

# 【jq】linux json 查詢工具(查詢篇)

- data

```
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021"
  },
  "phoneNumber": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "fax",
      "number": "646 555-4567"
    }
  ],
  "gender": {
    "type": "male"
  }
}
```

- 取得資料(age)

```
[root@jylee-stres-stest250 tmp]# cat data.json | jq .age
```

```
25
```

- 取得資料(city) (address > city)

```
[root@jylee-stres-stest250 tmp]# cat data.json | jq .address.city
"New York"
```

- 輸出不帶雙引號 -r :

```
[root@jylee-stres-stest250 tmp]# cat data.json | jq .firstName
"John"
```

```
[root@jylee-stres-stest250 tmp]# cat data.json | jq -r .firstName
John
```

- 取得資料(city) (address > city)

```
[root@jylee-stres-stest250 tmp]# cat data.json | jq .address.city
"New York"
```

- 取得電話

```
# 取得電話陣列
[root@jylee-stres-stest250 tmp]# cat data.json | jq .phoneNumber
[
  {
    "type": "home",
    "number": "212 555-1234"
  },
  {
    "type": "fax",
    "number": "646 555-4567"
  }
]
```

```

}
]

#取得第一組電話
[root@jylee-stres-stest250 tmp]# cat data.json | jq .phoneNumber[0]
{
  "type": "home",
  "number": "212 555-1234"
}

#取得第一組電話( type + number )
[root@jylee-stres-stest250 tmp]# cat data.json | jq -r '.phoneNumber[0] | .type+" "+.number'
home,212 555-1234

#取得所有type
[root@jylee-stres-stest250 tmp]# cat data.json | jq '.phoneNumber[] | .type'
"home"
"fax"

```

- 使用map,join重組資料

```

[root@jylee-stres-stest250 tmp]# cat data.json | jq '.phoneNumber | map(.type)'
[
  "home",
  "fax"
]
[root@jylee-stres-stest250 tmp]# cat data.json | jq '.phoneNumber | map(.type) | join(",")'
"home,fax"

```

- 使用select 過濾資料

```

[root@jylee-stres-stest250 tmp]# cat sample.json | jq -r '.phoneNumber[] | select(.type == "home") '
{
  "type": "home",
  "number": "212 555-1234"
}

[root@jylee-stres-stest250 tmp]# cat sample.json | jq -r '.phoneNumber[] | select(.type == "home") | .number'
212 555-1234

```

- 依key取出值

```

# 回傳物件
[root@jylee-stres-stest250 tmp]# echo '{"a":1, "ip":["10.10.10.1","10.10.10.2"], "c":3, "d":4}' | jq '{ip, d}'

{
  "ip": [
    "10.10.10.1",
    "10.10.10.2"
  ],
  "d": 4
}

# 回傳傳列
[root@jylee-stres-stest250 tmp]# echo '{"a":1, "ip":["10.10.10.1","10.10.10.2"], "c":3, "d":4}' | jq ' [{ip, d}]'

[
  {
    "ip": [
      "10.10.10.1",
      "10.10.10.2"
    ],
    "d": 4
  }
]

```

# jq\_array\_to\_shell

---

data

```
{
  "values": [
    {
      "email": "user1@domain.com",
      "id": "USER1_ID"
    }, {
      "email": "user2@domain.com",
      "id": "USER2_ID"
    }
  ]
}
```

ss.sh

```
while IFS=' ' read -r email id; do
  echo "$email"
  echo "$id"
done <<EOF
$(jq -r '.values[] | "\(.email),\(.id)"' data2.json)
EOF
```

```
[root@jylee-stres-stest250 tmp]# sh ss.sh
user1@domain.com
USER1_ID
user2@domain.com
USER2_ID
```

<https://stedolan.github.io/jq/manual/#Invokingjq>

---

🕒修訂版本 #3

★由 treeman 建立於 23 🕒@🕒🕒 2021 23:53:47

🔧由 treeman 更新於 9 🕒E🕒🕒 2022 02:35:33