REST API REPORT

Jack kelly

K00236610 K00236610

Contents

NOTE: For the API Documentation I have provided screenshots of each function in the documentation I created with Swagger and HAL Explorer. I used the HAL explorer for functions that contain a lot of JSON Data. Swagger UI works for smaller functions like getting a beer by an ID. Swagger UI is slower to load data than the HAL explorer

Start-up URLs:

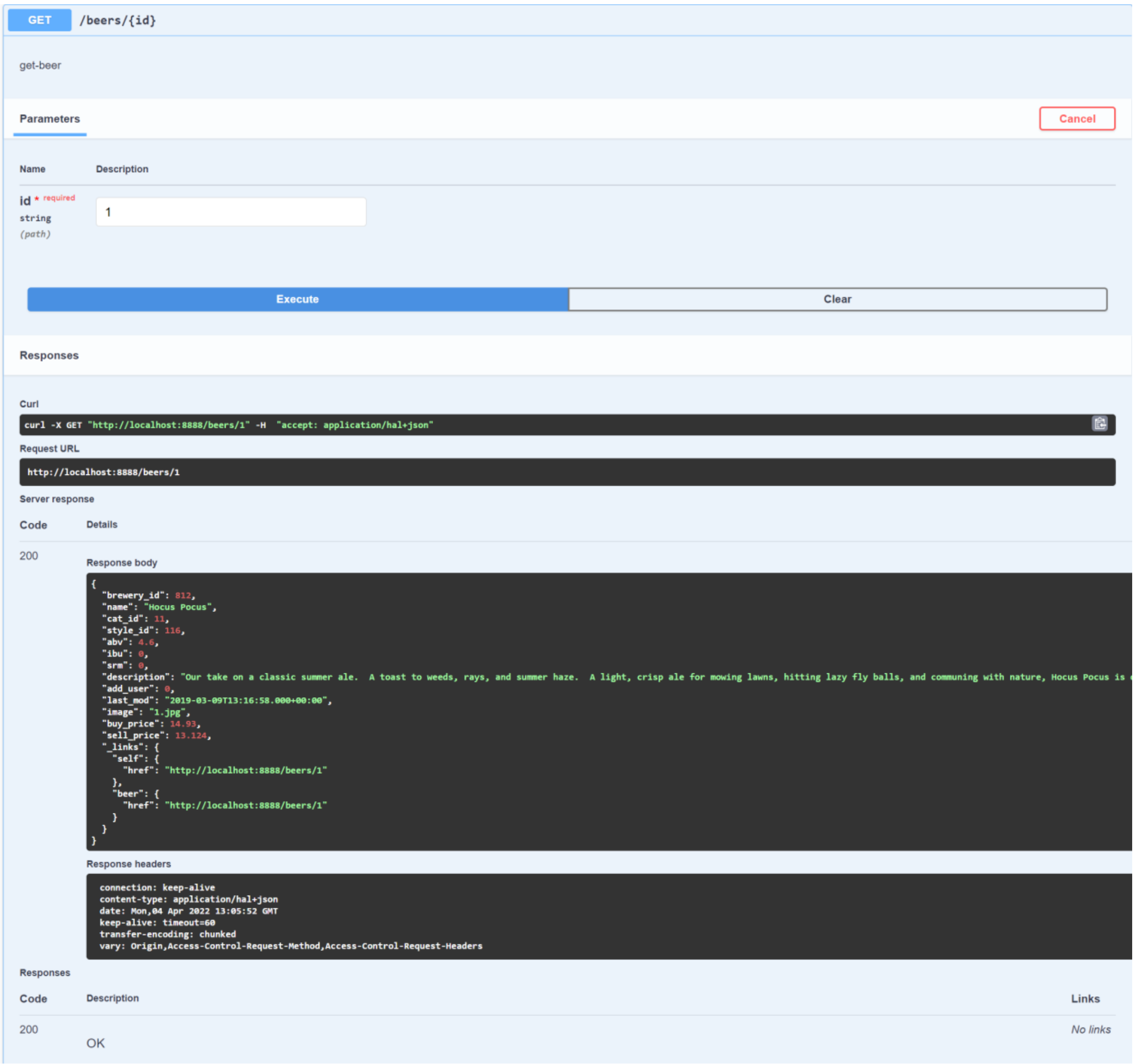
Hal Explorer: http://localhost:8888/explorer/index.html#uri=/

