

jsontools

Handy command line tools for dealing with json data.

To use it, add `jsontools` to the plugins array in your zshrc file:

```
plugins=(... jsontools)
```

Usage

Usage is simple... just take your json data and pipe it into the appropriate jsontool:

- `pp_json` : pretty prints json.
- `is_json` : returns true if valid json; false otherwise.
- `urlencode_json` : returns a url encoded string for the given json.
- `urldecode_json` : returns decoded json for the given url encoded string.

Supports NDJSON (Newline Delimited JSON)

The plugin also supports [NDJSON](#) input, which means all functions have an alternative function that reads and processes the input line by line. These functions have the same name except using `ndjson` instead of `json` :

```
pp_ndjson, is_ndjson, urlencode_ndjson, urldecode_ndjson.
```

Examples

- **pp_json:**

```
# curl json data and pretty print the results
curl https://coderwall.com/bobwilliams.json | pp_json
```

- **is_json:**

```
# validate if file's content conforms to a valid JSON schema
$ is_json < data.json
true
# shows true / false and returns the proper exit code
$ echo $?
0
```

- **urlencode_json:**

```
# json data directly from the command line
$ echo '{"b":2, "a":1}' | urlencode_json
%7B%22b%22:2,%20%22a%22:1%7D
```

- **urldecode_json:**

```
# url encoded string to decode
$ echo '%7B%22b%22:2,%20%22a%22:1%7D' | urldecode_json
{"b":2, "a":1}
```

- **pp_ndjson:**

```
# echo two separate json objects and pretty print both
$ echo '{"a": "b"}\n{"c": [1,2,3]}' | pp_ndjson
{
  "a": "b"
}
{
  "c": [
    1,
    2,
    3
  ]
}
```