API Documentation

Version: 1.0

Base URL: http://localhost:8081/

# Airport Controller

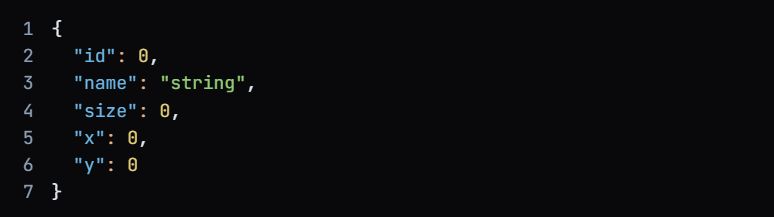
## 1. Create Airport

Method: POST

Endpoint: /api/airport/create

Parameters

* data (object, required): The airport data.

**Example Value:**  


**Response Codes**

* 200: OK

**Example Value:**  


* 201: Created
* 401: Unauthorized
* 403: Forbidden
* 404: Not Found

## 2. Delete Airport

Method: DELETE

Endpoint: /api/airport/delete/{id}

**Parameters**

* id (integer, required): The ID of the airport.

**Response Codes**

* 200: OK
* 204: No Content
* 401: Unauthorized
* 403: Forbidden

## 3. Get Airport

Method: GET

Endpoint: /api/airport/get

**Parameters**

* id (integer, optional): The ID of the airport.
* name (string, optional): The name of the airport.
* size (integer, optional): The size of the airport.
* x (number, optional): X Coordinate.
* **y (number, optional): Y Coordinate.**

**Response Codes**

* 200: OK

Example Value:  


* 401: Unauthorized
* 403: Forbidden
* 404: Not Found

## 4. Update Airport

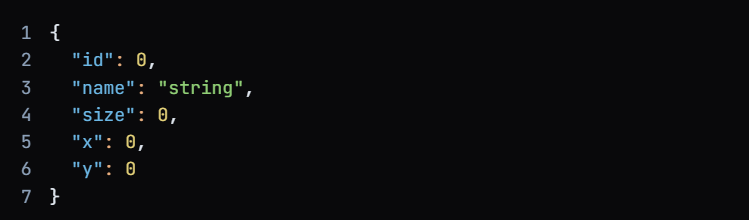
Method: POST

Endpoint: /api/airport/update

**Parameters**

* data (object, required): The airport data.
* update (string, required): Update string.

**Example Value:**

****

**Response Codes**

* 200: OK
* 401: Unauthorized
* 403: Forbidden

# Dispatcher Controller

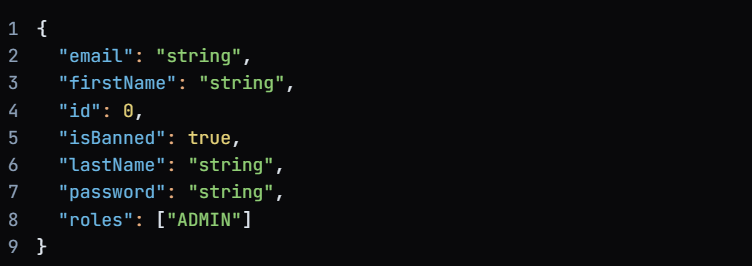
## 1. Create Dispatcher

Method: POST

Endpoint: /api/dispatcher/create

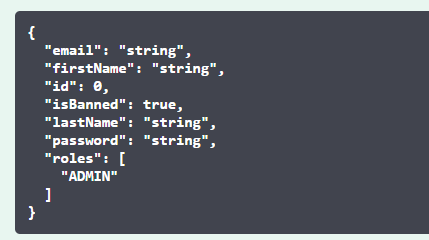
**Parameters**

* data (object, required): The dispatcher data.

**Example Value  
**

**Response Codes**

* 200: OK

**Example Value:**   


* 201: Created
* 401: Unauthorized
* 403: Forbidden
* 404: Not Found

## 2. Delete Dispatcher

Method: DELETE

Endpoint: /api/dispatcher/delete/{id}

**Parameters**

* id (integer, required): The ID of the dispatcher.

**Response Codes**

* 200: OK
* 204: No Content
* 401: Unauthorized
* 403: Forbidden

## 3. Get Dispatcher

Method: GET

Endpoint: /api/dispatcher/get

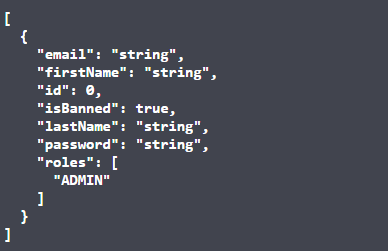
**Parameters**

* email (string, optional): Email of the dispatcher.
* firstName (string, optional): First name of the dispatcher.
* id (integer, optional): The ID of the dispatcher.
* isBanned (boolean, optional): Is the dispatcher banned?
* lastName (string, optional): Last name of the dispatcher.
* password (string, optional): Password of the dispatcher.
* roles (string, optional): Roles of the dispatcher.

**Response Codes**

* 200: OK

**Example Value:**

****

* 401: Unauthorized
* 403: Forbidden
* 404: Not Found

## 4. Update Dispatcher

Method: POST

Endpoint: /api/dispatcher/update

**Parameters**

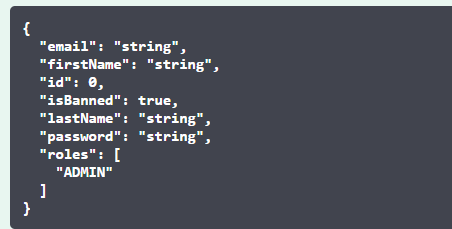
* data (object, required): The dispatcher data.
* update (string, required): Update string.