Swagger UI: http://localhost:8888/swagger-ui/index.html?configUrl=/v3/api-docs/swagger-config#/

API function screenshots and documentation

Example using swagger UI

Get a beer by ID



Name	Description
id * required string *(path)*	1

Curl

```
curl -X GET "http://localhost:8888/beers/1" -H  "accept: application/hal+json"
```

Request URL

```
http://localhost:8888/beers/1
```

Server response

Code | Details

200

Response body

```
{
  "brewery_id": 812,
  "name": "Hocus Pocus",
  "cat_id": 11,
  "style_id": 116,
  "abv": 4.6,
  "ibu": 0,
  "srm": 0,
  "description": "Our take on a classic summer ale.  A toast to weeds, rays, and summer haze.  A light, crisp ale for mowing lawns, hitting lazy fly balls, and communing with nature, Hocus Pocus is
  "add_user": 0,
  "last_mod": "2019-03-09T13:16:58.000+00:00",
  "image": "1.jpg",
  "buy_price": 14.93,
  "sell_price": 13.124,
  "_links": {
    "self": {
      "href": "http://localhost:8888/beers/1"
    },
    "beer": {
      "href": "http://localhost:8888/beers/1"
    }
  }
}
```

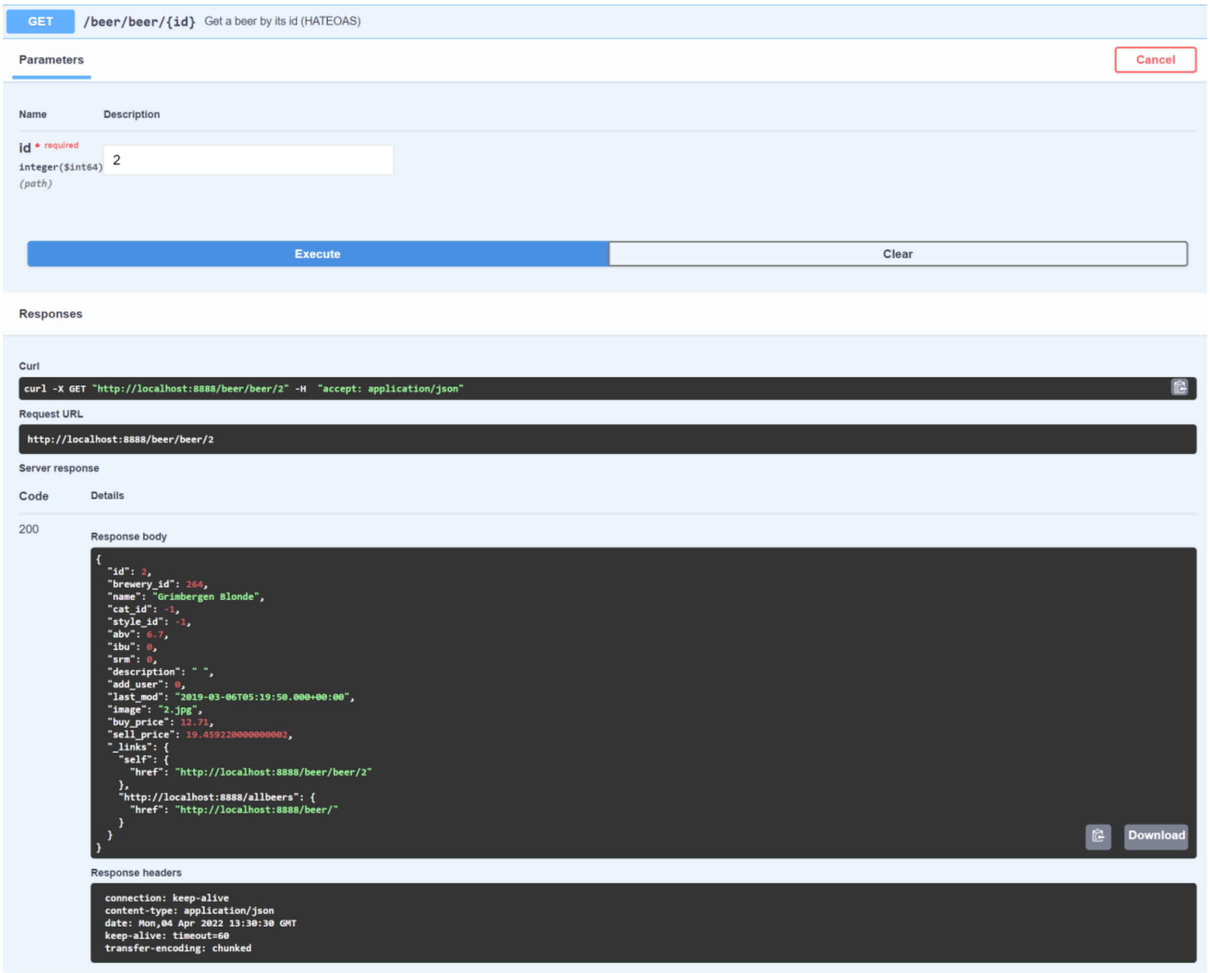
Response headers

```
connection: keep-alive
content-type: application/hal+json
date: Mon,04 Apr 2022 13:05:52 GMT
keep-alive: timeout=60
transfer-encoding: chunked
vary: Origin,Access-Control-Request-Method,Access-Control-Request-Headers
```

