

Introduction to jq: grep for JSON

SeaGL

2021Nov05 @ 10:50

SeaGL Portal

der.hans (https://floss.social/@FLOX_advocate)

CDE Manager

Object Rocket, a rackspace company

<https://www.ObjectRocket.com/>

Yes, I'm hiring :)

ObjectRocket

<https://www.objectrocket.com/careers/>

Rackspace Technologies

<https://rackspace.jobs/>

Upcoming talks

- SeaGL, today at 13.15
 - Nextcloud: Your Own Piece of the Cloud
- Tux-Tage
 - Persönliche Sicherheit & Datenschutz mit Passwortmanager
- GeekBeacon Fest
 - Fediverse: Decentralized Social Networking and Services
 - moved to February

jq == ginsu for JSON

it slices, it dices, it splices

it mashes, it bashes, it amasses

EWRONGSCHTICK

Hey, that's Gallagher, not Ginsu

Yup :)

jq for JSON data

"lightweight and flexible command-line JSON processor"

- read/parse
- transform
- filter
- map

parse and filter

- grep

what is JSON?

- structured data
- not line delineated

example JSON

```
{ "conference": "SeaGL", "speaker": "der.hans" }
```

```
{ "speaker", "der.hans", "conference": "SeaGL" }
```


example JSON

```
{  
  "conference": "SeaGL",  
  "speaker": "der.hans"  
}
```

Digital Rights Protocol example

```
{
  "meta": {
    "version": "0.4"
  },
  "regime": "ccpa",
  "exercise": [
    "sale:opt-out"
  ],
  "relationships": ["customer", "marketing"],
  "identity": "jwtstring",
  "status_callback": "https://dsr-agent.example.com/update_status"
}
```

find the value to a key

```
{
  "meta": {
    "version": "0.4"
  },
  "regime": "ccpa",
  "exercise": [
    "sale:opt-out"
  ],
  "relationships": ["customer", "marketing"],
  "identity": "jwtstring",
  "status_callback": "https://dsr-agent.example.com/update_status"
}
```

```
$ cat ex_digital_rights_protocol.json | jq .identity
"jwtstring"
```

"grep" for a key and its value

```
{
  "meta": {
    "version": "0.4"
  },
  "regime": "ccpa",
  "exercise": [
    "sale:opt-out"
  ],
  "relationships": ["customer", "marketing"],
  "identity": "jwtstring",
  "status_callback": "https://dsr-agent.example.com/update_status"
}
```

```
$ cat ex_digital_rights_protocol.json | jq '. | {identity}'
{
  "identity": "jwtstring"
}
```

key-value as value

```
{
  "meta": {
    "version": "0.4"
  },
  "regime": "ccpa",
  "exercise": [
    "sale:opt-out"
  ],
  "relationships": ["customer", "marketing"],
  "identity": "jwtstring",
  "status_callback": "https://dsr-agent.example.com/update_status"
}
```

```
$ cat ex_digital_rights_protocol.json | jq -r '. | {meta}'
{
  "meta": {
    "version": "0.4"
  }
}
```

the value of the key-value of the value

```
{
  "meta": {
    "version": "0.4"
  },
  "regime": "ccpa",
  "exercise": [
    "sale:opt-out"
  ],
  "relationships": ["customer", "marketing"],
  "identity": "jwtstring",
  "status_callback": "https://dsr-agent.example.com/update_status"
}
```

```
$ cat ex_digital_rights_protocol.json | jq -r '.meta.version'
0.4
```

```
// "meta.version": null
// }
// ----
```

multi-grep

er, grep for multiple items

```
{
  "meta": {
    "version": "0.4"
  },
  "regime": "ccpa",
  "exercise": [
    "sale:opt-out"
  ],
  "relationships": ["customer", "marketing"],
  "identity": "jwtstring",
  "status_callback": "https://dsr-agent.example.com/update_status"
}
```

```
$ cat ex_digital_rights_protocol.json | jq '. | {meta, identity}'
{
  "meta": {
    "version": "0.4"
  },
  "identity": "jwtstring"
}
```


the dot

```
$ echo '{ "conference": "SeaGL", "speaker": "der.hans" }'  
{ "conference": "SeaGL", "speaker": "der.hans" }
```

```
$ echo '{ "conference": "SeaGL", "speaker": "der.hans" }' | jq .  
{  
  "conference": "SeaGL",  
  "speaker": "der.hans"  
}
```

line approximates document

```
$ cat ex_multiple_documents.json
[
  { "conf": "SeaGL", "spkr": "der.hans", "topic": "Nextcloud" },
  { "conf": "SeaGL", "spkr": "der.hans", "topic": "jq", "day": "Fr" }
]
```

```
$ cat ex_multiple_documents.json | jq '.[ ] | .topic'
"Nextcloud"
"jq"
```

