Jq

Jq (https://stedolan.github.io/jq) és una comanda de terminal que permet consultar i manipular documents JSON. Seria "l'equivalent al grep/cut/sed/awk" per processar informació amb aquest tipus de format. A continuació mostrem alguns exemples d'ús, implementats sobre un fitxer anomenat "prova.json" amb el següent contingut:

```
{
  "name": "something",
  "version": "0.0.0",
  "array": [1,2],
  "params": {
    "name1": "value1",
    "name2": "value2"
  }
}
```

jq "." prova.json	Mostra tot el JSON (el símbol "." representa l'arrel)
jq ".name" prova.json	Mostra el valor de la clau "name" (de primer nivell)
jq ".array" prova.json	Mostra els valors de l'array "array" (de primer nivell) NOTA:També es pot escriure *jq ".array[]" prova.json*. La sortida és un poc diferent
jq ".array[0]" prova.json	Mostra el valor del primer element de l'array "array" NOTA:També es poden escriure rangs; per exemple, el filtre *".array[1:4]"* seleccionaria des de l'element 1 (inclós) fins el 4 (sense incloure)
jq ".params" prova.json	Mostra els valors de l'objecte "params" (de primer nivell)
jq ".params.name1" prova.json	Mostra el valor de l'element "name1" de l'objecte "params"
jq ".params | length " prova.json	Mostra la quantitat d'elements de l'objecte (o array) indicat
jq ". | {uncampo:.version, otro:.array}" prova.json	Crea un objecte JSON nou format per elements concrets de l'objecte original. En aquest exemple, concretament crea un objecte de dos elements, un anomenat "uncampo" amb el valor que tenia l'element "version" de l'objecte original i un altre anomenat "otro" amb el valor que tenia l'element "array" de l'original NOTA:També es poden escollir elements concrets d'un array, així: *jq ". | {uncampo:.version, otro:.array[1]}" prova.json* NOTA: Un exemple pràctic: *lshw -c memory -json | jq ". | {MEMORIA: .size}"*
jq '.| select(.name == "anything")' prova.json	Mostra tots els objectes JSON complets (en l'exemple només tenim un) que tinguin el seu element "name" amb el valor "anything" (en l'exemple no en tenim cap)
jq '.| select(.name == "anything") | .version' prova.json	Mostra només el valor de l'element "version" de tots els objectes filtrats pel valor del seu camp ".name".
jq ".params.name3 = [5,6]" prova.json	Afegeix un nou element anomenat "name3" al subobjecte "params" de l'objecte JSON original. Aquest nou element en aquest cas és un array amb dos valors (5 i 6).
jq '.version = "1.0.0"' prova.json	Canvia el valor de l'element simple indicat per l'especificat
jq ".name = .params.name1" prova.json	Canvia el valor de l'element simple indicat pel valor d'un altre element de l'objecte
jq '.version == "1.0.0"' prova.json	Comprova si l'element simple indicat té el valor especificat (retorna *true* o *false*)
jq ".array |= .+[3]" prova.json	Afegeix un nou element a l'array. NOTA: |= will reassign the value and .+ [...] adds a new item.
jq "del(.params,.array)" prova.json	Elimina els elements indicats

NOTA: Otras funciones interesantes de *jq* son "*.nombreArray | min*" (devuelve el elemento del array con valor mínimo), "*.nombreArray | max*" (devuelve el elemento del array con valor máximo), "*.nombreArray | unique*" (devuelve un array con los elementos sin repetir), "*.nombreArray | sort*" (devuelve un array ordenado a partir de un array original desordenado), "*.nombreArray | sort[]*" (devuelve una de valores ordenados lista para entubar a partir de un array desordenado), ". | *keys*" (muestra solo los nombres de los elementos), "*.nombreArray[0] | keys*", "*.nombreArray[0] | .campo*" (último es -1), "*nombreArray[] | .campo*", etc.

NOTA: Una especificació estàndar molt més complexa (i completa) que permet consultar i manipular informació en format JSON és http://jsoniq.org