Responses

Code	Description	Links
200	OK	*No links*

Get a beer by ID using HATEAS principles



id * required
integer($int64)
(path)

2

Curl

```
curl -X GET "http://localhost:8888/beer/beer/2" -H  "accept: application/json"
```

Request URL

```
http://localhost:8888/beer/beer/2
```

Server response

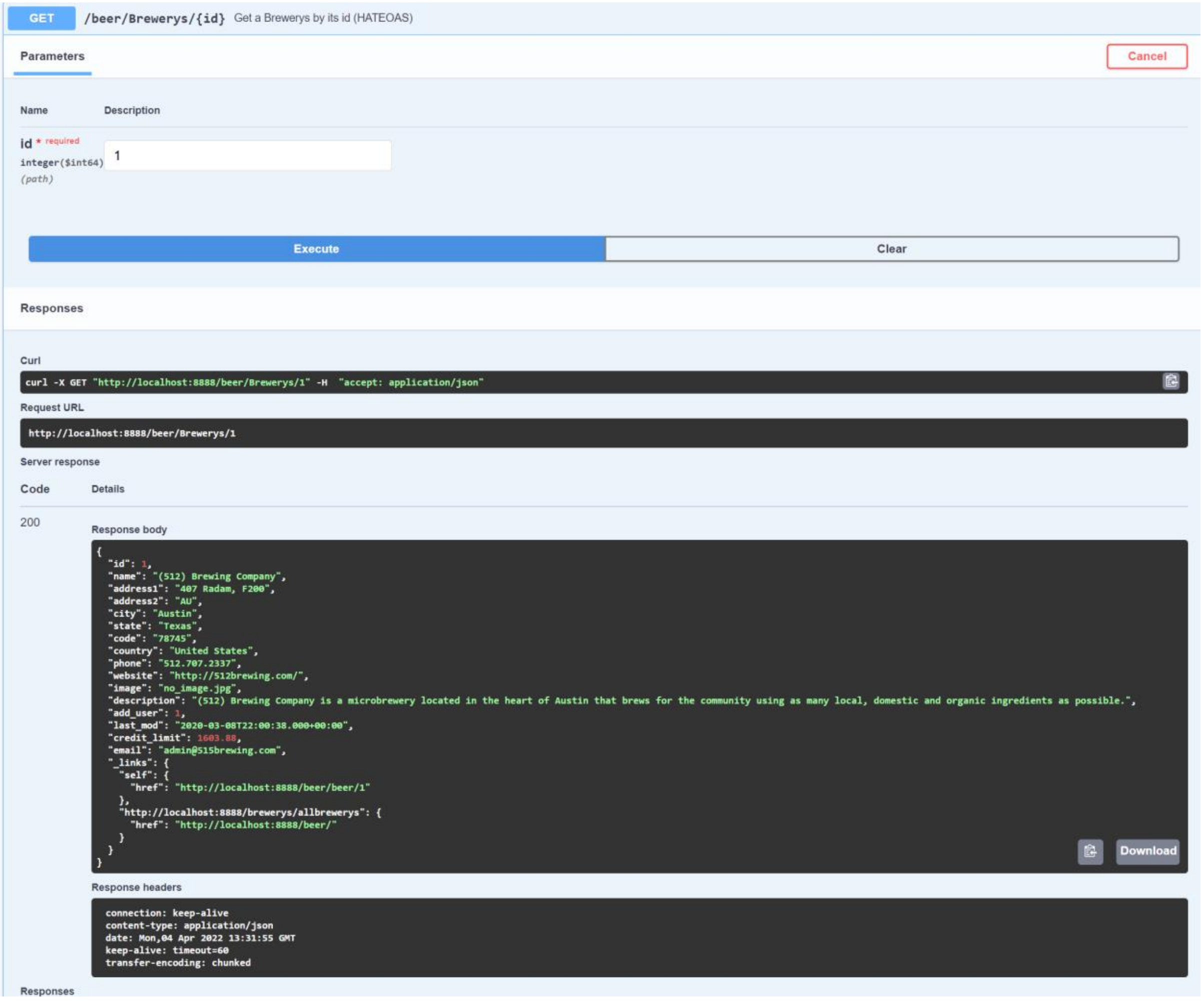
Code	Details
200	Response body

```
{
  "id": 2,
  "brewery_id": 264,
  "name": "Grimbergen Blonde",
  "cat_id": -1,
  "style_id": -1,
  "abv": 6.7,
  "ibu": 0,
  "srm": 0,
  "description": " ",
  "add_user": 0,
  "last_mod": "2019-03-06T05:19:50.000+00:00",
  "image": "2.jpg",
  "buy_price": 12.71,
  "sell_price": 19.459220000000002,
  "_links": {
    "self": {
      "href": "http://localhost:8888/beer/beer/2"
    },
    "http://localhost:8888/allbeers": {
      "href": "http://localhost:8888/beer/"
    }
  }
}
```

Response headers

```
connection: keep-alive
content-type: application/json
date: Mon,04 Apr 2022 13:30:30 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

Get a Brewerys by ID using HATEAS principles



```
curl -X GET "http://localhost:8888/beer/Brewerys/1" -H  "accept: application/json"
```

