

Tech Assignment - Build a URL Shortener

Requirements

1. Develop a back-end (REST / HTTP API) server which:
 - a. returns a short URL when given a specified URL
 - b. looks up or redirects to the specified URL when given a short URL
2. Develop a simple front-end (website / application) which allows users to easily create new shortened URLs.

Examples

- Request - Shortened URL: `http://{server}/a1b2c3d4`
- Response - Original URL: <https://developer.gov.sg/>

Options

Looking for a challenge, or have some time to spare?

Try implementing one (or more) of these:

- **I've got style** -- Show us how you would style the front-end to make it easier and more attractive to use, or make it accessible to mobile users with a responsive layout.
- **Elephants never forget** -- Persist your shortened URLs and data with a relational or document-oriented database of your choice.
- **Push to production** -- Deploy your solution to the public cloud, or your own servers, or docker compose file to spin up.
- **Tried and tested** -- Write a few unit / functional tests, or even an API canary.
- **I ship containers** -- Containerise your application for easy deployment and testing.
- **I'm clean** -- Show us how you organize your code to make it easier to read and maintain.
- **I'm efficient** - Show us the shortest path to a solution with minimum time and space complexity.

Submission format

- Upload your code to a git repository, such as github.com
- In the repository, include relevant information that you would like us to know!

What's Next?

We might ask you to join us for a interview shortly after.

Don't worry, it means we like your work and want to see you in action!

One of our engineers will pair with you to implement a small new feature.

FAQs

1. What languages / technology stack can I use?

We recommend that you use what you're familiar and comfortable with. This is vital for the pair programming stage.

2. Can I use frameworks (like Express, Hapi.js, Spring, Django, etc.)

Yes, you can! We want you to implement the business logic yourself, though.

3. Can I use {{advanced technologies}} (e.g. serverless, kubernetes, proprietary application stacks)?

Yes, although we don't expect you to, and a working product demonstrates your skill better than an incomplete one. Just make sure you're genuinely comfortable with what you use.

4. Can I use a third-party URL shortening service, or existing code?

The short answer: no. The long answer: nooooooooo.

5. Can I discuss this with a friend, or have someone else work with me on this?

No, that would be unfair. You will also find it difficult to modify the code or explain your program during the interview. If your friend would like to apply, though, they're welcome to drop us an email. :)