

Assignment Two

API Documentation

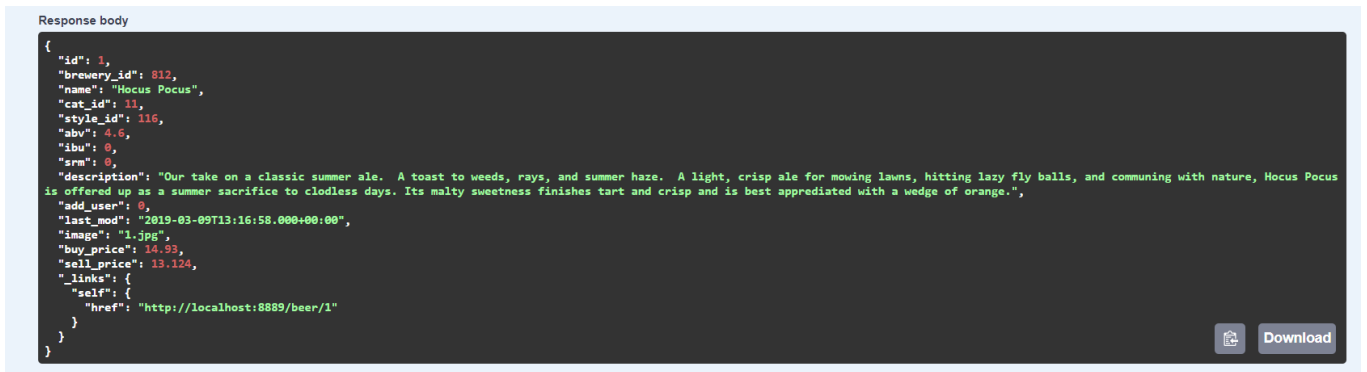
<http://localhost:8888/swagger-ui/index.html>

Evaluation

I am using the swagger UI to show each functionality of the API that has been created and working so far(Postman is also optional for this as well).

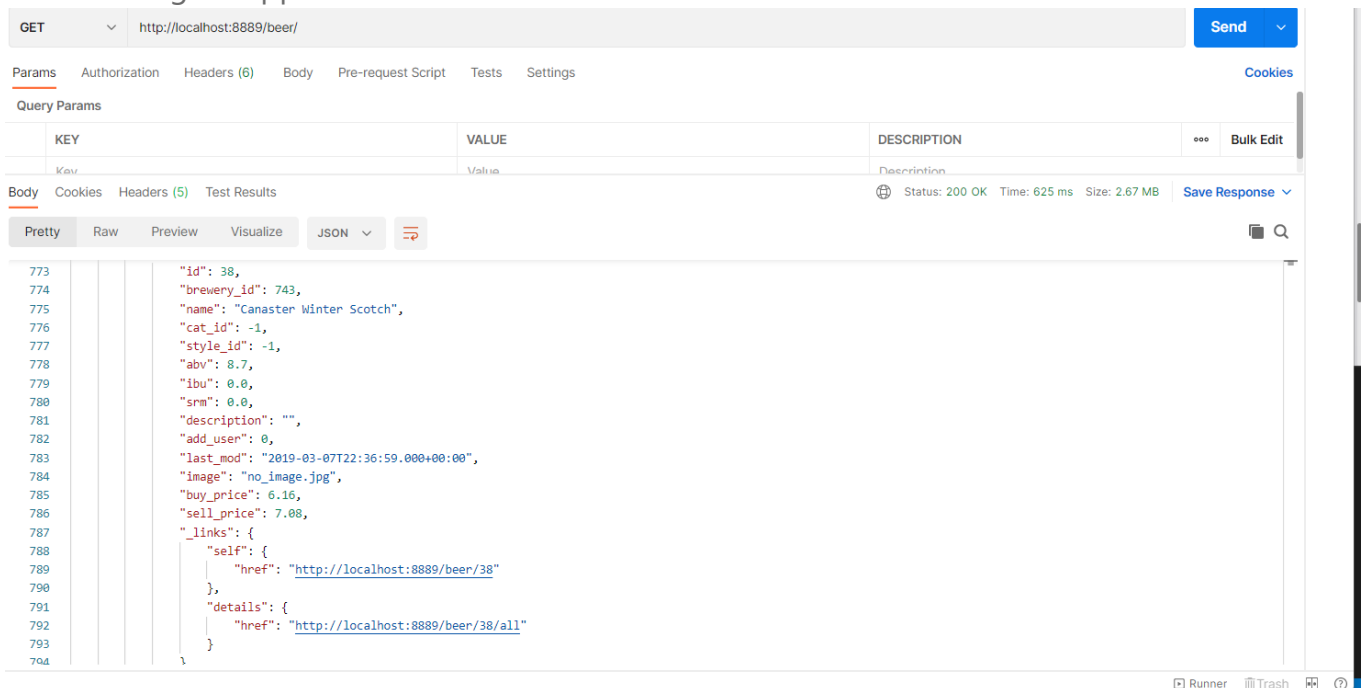
The screenshot displays the Swagger UI interface in a web browser. The address bar shows the URL `localhost:8888/swagger-ui/index.html/#/v3/api-docs`. The interface features a top navigation bar with the Swagger logo and a search bar. Below this, the 'OpenAPI definition' section is visible, indicating the API version is v0.0.0. The 'Servers' section shows the default server URL as `http://localhost:8888`. The main content area lists two API controllers: 'brewery-controller' and 'beer-hateoas-controller'. The 'brewery-controller' section lists several endpoints: `GET /brewery/{id}` (Get a brewery by its id), `PUT /brewery/{id}`, `DELETE /brewery/{id}`, `GET /brewery` (Get a list of breweries), `POST /brewery`, and `GET /brewery/qrcode/{breweryId}` (Get a brewery's qr code). The 'beer-hateoas-controller' section shows the `GET /beer/{id}` endpoint (Get a beer by its id). This endpoint is expanded, showing its parameters: a required integer path parameter `id` with a value of `1`. Below the parameters, there are 'Execute' and 'Clear' buttons. The 'Responses' section shows the curl command for the request: `curl -X 'GET' \ 'http://localhost:8888/beer/1' \ -H 'accept: application/hal+json'`. The 'Request URL' is `http://localhost:8888/beer/1`. The 'Server response' section shows a `200` status code and a 'Response body'.

