**Note Application**

**Overview**

This document outlines the design and implementation details of a note application built using Java, Spring Boot, MongoDB, and Docker. The application provides RESTful endpoints for managing user’s notes.

**Technical Stack**

Programming Language: Java17

Backend Framework: Spring Boot 3

Database: MongoDB

Build System: Maven

Containerization: Docker

Testing Frameworks: Junit, Mockito

**Rest Endpoints**

**1. Save Note**

POST: http://localhost:9091/note/save

REQUEST

{

"title":"8.Prashob",

"text":"Text is the text of the century to test the statistics",

"tags":[]

}

RESPONSE:

{

"responseCode": 200,

"responseMsg": "SUCCESS"

}

**2. Update Note**

PUT:http://localhost:9091/note/update/{idOfnote}

REQUEST

{

"title":"",

"text":"This is the updated text saving",

"tags":["IMPORTANT","PERSONAL"]

}

RESPONSE

{

"responseCode": 200,

"responseMsg": "SUCCESS"

}

**3. List Notes**

GET: http://localhost:9091/note/list?tags=BUSINESS&page=0

RESPONSE

{

"responseCode": 200,

"responseMsg": "SUCCESS",

"data": [

{

"id": "65d9b3cc9ca61a5bb0fbfd24",

"title": "4.Prashob,BUSINESS IMPORTANT Doc",

"createdDate": "2024-02-24T13:15:56.763"

},

{

"id": "65d9b3ac9ca61a5bb0fbfd22",

"title": "2.Prashob, BUSINESS Doc",

"createdDate": "2024-02-24T13:15:24.433"

}

]

}

**4. Delete Note**

DELETE: http://localhost:9091/note/delete/65d9c10d9d7dd52d7979d1fd

RESPONSE:

{

"responseCode": 200,

"responseMsg": "SUCCESS"

}

**5. Note Statistics**

GET: http://localhost:9091/note/{id}/stat

{

"responseCode": 200,

"responseMsg": "SUCCESS",

"data": {

"the": 3,

"century": 1,

"test": 1,

"of": 1,

"is": 1,

"Text": 1,

"to": 1,

"text": 1,

"statistics": 1

}

}

**Usage**

Clone the repository:

Build the Docker image for noteapp application: docker build -t noteapp .

Navigate to project directory and run the Docker compose file

Access the API endpoints at http://localhost:9091/api/notes