

# URL Shortener

Developer: Naitik Dodia

Mentor: Rajat Agarwal

---

## Summary

1. Overview
2. /get-short-url
3. /get-long-urls
4. /delete-expired/<days>
5. /served-clicks

## Overview

URL shortener is a microservice to convert a long url to a short url. The services provided by this api are:

- 1) Convert a long url to short url ( with optional domain )
- 2) Get a long url (that was previously shortened) using the short url
- 3) Delete all expired urls ( default expiry time is 1 day)
- 4) Get a report of number of urls converted from a short url to long url

## Technical specifications

- 1) Programming language: Python
- 2) Web-Framework: Flask
- 3) Cache - Redis
- 4) Database - mysql

## APIs

### /get-short-url

- a. POST api with json body.
- b. Body json must contain "url" field. "domain" field is optional.

- c. This will return a short url which will have the “shurl.com” as default domain or the specified domain and alphanumeric characters after this domain for example “shurl.com/12abc”. Here 12abc becomes the id of the url.
- e. The alphanumeric characters are generated by converting a running counter of urls to a BASE62 number
- f. If there are any free short url ids (due to expiration) then they will automatically be reused using the id pool.
- g. If the long url already exists with some different domain specified before then a new short url with the presently specified domain will be generated but with the same id.
- h. If there exists the same long url and domain specified, then the same short url will be returned.

### /get-long-urls

- a. GET api with json body.
- b. Body json must contain “url” field.
- c. This will return the long url that was previously requested to shorten using /get-short-url. If not then it will return None object.

### /delete-expired/<days>

- a. GET api.
- b. Accepts number argument in <days> variable.
- c. Deletes the urls (both long and short) which were recently used <days> number of days ago.
- d. The now available free ids for making short urls are pushed to the id pool which will be utilized prior to generate new id.

### /served-clicks

- a. GET api.
- b. Returns a json response containing the number of clicks for each short url and it also has a key named “TOTAL CLICKS” which contains the total number of clicks.



## Database

### **[urls] table:**

id	Integer primary key, auto increment
long_url	text
short_url	varchar
hash_long_url	varchar
domain	varchar
added_datetime	datetime
click_count	integer

### **[url\_pool] table:**

short_url	varchar
-----------	---------

### **[running\_id] table:**

current_id	Integer
------------	---------