Let's start by discussing the `getBeerById` function. So one running it will display all the details of a single beer by typing in its id. Then once executing it will display the following:



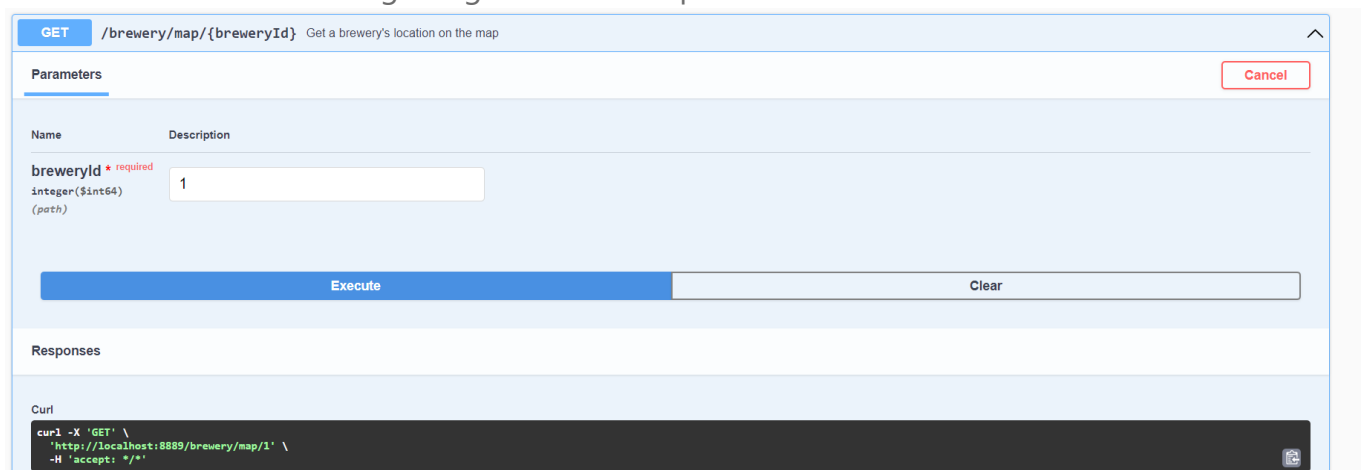
It will also include a link.

