

## Mock-API Manager: Essential cURL Commands

The interface displays three panels:

- 1. Get Environments (List All)**: Shows a terminal window with two curl commands. The first command lists environments, and the second shows a detailed environment configuration (id: E-commerce, name: E-commerce, status: active). A screenshot of a database table shows rows for environments.
- 2. Get Endpoints (List "Zones")**: Shows a terminal window with two curl commands. The first command lists environments, and the second shows endpoint details for environment ID 123. A screenshot of a database table shows rows for endpoints.
- 3. Update Environment (ON/OFF)**: Shows a terminal window with a curl PUT command to update an environment's status to inactive. A large red button labeled "OFF" is shown, along with a red X icon over a server icon.

Base URL: <http://lockshot.3000/api>

# Mock-API Manager Management API: cURL Reference

This guide assumes your Node.js backend server is running locally on the default port.

Assumption Value

Base URL <http://localhost:3000>

API Prefix /api

## 1. Retrieve All Environments (The "Environment" List)

This command is used to fetch a list of all top-level environments currently configured in the database. This gives you a high-level view of every active and inactive project or context the server is managing.

Detail Description

**Method** GET

**Route** /api/environments

**Purpose** Get the ID, name, status, and configuration details for all environments.

### Command Syntax

Bash

```
curl -X GET 'http://localhost:3000/api/environments'
```

## 2. Retrieve Endpoints for a Specific Environment (The "Zones" List)

This command allows you to drill down into a specific environment (identified by its unique ID) and retrieve all the mock **Endpoints**—the actual URL paths—configured for it. You referred to this as retrieving the "zones."

Detail Description

**Method** GET

**Route** /api/environments/:id/endpoints

**Purpose** Get the full list of mock paths, HTTP methods, and associated response data for a single environment.

### Command Syntax

Remember to replace `:id` with the actual Environment ID you retrieved from the first command.

Bash

```
# Example: Fetching all endpoints for Environment ID 123
curl -X GET 'http://localhost:3000/api/environments/123/endpoints'
```

---

## 3. Update an Environment's State (The "ON/OFF" Action)

This command uses the **PUT** method to modify the details of an existing environment. The most common use case is toggling the environment's activity status, which you described as the "ON/OFF" action.

Detail Description

**Method** PUT

**Route** /api/environments/:id

**Purpose** Modify any field on a specific environment, such as its name, prefix, or its active state (`is_active`).

### Command Syntax

This command requires sending a JSON payload (`-d`) and specifying the content type (`-H`) to tell the server what data to expect.

#### Example A: Switching the Environment OFF

This sets the `is_active` status to `false`, effectively taking the entire mock environment offline until it's turned back on.

Bash

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

```
-d '{"is_active": false}' \
'http://localhost:3000/api/environments/123'
```

### Example B: Renaming and Switching the Environment ON

This updates multiple fields, including setting `is_active` to `true`.

Bash

```
curl -X PUT \
-H "Content-Type: application/json" \
-d '{"name": "E-commerce Backend - Production Mock", "is_active": true}' \
'http://localhost:3000/api/environments/123'
```