

1. POST /shorten

- **Description:** Shortens a given long URL. Optionally, a custom ID can be provided.
- **Method:** POST
- **Request Body:**
 - `longUrl` (string): The original long URL to shorten (required).
 - `customId` (string, optional): Custom short URL identifier. Must be alphanumeric and no longer than 9 characters.
- **Response:**
 - **Status:** 200 OK on success, 400 Bad Request on invalid URL, 409 Conflict on duplicate custom ID.

Body:

```
{
  "shortUrl": "http://localhost:3000/{shortId}"
}
```

- **Usage:**
 - Request an endpoint to shorten a long URL. If `customId` is provided, it will use it; otherwise, a unique ID is generated automatically.

Example Request:

```
{
  "longUrl": "https://www.example.com",
  "customId": "example123"
}
```

Example Response:

```
{
  "shortUrl": "http://localhost:3000/example123"
}
```

2. GET /:shortId

- **Description:** Redirects to the original URL associated with the short URL.
- **Method:** GET
- **URL Parameter:**
 - `shortId` (string): The shortened ID corresponding to the long URL.

- **Response:**
 - **Status:** 302 Found for successful redirection, 404 Not Found if the short URL does not exist.
- **Usage:**
 - When the user accesses the short URL, the server will redirect to the original long URL.

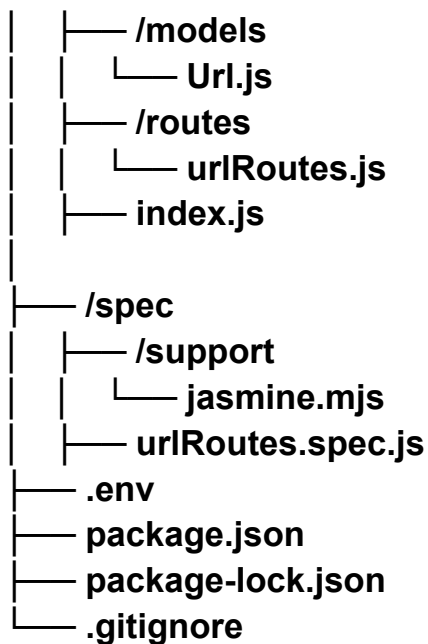
Example Request:

GET http://localhost:3000/example123

- **Example Response:**
 - Redirects the browser to: https://www.example.com

Project Structure Overview

/back-url-short



Setup Instructions

1. Clone the Repository

- git clone <https://github.com/MrkingOS/back-url-short.git>

2. Install Dependencies

- cd back-url-short
- npm install

3. Set Up Environment Variables

- **PORT**=3000
- **MONGO_URI**=mongodb://localhost:27017/url-shortener

4. Run the Development Server

- **npm start**