Next is displaying all the beers along with their details and two different links. Because of the Swagger issue for not executing the function the Postman would be the next best option just in case this might happen.



As shown in the image above, the getBeers function will display all the beers along with their details and two different links which are the self link and another that would display a beer's name and description as well as the associated brewery.

Now then let's discuss the getting breweries map location function.



By filling an id to any brewery then executing it will display an html script brewery's location a

map along with its details.

Server response

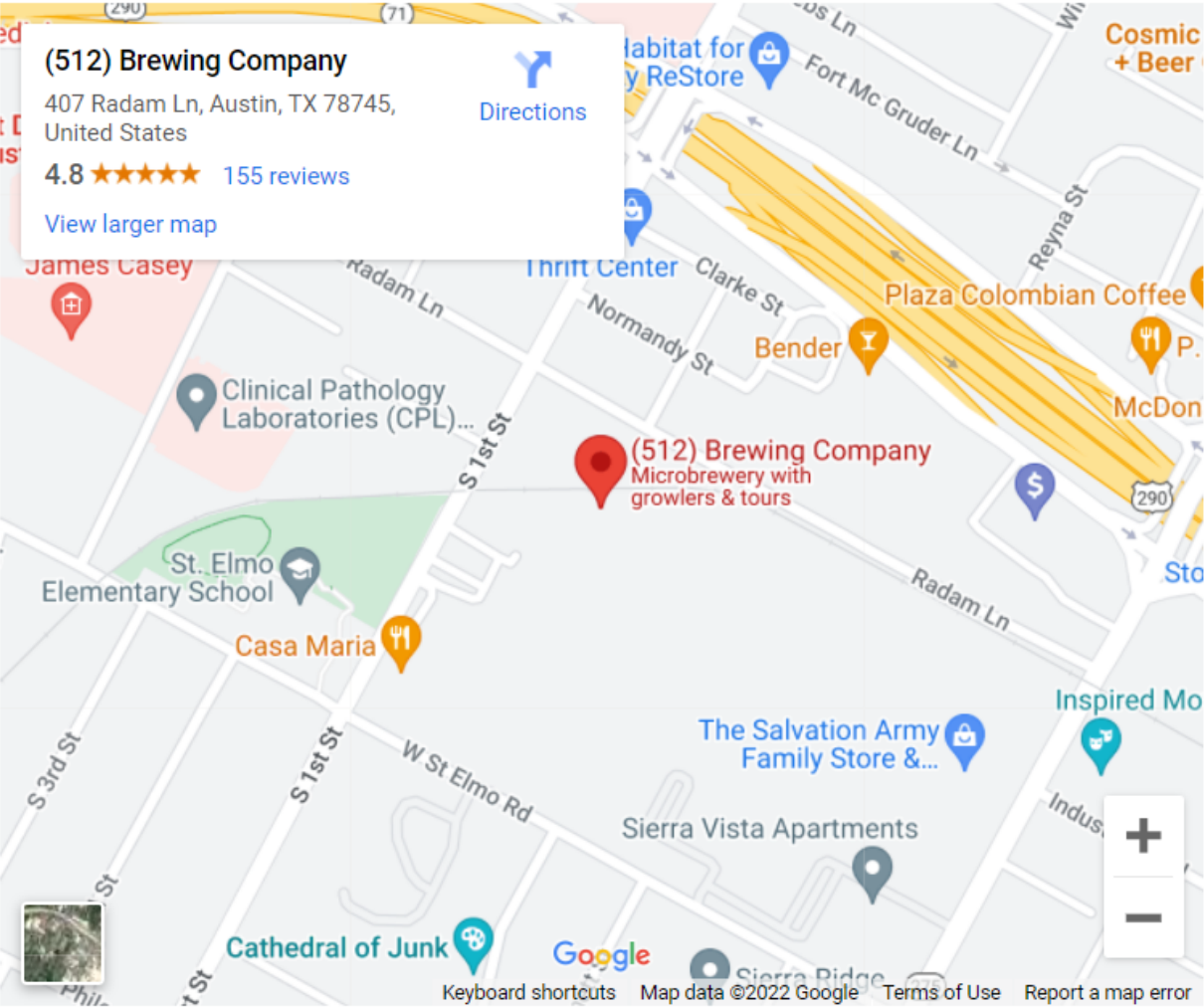
Code	Details
200	<div><div>Response body</div><div><div><html><body>ch2:(512) Brewing Company407 Radam, F200AUAustin</h2><iframe width="600" height="500" id="gmap_canvas" src="https://maps.google.com/maps?q=(512) Brewing Company407 Radam, F200AUAustin78745United States&output=embed" frameborder="0" scrolling="no" marginheight="0" marginwidth="0"></div><div><div>Download</div></div></div><div><div>Response headers</div><div><div>connection: keep-alive</div><div>content-length: 293</div><div>content-type: text/plain; charset=UTF-8</div><div>date: Thu, 31 Mar 2022 20:22:36 GMT</div><div>keep-alive: timeout=60</div></div></div></div>

