

Service Layer: Crystals and Tinks

Arty Ruiz

05 November 2023

Overview

I will use the Express library from NodeJS. I will intend to use a REST API which should use the frontend to call in to create and manipulate my data.

Specifications

I have listed all routes and endpoints in my service layer. I currently have a placeholder URL, the URL will simply take you to where my app will be hosted, so it shouldn't work currently. Once everything works as planned, it will be hosted in Heroku, as it has a free function and I think works best for creating apps at my current level.

Get logged in:

Method: GET

URL: <https://crystalsandtinks-6273b29a4dfb.herokuapp.com>

Purpose: When a user logs in to the app this endpoint will be called from the front end, and the data returned will be used to render the app's home screen.

Example request:

Curl -request GET -url
/ARTURORUIZ18/crystalsntinkss/1.0.0/user/login?username=&password=

Success response:

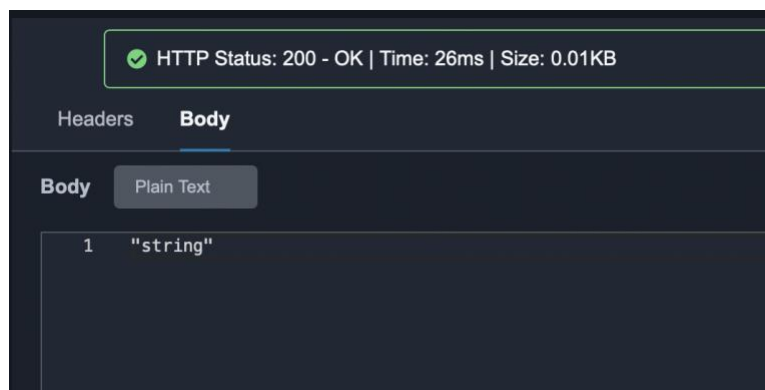


Diagram:



Signup and Create an Account

Method: GET

URL: <https://crystalsandtinks-6273b29a4dfb.herokuapp.com/api/signup>

Example request:

```
curl -X 'POST' \
  'https://crystalsandtinks.swagger.io/v2/user' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "id": 0,
    "username": "string",
    "firstName": "string",
    "lastName": "string",
    "email": "string",
    "password": "string",
    "phone": "string",
    "dob": int
  }'
```

Success Response:

200	Response body
	<pre>{ "code": 200, "type": "unknown", "message": "1843938241" }</pre>
	Response headers
	<pre>access-control-allow-headers: Content-Type,api_key,Authorization access-control-allow-methods: GET,POST,DELETE,PUT access-control-allow-origin: * content-type: application/json date: Mon, 06 Nov 2023 01:51:23 GMT server: Jetty(9.2.9.v20150224)</pre>
Responses	
Code	Description
default	successful operation

Diagram:



Search for Crystals by Name

Method: GET

URL: <https://crystalsandtinks-6273b29a4dfb.herokuapp.com/api/crystalsname>

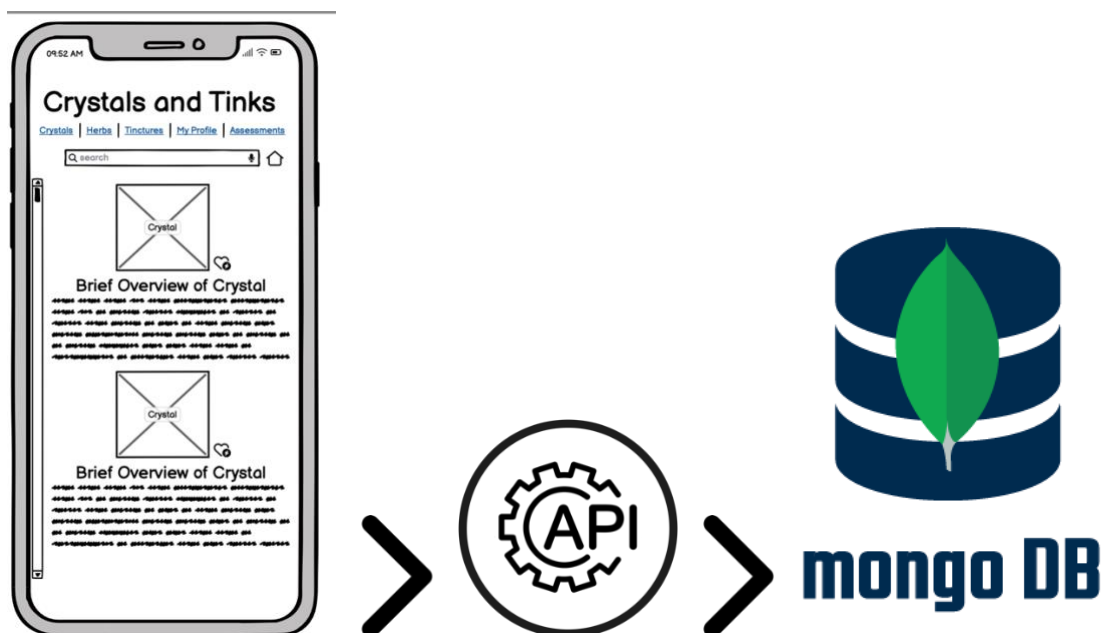
Example request:

```
curl -X 'GET' \
  'https://crystalsandtinks-6273b29a4dfb.herokuapp.com/api/crystalsname' \
  -H 'accept: application/json'
```

