# Backend API Documentation

## Introduction

This document provides an overview of the Backend API that facilitates communication between the Roblox environment and the server. The API allows for creating, updating, retrieving, and deleting transporter data, enabling synchronization between the Roblox game and the backend server.

## Base URL

The base URL for the backend API is: **http://localhost:3000**

## Endpoints

1. Retrieve Transporters

**GET ‘/transporters’**

Returns an array of transporter positions.

## Response

- Status: 200 OK

- Body: Array of transporter objects, where each object has the following properties:

- **name**: The name of the transporter.

- **location**: The current location of the transporter as a string in the format “**x, y, z**”.

2. Retrieve a Specific Transporter

**GET ‘/transporters/:id’**

Retrieves the position of a specific transporter by its ID.

## Parameters

- **:id**: The name or ID of the transporter.

## Response

- Status: 200 OK

- Body: Object representing the transporter, with the following properties:

- **name**: The name of the transporter.

- **location**: The current location of the transporter as a string in the format “**x, y, z**”.

- Status: 404 Not Found

- Body: **{ message: ‘Transporter not found’ }**

3. Create Transporter

**POST ‘/transporters’**

Creates a new transporter and adds its position to the list.

## Request Body

- Object representing the transporter, with the following properties:

- **name**: The name of the transporter.

- **location**: The initial location of the transporter as a string in the format **“x, y, z”**.

## Response

- Status: 201 Created

- Body: **{ message: ‘Transporter created successfully’, id: transporterName }**

- Status: 400 Bad Request

- Body: **{ error: ‘Transporter data is required’ }**

4. Update Transporter Position

**PUT ‘/transporters/:id’**

Updates the position of a specific transporter.

## Parameters

- **:id**: The name or ID of the transporter.

Request Body

- Object representing the updated transporter position:

- **name**: The name of the transporter.

- **location**: The new location of the transporter as a string in the format “**x, y, z”.**

## Response

- Status: 200 OK

- Body: **{ message: ‘Transporter updated successfully’ }**

- Status: 404 Not Found

- Body: **{ message: ‘Transporter not found’ }**

5. Delete Transporter

**DELETE ‘/transporters/:id’**

Deletes a specific transporter and removes its position from the list.

## Parameters

- **:id**: The name or ID of the transporter.

**Response**

- Status: 200 OK

- Body: **{ message: ‘Transporter deleted successfully’ }**

- Status: 404 Not Found

- Body: **{ message: ‘Transporter not found’ }**

## Conclusion

The Backend API allows for seamless interaction between the Roblox environment and the server, enabling real-time synchronization of transporter positions. Developers can utilize these endpoints to create, update, retrieve, and delete transporter data, ensuring accurate and consistent representation of transporters within the game world.