System Design: Tiny URL

# Requirements

## Functional Requirements

1. Given a URL, our service should generate a shorter and unique alias of it. This is called short link. This short link should be short enough to be copied and pasted into applications.
2. When users access short link, our service should be able to redirect them to original link.
3. Users should be able to pick up a custom short link for their URL.
4. Links will expire after a standard default timespan. Users should be able to specify the expiration time.

## Non-functional Requirements

1. The system should be highly available. This is required because if the service is down, all the URL redirections will start failing.
2. URL redirection should happen real-time with minimum latency.
3. Short links should not be predictable.

## Extended Requirements

1. Analytics: How many times redirection happened
2. Our service should be available through REST APIs by other services.

# Capacity Estimation and Constraints

TinyURL will be a read-heavy system, meaning there will be more read requests than write requests. Let the read : write ratio be 100 : 1.

## Traffic estimates:

# System APIs

We can have SOAP / REST APIs to expose functionality of a service.

Following can be the APIs for creating and deleting short-links.

## createURL(api\_dev\_key, original\_URL, custom\_alias, user\_id, expiration\_date):

Parameters:

1. api\_dev\_key (String) : The API key of registered account. Used to prevent DDOS attacks by throttling users based on their allocated quota.
2. original\_URL (String) : Original URL to be shortened
3. custom\_alias (String): Optional custom alias if requested by user
4. user\_id (String): Optional, can be used to encode for generating short urls
5. expiration\_date (String): Optional expiration date for shortened URL

Returns: (String) Success : short-link, else, error code

## deleteURL(api\_dev\_key, short\_link, user\_id):

Parameters:

1. api\_dev\_key (String) : The API key of registered account. Used to prevent DDOS attacks by throttling users based on their allocated quota.
2. Short\_link (String): Short\_link to be deleted
3. user\_id (String): to authenticate if given short-URL belongs to the same user\_id or not

Returns: (String) Success or Failure message