json-dotenv Documentation

Release 0.0.29

Adrien Delle Cave

Apr 26, 2023

Contents

1 Table of contents 3

2 Installation 5

3 Usage 7

4 Commands 9

json-dotenv is a free and open-source, we develop it to manipulate and extract envfiles in json format.

CHAPTER 1

Table of contents

1. *Installation*
2. *Usage*
3. *Commands*

CHAPTER 2

Installation

```
pip install json-dotenv
```

CHAPTER 3

Usage

```
usage: json-dotenv [-h] [--allow-envvar] [-c {list,keys,get,set,unset}]
                   [-k KEY] [-v VALUE] [-f FILE] [--force]
                   [-l {critical,error,warning,info,debug}]
                   [--logfile LOGFILE] [-o OUTPUT] [-q {always,never,auto}]
                   [--format {env,json}]
                   [{list,keys,get,set,unset}]

positional arguments:
  {list,keys,get,set,unset}
                        Commands: list, keys, get, set, unset

optional arguments:
  -h, --help            show this help message and exit
  --allow-envvar        Allow environment variables expansion, instead of
                        False
  -c {list,keys,get,set,unset}
                        Commands: list, keys, get, set, unset, instead of list
                        (deprecated)
  -k KEY, --key KEY     variable name to set or unset
  -v VALUE, --value VALUE
                        variable value to set
  -f FILE               Location of the environment file or from stdin (-),
                        instead of .env
  --force               Force the output even if there is an error
  -l {critical,error,warning,info,debug}, --loglevel {critical,error,warning,info,
↪debug}
                        Emit traces with LOGLEVEL details, must be one of:
                        critical, error, warning, info, debug
  --logfile LOGFILE     Use log file <logfile> instead of /var/log/json-dotenv
                        /json-dotenv.log
  -o OUTPUT             Output result in file or to stdout
  -q {always,never,auto}
                        Whether to quote or not the variable values, instead
                        of always. This does not affect parsing
```

(continues on next page)

(continued from previous page)

```
  --format {env,json}   Output format env or json, instead of json
```

CHAPTER 4

Commands

List all environment variables in file foo.env:

```
json-dotenv list -f foo.env
```

```
{
  "LANG": "en_US.utf8",
  "PATH": "/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games",
  "MONIT_DOCKER_CONFIG": "vars:\n  base_url_unix: unix:///var/run/docker.sock\n  base_
↪url_https: https://127.0.0.1:2376/\n  tls_verify: true\nclients:\n  '@import_client
↪':\n    - clients.yml.example\n  local_https:\n    config:\n      base_url: ${vars[
↪'base_url_https']}\n      tls:\n        verify: ${vars['tls_verify']}\n  foo_
↪https:\n    '@import_vars': foo_https.vars.yml.example\n    config:\n      base_
↪url: ${vars['base_url_https']}\nctn-groups:\n  php:\n    match:\n      - 'name:foo-
↪php*'\n      - 'image:*/php-fpm/*'\n      - 'label:*php-fpm*'\n  nodejs:\n  ␣
↪match:\n      - 'id:4c01db0b339c'\n      - 'name:node*'\nconditions:\n  mem_gt_
↪10pct_and_cpu_gt_60pct:\n    expr:\n      - mem_percent > 10\n      - cpu_percent >␣
↪60\n  mem_usage_100MiB:\n    expr:\n      - mem_usage > 100 MiB\n  status_not_
↪running:\n    expr:\n      - status not in (pause,running)\ncommands:\n  start_
↪pause:\n    exec:\n      - start\n      - (echo 'foo' > /tmp/bar)\n      - pause\n ␣
↪pause_restart:\n    exec:\n      - pause\n      - restart\n  remove_force:\n  ␣
↪exec:\n      - remove:\n          kwargs:\n            force: true",
  "SHELL": "/bin/bash",
  "AUTON_CONFIG": "general:\n  listen_addr:   0.0.0.0\n  listen_port:   8666\n  max_
↪workers:   5\n  max_requests:  5000\n  max_life_time: 3600\n  lock_timeout:  60\n ␣
↪charset:       utf-8\n  content_type:  'application/json; charset=utf-8'\n  #auth_
↪basic:      'Restricted'\n  #auth_basic_file: '/etc/auton/auton.passwd
↪'\nendpoints:\n  si.corp-ansible:\n    plugin: subproc\n    config:\n      prog:␣
↪ansible-playbook\n      timeout: 3600\n  si.corp-terraform:\n    plugin: subproc\n ␣
↪  config:\n      prog: terraform\n      timeout: 3600\n  curl:\n    plugin:␣
↪subproc\n    config:\n      prog: curl\n      timeout: 3600\nmodules:\n  job:\n  ␣
↪routes:\n      run:\n        handler:   'job_run'\n        regexp:    '^run/(?P
↪<endpoint>[^\\/]+)/(?P<id>[a-z0-9][a-z0-9\\-]{7,63})$'\n        safe_init: true\n  ␣
↪     auth:      false\n        op:        'POST'\n      status:\n        handler:
↪'job_status'\n        regexp:    '^status/(?P<endpoint>[^\\/]+)/(?P<id>[a-z0-9][a-
↪z0-9\\-]{7,63})$'\n        auth:      false\n        op:        'GET'"
```

