The jq Cookbook

Sample

Dorian Pula

Version 0.5.0, 2023-12-01

Table of Contents

Introduction to JSON and jq

A developer working on modern web applications and websites has to contend with many different technologies. JSON is one such technology, which has become a standard for communicating between apps and servers via REST APIs. It is also a commonly used configuration format especially in the Node world.

JSON's ubiquity came about thanks to Javascript's own ubiquity, the brevity of the JSON spec and the ease of using JSON compared to other formats such as XML. The tale of how Javascript (and not Visual Basic or some other language) became the standard for frontend web development is too long for this book to cover. There are a few interesting interviews with Brendan Eich (the inventor of Javascript) about how that came about. [1]

As for the JSON specification itself, Douglas Crockford who popularized JSON insisted on keeping the specification minimal. This in turn meant that libraries that implemented JSON serialization/deserialization could be relatively easily implemented. [2] as there a number of edge-cases that quite difficult to solve. Always make sure to do due diligence when using a serialization library, avoid the temptation of writing your own, and don't blindly accept arbitrary untrusted JSON data off the web; since that could lead to exploits such as JSON injection.

While JSON is compact compared to other formats like XML, it isn't uncommon to receive sprawling amounts of JSON from a REST API. Without a tool, it can be difficult to parse such JSON. With this book I hope to introduce just such a tool called `jq`. Hopefully by the end of the book, I can both persuade you to incorporate `jq` into your toolbox, and have you feel comfortable using it. Also I will add examples to using `jq` inspired tools that let you work with XML, YAML and TOML in a similar manner as with JSON.

Lastly I hope that you will find this book *The jq Cookbook* a useful reference when working with `jq`.

About the Author and the Book

As the author of The jq Cookbook, I should introduce myself and my reasons for writing this book.

I am Dorian Pula, and I worked as a full-stack web application developer before transitioning into the DevOps space. Over the years I developed and maintained various web applications and REST APIs in various languages (Python, Java, PHP, Javascript and Rust to name a few), but they all shared a common format for communication: JSON. Even with DevOps tools like Ansible, Docker, Terraform and Kubernetes so on, JSON is very much the lingua franca of the distributed computing. So it pays off to become proficient and efficient with working with JSON, especially large complex JSON. I believe that `jq` is one of the best tools for this.

Over the years, I learned about and worked with `jq` to help me work with JSON better. This book originated as a blog post, and a lightning talk I gave at the local Python Toronto meetup. I wanted to expand upon that brief content, and document my gained experience, in the form of an easy to reference cookbook.

I hope that this book acts both an introduction to `jq` and as reference to using its query language. It is my goal that both the tool and my book will add to your toolkit as a developer, data scientist or

DevOps practitioner whenever you need to work with JSON, XML, YAML or TOML.

The Audience

The jq Cookbook is targeted towards DevOps engineers, developers or system admins who are familiar with using bash (or a bash-compatible command line), and need to work with JSON from REST APIs. However the data engineer may find it useful to manipulate data in a JSON format.

Conventions

I use a few conventions throughout the book to denote examples and comments.

Code examples will appear in boxes like so:

```
command --argument --key=value | another-command
```

Code examples with output will have `$` denoting where the command gets entered, and the resulting program output is displayed below.

```
$ command --argument --key=value | another-command
The program output appears here.
```

Aside tips or warnings will appear as sections:

💡 This is a useful tip or aside.

🔥 This is a *tricky* part, that you ought to look out for.

⚠ This is something that can get you into trouble.

Exercises and Examples

I include a few optional but highly recommended exercises as part of each chapter or section. More complex exercises are denoted with an asterisk (*).

These exercises rely on a freely available examples API https://jq-cookbook.com/examples/v1/ that I have created as a companion to this book. You'll be able to a

```
cat path/to/examples/hello-world.json
```

Reader Feedback

This is a early release of this book. This will enable me to get reader feedback early on, so that I can

```
}
```

Getting Help

Before jumping into querying, I want to mention a few ways to get help when using `jq`. You can find a reference section with common usages at the end of this book.

Another good resource is the manual on the jq project page.

As with any good UNIX CLI utility, `jq` comes with its own help in two formats: the CLI arguments help screen and a comprehensive man page.

You can invoke the help screen by running `jq` with the `--help` (or `-h`) argument:

```
jq --help
```

Also you can access the UNIX man page using: `man jq`. The man page is quite comprehensive, as it includes much of the manual documentation available on the `jq` project's website.