

# JSON: use jq to edit specific values in nested arrays

Asked 8 years, 4 months ago   Modified 8 years, 4 months ago   Viewed 9k times



I'm trying to update values within an array inside an array using the utility jq. I've pasted the sample json below.

3



More specifically: Within the sheets array, and then within the formulas array, I'd like to change each columnName with a value of "MONTH" to "YEAR". I'd like to do the same for within the sheets array, within the columnStyles array, change each incidence of "MONTH" also to "YEAR"



This jq filter gets me the list of columnNames.

```
.sheets[1] | .formulas[] | .columnName
```

How can I edit the entire file in place by just updating the values I want? Do I use map with if?

And what if I wanted to edit a portion of a value? For example, in a formulaString property, just changing the part of the string that contains MONTH but leaving the rest intact?

```
{
  "version": "6.1.1",
  "className": "xyz",
  "sheets": [
    {
      "name": "Pass1",
      "sheetId": "95e6c2cd-abbe-46c1-8012-bdf37438b9b7",
      "keep": true,
      "formulas": [
        {
          "columnName": "SAMPLE_PROVIDER",
          "columnId": "0",
          "columnIndex": 0,
          "formulaString": "\u003dGROUPBY(#Raw!SAMPLE_PROVIDER)"
        },
        {
          "columnName": "MONTH",
          "columnId": "1",
          "columnIndex": 1,
          "formulaString": "\u003dGROUPBY(#Raw!MONTH)"
        }
      ],
      "columnStyles": [
        {
          "columnId": "0",
          "name": "SAMPLE_PROVIDER",
          "width": 206,
```

```

        "thousandSeparator": true
      },
      {
        "columnId": "1",
        "name": "MONTH",
        "width": 100,
        "thousandSeparator": true
      }
    ],
    "nextColumnId": 2
  },
  {
    "name": "Transform1",
    "sheetId": "49071c1c-fa84-4ae3-92c1-b63175a6b26c",
    "keep": true,
    "formulas": [
      {
        "columnName": "SAMPLE_PROVIDER",
        "columnId": "0",
        "columnIndex": 0,
        "formulaString": "\u003d#Pass1!SAMPLE_PROVIDER"
      },
      {
        "columnName": "MONTH",
        "columnId": "1",
        "columnIndex": 1,
        "formulaString": "\u003d#Pass1!MONTH"
      }
    ],
    "columnStyles": [
      {
        "columnId": "0",
        "name": "SAMPLE_PROVIDER",
        "width": 179,
        "thousandSeparator": true
      },
      {
        "columnId": "1",
        "name": "MONTH",
        "width": 100,
        "thousandSeparator": true
      }
    ],
    "nextColumnId": 3
  }
],
"advancedSchedulingInUse": true,
"errorHandlingMode": "IGNORE"
}

```

arrays json jq

Share

Improve this question

Follow

edited Aug 16, 2016 at 18:25



chepner

529k ● 76 ● 581 ● 731


asked Aug 16, 2016 at 18:07



bweiss

33 ● 1 ● 3

### 3 Answers

Sorted by: Highest score (default) 



5



To change the `columnName` field in the desired containers, you can use

```
jq '(.sheets[] | .formulas[]? | .columnName | select(=="MONTH")) |= "YEAR"'  
tmp.json
```

(The `?` avoids an error if there is no key `formula`.)

To replace `MONTH` with `YEAR` in formula strings, replace each `formulaString` value with a possible modified string returned by `sub`.

```
jq '(.sheets[] | .formulas[]? | .formulaString) |= sub("MONTH"; "YEAR")'  
tmp.json
```

(`sub` requires `jq` 1.5, compiled with the Oniguruma library.)

To combine these into a single `jq` filter? I'm not sure yet; I have only a tenuous understanding of why either one alone works.

Share

edited Aug 16, 2016 at 19:27

answered Aug 16, 2016 at 18:26

Improve this answer



[chepner](#)


529k ● 76 ● 581 ● 731

Follow

Which version of `jq` are you using? – [hek2mgl](#) Aug 16, 2016 at 18:40

Version 1.5 (which is required for the `sub` function). – [chepner](#) Aug 16, 2016 at 18:42

Ah I see, after using the *right* syntax, meaning `sub(a; b)` instead of `sub(a, b)` I got: `jq` was compiled without ONIGURUMA regex library. `match/test/sub` and related functions are not available. – [hek2mgl](#) Aug 16, 2016 at 18:42

I think the semicolon is used to separate arguments because `,` is an operator that feeds its input to the filters on its left and right (if that's what you mean). (Since `"MONTH"` and `"YEAR"` are just filters that ignore their input and return strings.) – [chepner](#) Aug 16, 2016 at 18:45 

Ah! I remember that now. The `,` separates multiple filters. Thanks! – [hek2mgl](#) Aug 16, 2016 at 18:48



It looks like you're updating more than just fields in the formulas arrays, but a little bit of everything.

4

If you want to indiscriminately change all occurrences of the string "MONTH" to "YEAR", you could do this:

```
(.. | strings) |= sub("MONTH"; "YEAR")
```

Share Improve this answer Follow

answered Aug 16, 2016 at 19:11



Jeff Mercado

135k ● 33 ● 266 ● 279

▲

0

This may be a task for walk/1.

(If your jq does not have walk/1, then you can copy its definition from <https://github.com/stedolan/jq/blob/master/src/builtin.jq>)

▼

For example, if you want to change "MONTH" to "YEAR" whenever "MONTH" appears as the value of a key in an object, then the following would do the job:

```
jq 'walk(if type == "object"
      then with_entries(.value |= (if . == "MONTH" then "YEAR" else . end))
      else . end)' input.json
```

Equivalently:

```
jq 'walk(if type == "object"
      then with_entries(if .value == "MONTH" then .value = "YEAR" else .
end)
      else . end)' input.json
```

These can easily be modified in accordance with similar requirements.

Share Improve this answer Follow

answered Aug 17, 2016 at 10:32



peak

116k ● 20 ● 175 ● 207

Curious as to why you refer to walk as walk/1 ? – Ed Randall Aug 10, 2023 at 7:28

- 1 The /N indicates the arity is N. That's the way the built-in function builtins reports the built-in filters, for good reason. – peak Aug 10, 2023 at 14:33