Rahul Rajagopalan
6.170

Link Shortener Design Notes

The link shortener is primarily a server-side application. JavaScript makes AJAX requests to the server and displays the information it receives, but this is almost entirely presentation logic. The server handles the heavy lifting with regards to generating URLs and associating them with users. This makes it easier to transfer the application data to a relational database (since the database is on the server), and also allows me to test all application logic using Python's unittest module.

If not specified by the user, URLs were shortened by generating a random alphanumeric string of 5 characters, checking for collisions, and retrying if necessary. This produces short URLs, and the possible set of characters is still quite large (5 ** (len(alphabet) + len(digits)), so performance issues because of too many tries is not a huge issue.

The application uses relative URLs and the url_for function whenever links are necessary, so it can be moved to a different domain without having to modify the code.

Analytics were collected by maintaining a log of visits to shortened URLs in the database. This can be done because the visits are routed through the application like any other request. Because shortened URLs are unique, it is possible to associate the visit with the URL and in turn the user who created the shortened URL. In addition, the server routes invalid shortened URLs to an error page.

Users' information was kept private via a simple login with password system. Most of the code from users.py, originally created in the previous assignment was reusable – it was only necessary to modify it to use sqlite3 instead of shelve.