

Rexx/JSON

2019 International Rexx Symposium
Hursley, United Kingdom

Mark Hessling
25 September 2019



Rexx/JSON

- Enables a Rexx program to manipulate JSON objects
- Enables direct and compatible interface to web applications that use JavaScript
- Provides a mechanism to use data objects with a "classic" Rexx program



Rexx/JSON - JSON

- JSON is JavaScript Object Notation
 - Human readable data interchange format
 - A collection of attribute/value pairs
 - Light weight replacement for XML
- Datatypes
 - String (in double quotes)
 - Number
 - Boolean (true or false)
 - Array (ordered list within []) - can be nested
 - Object (unordered list within { }) - can be nested



Rexx/JSON - Example

- Text representation of "*_mark*" object

```
{
  "name": "Mark", "country": "AU",
  "pets":
  [
    {
      "name": "Donal", "age": 10
    },
    {
      "name": "Jenna", "age": 11
    }
  ]
}
```

- In JavaScript the age of Donal is: `mark.pets[0].age`



Rexx/JSON - Create Items

- Create and populate the "*_mark*" object using Rexx/JSON

```
_mark = JSONCreateObject()  
Call JSONAddStringToObject _mark, 'name', 'Mark'  
Call JSONAddNumberToObject _mark, 'id', 23  
_pets = JSONCreateArray()  
_donal = JSONCreateObject( 'name', 'Donal' )  
Call JSONAddNumberToObject _donal, 'age', 10  
Call JSONAddObjectToArray _pets, _donal  
_jenna = JSONCreateObject( 'name', 'Jenna' )  
Call JSONAddNumberToObject _jenna, 'age', 11  
Call JSONAddObjectToArray _pets, _jenna  
Call JSONAddArrayToObject _mark, 'pets', _pets
```



Rexx/JSON - Conversion

- Stringifying is to convert a JSON object to a compact string format for transmission

```
Say JSONStringify( _mark )
```

```
{"name":"Mark","id":23,"pets":  
[{"name":"Donal","age":10}, {"name":"Jenna","age":11}]}
```

- Printing is to convert a JSON object to a formatted string much like Example slide

```
Say JSONPrint( _mark )
```



Rexx/JSON - Conversion

- Parsing is converting a JSON "*stringified*" object to a Rexx/JSON object

- `_json = JSONParse(string)`

```
_mark1 = JSONParse( {"name": "Mark", "id": 23, "pets":  
[ {"name": "Donal", "age": 10}, {"name": "Jenna", "age": 11} ] } )
```

- *The _mark1 object will have the same contents as if we had used the JSONCreatexxx() functions above*



Rexx/JSON - Extracting

- Various functions for extracting items from a JSON object or array

- JSONGetObjectItem(object, item)

```
_name = JSONGetObjectItem( _mark, 'name' )
```

The _name object contains details about the 'name' item which is a string object

- JSONGetObjectItemValue(object[, item])

```
name = JSONGetObjectItemValue( _name )
```

is equivalent to:

```
name = JSONGetObjectItemValue( _mark, 'name' )
```

name = 'Mark'



Rexx/JSON - Extracting

- JSONGetObjectItem(object, item)

```
_pets = JSONGetObjectItem(_mark, 'pets' )
```

The _pets object is an array containing both Jenna and Donal objects

- JSONGetArrayItem(array, index)

```
_donal = JSONGetArrayItem(_pets, 1 )
```

- *_donal is an object containing details about Donal*



Rexx/JSON - Extracting

- JSONGetArraySize(array)

```
count = JSONGetArraySize(_pets ) -- count = 2
Do i = 1 To JSONGetArraySize( _pets )
    _pet = JSONGetArrayItem( _pets, i )
    Say 'Pet age:' JSONGetObjectItemValue( _pet, 'age' )
End
```

Can also be used on objects

```
count = JSONGetArraySize(_mark ) -- count = 3
```



Rexx/JSON - Miscellaneous

- JSONDatatype()

- return datatype of item

```
type = JSONDataType( _mark ) -- 'object'
```

- JSONDelete()

- Deletes all memory associated with the specified items and all items owned by the item

```
Call JSONDelete _mark
```



Rexx/JSON - Helper Functions

- JSONArrayToStem()

- convert JSON Array to Rexx "array"

```
count = JSONArrayToStem( _pets, '_pet.' )  
/*_pet.1 will be a JSON object */
```

- JSONObjectToStem()

- convert JSON Object to Rexx compound variables

```
count = JSONObjectToStem( _pet.2, 'jenna.' )  
/* jenna.name = 'Jenna', jenna.age = '11' */
```



Rexx/JSON - Helper Functions

- JSONCreateArrayFromStem()

- convert Rexx "array" to JSON object

```
id.0 = 2; id.1 = 3; id; id.2 = 5;
```

- `_arr = JSONCreateArrayFromStem('id.')`

- JSONCreateObjectFromStem()

- convert Rexx compound variables to JSON Object

```
pet.age = 11; pet.name = 'Jenna';
```

```
_obj = JSONCreateObjectFromStem( 'pet.' )
```

- All JSON items created are strings



Questions

