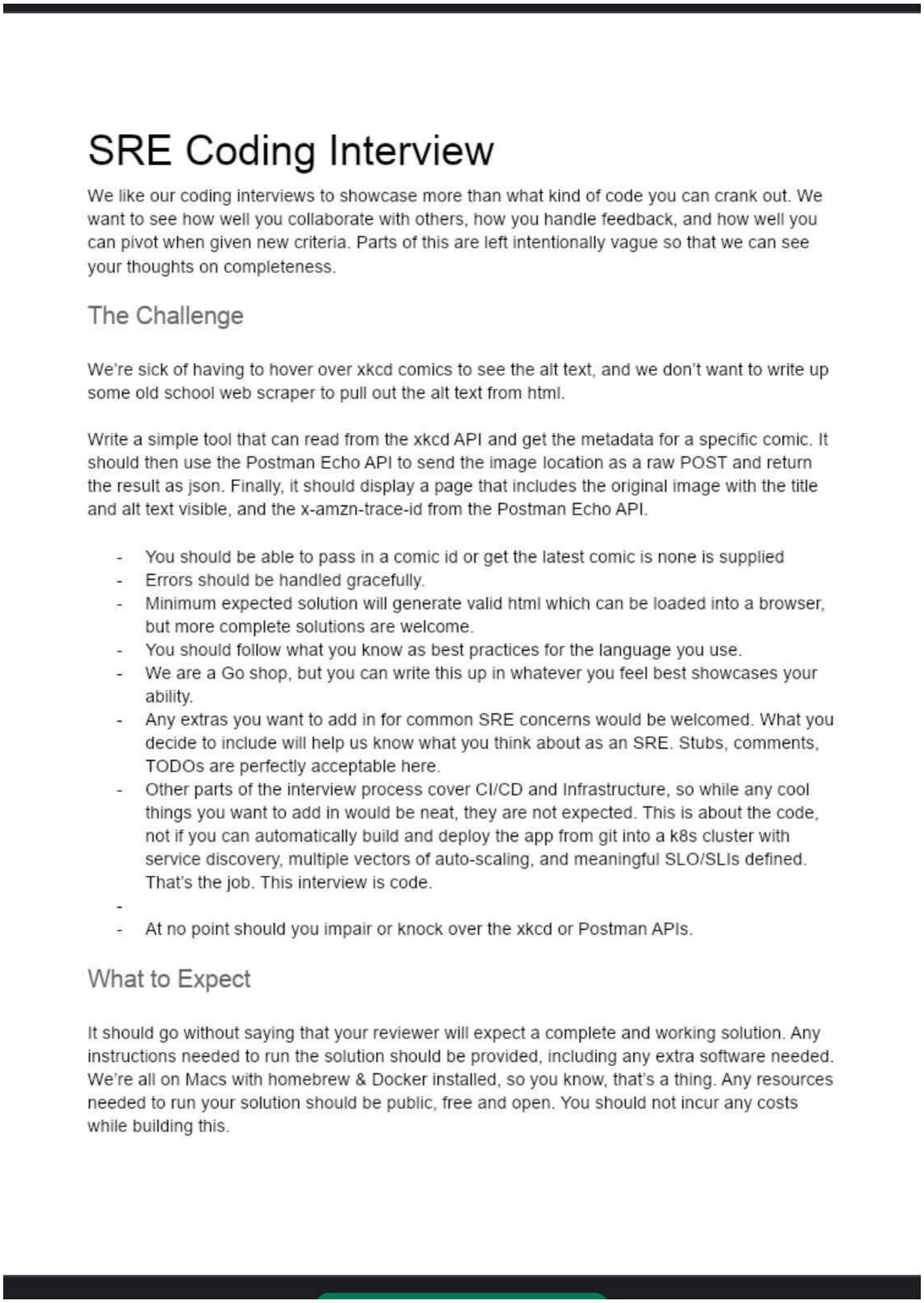

SRE Coding Interview

We like our coding interviews to showcase more than what kind of code you can crank out. We want to see how well you collaborate with others, how you handle feedback, and how well you can pivot when given new criteria. Parts of this are left intentionally vague so that we can see your thoughts on completeness.

The Challenge

We're sick of having to hover over xkcd comics to see the alt text, and we don't want to write up some old school web scraper to pull out the alt text from html.

Write a simple tool that can read from the xkcd API and get the metadata for a specific comic. It should then use the Postman Echo API to send the image location as a raw POST and return the result as json. Finally, it should display a page that includes the original image with the title and alt text visible, and the x-amzn-trace-id from the Postman Echo API.

- You should be able to pass in a comic id or get the latest comic is none is supplied
- Errors should be handled gracefully.
- Minimum expected solution will generate valid html which can be loaded into a browser, but more complete solutions are welcome.
- You should follow what you know as best practices for the language you use.
- We are a Go shop, but you can write this up in whatever you feel best showcases your ability.
- Any extras you want to add in for common SRE concerns would be welcomed. What you decide to include will help us know what you think about as an SRE. Stubs, comments, TODOs are perfectly acceptable here.
- Other parts of the interview process cover CI/CD and Infrastructure, so while any cool things you want to add in would be neat, they are not expected. This is about the code, not if you can automatically build and deploy the app from git into a k8s cluster with service discovery, multiple vectors of auto-scaling, and meaningful SLO/SLIs defined. That's the job. This interview is code.
-
- At no point should you impair or knock over the xkcd or Postman APIs.

What to Expect

It should go without saying that your reviewer will expect a complete and working solution. Any instructions needed to run the solution should be provided, including any extra software needed. We're all on Macs with homebrew & Docker installed, so you know, that's a thing. Any resources needed to run your solution should be public, free and open. You should not incur any costs while building this.