```
http://localhost:8888/beer/Brewerys/1
```

```
{
  "id": 1,
  "name": "(512) Brewing Company",
  "address1": "407 Radam, F200",
  "address2": "AU",
  "city": "Austin",
  "state": "Texas",
  "code": "78745",
  "country": "United States",
  "phone": "512.707.2337",
  "website": "http://512brewing.com/",
  "image": "no_image.jpg",
  "description": "(512) Brewing Company is a microbrewery located in the heart of Austin that brews for the community using as many local, domestic and organic ingredients as possible.",
  "add_user": 1,
  "last_mod": "2020-03-08T22:00:38.000+00:00",
  "credit_limit": 1603.88,
  "email": "admin@515brewing.com",
  "_links": {
    "self": {
      "href": "http://localhost:8888/beer/beer/1"
    },
    "http://localhost:8888/brewerys/allbrewerys": {
      "href": "http://localhost:8888/beer/"
    }
  }
}
```

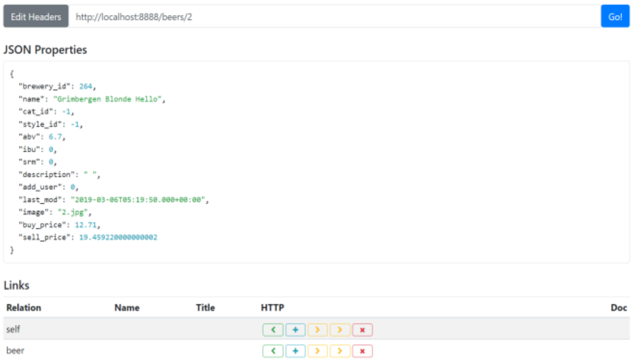
```
connection: keep-alive
content-type: application/json
date: Mon,04 Apr 2022 13:31:55 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

Examples Using HAL Explorer

PUT Request a beer by ID

Change the beer name to include hello

Hello, is just test data



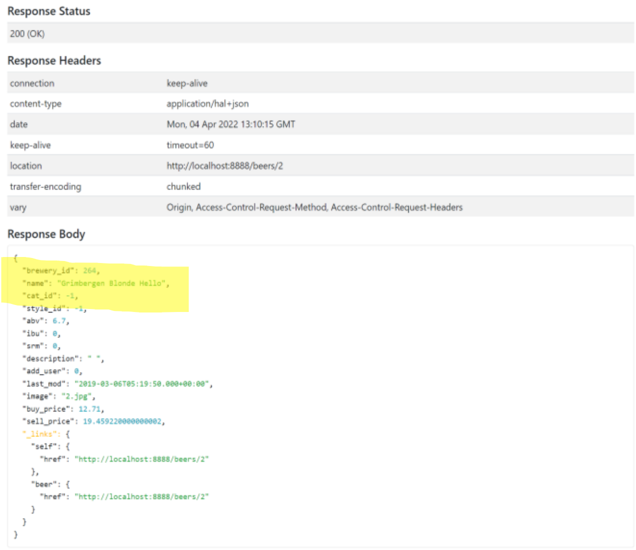
http://localhost:8888/beers/2

JSON Properties

```
{
  "brewery_id": 264,
  "name": "Grimbergen Blonde Hello",
  "cat_id": -1,
  "style_id": -1,
  "abv": 6.7,
  "ibu": 0,
  "srm": 0,
  "description": " ",
  "add_user": 0,
  "last_mod": "2019-03-06T05:19:50.000+00:00",
  "image": "2.jpg",
  "buy_price": 12.71,
  "sell_price": 19.459220000000002
}
```

Links

Relation	Name	Title	HTTP	Doc
self			< + > > ×	
beer			< + > > ×	



Response Status

200 (OK)

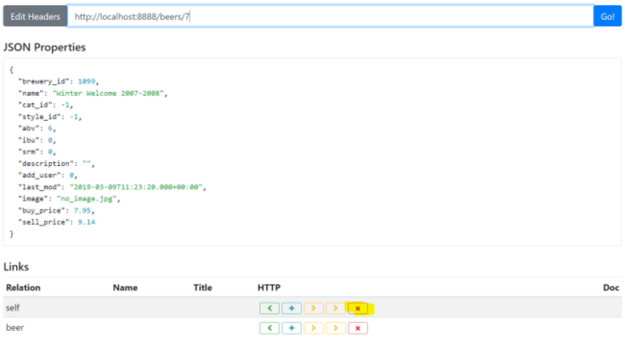
Response Headers

connection	keep-alive
content-type	application/hal+json
date	Mon, 04 Apr 2022 13:10:15 GMT
keep-alive	timeout=60
location	http://localhost:8888/beers/2
transfer-encoding	chunked
vary	Origin, Access-Control-Request-Method, Access-Control-Request-Headers

