QuickLinks Design Document

Contact on Github at @MasterChief-John-117

What is this?

A simple, lightweight link shortening API for use on a personal web server

Principles

1. All primary endpoints MUST accept GET requests to ensure web browsers can be used to interact with it

Goals

1. Be simple and easy to learn
2. Provide fastest possible response times
3. Utilize native HTTP features for redirection as to be non-browser-reliant
4. Provide simple analytics information upon request

Non-Goals

1. Provide a graphical front-end
2. Allow for IP/Client blocking or require authentication
    a. These can be implemented at the webserver level and should not be implemented here

Requirements

Use cases

1. Adding a new URL via browser
    1.1. User visits website to add link
    1.2. The website gives him the new shortened link in a form that does NOT redirect
2. Visiting a shortlink
    2.1. The User visits the provided shortlink
        2.1.1. If the link does not exist he should be clearly notified
    2.2. The user is redirected to the long form of the page

Other Requirements

1. The same URL added multiple times should not have the same shortlink
2. Two URLs added at the same time should not have the same shortlink

Design

Algorithms

1. Link generation
   1.1. Identical links should NEVER be generated
   1.2. The link should be alphanumeric and both rememberable and easy to distribute
   1.3. The link should be three to five English words randomly selected from a words list
   1.4. The first letter of each word should be capitalized
   1.5. After generation uniqueness should be checked

Data Structures

ShortLinks should be stored with the original URL, analytical information, and a unique string different from the shortlink that can be used to retrieve analytics

Analytic information should include total page views and unique visitors

Databases

The database should be automatically set up and usable on any system without any admin interaction