**URL Shortener Project**

**Project Overview**

This project is a Node.js and Express-based URL Shortener backend. It allows users to:

- Create short URLs for any valid link

- Redirect to the original URL using the shortcode

- Track usage statistics for each short URL

- Log all activity to a remote logging service

**How We Solved the Project**

**1. Project Setup**

- Initialized a Node.js project and installed Express.

- Set up the main server and routing files.

- Implemented in-memory storage for short URLs and their statistics.

**2. Implemented API Endpoints**

**a. Create Short URL**

**-Endpoint:** `POST http://localhost:3000/shorturl`

**-Request Body:**

```json

{

"url": "https://www.google.com",

"valid": 60

}

```

**Response:**

```json

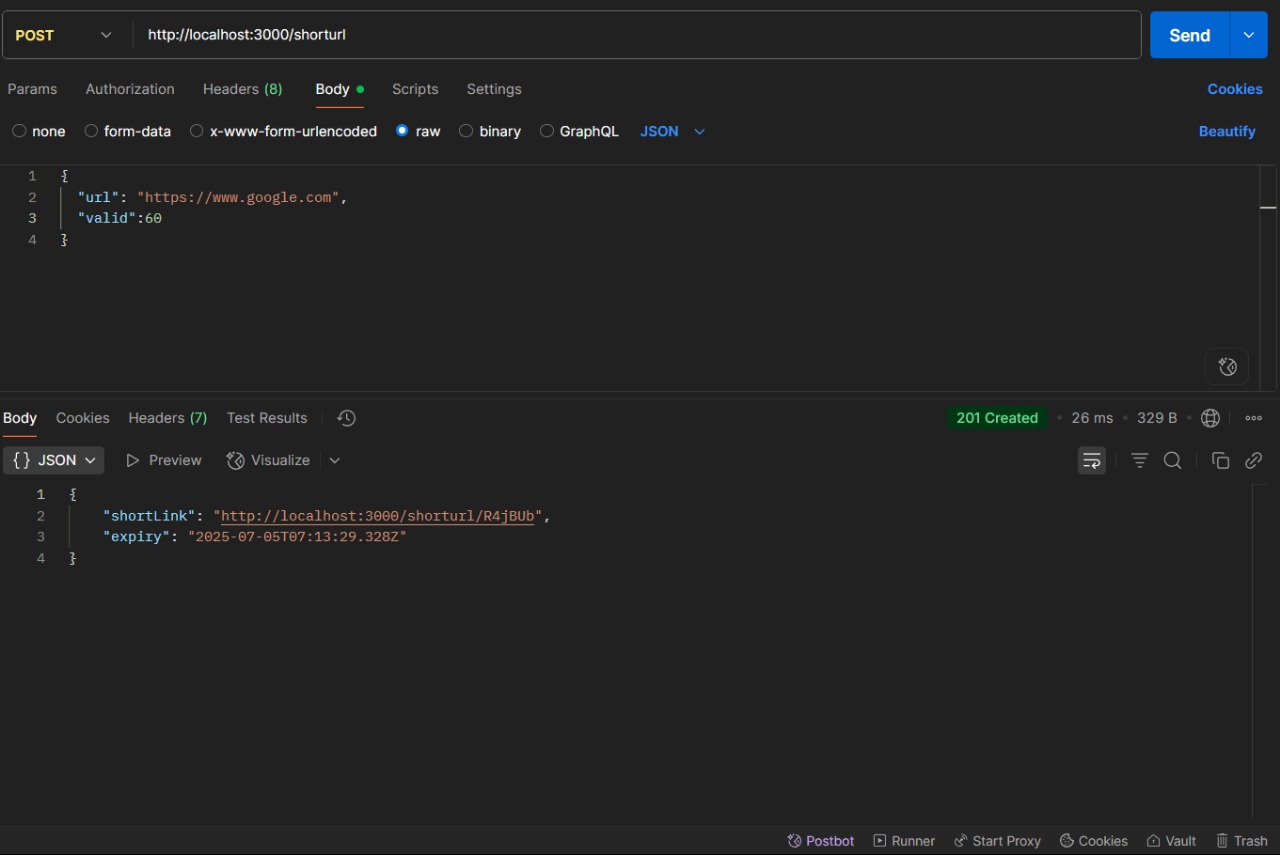
{

"shortLink": "http://localhost:3000/shorturl/R4jBUb",

"expiry": "2025-07-05T07:13:29.328Z"

}

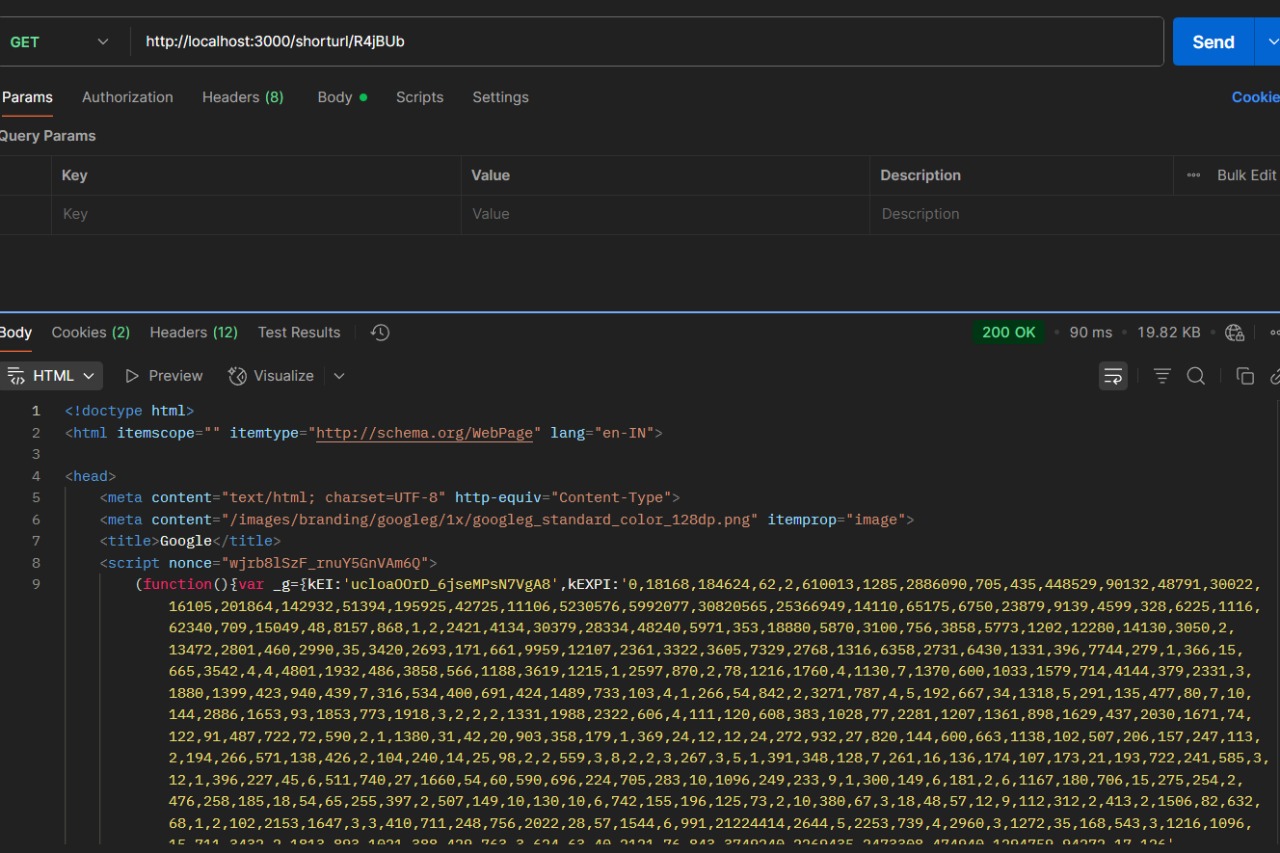
``



**b. Redirect to Original URL**

**- Endpoint:** `GET http://localhost:3000/shorturl/R4jBUb`

**- Behavior:** Redirects to the original URL (e.g., Google).



**c. Get URL Statistics**

**- Endpoint:** `GET http://localhost:3000/shorturl/stats/R4jBUb`

**- Response:**