Response Body

```
{
  "brewery_id": 264,
  "name": "Grimbergen Blonde Hello",
  "cat_id": -1,
  "style_id": -1,
  "abv": 6.7,
  "ibu": 0,
  "srm": 0,
  "description": " ",
  "add_user": 0,
  "last_mod": "2019-03-06T05:19:50.000+00:00",
  "image": "2.jpg",
  "buy_price": 12.71,
  "sell_price": 19.459220000000002,
  "_links": {
    "self": {
      "href": "http://localhost:8888/beers/2"
    },
    "beer": {
      "href": "http://localhost:8888/beers/2"
    }
  }
}
```

Delete Request a beer by ID



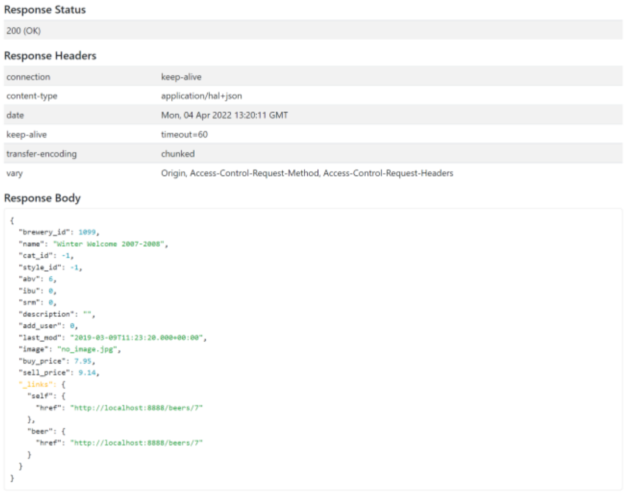
http://localhost:8888/beers/7

JSON Properties

```
{
  "brewery_id": 1099,
  "name": "Winter Welcome 2007-2008",
  "cat_id": -1,
  "style_id": -1,
  "abv": 6,
  "ibu": 0,
  "srm": 0,
  "description": "",
  "add_user": 0,
  "last_mod": "2019-03-09T11:23:20.000+00:00",
  "image": "no_image.jpg",
  "buy_price": 7.95,
  "sell_price": 9.14
}
```

Links

Relation	Name	Title	HTTP	Doc
self			< + > > ×	
beer			< + > > ×	



Response Status

200 (OK)

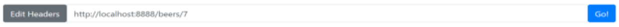
Response Headers

connection	keep-alive
content-type	application/hal+json
date	Mon, 04 Apr 2022 13:20:11 GMT
keep-alive	timeout=60
transfer-encoding	chunked
vary	Origin, Access-Control-Request-Method, Access-Control-Request-Headers

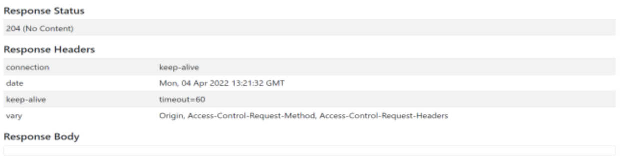
Response Body

```
{
  "brewery_id": 1099,
  "name": "Winter Welcome 2007-2008",
  "cat_id": -1,
  "style_id": -1,
  "abv": 6,
  "ibu": 0,
  "srm": 0,
  "description": "",
  "add_user": 0,
  "last_mod": "2019-03-09T11:23:20.000+00:00",
  "image": "no_image.jpg",
  "buy_price": 7.95,
  "sell_price": 9.14,
  "_links": {
    "self": {
      "href": "http://localhost:8888/beers/7"
    },
    "beer": {
      "href": "http://localhost:8888/beers/7"
    }
  }
}
```

