

# Introduction to jq: grep for JSON

LinuxFest Northwest

2022Apr24 @ 15:00

Big Blue Button

der.hans ([https://floss.social/@FLOX\\_advocate](https://floss.social/@FLOX_advocate))

Database Support Manager

Object Rocket, a Rackspace Technology Company

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

Yes, I'm hiring :)

Rackspace Technologies

<https://rackspace.jobs/>

# Finding Hans

## My Conferences and Presentations

- <https://www.SpiralArray.com/talks>

## Social Media / Fediverse

- FLOX\_advocate on Mastodon, [https://floss.social/@FLOX\\_advocate](https://floss.social/@FLOX_advocate)
- LuftHans on Libera.chat IRC, usually in #SeaGL and #PLUGaz



## Open Calls for Participation

- SeaGL volunteering, [https://seagl.org/get\\_involved.html](https://seagl.org/get_involved.html)

# 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

# Associative JSON

```
2 + 3 == 3 + 2
```

```
2 * 3 * 4 == 4 * 3 * 2 == 3 * 4 * 2
```

```
{ "conference": "LinuxFest Northwest", "speaker": "der.hans" }
```

```
{ "speaker", "der.hans", "conference": "LinuxFest Northwest" }
```

```
{  
  "conference": "LinuxFest Northwest",  
  "speaker": "der.hans"  
}
```



# Give Me a C

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

### *payload from a JSON Web Token (JWT)*

```
$ echo "eyJhY2NvdW50IjoiYWRtaW4iLCJuYmYiOjAwMDAwMDAwMDAsImV4cCI6OTk5OTk5OTk5OXA0=" | base64url  
-d; echo  
{"account":"admin","nbf":0000000000,"exp":9999999999}
```

### *decode, then show re-encode matches*

```
$ jwtpayload='eyJhY2NvdW50IjoiYWRtaW4iLCJuYmYiOjAwMDAwMDAwMDAsImV4cCI6OTk5OTk5OTk5OXA0='  
$ echo "$jwtpayload" | base64url -d; echo  
{"account":"admin","nbf":0000000000,"exp":9999999999}  
$  
$ echo -n '{"account":"admin","nbf":0000000000,"exp":9999999999}' | base64url | grep -E  
"^$jwtpayload$"  
eyJhY2NvdW50IjoiYWRtaW4iLCJuYmYiOjAwMDAwMDAwMDAsImV4cCI6OTk5OTk5OTk5OXA0=
```

### *show account from payload*

```
$ echo "$jwtpayload" | base64url -d | jq .account  
"admin"  
$
```

# JSON Extended Sample

```
{
  "conference": "LinuxFest Northwest",
  "time": {
    "date": "2022Apr24",
    "time": "15:00"
  },
  "tags": [ "LinuxFest Northwest", "jq" ],
  "speaker": "der.hans"
}
```

# Value Lookup

```
{
  "conference": "LinuxFest Northwest",
  "time": {
    "date": "2022Apr24",
    "time": "15:00"
  },
  "tags": [ "LinuxFest Northwest", "jq" ],
  "speaker": "der.hans"
}
```

```
$ cat resources/ex_JSON_extended.json | jq .conference
"LinuxFest Northwest"
```

```
{
  "conference": "LinuxFest Northwest",
  "time": {
    "date": "2022Apr24",
    "time": "15:00"
  },
  "tags": [ "LinuxFest Northwest", "jq" ],
  "speaker": "der.hans"
}
```

```
$ cat resources/ex_JSON_extended.json | jq .time
{
  "date": "2022Apr24",
  "time": "15:00"
}
```

```
$ cat resources/ex_JSON_extended.json | jq '.time | .date'
"2022Apr24"
```

# "grep" Key Value Pair

```
{
  "conference": "LinuxFest Northwest",
  "time": {
    "date": "2022Apr24",
    "time": "15:00"
  },
  "tags": [ "LinuxFest Northwest", "jq" ],
  "speaker": "der.hans"
}
```

```
$ cat resources/ex_JSON_extended.json | jq '. | {conference}'
{
  "conference": "LinuxFest Northwest"
}
```

# Key Value as Value

```
{
  "conference": "LinuxFest Northwest",
  "time": {
    "date": "2022Apr24",
    "time": "15:00"
  },
  "tags": [ "LinuxFest Northwest", "jq" ],
  "speaker": "der.hans"
}
```

```
$ cat resources/ex_JSON_extended.json | jq '. | {time}'
{
  "time": {
    "date": "2022Apr24",
    "time": "15:00"
  }
}
```

# the value of the key-value of the value

```
{
  "conference": "LinuxFest Northwest",
  "time": {
    "date": "2022Apr24",
    "time": "15:00"
  },
  "tags": [ "LinuxFest Northwest", "jq" ],
  "speaker": "der.hans"
}
```

```
$ cat resources/ex_JSON_extended.json | jq '.time.date'
"2022Apr24"
```



# Quoteless

Use `--raw-value / -r` to forgo JSON formatting for a value

```
{
  "conference": "LinuxFest Northwest",
  "time": {
    "date": "2022Apr24",
    "time": "15:00"
  },
  "tags": [ "LinuxFest Northwest", "jq" ],
  "speaker": "der.hans"
}
```

```
$ cat resources/ex_JSON_extended.json | jq -r .speaker
der.hans
```

# multi-grep



## er, grep for multiple items

```
{
  "conference": "LinuxFest Northwest",
  "time": {
    "date": "2022Apr24",
    "time": "15:00"
  },
  "tags": [ "LinuxFest Northwest", "jq" ],
  "speaker": "der.hans"
}
```

```
$ cat resources/ex_JSON_extended.json | jq '{conference, time}'
{
  "conference": "LinuxFest Northwest",
  "time": {
    "date": "2022Apr24",
    "time": "15:00"
  }
}
```