**Example Value  
**

**Response Codes**

* 200: OK

**Example Value:**

****

* 201: Created
* 401: Unauthorized
* 403: Forbidden
* 404: Not Found

# Flight Controller

## 1. Create Flight

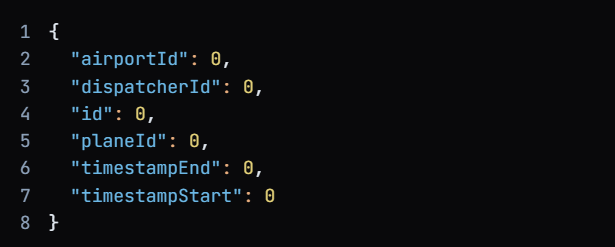
Method: POST

Endpoint: /api/flights/create

**Parameters**

* data (object, required): The flight data.

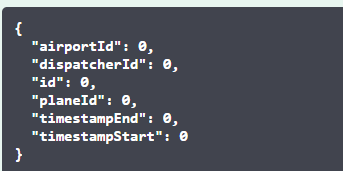
**Example Value**

****

**Response Codes**

* 200: OK

**Example Value:**

****

* 201: Created
* 401: Unauthorized
* 403: Forbidden
* 404: Not Found

## 2. Delete Flight

Method: DELETE

Endpoint: /api/flights/delete/{id}

**Parameters**

* id (integer, required): The ID of the flight.

**Response Codes**

* 200: OK
* 204: No Content
* 401: Unauthorized
* 403: Forbidden

**3. Get Flight**

Method: GET

Endpoint: /api/flights/get

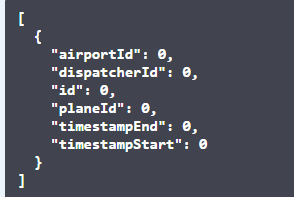
**Parameters**

* airportId (integer, optional): The ID of the airport.
* dispatcherId (integer, optional): The ID of the dispatcher.
* id (integer, optional): The ID of the flight.
* planeId (integer, optional): The ID of the plane.
* timestampEnd (integer, optional): End timestamp.
* timestampStart (integer, optional): Start timestamp.

**Response Codes**

* 200: OK

**Example Value:**

****

* 401: Unauthorized
* 403: Forbidden
* 404: Not Found

## 4. Update Flight

Method: POST

Endpoint: /api/flights/update

**Parameters**

* data (object, required): The flight data.
* update (string, required): Update string.

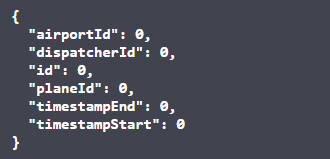
**Example Value**

****

**Response Codes**

* 200: OK

**Example Value:**

****

* 201: Created
* 401: Unauthorized
* 403: Forbidden
* 404: Not Found

# Plane Controller

## 1. Create Plane

Method: POST

Endpoint: /api/plane/create

**Parameters**

* data (object, required): The plane data.

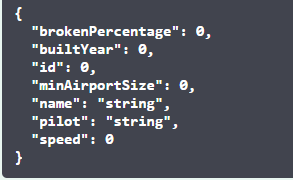
**Example Value:**



**Response Codes**

* 200: OK

**Example Value:**



* 201: Created
* 401: Unauthorized
* 403: Forbidden
* 404: Not Found

## 2. Delete Plane

Method: DELETE

Endpoint: /api/plane/delete/{id}

**Parameters**

* id (integer, required): The ID of the plane.

**Response Codes**

* 200: OK
* 204: No Content
* 401: Unauthorized
* 403: Forbidden

## 3. Get Plane

Method: GET

Endpoint: /api/plane/get

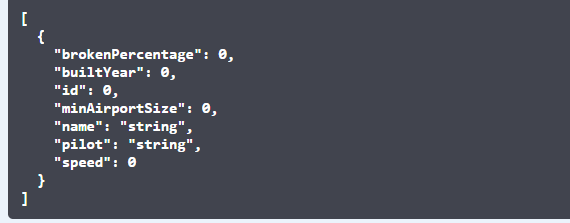
**Parameters**

* brokenPercentage (integer, optional): Broken percentage.
* builtYear (integer, optional): Built year.
* id (integer, optional): The ID of the plane.
* minAirportSize (integer, optional): Minimum airport size.
* name (string, optional): Name of the plane.
* pilot (string, optional): Pilot name.
* speed (integer, optional): Speed.

**Response Codes**

* 200: OK

**Example Value:**



* 401: Unauthorized
* 403: Forbidden
* 404: Not Found

## 4. Update Plane

Method: POST

Endpoint: /api/plane/update

**Parameters**

* data (object, required): The plane data.
* update (string, required): Update string.

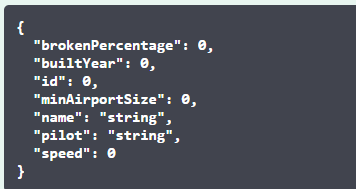
**Example Value:**



**Response Codes**

* 200: OK

**Example Value:**



* 201: Created
* 401: Unauthorized
* 403: Forbidden
* 404: Not Found

# Models

## AirportModel

* id: integer($int64)
* name: string
* size: integer($int32)
* x: number($double)
* y: number($double)

## DispatcherModel

* email: string
* firstName: string
* id: integer($int64)
* isBanned: boolean
* lastName: string
* password: string
* roles: array

## FlightModel

* airportId: integer($int64)
* dispatcherId: integer($int64)
* id: integer($int64)
* planeId: integer($int64)
* timestampEnd: integer($int64)
* timestampStart: integer($int64)

## PlaneModel

* brokenPercentage: integer($int32)
* builtYear: integer($int32)
* id: integer($int64)
* minAirportSize: integer($int32)
* name: string
* pilot: string