After the Delete button is pressed the beer is deleted



http://localhost:8888/beers/7



Response Status

204 (No Content)

Response Headers

connection	keep-alive
date	Mon, 04 Apr 2022 13:21:32 GMT
keep-alive	timeout=60
vary	Origin, Access-Control-Request-Method, Access-Control-Request-Headers

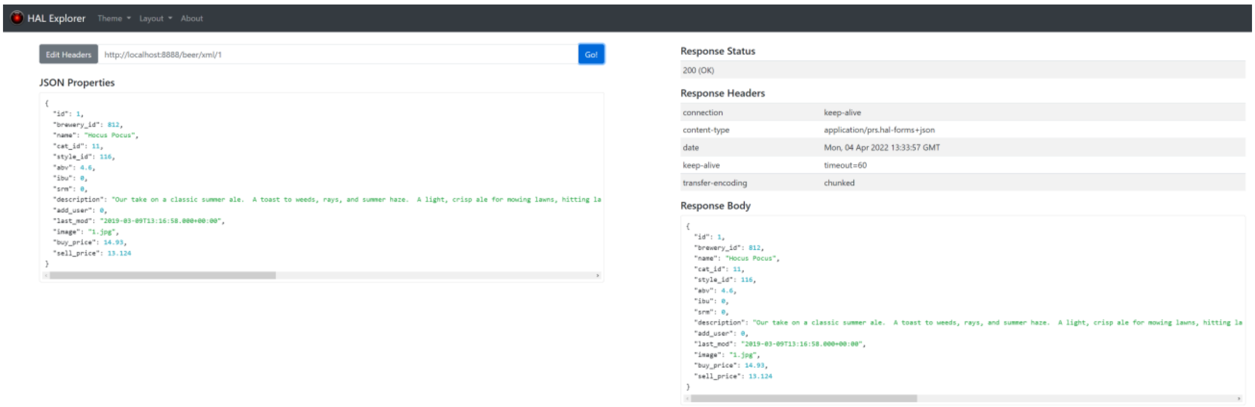
Response Body

Getting a beer by ID in XML Format for the different language question

The XML format is displayed in the response body

Link just displays the XML without using HAL Explorer

http://localhost:8888/beer/xml/1



http://localhost:8888/beer/xml/1

JSON Properties

```
{
  "id": 1,
  "brewery_id": 812,
  "name": "Hocus Pocus",
  "cat_id": 11,
  "style_id": 116,
  "abv": 4.6,
  "ibu": 0,
  "srm": 0,
  "description": "Our take on a classic summer ale.  A toast to weeds, rays, and summer haze.  A light, crisp ale for mowing lawns, hitting la
  "add_user": 0,
  "last_mod": "2019-03-09T13:16:58.000+00:00",
  "image": "1.jpg",
  "buy_price": 14.93,
  "sell_price": 13.124
}
```

Response Status

200 (OK)