Responses

Code	Description	Links
200	OK	No links

When displaying it on the browser:

(512) Brewing Company407 Radam, F200AUAustin



The QR code function (which is after the map) will display a QR code of the brewery's contact details. Similar to the map by typing a brewery's id then execute will display the QR code then by scanning the code with brewery's contact details gets added to your phone.


Curl

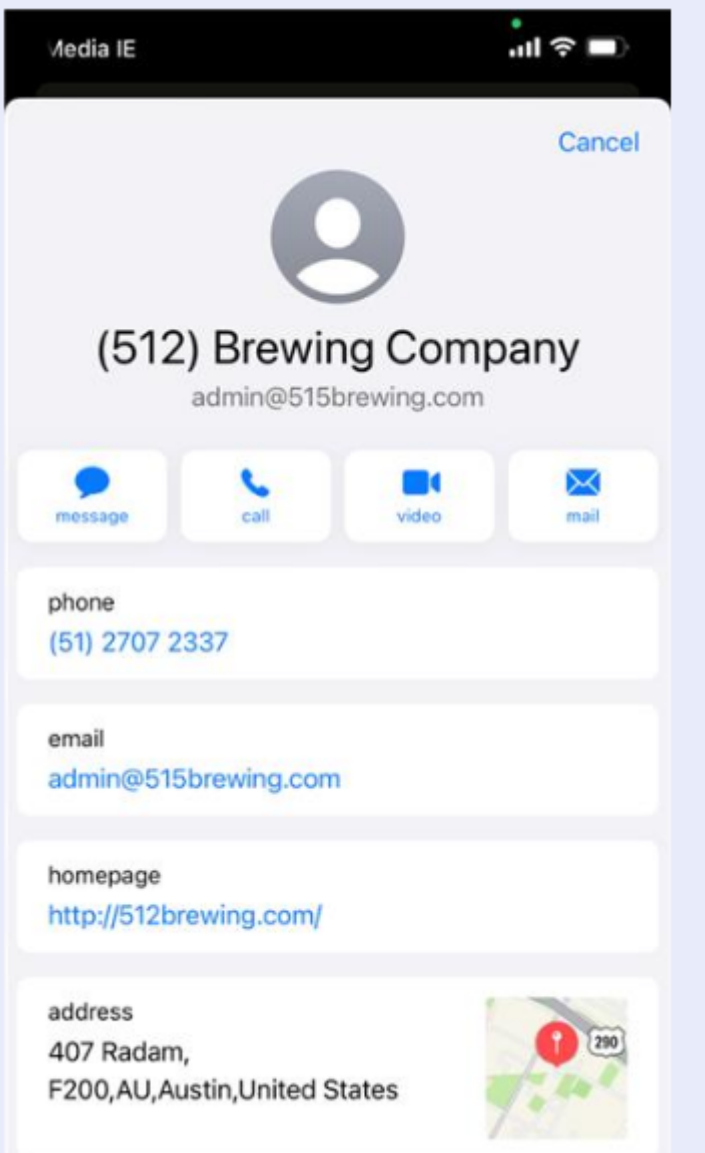
```
curl -X 'GET' \
'http://localhost:8889/brewery/qrcode/1' \
-H 'accept: image/png'
```

