DNS URL shortener

Table of Contents

DNS TXT records is an unconventional way to provide short links without relying on a web-based shortening service.

1. Understand the TXT Record Format

DNS TXT records are designed to hold arbitrary text associated with a domain. We'll use these records to store mappings of shortened paths (like `/abc`) to the full URLs they represent (like `https://example.com/full-page`).

2. Choose Your Domain

Pick a domain name that will serve as the base for your shortened URLs. Let's say you own `example.com`.

When someone enters a shortened URL like `https://example.com/abc`, we'll use DNS to retrieve a TXT record that provides the full URL associated with that path.

3. Define a URL Mapping Scheme

To set up your mappings, decide on a way to encode short paths into DNS queries. Each shortened path (like `/abc`) will have its own TXT record. You might use subdomains for each path, like `abc.example.com`, to store mappings in a structured way.

> `abc.example.com` will point to `https://longwebsite.com/specific-page`

4. Create TXT Records for Each Mapping

Using your DNS provider's API or dashboard, add TXT records for each short path you want to create. The TXT field should contain the full URL.

Table 1. Example TXT record

Record Type:	TXT
Name:	abc (in your DNS provider's setup, this often becomes `abc.example.com`)
Value:	`https://longwebsite.com/specific-page`

If you add a few TXT records like this, each one will represent a new short link.

5. Set Up a DNS Query Tool (Client)

Since DNS is not typically used as a URL shortener, you'll need a client-side script or tool to interpret these TXT records.

Script

Write a script (for instance in Python) that takes a short path as input (like `abc`) and performs a DNS lookup for the TXT record at `abc.example.com`.

Command Line Tool

Use `dig` or `nslookup`

```
dig +short TXT abc.example.com
```

6. Resolve and Redirect

Now, you need a way to automate the redirection process wich can be for instance:

- Local script
- Lightweight web API

6.1. Local Script

Use a Python or JavaScript script that fetches the TXT record, extracts the full URL, and then redirects the user in a browser (or opens the link on the command line).

6.2. Lightweight Web API

If you want it to work more like a traditional URL shortener, create a minimal web server that:

- Accepts a short URL path (e.g., `/abc`).

- Looks up the corresponding TXT record.
- Returns a redirect to the full URL found in the TXT record.

6.2.1. Access Your URL Shortener

With this setup, you can create new short URLs just by adding TXT records to your DNS configuration. When someone visits `https://example.com/abc`, the server retrieves the correct TXT record and redirects them to the full URL.

7. Enhancements and Considerations

Caching

DNS records often have TTL (Time to Live) values that affect how quickly updates take effect. Set a lower TTL if you want the changes to propagate faster.

Character Limits

Some DNS providers have character limits for TXT records, so very long URLs might be tricky. Splitting across multiple TXT records could be a workaround, though that adds complexity.

Security

Ensure your code handles errors gracefully and verify the URLs to avoid redirection issues.