

← [course home \(/table-of-contents#section\\_system-design\\_question\\_url-shortener\)](/table-of-contents#section_system-design_question_url-shortener)

## Design a URL shortener

You know, like bit.ly.

Let's call it ca.ke!

**Step 1 is to scope the project.** System design questions like this are usually intentionally left open-ended, so you have to ask some questions and make some decisions about exactly what you're building to get on the same page as your interviewer.

So, what are we building? What features might we need?

### Features

**Is this a full web app, with a web interface?** No, let's just build an API to start.

**Since it's an API, do we need authentication or user accounts or developer keys?** No, let's just make it open to start.

**Can people modify or delete links?** Let's leave that out for now.

**If people can't delete links...do they persist forever? Or do we automatically remove old ones?** First, it's worth considering what policies we could use for removing old ones:

1. We could remove links that were *created* some length of time ago...like 6 months.
2. We could remove links that haven't been *visited* in some length of time...like 6 months.

(2) seems less frustrating than (1). Are there cases where (2) could still frustrate users? If a link is on the public web, it's likely to get hit somewhat regularly, at least by spiders. But what if it's on the private web (e.g. an internal "resources" page on a private university intranet)? Or...what if someone printed a bunch of *pamphlets* that had the URL on it, didn't give out any pamphlets for a

few months, then started giving them out again? That seems like a pretty reasonable thing that might happen (putting a URL on a printed piece of paper is a great reason to use a link shortener!) and having the link suddenly stop working would be quite frustrating for the user. Worse, what if a *book* already had the shortlink printed in a million copies? So let's let links exist forever.

**Should we let people *choose* their shortlink, or just always auto-generate it?** For example, say they want `ca.ke/parkers-resume`. Let's definitely support that.

**Do we need analytics, so people can see how many people are clicking on a link, etc?**

Hmmm, good idea. But let's leave it out to start.

It's okay if your list of features was different from ours. Let's proceed with these requirements so we're working on the same problem.

**Next step: Design goals.** If we're designing something, we should know what we're optimizing for! What are we optimizing for?

---

Want more coding interview help?

Check out **interviewcake.com** for more advice, guides, and practice questions.