needed. Here the array 'friends', containing friend objects (id and name), has been expanded to 4 columns: *friends. id, friends.name, friends.id.1, friends.name.1.* If there had been 3 friends in one of the arrays, there would have been 6 columns after the importation in SPSS.

3. If a field is missing from an object, it is filled with a null value. Here the second object lacks *age* and the first object lacks *isActive* and *weight*. This is also true in arrays. For example the second object has only one friend, so its *friend.id.1* and *friend.name.1* fields are filled with null values.

   This also means that if only one object has a given field, a column will be created for this field which will contain mostly null values. An inconsistent document structure will therefore lead to a SPSS table difficult to interpret.

4. The columns are created in the order in which the fields are found. For example, the second object has 3 "new" fields relative to document 1: *isActive*, *weight* and a third element in the array *likes.* These are

added at the end of the table. The important consequence is that consecutive field in an object may not be consecutive in the table, depending on the other objects. In particular, array elements like *likes1, likes2, likes3* may not be consecutive.

Since SPSS Modeler works with tables, there is necessarily a loss of structure when importing noSQL data with embedded objects and arrays. This is dealt with by expanding every sub-object or sub-array and adding an index number, so that the document fit on one row.

Since the fields can vary from one document to another, NULL fields are often generated in SPSS when merging the results in a single table.

Important Links

Learn

- Learn more about SPSS software.
- Visit developerWorks Business analytics for more technical analytics resources for developers.
- The Comprehensive R Archive Network is the main site for the R project and each R package. The help pages and manuals that are associated with optimx, nlmrt, and Rcgmin are detailed. Numerous references are provided.
- Read "Do I need to learn R?" (Catherine Dalzell, developerWorks, September 2013) to learn why R is a valuable tool for data analytics that was expressly designed to reflect the way that statisticians think and work.
- "Calling R from SPSS" describes how to use R code inside IBM SPSS Modeler 16.
- Read "Create new nodes for IBM SPSS Modeler 16 using R" to learn how to create new extensions easily.

Discuss

- Visit the IBM SPSS Community to share tips and experiences with other IBM SPSS developers.
- Follow developerWorks on Twitter to be among the first to hear about new resources.

Appendix – JSON Sample Data


```
{
 "id_array":[
  {
   "_id": "53ea1d4734e8189d0385473b",
   "age": 26,
   "name": "Karyn Craig",
   "likes": [
    "cats",
    "sleeping"
   ],
   "friends": [
    {
     "id": 0,
     "name": "Lane Castaneda"
    },
    {
     "id": 1,
     "name": "Cline Mccoy"
    }
   ]
  },
  {
   "_id": "53ea19146bc5d2192ef92733",
   "name": "Connie Barron",
   "isActive": false,
   "weight": 61.345,
```

```
    "likes": [
      "dogs",
      "jogging",
      "music"
    ],
    "friends": [
      {
        "id": 0,
        "name": "Claire Phelps"
      }
    ]
  }
 ]
}
```