I Wish I Knew How To …

Begin Programming

JSON with Xojo

November 2015 Edition (1.1)

By Eugene Dakin

Table of Contents

If the user successfully selected a file then the JSONFile Exists argument will be true. Loading a JSON file has the same steps as loading a text file, a TextInputStream variable is created and a string variable (JSONData) is created to hold the string data from the file.

A try-catch statement is used to try and execute code, and if there is a problem (error) then the catch portion of the statement is executed. TextInputStream calls the open method for the JSONfile and then places the raw string data into the JSONStream variable. All of the data is read from the JSONStream and is placed in the string JSONData. Larger files may take longer to load, which is why loading with a stream of data is important.

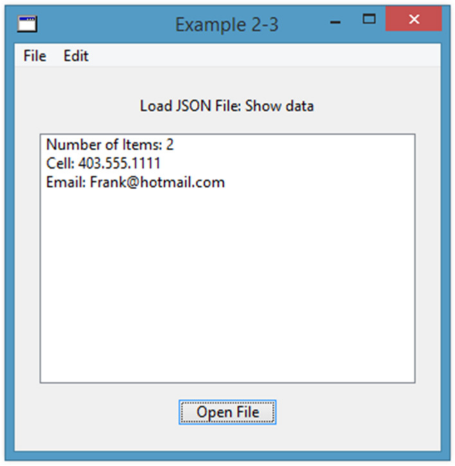
All rows in Listbox1 are deleted which removes previous data if Listbox1 was pressed twice. The JSON item uses the Count function to provide the number of JSON Name:Value pairs and the ToText portion converts the data to a text format which is viewed in Listbox1.

This example shows how to count the number of Name:Value pairs and show the result in a Listbox.

Show JSON Data

This example shows how to display the Name:Value pair in a listbox with Xojo. The sample file contains only Parent data without Child or GrandChild data, meaning there is only one node. Opening the JSON file SimpleList will give the following information in Example 2-3.

Figure 5. Example 2-3: Showing JSON Data in a Listbox



```
Number of Items: 2
Cell: 403.555.1111
Email: Frank@hotmail.com
```

Using code from the previous example, the pushbutton action event opens a folder item for the user to choose the JSON file. When the JSON file is selected and opened, the number of items in the parent node are counted (2) and the two Name:Value pairs are shown in Listbox1.

Code 9. SimpleList.JSON File

```
{
  "Cell": "403.555.1111",
  "Email": "Frank@hotmail.com"
}
```

There are two Name:Value pairs that are separated by a comma and all the data is surrounded by two curly brackets.

Xojo code to read the node is shown below.

Code 10. Example 2-3: Showing Parent Node Information

```
//Holds JSON data
Dim JSONData as String

//Get JSON File
Dim JSONFile as FolderItem
JSONFile = GetOpenFolderItem("")

'Check to see if the file exists
If JSONFile.Exists Then
  'Try to load the file
  Dim JSONStream as TextInputStream
  Try //loading the file data
    JSONStream = TextInputStream.Open(JSONFile)
    JSONData = JSONStream.ReadAll
    JSONStream.Close
  Catch e as JSONException
    MsgBox("Error: " + e.Message)
    Return
  End
Else
  MsgBox("JSON file does not exist")
  Return
End If
```

```
//Load the JSON data
Dim js as New JSONItem(JSONData)
//Show number of items
Listbox1.AddRow  "Number of Items: " + js.Count.ToText

//Show JSON Name/Value pairs
Dim JSONName, JSONValue as String
Dim i as Integer
For i = 0 to js.Count-1
  JSONName = js.Name(i) //Get Name
  JSONValue = js.Value(JSONName) //Get Value
  Listbox1.AddRow JSONName + ": " + JSONValue //Show both
Next
```

The first few lines of code create a variable (JSONFile) from a FolderItem. The GetFolderItem command opens the folder item so the user can choose and select a file. In this example find and select the SimpleList JSON file that is in the Chapter 2 folder.

If the user successfully selected a file then the JSONFile Exists argument will be true. Loading a JSON file has the same steps as loading a text file, a TextInputStream variable is created and a string variable (JSONData) is created to hold the string data from the file.

A try-catch statement is used to try and execute code, and if there is a problem (error) then the catch portion of the statement is executed. TextInputStream calls the open method for the JSONfile and then places the raw string data into the JSONStream variable. All of the data is read from the JSONStream and is placed in the string JSONData. Larger files may take longer to load, which is why loading with a stream of data is important.

All rows in Listbox1 are deleted which removes previous data if Listbox1 was pressed twice. The JSON item uses the Count function to provide the number of JSON Name:Value pairs and the ToText portion converts the data to a text format which is viewed in Listbox1.

Two string variables are created to hold the name and value of each data pair. The name is first required because retrieving the value requires the name. A for-next loop is created from zero to the number of parent items with the Count function. The first name in the array (js.Name(i)) is passed to the JSONName string. Using the JSONName string, the value (js.Value(JSONName)) is then placed in the string variable JSONValue. A row is added to the listbox with the name (JSONName) and the value of the name (JSONValue). The for-next loop continues until the last JSON node has been read and the Name:Value data pair shown in the listbox.

This example shows how to read and show the data from a single JSON node which has two data pairs.

Index

The 'I Wish I Knew' series contains technical data and advice that makes sense and contains practical and numerous examples with explanations to allow you to ease into the steep programming curve. You can extend Xojo applications today!

This book "I Wish I Knew How to … Begin Programming JSON with Xojo" shows you how to create, parse, and read JSON data. There is much JSON data that is free, and for a fee there is a very large repository of information. This book is more than a cheat sheet, it provides an introduction to the basics of working with JSON in Xojo.

The book is written as a guide and reference to Xojo programmers who program Desktop Applications in Windows, Mac, or Linux.

There are 8 chapters and contains over 170 pages with over 50 example programs. Examples were tested with Xojo 2015 r2.4 and Windows 8.1, 10, OSX Yosemite 10.10.4, and Ubuntu 15.04-32bit.

Examples include parsing, creating, walking through children, arrays and more. Both Classic and New Framework examples are provided. Many screenshots have been added to show the results of the code with an index to help find topics quickly.

This is one of many books at Great White Software. This book can be purchased at http://great-white-software.com/rblibrary/ where many great Xojo resources are available.

Happy programming!

Eugene

Eugene Dakin MBA, Ph.D., P.Chem., is an author of Xojo and Real Studio reference materials and has many years of experience in the programming industry. Another great reference book is *I Wish I Knew How To … Program SQLite*.

ISBN: 978-1-927924-13-6