JSON schema

- key-value pair: "label" : "value"
- document/object: "{ ... }"
- array: "[...]"
- value can be any of the above

value match

```
$ cat ex_multiple_documents.json | jq '.[ ] | select(.topic == "Nextcloud")'  
{  
  "conf": "SeaGL",  
  "spkr": "der.hans",  
  "topic": "Nextcloud"  
}
```

match and reduce keys

```
$ cat ex_multiple_documents.json | jq '.[ ] | select(.conf == "SeaGL") | [{spkr, topic}]'
[
  {
    "spkr": "der.hans",
    "topic": "Nextcloud"
  }
]
[
  {
    "spkr": "der.hans",
    "topic": "jq"
  }
]
```

match with or

```
$ cat ex_multiple_documents.json | jq '.[[] | select(.topic == "Nextcloud" or .topic ==
"jq") | [{spkr, topic}]'
[
  {
    "spkr": "der.hans",
    "topic": "Nextcloud"
  }
]
[
  {
    "spkr": "der.hans",
    "topic": "jq"
  }
]
```

variable input

```
$ cat ex_multiple_documents.json | jq --arg topic "jq" '.[ ] | select(.topic == $topic) |
[ {spkr, topic}]'
[
  {
    "spkr": "der.hans",
    "topic": "jq"
  }
]
```

remove a key

```
$ cat ex_multiple_documents.json | jq '.[ ] | del(.spkr)'
{
  "conf": "SeaGL",
  "topic": "Nextcloud"
}
{
  "conf": "SeaGL",
  "topic": "jq",
  "day": "Fr"
}
```


Firefox: list addons

```
$ cat ~/.mozilla/firefox/randstri.profile/addons.json | jq -c '.addons[] | {name, version}'
{"name":"HTTPS Everywhere","version":"2019.1.31"}
{"name":"FoxyProxy Standard","version":"6.3"}
{"name":"uMatrix","version":"1.3.16"}
{"name":"Deutsch (DE) Language Pack","version":"67.0buildid20190322012752"}
{"name":"Firefox Lightbeam","version":"2.1.0"}
{"name":"Save Page WE","version":"13.4"}
{"name":"English (South African) Language Pack","version":"67.0buildid20190322012752"}
{"name":"Crappy Firetitle","version":"0.1.12"}
{"name":"Firefox Multi-Account Containers","version":"6.2.5"}
```

Nextcloud deck

```
$ ncapibase='https://example.com/index.php/apps/'
$ deckapibase="$ncapibase/apps/deck/api/v1.0/"
$ alias ncdeckget='curl -s -u "user:randomstring" -H "OCS-APIRequest: true" -H "Accept: application/json" -XGET "$deckapibase/boards/7/stacks/21"'
```

```
$ ncdeckget | jq '.cards[] | {title,description}'
{
  "title": "Example Task 3",
  "description": "SeaGL example \n\n with multiple lines"
}
{
  "title": "do the things",
  "description": "do them, do them"
}
```

Case-insensitive matching

```
$ ncdeckget | jq --arg conf "seagl" '.cards[] | {title,description} |
select(.description | test(".*$conf","i"))'
{
  "title": "Example Task 3",
  "description": "SeaGL example \n\n with multiple lines"
}
```

Regex matching

- test, match, capture

```
$ ncdeckget | jq '.cards[] | {title,description} | select(.description | test("lines$";  
"i"))'  
{  
  "title": "Example Task 3",  
  "description": "SeaGL example \n\n with multiple lines"  
}
```

Regex matching 2

```
$ ncdeckget | jq '.cards[] | {title,description} | select(.description | test("(m|s)$";
"i"))'
{
  "title": "Example Task 3",
  "description": "SeaGL example \n\n with multiple lines"
}
{
  "title": "do the things",
  "description": "do them, do them"
}
```

Contacting der.hans

der.hans on Social Media / Fediverse

- FLOX_advocate on Mastodon
 - https://floss.social/@FLOX_advocate
- LuftHans on Libera.chat IRC
 - usually in #SeaGL, #PLUGaz and #LOPSA



Thank You

Credits

Resources

- <https://github.com/consumer-reports-digital-lab/data-rights-protocol#303-schema-status-of-a-data-subject-exercise-request>