

Python Problem for Candidates

Basic Defang/Refang API (estimated 4hr)

Our developers have noticed that people are accidentally clicking on URLs when they mean to copy and paste them. One of our engineers proposed a “defanger” module (and a “refanger” module) that would allow URLs to be escaped (i.e. <https://twitter.com> -> `hxxp:\V\twitter[dot]com`). But some of our employees copy a lot of URLs! They want an API endpoint where you can supply a URL and the response will be the defanged URL. Similarly they want to be able to provide a defanged URL and receive the original URL. The module must be python 3.7>= compatible.

Requirements

At minimum the API endpoint should support the following endpoints

- `/api/defang` (i.e. <https://twitter.com> -> `hxxp:\V\twitter[dot]com`)
- `/api/refang` (i.e. `hxxp:\V\twitter[dot]com` -> <https://twitter.com>)

While the following are not necessarily requirements, here are some things to consider. There are no right or wrong answers, but consider them as you’re designing/writing out your solution:

- Should you sanitize or verify input? How?
- What HTTP methods should be used? GET? POST?
- What if we want to provide a list of URLs?
- What are some other URLs that clients auto turn into hyperlinks? Different protocols? Fully qualified domain names? Should we take care of them?
- What should your API response be? JSON? XML? Plaintext?
- How are you testing your code?

Example Usage

E.g

```
curl https://mydefanger.com/api/defang/https%3A%2F%2Ftwitter.com
{"response": "hxxp:\V\twitter[dot]com"}
```

Boilerplate Code

The code below should help you get started with your API endpoint. You do not need to use the following, it only serves as a general guideline to get started.

```
from flask import Flask
from flask_restful import Api, Resource
```

```
app = Flask(__name__)
api = Api(app, prefix='/api')

class DefangResource(Resource):
    def get(self, path):
        return {}

if __name__ == '__main__':
    api.add_resource(DefangResource, '/defang/<path:path>')
    app.run()
```

Completion and Delivery

When you are finished with your problem, there are two ways to turn it in:

1. Zip/tar your entire work and email to the team member you received the assignment from.
2. Provide a link to a version control repository (Github, Gitlab, Bitbucket, etc.).