Success Response:

Code	Description
200	successful operation
	Example Value Model
	<pre>{ "id": 0, "category": { "id": 0, "name": "string" }, "name": "string" }</pre>

Diagram:



Search for Crystals by Color

Method: GET

URL: <https://crystalsandtinks-6273b29a4dfb.herokuapp.com/api/crystalscolor>

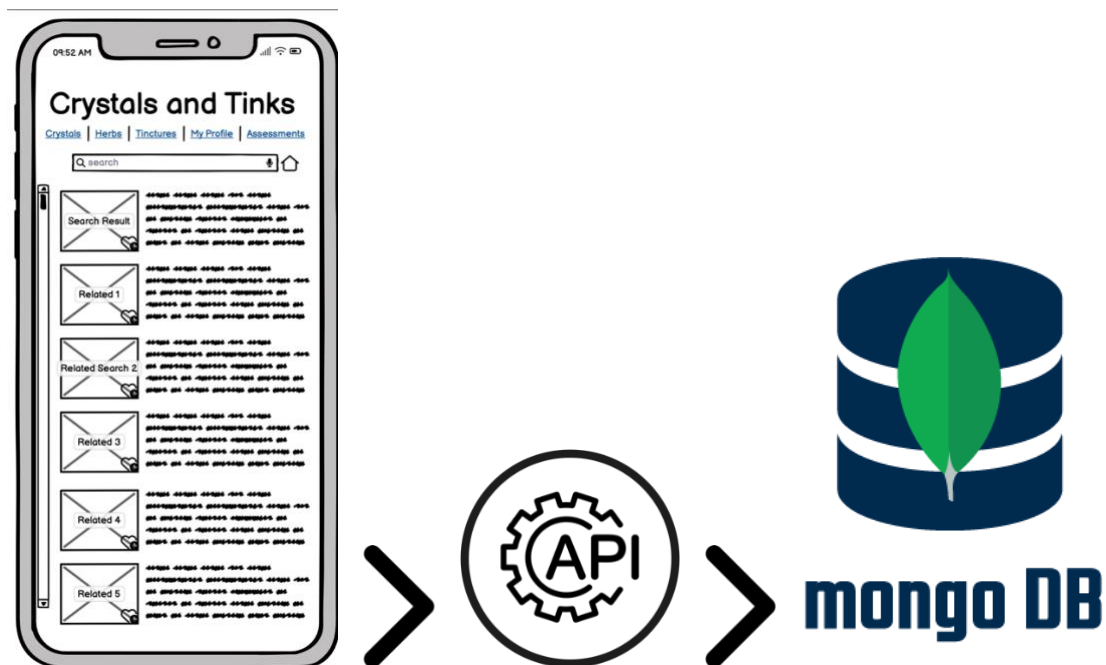
Example request:

```
curl -X 'GET' \
  'https://crystalsandtinks-6273b29a4dfb.herokuapp.com/api/crystalscolor' \
  -H 'accept: application/json'
```

Success Response:

Code	Description
200	successful operation
Example Value	Model
<pre>{ "id": 0, "category": { "id": 0, "name": "string" }, "name": "string" }</pre>	

Diagram:



