

Implement your own short url

Short URL or tiny URL is an URL used to represent a long URL. For example, <http://tinyurl.com/45lk7x> will be redirect to <http://www.snippetit.com/2008/10/implement-your-own-short-url>.

There are 2 main advantages of using short URL:

- Easy to remember - Instead of remember an URL with 50 or more characters, you only need to remember a few (5 or more depending on application's implementation).
- More portable - Some systems have limitation on the total character to be used. For example SMS (short message services) and Twitter. When you want to send an 50 characters' URL to a friend via SMS, you only left 90 characters for your message.

Implement your own short URL is simple as 1, 2 and 3:

1. Define your own URL mapping algorithm.
2. Have a database to store the mapped URL.
3. Implement your URL mapping algorithm.

I'm not sure how the others create URL mapping algorithm, but at here, I will show my simple and fast's short URL implementation. The following is the specification of my implementation:

The system uses 6 characters of short code to represent an URL of any length. The valid characters for the short codes are ASCII 'a' to 'z' and '0' to '5' where each character contains 2^5 (32) states. 6 characters of short code can used to map 32^6 (1,073,741,824) URLs so it is unlikely to be used up in the near future.

First, you will need a database to store and retrieve your mapped URL. Below is the suggested database schema:

```
CREATE TABLE mappedURL (  
  shortCode char(6) not null,  
  lognURL text not null,  
  PRIMARY KEY shortCodeInd (shortCode),  
);
```

Second, you will need to define an algorithm to map the long URL into short code. Below is the suggested algorithm:

```
loop1: while true
  calculate md5 of the URL
  loop2: from 1st 4 bytes to 4th 4 bytes of md5 result
    cast the 4 bytes to an integer
    loop3: for shortCodeChar[0] to shortCodeChar[5]
      use 1st 5 bits of the integer to find the value in codeMap
      remove 5 bits from the integer
    end loop3
    save shortCodeChar as shortCode
    if shortCode does not exist in database
      insert the short code and original URL into database
      break loop1:
    else
      retrieve the stored URL from database
      if original URL equals to URL stored in database
        break loop1:
      end if
    end if
  end loop2
  insert '-' at the front of the URL
end loop1
return shortCode
```

Note: codeMap contains value of valid characters from 'a' to 'z' (index 0 to 25) and '0'-'5' (index 26 to 31).

Third, you will need to create a page that receive a short code and redirect browser to the mapped URL in database:

```
if valid short code and exists in database
  redirect browser to the mapped URL
else
  show error or redirect to an error page
end if
```

And that's all. You can use this implementation within your own domain as well to shorten the long URL of your blog before you send it to your friend. For example:

<http://domain/long-long-long-long-url.html> to <http://go.domain/abcdef>

Please note that there will be no source code provided in this page because you can implement it in any web languages (e.g. PHP, ASP or JSP) easily.

本文转载自 : <http://www.snippetit.com/2008/10/implement-your-own-short-url/>

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。
本文链接: **【】**（**）**