(continues on next page)

(continued from previous page)

```
}
```

List all environment variables name in file foo.env:

```
json-dotenv keys -f foo.env
```

```
[
  "LANG",
  "PATH",
  "MONIT_DOCKER_CONFIG",
  "SHELL",
  "AUTON_CONFIG"
]
```

Get foo.env contents from stdin and set variables AUTON_CONFIG=bar and toto=titi:

```
cat foo.env | json-dotenv set -f - -k AUTON_CONFIG -v bar -k toto -v titi
```

```
{
  "LANG": "en_US.utf8",
  "PATH": "/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games",
  "MONIT_DOCKER_CONFIG": "vars:\n  base_url_unix: unix:///var/run/docker.sock\n  base_
↪url_https: https://127.0.0.1:2376/\n  tls_verify: true\nclients:\n  '@import_client
↪':\n    - clients.yml.example\n  local_https:\n    config:\n      base_url: ${vars[
↪'base_url_https']}\n      tls:\n        verify: ${vars['tls_verify']}\n  foo_
↪https:\n    '@import_vars': foo_https.vars.yml.example\n    config:\n      base_
↪url: ${vars['base_url_https']}\nctn-groups:\n  php:\n    match:\n      - 'name:foo-
↪php*'\n      - 'image:*/php-fpm/*'\n      - 'label:*php-fpm*'\n  nodejs:\n    ␣
↪match:\n      - 'id:4c01db0b339c'\n      - 'name:node*'\nconditions:\n  mem_gt_
↪10pct_and_cpu_gt_60pct:\n    expr:\n      - mem_percent > 10\n      - cpu_percent >␣
↪60\n  mem_usage_100MiB:\n    expr:\n      - mem_usage > 100 MiB\n  status_not_
↪running:\n    expr:\n      - status not in (pause,running)\ncommands:\n  start_
↪pause:\n    exec:\n      - start\n      - (echo 'foo' > /tmp/bar)\n      - pause\n ␣
↪pause_restart:\n    exec:\n      - pause\n      - restart\n  remove_force:\n    ␣
↪exec:\n      - remove:\n          kwargs:\n            force: true",
  "SHELL": "/bin/bash",
  "AUTON_CONFIG": "bar",
  "toto": "titi"
}
```

Get variables LANG and PATH from foo.env:

```
json-dotenv get -f foo.env -k LANG -k PATH
```

```
{
  "LANG": "en_US.utf8",
  "PATH": "/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games"
}
```

Unset variables MONIT_DOCKER_CONFIG and AUTON_CONFIG from file foo.env (file not modified):

```
json-dotenv unset -f foo.env -k MONIT_DOCKER_CONFIG -k AUTON_CONFIG
```

```
{
  "LANG": "en_US.utf8",
  "PATH": "/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games",
  "SHELL": "/bin/bash"
}
```

Set variables TOTO and BAR and output result in file bar.json:

```
json-dotenv set -f '' -k TOTO -v tutu -k BAR -v foo -o bar.json
```

Set variables TOTO and BAR and output result in file bar.env (environment variables format):

```
json-dotenv set -f '' -k TOTO -v tutu -k BAR -v foo --format env -o bar.env
```