# Table of Contents

[**Table of Contents 1**](#_7d6p986sx688)

[**1. Status Codes 1**](#_hjhrezz9m9wf)

[**2. Message Structure (MongoDB) 1**](#_9c3c6mhspczt)

[**3. API Endpoints 1**](#_4qhukgq306pr)

[- Get Message List 1](#_ckapsz5pd223)

[Parameters 1](#_vuyymf3ggny6)

[Example 2](#_c0on8fusxnt8)

[Example Response 2](#_9uq8g8c0yxp7)

[- Send Message 2](#_ratplyke8iz8)

[Parameters 2](#_mxdmt4q4w1dh)

[Example 3](#_nejpbw1p83gr)

[Example Response 3](#_9bx69a34nvis)

[- Delete Message (Admin) (To be done after implementation of the first 2) 3](#_ff55fguitn4q)

[Parameters 3](#_tixq12wl4yo7)

[Example 3](#_yve2l4jyz5fr)

[Example Response 3](#_149w9nru6mqh)

# Status Codes

| Status Code | Explanation |
| --- | --- |
| 200 - OK | The request was successfully processed (but not necessarily accepted, see individual endpoints for details) |
| 201 - Created | Resource is successfully created |
| 400 - Bad Request | User did not provide all necessary parameters, or the provided parameters are not of the expected type |

# Message Structure (MongoDB)

{

“\_id”: [Object ID], //Unique identifier

“userID”: [user ID], //sender

“messageTimestamp”: [UNIX timestamp], //API receives datetime string but DB should be storing timestamps

“messageBody”: [message body string]

}

# API Endpoints

## Get Message List

GET /messages

### Parameters

| **Name** | **Type** | **Location** | **Description** |
| --- | --- | --- | --- |
| start (optional) | String | Query | Datetime string (in UTC) representing the lower bound of the search’s time range |
| end (optional) | String | Query | Datetime string (in UTC) representing the upper bound of the search’s time range |
| userId (optional) | String[] | Query | A string representing which sender ID(s) to search for (can be present in the query parameter multiple times) |
| message (optional) | String[] | Query | A string representing what string(s) to match for in messages (can be present in the query parameter multiple times) |
| page (default:1) | Integer | Query | An integer representing the number of page to fetch (items per page: 50) |

### Example

GET “[uri]/messages?start=2023-07-15T10%3A10%3A10&end=2023-07-24T10%3A10%3A10&userId=132&userId=321&message=hi”

### Example Response

[{

“\_id”: [Object ID], //Unique identifier of the message

“userID”: [user ID], //sender

“messageTime”: [datetime string], //datetime of the stored timestamp (in UTC)

“messageBody”: [message body string] //message body

},{

“\_id”: [Object ID],

“userID”: [user ID],

“messageTime”: [datetime string],

“messageBody”: [message body string]

}]

## Send Message

POST /messages

### Parameters

| **Name** | **Type** | **Location** | **Description** |
| --- | --- | --- | --- |
| userId | String | Message Body | A string representing which user to send the message as |
| message | String | Message Body | A string representing the message to be sent |

### Example

POST “[uri]/messages” -d ‘{

“userId”: 1234,

“message”: “Hi”

}’

### Example Response

{

“result”: true //boolean indicating whether API call is successful or not

}

## Delete Message (Admin) (To be done after implementation of the first 2)

DELETE /messages/:id

### Parameters

| **Name** | **Type** | **Location** | **Description** |
| --- | --- | --- | --- |
| messageId | String | Query | A string representing the message ID |

### Example

DELETE “[uri]/messages/1231”

### Example Response

{

“result”: true //boolean indicating whether API call is successful or not

}