# the dot

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

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

# Arrays

```
$ cat resources/ex_multiple_documents.json
[
  { "conf": "LinuxFest Northwest", "spkr": "der.hans", "topic": "Fediverse" },
  { "conf": "LinuxFest Northwest", "spkr": "der.hans", "topic": "jq", "date": "2022Apr24" }
]
```

```
$ cat resources/ex_multiple_documents.json | jq '.[[] | .topic'
"Fediverse"
"jq"
```

# JSON schema

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

# Value Match

```
$ cat resources/ex_multiple_documents.json
[
  { "conf": "LinuxFest Northwest", "spkr": "der.hans", "topic": "Fediverse" },
  { "conf": "LinuxFest Northwest", "spkr": "der.hans", "topic": "jq", "date": "2022Apr24" }
]
```

```
$ cat resources/ex_multiple_documents.json | jq '.[[] | select(.topic == "Fediverse")'
{
  "conf": "LinuxFest Northwest",
  "spkr": "der.hans",
  "topic": "Fediverse"
}
```

# Match and Reduce Keys

```
$ cat resources/ex_multiple_documents.json
[
  { "conf": "LinuxFest Northwest", "spkr": "der.hans", "topic": "Fediverse" },
  { "conf": "LinuxFest Northwest", "spkr": "der.hans", "topic": "jq", "date": "2022Apr24" }
]
```

```
$ cat resources/ex_multiple_documents_token.json | jq '.[[] | select(.conf == "LinuxFest Northwest") | {spkr, topic}]'
{
  "spkr": "der.hans",
  "topic": "Fediverse"
}
{
  "spkr": "der.hans",
  "topic": "jq"
}
```



# Match With "or", Compactly

```
$ cat resources/ex_multiple_documents.json
[
  { "conf": "LinuxFest Northwest", "spkr": "der.hans", "topic": "Fediverse" },
  { "conf": "LinuxFest Northwest", "spkr": "der.hans", "topic": "jq", "date": "2022Apr24" }
]
```

```
$ cat resources/ex_multiple_documents_token.json | jq -c '.[ ] | select(.topic == "jq" or .topic == "Fediverse") | {spkr, topic}'
{"spkr":"der.hans","topic":"Fediverse"}
{"spkr":"der.hans","topic":"jq"}
```

# variable input

```
$ cat resources/ex_multiple_documents.json
[
  { "conf": "LinuxFest Northwest", "spkr": "der.hans", "topic": "Fediverse" },
  { "conf": "LinuxFest Northwest", "spkr": "der.hans", "topic": "jq", "date": "2022Apr24" }
]
```

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

# Remove Key

```
$ cat resources/ex_multiple_documents.json
[
  { "conf": "LinuxFest Northwest", "spkr": "der.hans", "topic": "Fediverse" },
  { "conf": "LinuxFest Northwest", "spkr": "der.hans", "topic": "jq", "date": "2022Apr24" }
]
```

```
cat resources/ex_multiple_documents_token.json | jq '.[ ] | del(.spkr)'
{
  "conf": "LinuxFest Northwest",
  "topic": "Fediverse"
}
{
  "conf": "LinuxFest Northwest",
  "topic": "jq",
  "date": "2022Apr24"
}
```

# 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": "LinuxFest Northwest example \n\n with multiple lines"  
}  
{  
  "title": "do the things",  
  "description": "do them, do them"  
}
```

# Case-insensitive Matching

```
$ cat resources/ex_multiple_documents.json
[
  { "conf": "LinuxFest Northwest", "spkr": "der.hans", "topic": "Fediverse" },
  { "conf": "LinuxFest Northwest", "spkr": "der.hans", "topic": "jq", "date": "2022Apr24" }
]
```

```
$ cat resources/ex_multiple_documents_token.json | jq '.[ ] | {topic,spkr} | select(.spkr | test("HANS"; "i"))'
{
  "topic": "Fediverse",
  "spkr": "der.hans"
}
{
  "topic": "jq",
  "spkr": "der.hans"
}
```

# Regex matching

```
$ cat resources/ex_multiple_documents.json
[
  { "conf": "LinuxFest Northwest", "spkr": "der.hans", "topic": "Fediverse" },
  { "conf": "LinuxFest Northwest", "spkr": "der.hans", "topic": "jq", "date": "2022Apr24" }
]
```

```
cat resources/ex_multiple_documents_token.json | jq '.[[] | {topic,spkr} | select(.topic |
test("^jq$"; "i"))'
{
  "topic": "jq",
  "spkr": "der.hans"
}
```

# Regex matching 2

```
$ cat resources/ex_multiple_documents.json
[
  { "conf": "LinuxFest Northwest", "spkr": "der.hans", "topic": "Fediverse" },
  { "conf": "LinuxFest Northwest", "spkr": "der.hans", "topic": "jq", "date": "2022Apr24" }
]
```

```
$ cat resources/ex_multiple_documents_token.json | jq '.[ ] | {topic,spkr} | select(.topic |
test("(Q|T)$"; "i"))'
{
  "topic": "Fediverse",
  "spkr": "der.hans"
}
{
  "topic": "jq",
  "spkr": "der.hans"
}
```



# Thank You

# Resources

- jq homepage
  - <https://github.com/stedolan/jq>
- Consumer Reports data rights protocol
  - <https://github.com/consumer-reports-digital-lab/data-rights-protocol#303-schema-status-of-a-data-subject-exercise-request>
- Nextcloud Deck API
  - <https://deck.readthedocs.io/en/latest/API/>