

Encode and Decode TinyURL New

Note: This is a companion problem to the **System Design** problem: **Design TinyURL**.

TinyURL is a URL shortening service where you enter a URL such as `https://leetcode.com/problems/design-tinyurl` and it returns a short URL such as `http://tinyurl.com/4e9iAk`.

Design the `encode` and `decode` methods for the TinyURL service. There is no restriction on how your encode/decode algorithm should work. You just need to ensure that a URL can be encoded to a tiny URL and the tiny URL can be decoded to the original URL.

Solution 1

```
class Solution {
public:

    // Encodes a URL to a shortened URL.
    string encode(string longUrl) {
        return longUrl;
    }

    // Decodes a shortened URL to its original URL.
    string decode(string shortUrl) {
        return shortUrl;
    }
};
```

written by [xplor1d](#) original link [here](#)

Solution 2

The questions states:

Note: Do not use class member/global/static variables to store states. Your encode and decode algorithms should be stateless.

Maybe I misunderstand the question, but does it means we are not allowed to use hashtable/database to store the original and shortened url? As I understand the url shortening services are rely on database and cache to store the hash to the original url, otherwise this is more like a string compression algorithm.

Any comments are highly appreciated.

written by [zhongyuan9817](#) original link [here](#)

Solution 3

below is the tiny url solution in java, also this is the similar method in industry. In industry, most of shorten url service is by database, one auto increasing long number as primary key. whenever a long url need to be shorten, append to the database, and return the primary key number. (the database is very easy to distribute to multiple machine like HBase, or even you can use the raw file system to store data and improve performance by shard and replica).

Note, it's meaningless to promise the same long url to be shorten as the same short url. if you do the promise and use something like hash to check existing, the benefit is must less than the cost.

Note: if you want the shorted url contains '0-9a-zA-Z' instead of '0-9', then you need to use 62 number system, not 10 number system(decimal) to convert the primary key number. like 123->'123' in decimal, 123->'1Z' in 62 number system (or '0001Z' for align).

```
public class Codec {
    List<String> urls = new ArrayList<String>();
    // Encodes a URL to a shortened URL.
    public String encode(String longUrl) {
        urls.add(longUrl);
        return String.valueOf(urls.size()-1);
    }

    // Decodes a shortened URL to its original URL.
    public String decode(String shortUrl) {
        int index = Integer.valueOf(shortUrl);
        return (index<urls.size())?urls.get(index):"";
    }
}
```

written by [qiu5](#) original link [here](#)

From [LeetCoder](#).