

MAGICJSON

Description

MAGICJSON is an application to process JSON Data and produces any output format.

It support currently either JSON files, XML files or HL7 message files.

For any source data other than JSON MAGICJSON generates at first a JSON representation of the source JSON before processing.

In order to control the output a GO template file can be used.

Workflow of MAGICJSON

- Read from input file or STDIN (-i)
- Convert to UTF-8 encoding if necessary
- Convert XML/HL7 to JSON representation
- Transform JSON via GO template to output (-t)
- Write output (-o)

Template usage

The JSON representation is the data anchor (i.e. the `.` operand) which is given to the executing template defined by the `-t` parameter.

Useful links

- <https://pkg.go.dev/text/template>
- <https://blog.gopheracademy.com/advent-2017/using-go-templates/>
- <https://blog.logrocket.com/using-golang-templates/>
- <https://www.digitalocean.com/community/tutorials/how-to-use-templates-in-go>

Usage

Flag	Default value	Only CmdLine	Description
i			Input filename. Use <code>.</code> to read from STDIN
ie			Input encoding
nb	false		no copyright banner
o			Output filename. Omit to print to STDOUT
oe			Output encoding
t			Template filename, directory or direct template content

Samples

Read from STDIN input

```
magicjson -i .
```

Encode an input data like a XML or HL7 message file to its JSON representation

```
magicjson -i nidek3.xml
```

Output to a file

```
magicjson -i nidek3.xml -o nidek3.json
```

Transform JSON, XML or HL7 by using a GO template file

```
magicjson -i nidek3.xml -t mytemplate.tpl -o nidek3.out
```

Transform JSON, XML or HL7 by using a GO template directly

```
magicjson -i nidek3.xml -t '{{ .Data.Company }}'  
magicjson -nb -i testdata/sample-hl7.hl7 -t '{{ GetValue "PID.5.1" }}'
```