**URL Shortener**

A simple URL shortening service built in Go.

**Overview**

This project provides a basic URL-shortening service implemented in Go. It allows users to shorten long URLs into more manageable and shareable links. The service also includes a redirect feature to redirect users from the shortened URL to the original long URL.

**Features**

1. Generate a short URL for any long URL.
2. Redirect a short URL back to the original long URL.
3. Lightweight and easy to run locally.

**Run the Application**

1. Open a terminal and navigate to the directory where main.go is saved.
2. Run the following command to start the server:

bash

Copy code

go run main.go

1. The server will start and listen on port 3000.

**3. Test the Application**

You can test the application using tools like curl, Postman, or a browser. Below are the steps for each endpoint:

**Root Endpoint (/)**

* **Purpose**: To verify the server is running.
* **Command**:

bash

Copy code

curl http://localhost:3000/

* **Expected Output**: Hello, world!

**Shorten a URL (/shorten)**

* **Purpose**: Generate a short URL for a given long URL.
* **Command**:

bash

Copy code

curl -X POST -H "Content-Type: application/json" \

-d '{"url":"https://example.com"}' \

http://localhost:3000/shorten

* **Expected Output** (JSON Response):

json

Copy code

{

"short\_url": "d9736711"

}

**Redirect to Original URL (/redirect/{shortURL})**

* **Purpose**: Redirect a short URL back to its original long URL.
* **Command**:

bash

Copy code

curl http://localhost:3000/redirect/d9736711

* **Expected Outcome**:
  + The server will redirect you to https://example.com.
  + If using a browser, enter the URL http://localhost:3000/redirect/d9736711 in the address bar to test redirection.

**4. Experiment with Different URLs**

* Use different URLs when testing the /shorten endpoint to see how short URLs are generated.
  + Example:

bash

Copy code

curl -X POST -H "Content-Type: application/json" \

-d '{"url":"https://github.com"}' \

http://localhost:3000/shorten

Output:

json

Copy code

{

"short\_url": "d9736711"

}

* Each unique URL generates a different short URL.

**Future Enhancements**

1. Add persistent storage using Redis or PostgreSQL.
2. Implement URL expiration.
3. Validate and sanitize input URLs.
4. Handle short code collisions.
5. Add analytics for tracking link usage.

Git hub repo: https://github.com/priyankathakur64/go-url-shortener

Reference Video: <https://www.youtube.com/watch?v=dVVJU-3eU1g>