



Full credit is given to the above companies including the OS that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'jose-fmt.1'

\$ man jose-fmt.1

JOSE-FMT(1) JOSE-FMT(1)

NAME

jose-fmt - Converts JSON between serialization formats

SYNOPSIS

jose fmt [OPTIONS]

OVERVIEW

This jose fmt command provides a mechanism for building and parsing JSON objects from the command line. It operates as a simple stack machine. All commands operate on the TOP item of the stack and, occasionally, the PREV item of the stack. Commands that require a specific type of value will indicate it in parentheses. For example:

"TOP (arr.)".

This program returns 0 on success or the index of the option which failed.

OPTIONS

- ? -0, --null : Assert TOP to be null
- ? -a, --append : Append TOP to the end of PREV (arr.)
- ? -a, --append : Set missing values from TOP (obj.) into PREV (obj.)

- ? -A, --array : Assert TOP to be an array
- ? -B, --boolean : Assert TOP to be a boolean
- ? -c, --copy : Deep copy TOP, push onto TOP
- ? -d NAME, --delete=NAME : Delete NAME from TOP (obj.)
- ? -d , --delete= : Delete # from TOP (arr.)
- ? -d -, --delete=- : Delete # from the end of TOP (arr.)
- ? -e, --empty : Erase all items from TOP (arr./obj.)
- ? -E, --equal : Assert TOP to be equal to PREV
- ? -f FILE, --foreach=FILE : Write TOP (obj./arr.) to FILE, one line/item
- ? -f -, --foreach=- : Write TOP (obj./arr.) to STDOUT, one line/item
- ? -F, --false : Assert TOP to be false
- ? -g NAME, --get=NAME : Get item with NAME from TOP (obj.), push to TOP
- ? -g , --get= : Get # item from TOP (arr.), push to TOP
- ? -g -, --get=- : Get # item from the end of TOP (arr.), push to TOP
- ? -i , --insert= : Insert TOP into PREV (arr.) at #
- ? -l, --integer : Assert TOP to be an integer
- ? -j JSON, --json=JSON : Parse JSON constant, push onto TOP
- ? -j FILE, --json=FILE : Read from FILE, push onto TOP
- ? -j -, --json=- : Read from STDIN, push onto TOP
- ? -l, --length : Push length of TOP (arr./str./obj.) to TOP
- ? -M , --move= : Move TOP back # places on the stack
- ? -N, --number : Assert TOP to be a number
- ? -o FILE, --output=FILE : Write TOP to FILE
- ? -o -, --output=- : Write TOP to STDOUT
- ? -O, --object : Assert TOP to be an object
- ? -q STR, --quote=STR : Convert STR to a string, push onto TOP
- ? -Q, --query : Query the stack by deep copying and pushing onto TOP
- ? -R, --real : Assert TOP to be a real
- ? -s NAME, --set=NAME : Sets TOP into PREV (obj.) with NAME
- ? -s , --set= : Sets TOP into PREV (obj.) at #
- ? -s -, --set=- : Sets TOP into PREV (obj.) at # from the end

- ? -S, --string : Assert TOP to be a string
- ? -t, --truncate= : Shrink TOP (arr.) to length #
- ? -t -, --truncate=- : Discard last # items from TOP (arr.)
- ? -T, --true : Assert TOP to be true
- ? -u FILE, --unquote=FILE : Write TOP (str.) to FILE without quotes
- ? -u -, --unquote=- : Write TOP (str.) to STDOUT without quotes
- ? -U, --unwind : Discard TOP from the stack
- ? -x, --extend : Append items from TOP to the end of PREV (arr.)
- ? -x, --extend : Set all values from TOP (obj.) into PREV (obj.)
- ? -X, --not : Invert the following assertion
- ? -y, --b64load : URL-safe Base64 decode TOP (str.), push onto TOP
- ? -Y, --b64dump : URL-safe Base64 encode TOP, push onto TOP

EXAMPLES

Extract the alg parameter from a JWE Protected Header:

```
$ jose fmt -j "$jwe" -Og protected -yOg alg -Su-
A128KW
```

List all JWKs in a JWKSet (one per line):

```
$ echo "$jwkset" | jose fmt -j- -Og keys -Af-
{"kty":"oct",...}
{"kty":"EC",...}
```

Change the algorithm in a JWK:

```
$ echo "$jwk" | jose fmt -j- -j "A128GCM" -s alg -Uo-
{"kty":"oct","alg":"A128GCM",...}
```

Build a JWE template:

```
$ jose fmt -j '{}' -cs unprotected -q A128KW -s alg -UUo-
{"unprotected":{"alg":"A128KW"}}
```

AUTHOR

Nathaniel McCallum <npmccallum@redhat.com>

08/09/2021

JOSE-FMT(1)