

# URL Shortener API Documentation

## ## Deployment

The API is deployed at: <https://url-shortener-api-zw6x.onrender.com>

## ## API Endpoints

### ### 1. Shorten a URL

**\*\*POST\*\*** `/shorten`` : Accepts a URL and returns a shortened URL.

**\*\*Request Body\*\***:

```
```json
{
  "originalUrl": "https://sharmanikhilportfolio3d.vercel.app/"
}
```
```

**\*\*Response (200 OK)\*\***:

```
```json
{
  "shortUrl": "https://url-shortener-api-zw6x.onrender.com/G3UdS6w8o"
}
```
```

### ### 2. Retrieve Original URL

**\*\*GET\*\*** `/:shortId`` : Redirects the user to the original URL when accessed with the shortId.

**\*\*Request URL\*\*:** <https://url-shortener-api-zw6x.onrender.com/G3UdS6w8o>

**\*\*Response (200 OK)\*\*:** Found. Redirecting to <https://sharmanikhilportfolio3d.vercel.app/>

### ### 3. Retrieve Statistics of URL

**\*\*GET\*\*** `/stats/:shortId` : Returns the usage statistics for a specific short URL, including:

- Total number of clicks.
- Timestamp of the last accessed time.

**\*\*Request\*\*:** <https://url-shortener-api-zw6x.onrender.com/stats/G3UdS6w8o>

**\*\*Response (200 OK)\*\*:**

```
```json
{
  "originalUrl": "https://sharmanikhilportfolio3d.vercel.app/",
  "clicks": 1,
  "lastAccessed": "2024-11-26T16:45:22.161Z"
}
```
```

## ## Testing Locally

Steps to Test the Application Locally:

1. **\*\*Run the Server\*\*:** Start the server by running the following command:

```
```bash
```

```
npm start
```

...

2. **Test Endpoints**: You can now test the following endpoints using Postman or curl:

- **POST**: `http://localhost:5000/shorten`

Request Body (JSON):

```
```json
{
  "originalUrl": "https://sharmanikhilportfolio3d.vercel.app/"
}
```
```

Response: Returns a shortened URL.

- **GET**: `http://localhost:5000/:shortId`

Request URL: Replace `:shortId` with the actual short URL identifier (e.g., `G3UdS6w8o`).

Response: Redirects to the original URL.

- **GET**: `http://localhost:5000/stats/:shortId`

Request URL: Replace `:shortId` with the actual short URL identifier.

Response: Returns the usage statistics of the URL (e.g., clicks and last accessed time).

Ensure the API is working by testing the various endpoints locally. Use Postman to check the response data and behavior of each endpoint.