JSON Query in FileMaker

Recently we got a request for JSONPath queries in FileMaker, so we looked into various ways on how to implement this. We found a way and got three new functions for you:

- MBS("JSON.Query"; json; query; flags)
- MBS("JSON.Search"; json; path)
- MBS("JSON.Replace"; json; path; ReplaceJSON)

Please try them, e.g. run a query against them.

Let us take this JSON:

```
{ "store": {
    "book": [
      { "category": "reference",
        "author": "Nigel Rees",
        "title": "Sayings of the Century",
        "price": 8.95
      },
      { "category": "fiction",
        "author": "Evelyn Waugh",
        "title": "Sword of Honour",
        "price": 12.99
      },
      { "category": "fiction",
        "author": "Herman Melville",
        "title": "Moby Dick",
        "isbn": "0-553-21311-3",
        "price": 8.99
      },
      { "category": "fiction",
        "author": "J. R. R. Tolkien",
        "title": "The Lord of the Rings",
        "isbn": "0-395-19395-8",
        "price": 22.99
      }
    ],
    "bicycle": {
      "color": "red",
      "price": 19.95
    }
  }
}
```

JSON.Search

Let's try JSON.Search, which takes a JMESPath path in a different query language, but can do other things like:

MBS("JSON.Search"; Test::JSON; "store.book[?price == '8.99']"; 0)
this finds the book entry matching the price:

```
[
    {
        "category": "fiction",
        "author": "Herman Melville",
        "title": "Moby Dick",
        "isbn": "0-553-21311-3",
        "price": 8.99
    }
]
```

or the following query, which looks for books below $10, takes the title and then sorts them:

MBS("JSON.Search"; Test::JSON; "store.book[?price < `10.0`].title | sort(@)"; 0)
returns an array with two names:

```
[
    "Moby Dick",
    "Sayings of the Century"
]
```

JSON.Query

Let's now try JSON.Query with a few queries and check results:

"$.store.book[?(@.price < 10)].author"
in a calculation looks like this:

MBS("JSON.Query"; Test::JSON; "$.store.book[?(@.price < 10)].author"; 0)
This query finds the two books with a price below 10 USD:

```
[
    "Nigel Rees",
    "Herman Melville"
]
```

Inside the query, you can make checks like for whether a key exists:

"$..book[?(@.isbn)].title"
and return the entries which have the key:

```
[
    "Moby Dick",
    "The Lord of the Rings"
]
```

Or find keys with a specific value, even including regular expressions:

"$.store.book[?(@.author =~ /Evelyn.*?/)]"
This looks for authors starting with Evelyn in name on the beginning:

```
[
    {
        "category": "fiction",
        "author": "Evelyn Waugh",
        "title": "Sword of Honour",
        "price": 12.99
    }
]
```

JSON.Replace

Let's try to replace something with our JSON.Replace function. We can run a query and replace a value.

MBS("JSON.Replace"; Test::JSON; "$.store.book[?(@.title == 'Sayings of the Century')].price"; 123)
This writes the new value there into the price field.

So we find an entry in the book list by title and then replace the value. Now you can just go down the structure and pick a note to replace:

MBS("JSON.Replace"; Kontakte::Feld; "$.store.bicycle.price"; 123)
or replace all prices:

MBS("JSON.Replace"; Kontakte::Feld; "$.store.book[*].price"; "3.5")
Please try and let us know.