Search for Crystals by Keywords

Method: GET

URL: <https://crystalsandtinks-6273b29a4dfb.herokuapp.com/api/crystalskeywords>

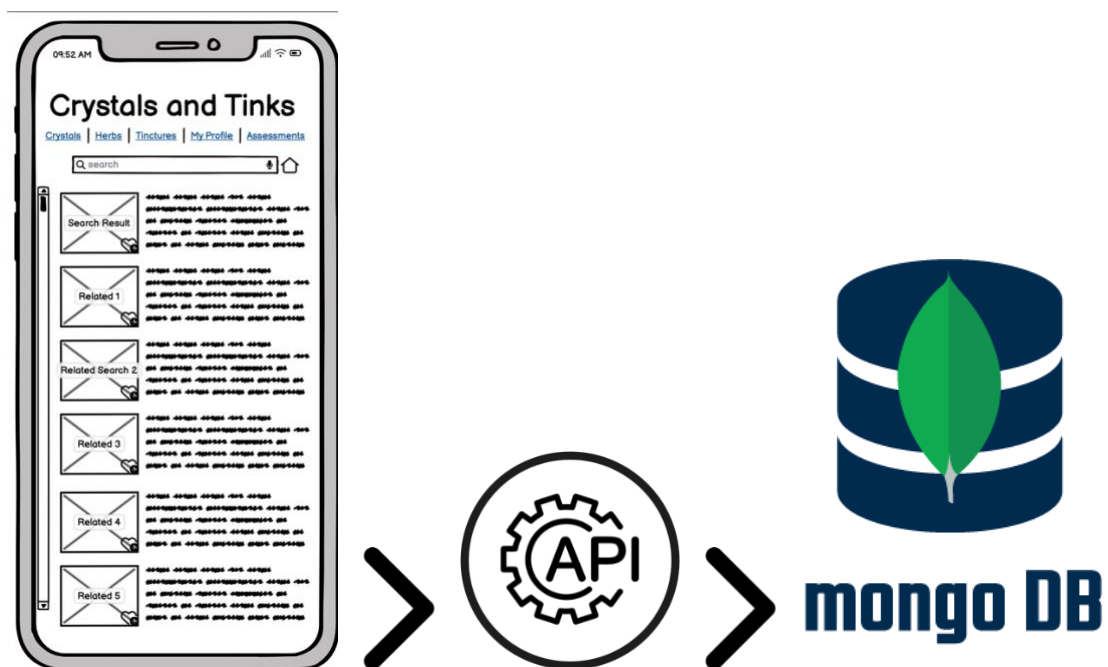
Example request:

```
curl -X 'GET' \
  'https://crystalsandtinks-6273b29a4dfb.herokuapp.com/api/crystalskeywords' \
  -H 'accept: application/json'
```

Success Response:

Code	Description
200	successful operation
	Example Value Model
	<pre>{ "id": 0, "category": { "id": 0, "name": "string" }, "name": "string" }</pre>

Diagram:



Search for Herbs by Name

Method: GET

URL: <https://crystalsandtinks-6273b29a4dfb.herokuapp.com/api/herbs>

Example request:

```
curl -X 'GET' \
  'https://crystalsandtinks-6273b29a4dfb.herokuapp.com/api/herbs' \
  -H 'accept: application/json'
```

Success Response:

Code	Description
200	successful operation
	Example Value Model
	<pre>{ "id": 0, "category": { "id": 0, "name": "string" }, "name": "string" }</pre>

Diagram:



Search for Herbs by Type

Method: GET

URL: <https://crystalsandtinks-6273b29a4dfb.herokuapp.com/api/herbstype>

Example request:

```
curl -X 'GET' \
  'https://crystalsandtinks-6273b29a4dfb.herokuapp.com/api/herbstype' \
  -H 'accept: application/json'
```

Success Response:

Responses	
Code	Description
200	successful operation
Example Value Model	
<pre>{ "code": 0, "type": "string", "message": "string" }</pre>	

Diagram:



Logout of the App

Method: GET

URL: <https://crystalsandtinks-6273b29a4dfb.herokuapp.com/api/logout>

Example request:

curl

curl -X 'GET' \

<https://crystalsandtinks-6273b29a4dfb.herokuapp.com/api/logout>

-H 'accept: application/json'

Success Response:

Server response	
Code	Details
200	<div>Response body</div> <pre>{ "code": 200, "type": "unknown", "message": "ok" }</pre> <div>Response headers</div> <pre>access-control-allow-headers: Content-Type,api_key,Authorization access-control-allow-methods: GET,POST,DELETE,PUT access-control-allow-origin: * content-type: application/json date: Mon,06 Nov 2023 02:15:53 GMT server: Jetty(9.2.9.v20150224)</pre>

Diagram:



Add image to Profile

Method: POST

URL: <https://crystalsandtinks-6273b29a4dfb.herokuapp.com/api/uploadimage>

Example request:

curl

curl -X 'GET' \

'<https://crystalsandtinks-6273b29a4dfb.herokuapp.com/api/uploadimage>'

-H 'accept: application/json'

Success Response:

