

```
$ jq
jq - commandline JSON processor [version 1.6]
```

```
Usage:      jq [options] <jq filter> [file...]
           jq [options] --args <jq filter> [strings...]
           jq [options] --jsonargs <jq filter> [JSON_TEXTS...]
```

jq is a tool for processing JSON inputs, applying the given filter to its JSON text inputs and producing the filter's results as JSON on standard output.

The simplest filter is `.`, which copies jq's input to its output unmodified (except for formatting, but note that IEEE754 is used for number representation internally, with all that that implies).

For more advanced filters see the jq(1) manpage ("man jq") and/or <https://stedolan.github.io/jq>

Example:

```
$ echo '{"foo": 0}' | jq .
{
  "foo": 0
}
```

For a listing of options, use `jq --help`.

```
// rootCmd represents the base command when called without any subcommands
var rootCmd = &cobra.Command{
    Use:     "cli-for-humans",
    Short:   "A brief description of your application",
    Long:    `A longer description that spans multiple lines and likely contains
examples and usage of using your application. For example:

Cobra is a CLI library for Go that empowers applications.
This application is a tool to generate the needed files
to quickly create a Cobra application.` ,
    // Uncomment the following line if your bare application
    // has an action associated with it:
    // Run: func(cmd *cobra.Command, args []string) { },
}
```

```
$ cat cmd/root.go
```