```json

{

"totalClicks": 2,

"url": "https://www.google.com",

"createdAt": "...",

"expiry": "...",

"expired": false,

"clicks": [

{

"timestamp": "...",

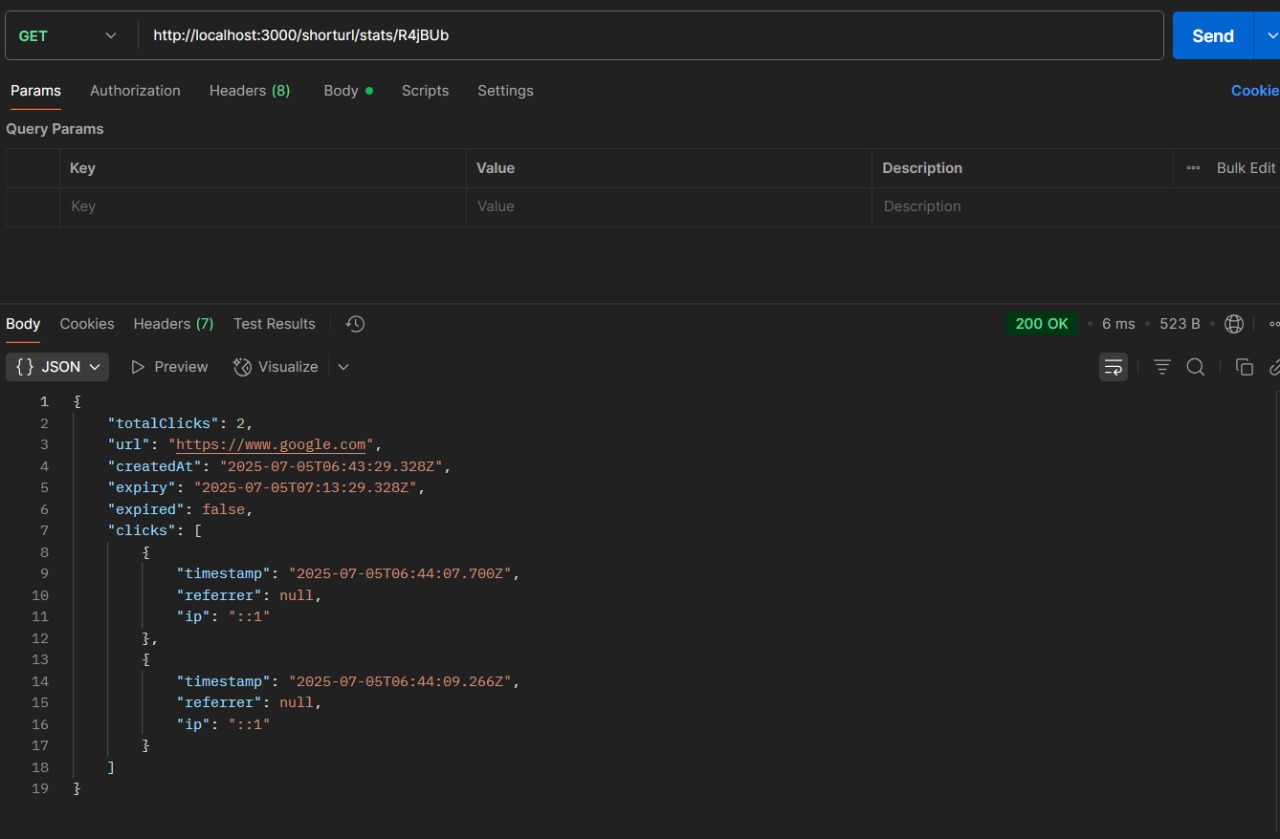
"referrer": null,

"ip": "::1"

}

]

}



**3. Logging Integration**

- Implemented a logger middleware that sends logs to a remote logging service using a Bearer token.

- Verified logging by checking for successful log submissions and error handling.

**Conclusion:**

All endpoints were tested and are working as expected. The project now supports URL shortening, redirection, statistics tracking, and centralized logging.