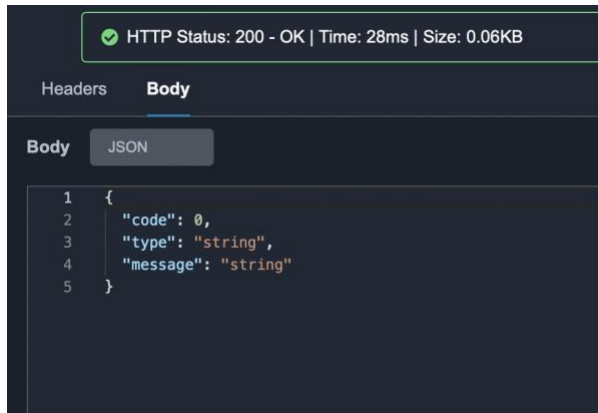


Diagram:



Take Assessment

Method: POST

URL: <https://crystalsandtinks-6273b29a4dfb.herokuapp.com/api/uploadimage>

Example request:

`curl -X 'POST' \`

```

'https://virtserver.swaggerhub.com/ARTURORUIZ18/crystalsntinkss/1.0.0/user/createWithArra
y' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '[
{
  "id": 0,

```

```
"type": "string",  
"ailments": "string",  
"userinput": "string",  
"recommendation": "string",  
"image": "string",  
"link": "string", }  
'
```

Success Response:

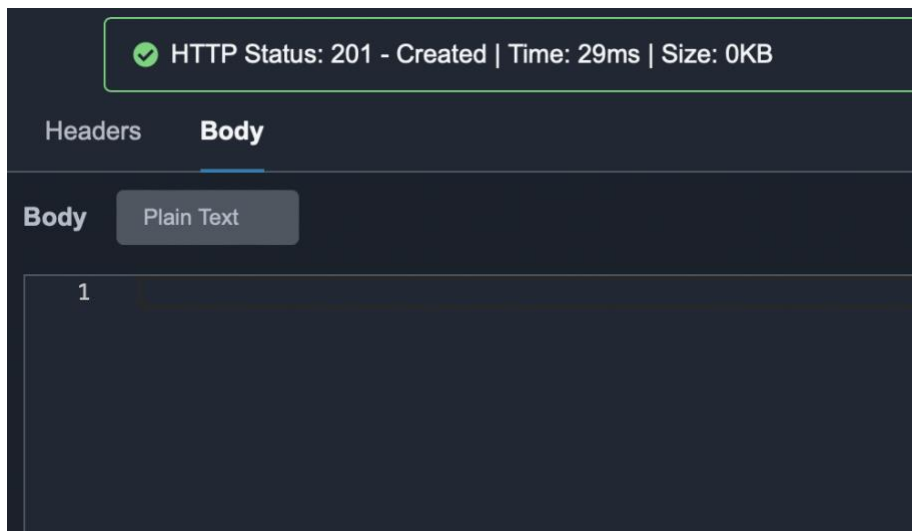


Diagram:



User updates Information in Profile

Method: PUT

URL: <https://crystalsandtinks-6273b29a4dfb.herokuapp.com/api/editprofile>

Example request:

```
curl -X 'PUT' \
  'https://virtserver.swaggerhub.com/ARTURORUIZ18/crystalsntinkss/1.0.0/user/artsy' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "id": 0,
    "username": "string",
    "firstName": "string",
    "lastName": "string",
    "email": "string",
    "password": "string",
    "phone": "string",
    "userStatus": 0
  }'
```

Success Response:

Responses	
Code	Description
200	successful operation
Example Value Model	
<pre>{ "code": 0, "type": "string", "message": "string" }</pre>	

Failed Response:

Code	Details	
400	<p>Error: response status is 400</p> <p>Response headers</p> <pre>access-control-allow-credentials: true access-control-allow-headers: X-Requested-With, Content-Type, Accept, Origin access-control-allow-methods: * access-control-allow-origin: * cache-control: no-cache content-encoding: gzip content-length: 20 content-type: application/json; charset=utf-8 date: Mon, 06 Nov 2023 02:32:51 GMT etag: W/"14-yIMMzzqGPkico39NpXK600BdB3s" expires: -1 server: Jetty(9.4.49.v20220914) status: 400 Bad Request x-powered-by: Express, Phusion Passenger(R) 6.0.18</pre>	
Responses		
Code	Description	Links
400	Invalid user supplied	<i>No links</i>
404	User not found	<i>No links</i>

Diagram:



All my endpoints demonstrated above are still ongoing. I struggled trying to figure out how to run mockups and if I needed to add any redirects for html pages. I think it is only the items that

need to connect to an api, and not so much my cloud image. I will probably work on images on the frontend and will use google cloud storage to store my images.