

json

[development](#), [addendum](#)

jq (shell)

selección

[ejemplo](#)

```
{
  "ACTION": "domain.list",
  "DATA": [
    {
      "TYPE": "slave",
      "AXFR_IPS": "none",
      "DOMAIN": "domain1.com",
      "TTL_SEC": 0,
      "SOA_EMAIL": "",
      "DOMAINID": 11111,
      "DESCRIPTION": "",
      "EXPIRE_SEC": 0,
      "RETRY_SEC": 0,
      "STATUS": 1,
      "LPM_DISPLAYGROUP": "",
      "REFRESH_SEC": 0
    },
    {
      "TYPE": "slave",
      "AXFR_IPS": "none",
      "DOMAIN": "domain2.net",
      "TTL_SEC": 0,
      "SOA_EMAIL": "",
      "DOMAINID": 22222,
      "DESCRIPTION": "",
      "EXPIRE_SEC": 0,
      "RETRY_SEC": 0,
      "STATUS": 1,
      "LPM_DISPLAYGROUP": "",
      "REFRESH_SEC": 0
    },
    {
      "TYPE": "slave",
      "AXFR_IPS": "none",
      "DOMAIN": "domain3",
      "TTL_SEC": 0,
      "SOA_EMAIL": "",
      "DOMAINID": 33333,
      "DESCRIPTION": "",
      "EXPIRE_SEC": 0,
      "RETRY_SEC": 0,
      "STATUS": 1,
      "LPM_DISPLAYGROUP": "",

```

```
    "REFRESH_SEC": 0
  },
],
"ERRORARRAY": []
}
```

selecciona segundo subdocumento del documento DATA

```
| jq '.DATA | .[2]'
```

selecciona todos los elementos de DOMAINID

```
| jq '.DATA | .[].DOMAINID'
```

selecciona elemento por criterio/filtro

```
| jq '.DATA[] | select(.DOMAIN == "domain2") | .DOMAINID'
```

/via: <https://github.com/stedolan/jq/issues/370>

script

```
sample='[{"name":"foo"}, {"name":"bar"}]'
```

```
for row in $(echo "${sample}" | jq -r '[] | @base64'); do
  _jq() {
    echo ${row} | base64 --decode | jq -r ${1}
  }

  echo $(_jq '.name')
```

```
done
```

/via: <https://starkandwayne.com/blog/bash-for-loop-over-json-array-using-jq/>

From:

<https://miguelangel.torresegea.es/wiki/> - miguel angel torres egea

Permanent link:

<https://miguelangel.torresegea.es/wiki/development:json:start?rev=1549352662>

Last update: **04/02/2019 23:44**