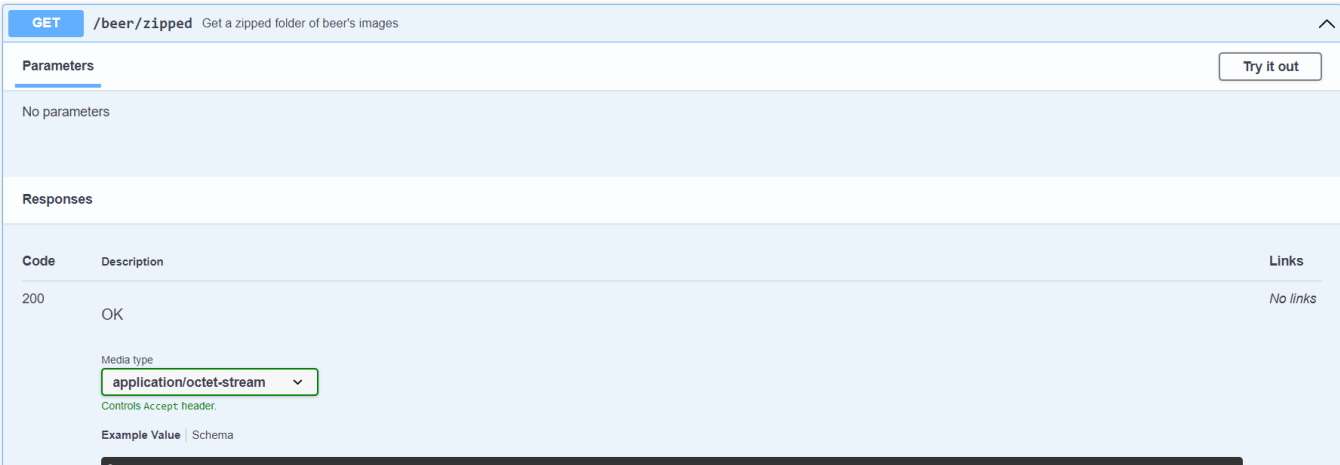
Request URL

```
http://localhost:8889/brewery/qrcode/1
```

Server response

Code	Details
200	<div>Response body</div> <div></div> <div>Response headers</div>

