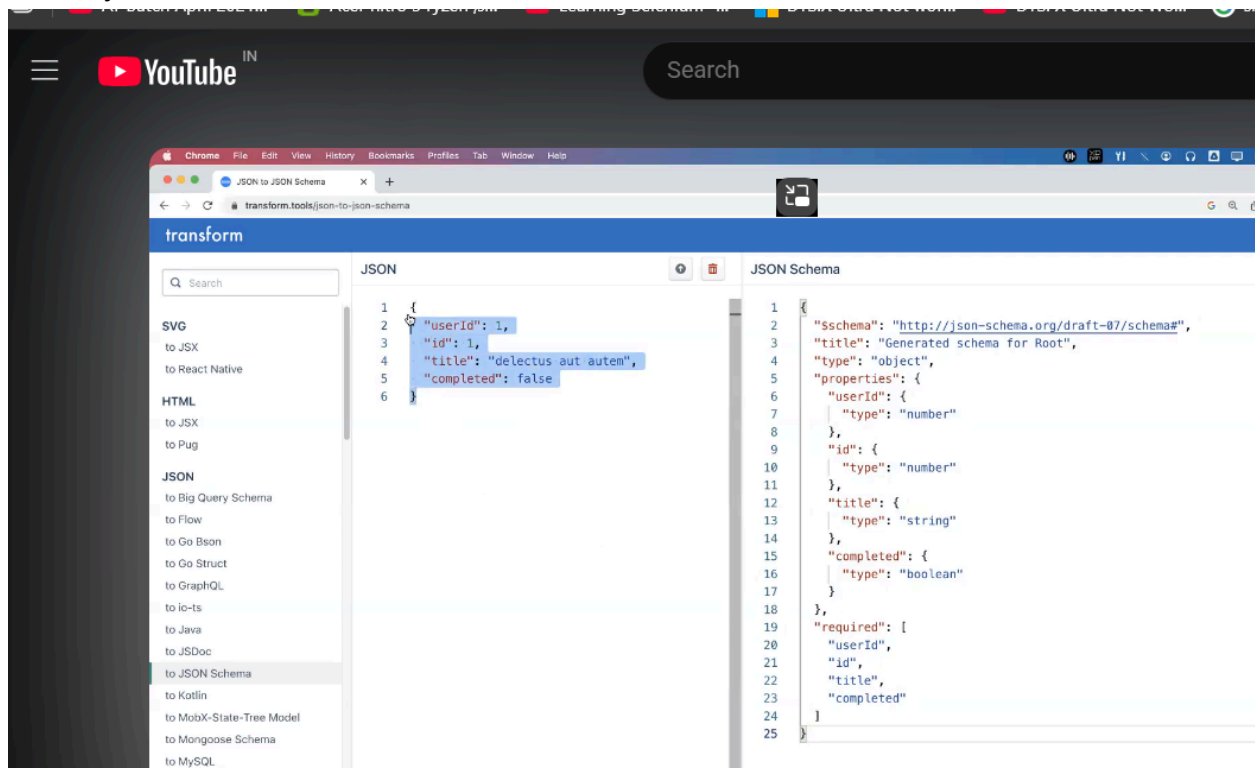


Json to json schema validator - use transform.



Paste the response in left side and we get json schema in right side –

ChromeFileEditViewHistoryBookmarksProfilesTabWindowHelp

JSON to JSON Schema

transform.tools/json-to-json-schema

transform

Q Search

SVG

to JSX

to React Native

HTML

to JSX

to Pug

JSON

to Big Query Schema

to Flow

to Go BSON

to Go Struct

to GraphQL

to io-ts

to Java

to JSDoc

to JSON Schema

to Kotlin

to MobX-State-Tree Model

to Mongoose Schema

to MySQL

to React PropTypes

Powered by Vercel

Created by @ritz078

JSON

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

```
{
  "id": 1,
  "title": "Fjallraven - Foldsack No. 1",
  "price": 109.95,
  "description": "Your perfect pack for everyday use and travels in cold weather.",
  "category": "men's clothing",
  "image": "https://fakestoreapi.com/img/61P5S6Rx4Y.jpg",
  "rating": {
    "rate": 3.9,
    "count": 120
  }
},
{
  "id": 2,
  "title": "Mens Casual Premium Slim Fit",
  "price": 22.3,
  "description": "Slim-fitting style, comfortable knit fabric, button closure.",
  "category": "men's clothing",
  "image": "https://fakestoreapi.com/img/71-3PjLcW2SL.jpg",
  "rating": {
    "rate": 4.1,
    "count": 259
  }
},
{
  "id": 3,
  "title": "Mens Cotton Jacket",
  "price": 55.99,
  "description": "great outerwear jackets with rain and wind repellents, in stock with 68 reviews.",
  "category": "men's clothing",
  "image": "https://fakestoreapi.com/img/80BJDN5AZX.jpg"
}
```

JSON Schema

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

```
{
  "category": {
    "type": "string"
  },
  "image": {
    "type": "string"
  },
  "rating": {
    "type": "object",
    "properties": {
      "rate": {
        "type": "number"
      },
      "count": {
        "type": "number"
      }
    },
    "required": [
      "rate",
      "count"
    ]
  },
  "required": [
    "id",
    "title",
    "price",
    "description",
    "category",
    "image",
    "rating"
  ]
}
```

```
1 // Install ajv library
2 //   npm install ajv      - command prompt/terminal
3
4 //import the library.
5 const Ajv=require('ajv')
6
7 //create object of ajv.
8 const avj=new Ajv()
9
10 describe("Schema validation",()=>{
11
12
13     it("schema validation against response",()=>{
14
15         cy.request(
16             { method: 'GET',
17               url : "https://fakestoreapi.com/products"
18             })
19
20         .then((response)=>{
```

codesnap.dev

Schema is pasted inside then.

```
1 //define the schema needed.
2     const schema = {
3         "$schema": "http://json-schema.org/draft-07/schema#",
4         "title": "Generated schema for Root",
5         "type": "array",
6         "items": {
7             "type": "object",
8             "properties": {
9                 "id": {
10                     "type": "number"
11                 },
12                 "title": {
13                     "type": "string"
14                 },
15                 "price": {
16                     "type": "number"
17                 },
18                 "description": {
19                     "type": "string"
20                 },
21                 "category": {
22                     "type": "string"
23                 },
24                 "image": {
25                     "type": "string"
26                 },
27                 "rating": {
28                     "type": "object",
29                     "properties": {
30                         "rate": {
31                             "type": "number"
32                         },
33                         "count": {
34                             "type": "number"
35                         }
36                     },
37                     "required": [
38                         "rate",
39                         "count"
40                     ]
41                 }
42             },
43             "required": [
44                 "id",
45                 "title",
46                 "price",
47                 "description",
48                 "category",
49                 "image",
50                 "rating"
51             ]
52         }
53     } // schema end
54
```

Write the below lines just after the line where we wrote schema end–

```
1 //first pass in the schema in compile method.
2     const validate= avj.compile(schema)
3     //then use the same variable and pass the actual response body.
4     //it will check if schema matches or not.
5     const isvalid=validate(response.body)
6     //assert.
7     expect(isvalid).to.be.true;
8
9     })
10
11 })
12
13
14
15
16 })
```