# Julia Evans

- About
- <u>Talks</u>
- Projects
- <u>Twitter</u>
- Github
- Favorites
- Zines
- RSS

# curl exercises

Recently I've been interested in how people learn things. I was reading Kathy Sierra's great book <u>Badass: Making Users Awesome</u>. It talks about the idea of *deliberate practice*.

The idea is that you find a small micro-skill that can be learned in maybe 3 sessions of 45 minutes, and focus on learning that micro-skill. So, as an exercise, I was trying to think of a computer skill that I thought could be learned in 3 45-minute sessions.

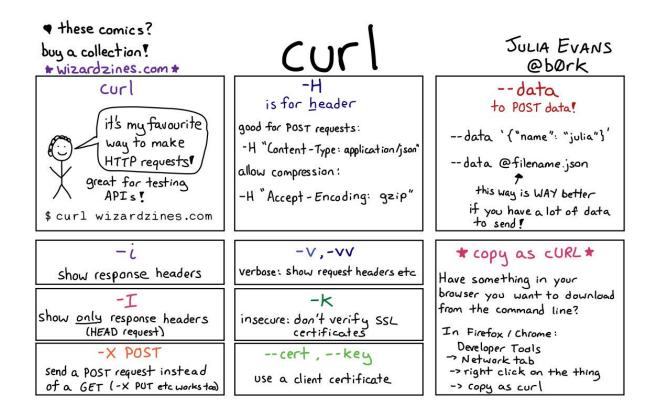
I thought that making HTTP requests with curl might be a skill like that, so here are some curl exercises as an experiment!

## what's curl?

curl is a command line tool for making HTTP requests. I like it because it's an easy way to test that servers or APIs are doing what I think, but it's a little confusing at first!

Here's a drawing explaining curl's most important command line arguments (which is page 6 of my <u>Bite Size Networking</u> zine). You can click to make it bigger.

1 of 3 6/20/21, 18:09



## fluency is valuable

With any command line tool, I think having fluency is really helpful. It's really nice to be able to just type in the thing you need. For example recently I was testing out the Gumroad API and I was able to just type in:

and get things working from the command line.

### 21 curl exercises

These exercises are about understanding how to make different kinds of HTTP requests with curl. They're a little repetitive on purpose. They exercise basically everything I do with curl.

To keep it simple, we're going to make a lot of our requests to the same website: <a href="https://httpbin.org">https://httpbin.org</a>. httpbin is a service that accepts HTTP requests and then tells you what request you made.

- 1. Request <a href="https://httpbin.org">https://httpbin.org</a>
- 2. Request <a href="https://httpbin.org/anything">httpbin.org/anything</a> will look at the request you made, parse it, and echo back to you what you requested. curl's default is to make a GET request.
- 3. Make a POST request to <a href="https://httpbin.org/anything">https://httpbin.org/anything</a>
- 4. Make a GET request to <a href="https://httpbin.org/anything">https://httpbin.org/anything</a>, but this time add some query parameters (set value=panda).
- 5. Request google's robots.txt file (www.google.com/robots.txt)
- 6. Make a GET request to <a href="https://httpbin.org/anything">https://httpbin.org/anything</a> and set the header User-Agent: elephant.

2 of 3 6/20/21, 18:09

- 7. Make a DELETE request to <a href="https://httpbin.org/anything">https://httpbin.org/anything</a>
- 8. Request <a href="https://httpbin.org/anything">https://httpbin.org/anything</a> and also get the response headers
- 9. Make a POST request to <a href="https://httpbin.org/anything">https://httpbin.org/anything</a> with the JSON body {"value": "panda"}
- 10. Make the same POST request as the previous exercise, but set the Content-Type header to application/json (because POST requests need to have a content type that matches their body). Look at the json field in the response to see the difference from the previous one.
- 11. Make a GET request to <a href="https://httpbin.org/anything">https://httpbin.org/anything</a> and set the header Accept-Encoding: gzip (what happens? why?)
- 12. Put a bunch of a JSON in a file and then make a POST request to <a href="https://httpbin.org/anything">https://httpbin.org/anything</a> with the JSON in that file as the body
- 13. Make a request to <a href="https://httpbin.org/image">https://httpbin.org/image</a> and set the header 'Accept: image/png'. Save the output to a PNG file and open the file in an image viewer. Try the same thing with different Accept: headers.
- 14. Make a PUT request to <a href="https://httpbin.org/anything">https://httpbin.org/anything</a>
- 15. Request <a href="https://httpbin.org/image/jpeg">https://httpbin.org/image/jpeg</a>, save it to a file, and open that file in your image editor.
- 16. Request <a href="https://www.twitter.com">https://www.twitter.com</a>. You'll get an empty response. Get curl to show you the response headers too, and try to figure out why the response was empty.
- 17. Make any request to <a href="https://httpbin.org/anything">https://httpbin.org/anything</a> and just set some nonsense headers (like panda: elephant)
- 18. Request <a href="https://httpbin.org/status/404">https://httpbin.org/status/200</a> and <a href="https://httpbin.org/status/200">https://httpbin.org/status/200</a>. Request them again and get curl to show the response headers.
- 19. Request <a href="https://httpbin.org/anything">https://httpbin.org/anything</a> and set a username and password (with -u username:password)
- 20. Download the Twitter homepage (<a href="https://twitter.com">https://twitter.com</a>) in Spanish by setting the Accept-Language: es-ES header.
- 21. Make a request to the Stripe API with curl. (see <a href="https://stripe.com/docs/development">https://stripe.com/docs/development</a> for how, they give you a test API key). Try making exactly the same request to <a href="https://httpbin.org/anything">https://httpbin.org/anything</a>.

Want a weekly digest of this blog?

Email address

#### Subscribe

Get your work recognized: write a brag document git exercises: navigate a repository

#### **Archives**

© Julia Evans. If you like this, you may like <u>Ulia Ea</u> or, more seriously, this list of <u>blogs I love</u>.

You might also like the <u>Recurse Center</u>, my very favorite programming community (<u>my posts about it)</u>

3 of 3 6/20/21, 18:09