NAME

cpanel_json_xs – Cpanel::JSON::XS commandline utility

SYNOPSIS

```
cpanel_json_xs [-v] [-f inputformat] [-t outputformat]
```

DESCRIPTION

cpanel_json_xs converts between some input and output formats (one of them is JSON).

The default input format is `json` and the default output format is `json-pretty`.

OPTIONS

-v Be slightly more verbose.

-f fromformat

Read a file in the given format from STDIN.

`fromformat` can be one of:

json – a json text encoded, either utf-8, utf16-be/le, utf32-be/le
json-nonref – json according to RFC 7159
json-relaxed – json with all relaxed options
json-unknown – json with allow_unknown
storable – a Storable frozen value
storable-file – a Storable file (Storable has two incompatible formats)
bencode – uses one of Net::BitTorrent::Protocol::BEP03::Bencode, Bencode or the broken Convert::Bencode, if available (used by torrent files, among others)
clzf – Compress::LZF format (requires that module to be installed)
eval – evaluate the given code as (non-utf-8) Perl, basically the reverse of ''-t dump''
yaml – loose YAML (requires YAML)
yaml-tiny – loose YAML (requires YAML::Tiny or CPAN::Meta::YAML)
yaml-xs – strict YAML 1.2 (requires YAML::XS)
yaml-syck – YAML (requires YAML::Syck)
cbor – CBOR (via CBOR::XS)
string – do not attempt to decode the file data
sereal – Sereal (via Sereal::Decoder)
none – nothing is read, creates an `undef` scalar – mainly useful with `-e`

-t toformat

Write the file in the given format to STDOUT.

`toformat` can be one of:

json, json-utf-8 – json, utf-8 encoded
json-pretty – as above, but pretty-printed with sorted object keys
json-stringify – as json-pretty with allow_stringify
json-relaxed – as json-pretty, but with the additional options
->allow_stringify->allow_blessed->convert_blessed->allow_unknown
->allow_tags->**stringify_infnan** (1)

json-utf-16le, json-utf-16be – little endian/big endian utf-16
json-utf-32le, json-utf-32be – little endian/big endian utf-32
storable – a Storable frozen value in network format
storable-file – a Storable file in network format (Storable has two incompatible formats)
bencode – uses one of Net::BitTorrent::Protocol::BEP03::Bencode, Bencode or the broken Convert::Bencode, if available (used by torrent files, among others)
clzf – Compress::LZF format
yaml – loose YAML (requires YAML)
yaml-tiny – loose YAML (requires YAML::Tiny or CPAN::Meta::YAML)

yaml-xs – strict YAML 1.2 (requires YAML::XS)
yaml-syck – YAML (requires YAML::Syck)
dump – Data::Dump
dumper – Data::Dumper
string – writes the data out as if it were a string
sereal – Sereal (via Sereal::Encoder)
none – nothing gets written, mainly useful together with -e

Note that Data::Dumper doesn't handle self-referential data structures correctly – use "dump" instead.

-e code

Evaluate perl code after reading the data and before writing it out again – can be used to filter, create or extract data. The data that has been written is in `$_`, and whatever is in there is written out afterwards.

EXAMPLES

```
cpanel_json_xs -t none <isitreally.json
```

"JSON Lint" – tries to parse the file *isitreally.json* as JSON – if it is valid JSON, the command outputs nothing, otherwise it will print an error message and exit with non-zero exit status.

```
<src.json cpanel_json_xs >pretty.json
```

Prettify the JSON file *src.json* to *dst.json*.

```
cpanel_json_xs -f storable-file <file
```

Read the serialized Storable file *file* and print a human-readable JSON version of it to STDOUT.

```
cpanel_json_xs -f storable-file -t yaml <file
```

Same as above, but write YAML instead (not using JSON at all :)

```
cpanel_json_xs -f none -e '$_ = [1, 2, 3]'
```

Dump the perl array as UTF-8 encoded JSON text.

```
<torrentfile cpanel_json_xs -f bencode -e '$_ = join "\n", map @$_, @{$_->{"annou
```

Print the tracker list inside a torrent file.

```
lwp-request http://cpantesters.perl.org/show/Cpanel-JSON-XS.json | cpanel_json_xs
```

Fetch the cpan-testers result summary `Cpanel::JSON::XS` and pretty-print it.

```
cpanel_json_xs -f yaml-xs -t yaml-tiny <META.yml   >MYMETA.yml
cpanel_json_xs -f yaml-tiny -t yaml-xs <MYMETA.yml >XSMETA.yml
cpanel_json_xs -f yaml -t yaml <XSMETA.yml #BOOM!
Error: YAML_LOAD_ERR_BAD_MAP_ELEMENT
```

Compare YAML en- and decoders, and see that YAML::XS generates unparsable YAML <https://github.com/ingydotnet/yaml-libyaml-pm/issues/9>

AUTHOR