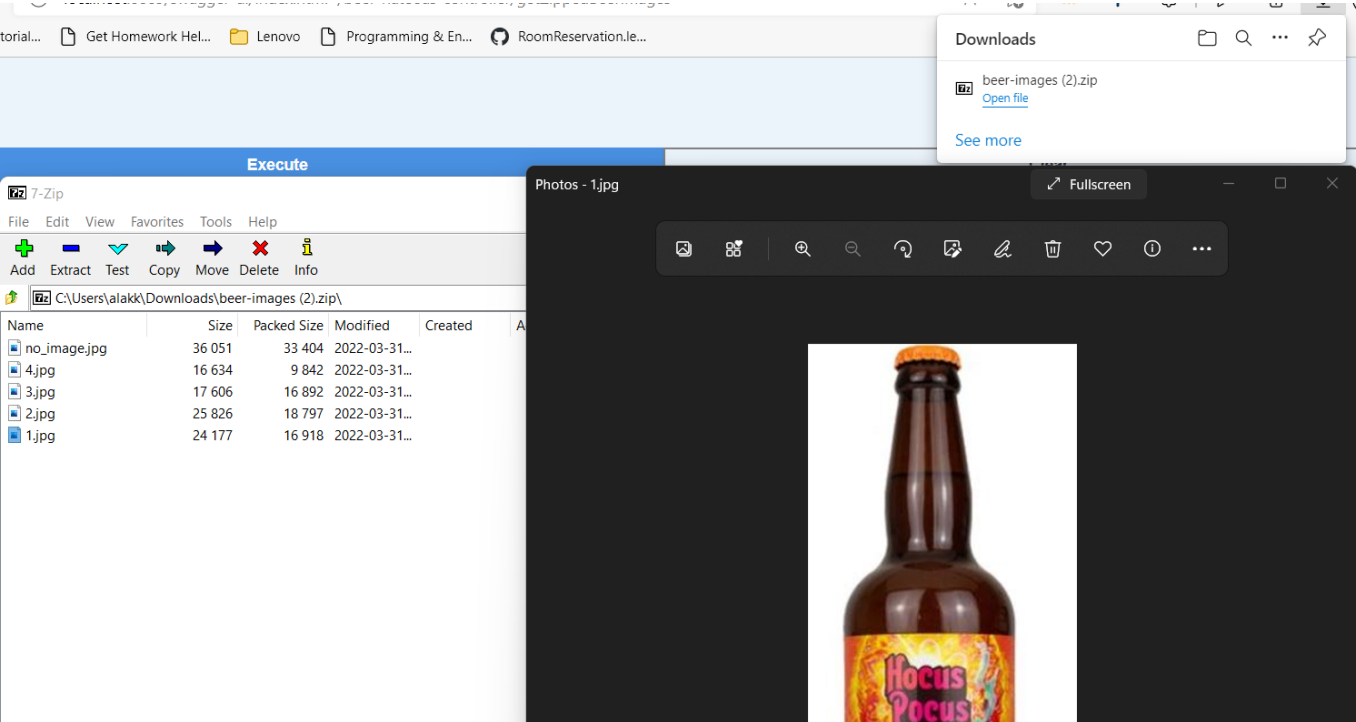




The function shown in the image above is used to a zipped folder for beers images. So when executing it will display a download link for the zipped folder.



When clicking on the download link it will download the zipped folder that contains these images.



So now moving on to demonstrate the getting beer's thumbnail or large images.

GET

/beer/image/{beerId}/{size}

Get beer image

^

Parameters

Try it out

Name	Description
size * required string (path)	Thubnail or Large
<input type="text" value="size"/>	
beerId * required integer(\$int64) (path)	beerId
<input type="text" value="beerId"/>	

Responses

In the image above shows then it consists of if/else feature that when choosing either large or thumbnail of a beer based on its id it will display the image of either of them. For large:

Parameters

Try it out

Name	Description
size * required string (path)	Thubnail or Large
<input type="text" value="large"/>	
beerId * required integer(\$int64) (path)	2
<input type="text" value="2"/>	

Execute

Clear

Responses

Response body



For thumbnail:

Name	Description
size * required string (path)	Thumbnail or Large
	<input type="text" value="thumbnail"/>
beerId * required integer(\$int64) (path)	<input type="text" value="2"/>
<div><div>Execute</div><div>Clear</div></div>	

Response body



After images comes the PDF. The issue with PDF's functionality however is that the data in the database is not consistent. So a fix was added to cope with the missing style i.e. -1

```
Style style = beer.getStyle_id() == -1 ? null : styleRepository.findById(beer.getStyle_id())
```



Hocus Pocus

4.6

13.124

Other Style

Our take on a classic summer ale. A toast to weeds, rays, and summer haze. A light, crisp ale for mowing lawns, hitting lazy

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8889/beer/pdf/1' \
  -H 'accept: application/pdf'
```

Request URL

```
http://localhost:8889/beer/pdf/1
```

Server response

Code	Details
200	<div>Response body Download file</div> <div>Response headers</div>

```
connection: keep-alive
content-disposition: attachment; filename="Hocus Pocus.pdf"
content-length: 25734
content-type: application/pdf
date: Thu, 31 Mar 2022 21:42:12 GMT
keep-alive: timeout=60
```

Responses

Similar to downloading the zipped folder of images it will download a pdf of the beer's description.

An Example of pagination is on the BeerController class. It allows you to download as much or little of the data as you need.

Benchmarks and Enhancements

Why is a benefit to add a REST API?

It separates the data from the presentation layer/logic and allows you to create a RESTful API that can be consumed by any client.

Hateaos implementation is not a great investment of development time. While it is claimed it separates the client / server coupling with the links, add one parameter to your REST API and you will have to change the client.

Future Enhancements:

Implement API Versioning If a client is not able to handle a certain version of the API, it can be changed to a different version.