


## Mock-API Manager: Essential cURL Commands


# Mock-API Manager: Essential cURL Commands

## 1. Get Environments (List All)




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

curl -X GET 'api/environments'
```

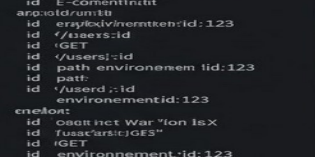


## 2. Get Endpoints (List "Zones")




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


curl -X GET 'http://localhost:3000/api/environments/123/endpoints'
```



## 3. Update Environment (ON/OFF)



```
curl -X PUT -H "Content-Type: application/json" -d '{"is_active": false}' method 'api/environments/123'
```



Base URL: <http://localhost: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
<b>Method</b>	GET
<b>Route</b>	/api/environments/:id/endpoints
<b>Purpose</b>	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
<b>Method</b>	PUT
<b>Route</b>	/api/environments/:id
<b>Purpose</b>	Modify any field on a specific environment, such as its name, prefix, or its active state ( <code>is_active</code> ).

### 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'
```