**Task API Documentation**

**(*This API allows you to manage tasks with basic CRUD operations. Ensure you follow the guidelines below for successful interaction.*)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description** |
| 1.0 | 01-Mar-2024 | Ramdas (Osumare.com) | Initial |

## NOTE: Base URL: <http://localhost:3000>

## 

Index

1. Retrive all tasks

[Request](#_kd6tjl5w5wjo)

[Response](#_8b7ijrpuamb8)

2. Retrive a specific task by ID

[Request](#_38pypefkb82)

[Response](#_47awg09ehquu)

3. Create a new task

[Request](#_xtszbus9yomc)

[Response](#_xuhh3y4ufz7)

4. Update an existing task by ID

Request

[Response](#_fvf6lulvf6wv)

5. Delete a task by ID

Request

[Response](#_fvf6lulvf6wv)

Conventions

[Status Codes](#_of6ke88xwee4)

## 

## 

Endpoints

## 1. Retrive all tasaks

Endpoint : ‘GET /tasks’

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **GET** | http:localhost:3000/tasks |

**api\_key**

api\_key Retrive a list of all tasks.

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | [  { "id": 1, "title": "Task 1", "description": "Description 1" },  { "id": 2, "title": "Task 2", "description": "Description 2" }  ] |
| 500 | {"error":"Internal Server Error." |

## 

## 2. Retrive a specific task by ID

Endpoint: ‘GET /tasks/:id’

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **GET** | Api/tasks/1 |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| GET | version | number |

**auth\_key**

The auth\_key that was given in response to /tasks/1

**Description**

Retrieves a specific task by its ID.

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | { "id": 1, "title": "Task 1", "description": "Description 1" } |
| 404 | {"error":"Not Found"} |
| 500 | {"error":"Internal Server Error"} |

## 

## 3. Create a new task

.

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/tasks |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| POST | <Task> | { "title": "New Task", "description": "New Description" } |

**Description**

Creates a new task.

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | **{ "id": 3, "title": "New Task", "description": "New Description" }** |
| 201 | Created |
| 400 | {"error":"Bad Request(Missing title or description)"} |
| 500 | {"error":"Internal Server Erro"} |

## 4. Update an existing task by ID

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **PUT** | api/tasks/:id |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Values** |
| URL\_PARAM | <task\_id> | { "title": "Updated Task", "description": "Updated Description" } |

**task\_id**

Id of the task you want the update.

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | **{ "id": 3, "title": "Updated Task", "description": "Updated Description" }** |
| 400 | {"error":"Bad Request(Missing title or description)"} |
| 400 | {"error":"Not Found"} |
| 500 | {"error":"Internal Server Error"} |

## 5. Delete a task by ID

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **DELETE** | api/task/:id |

**task\_id**

Id of the task you want the delete.

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | **{ "message": "Task deleted successfully" }** |
| 400 | {"error":"Not Found"} |
| 500 | {"error":"Internal Server Error"} |

Glossary

## Conventions

* **Client** - Client application.
* **Status** - HTTP status code of response.
* All the possible responses are listed under ‘Responses’ for each method. Only one of them is issued per request server.
* All response are in JSON format.

## Status Codes

All status codes are standard HTTP status codes. The below ones are used in this API.

2XX - Success of some kind

4XX - Error occurred in client’s part

5XX - Error occurred in server’s part

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 200 | OK |
| 201 | Created |
| 202 | Accepted (Request accepted, and queued for execution) |
| 400 | Bad request |
| 401 | Authentication failure |
| 403 | Forbidden |
| 404 | Resource not found |
| 405 | Method Not Allowed |
| 409 | Conflict |
| 412 | Precondition Failed |
| 413 | Request Entity Too Large |
| 500 | Internal Server Error |
| 501 | Not Implemented |
| 503 | Service Unavailable |