Logged Life

Service Layer

Patrick Mesey

10/30/2022

OVERVIEW

The web application will use Express Library from NodeJS for backend service and utilize an API on the frontend to call and create /manipulate data.

The service layers:

1. Route: This is for the paths to track down specific information.
2. Controller: This is pointed to by the route in order to handle the endpoints of HTTP requests.

Specifications:

1. Get cards:

   Method: GET

   URL: https://capstone.herokuapp.com/api/cards

   Purpose: Retrieve cards available in the database.

   Example request:

   curl -request GET -url https://capstone.herokuapp.com/api/cards

   Success Response:

   {title: “Hawaii”,

   message: “Beautiful Hawaiian River”,

   creator: “John Doe”,

   tags: [“Hawaii”, “River”, “Water”],

   selectedFile: 1225486.png,

   likeCount: {

   type: 1, },

createdAt: {

type: 10/30/2022,

}

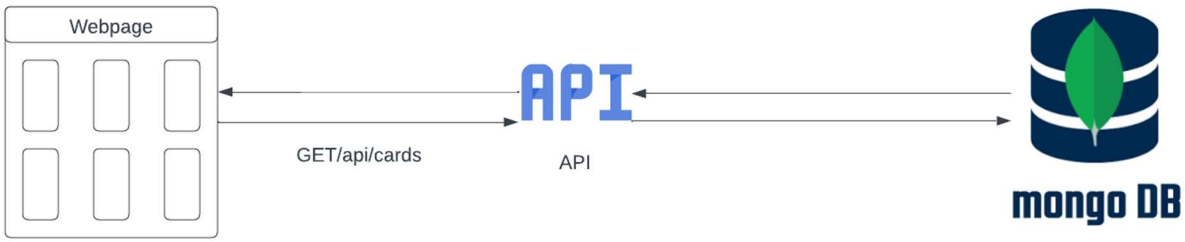
Error response example:

{

“status”: 404,

“message”: “Image not found”

}



2. Get cards by Tag:

Method: GET

URL: https://capstone.herokuapp.com/api/cards/:tag

Purpose: Retrieve cards available in the database containing a tag of the searched tag. This will give nearly the only result, but filtered during a search instead of all display during load.

Example request:

curl -request GET -url https://capstone.herokuapp.com/api/cards/river

Success Response:

{title: “Hawaii”,

message: “Beautiful Hawaiian River”,

creator: “John Doe”,

tags: [“Hawaii”, “River”, “Water”],

selectedFile: 1225486.png,

likeCount: {

type: 1, },

createdAt: {

type: 10/30/2022,

}

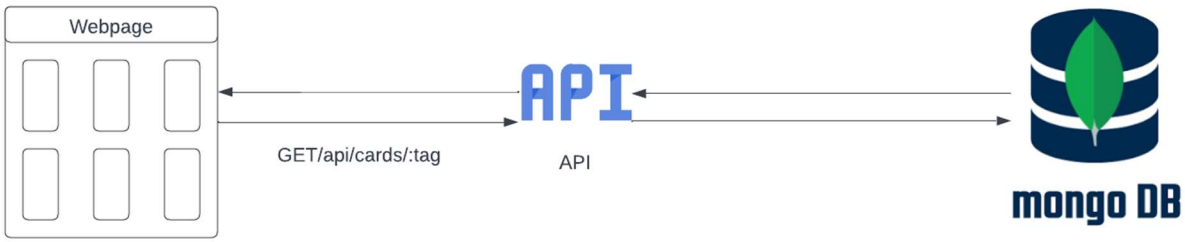
Error response example:

{

“status”: 404,

“message”: “No images containing tag”

}



3. Create a new card:

Method: POST

URL: https://capstone.herokuapp.com/api/cards

Purpose: After logged in, the user will be able to make a post on the main cards page without migrating to a different page. It is meant to keep this application as simple as possible. Clicking “submit” will direct that data to be processed and then displayed on the main cards page.

Example post:

-data

{

title: “Finland”,

message: “Best Food Every!!!”,

creator: “Martha”,

tags: [“Finland”, “International”, “Karelian”],

selectedFile: FinFood1.png,

likeCount: {

type: number,

default: 0,

},

createdAt: {

type: 10/30/2022,

}

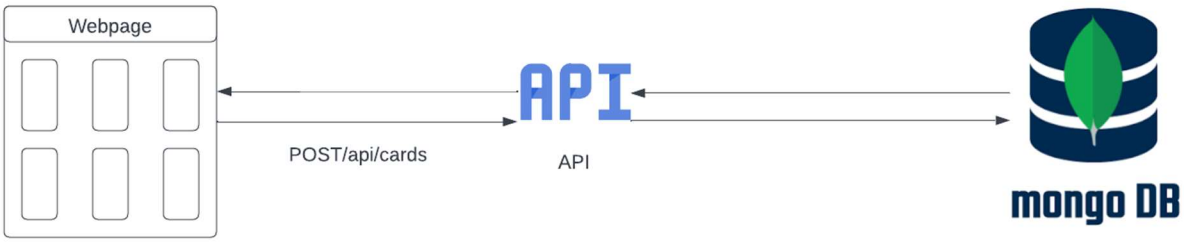
cardMessage(card) runs a cardMessage function with the details of the card to be processed.

Success Example:

Status (201): message: “Card Created Successfully”

Error Example:

Status(409): message: "Error: Card could not be created”



This is a simplification of potential endpoints. More are to be considered for the final during the creation of the application. I am not sure how to handle much of this but will start basic and continue to add on depending on feedback and barriers approached during the process.