Response Headers

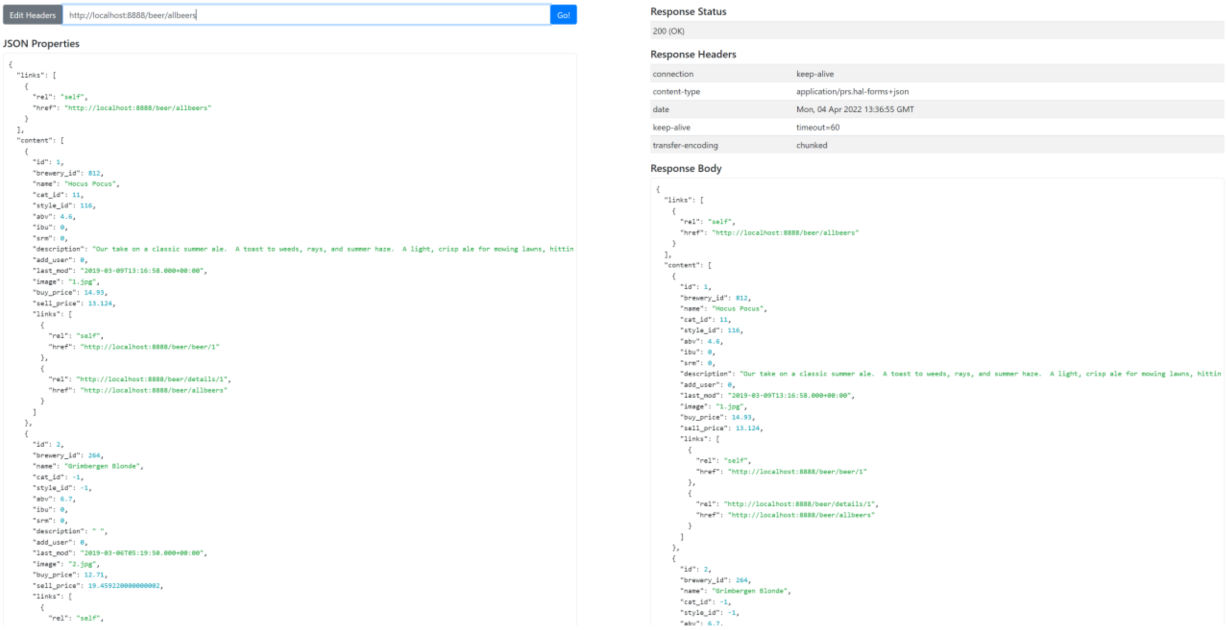
connection	keep-alive
content-type	application/prs.hal-forms+json
date	Mon, 04 Apr 2022 13:33:57 GMT
keep-alive	timeout=60
transfer-encoding	chunked

Response Body

```
{
  "id": 1,
  "brewery_id": 812,
  "name": "Hocus Pocus",
  "cat_id": 11,
  "style_id": 116,
  "abv": 4.6,
  "ibu": 0,
  "srm": 0,
  "description": "Our take on a classic summer ale.  A toast to weeds, rays, and summer haze.  A light, crisp ale for mowing lawns, hitting la
  "add_user": 0,
  "last_mod": "2019-03-09T13:16:58.000+00:00",
  "image": "1.jpg",
  "buy_price": 14.93,
  "sell_price": 13.124
}
```

Get all beers using HATEAS principles

Self-link and HREF link included in the JSON



http://localhost:8888/beer/allbeers

JSON Properties

```
{
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8888/beer/allbeers"
    }
  ],
  "content": [
    {
      "id": 1,
      "brewery_id": 812,
      "name": "Hocus Pocus",
      "cat_id": 11,
      "style_id": 116,
      "abv": 4.6,
      "ibu": 0,
      "srm": 0,
      "description": "Our take on a classic summer ale.  A toast to weeds, rays, and summer haze.  A light, crisp ale for mowing lawns, hittin
      "add_user": 0,
      "last_mod": "2019-03-09T13:16:58.000+00:00",
      "image": "1.jpg",
      "buy_price": 14.93,
      "sell_price": 13.124,
      "links": [
        {
          "rel": "self",
          "href": "http://localhost:8888/beer/beer/1"
        },
        {
          "rel": "http://localhost:8888/beer/details/1",
          "href": "http://localhost:8888/beer/allbeers"
        }
      ]
    },
    {
      "id": 2,
      "brewery_id": 264,
      "name": "Grimbergen Blonde",
      "cat_id": -1,
      "style_id": -1,
      "abv": 6.7,
      "ibu": 0,
      "srm": 0,
      "description": " ",
      "add_user": 0,
      "last_mod": "2019-03-06T05:19:50.000+00:00",
      "image": "2.jpg",
      "buy_price": 12.71,
      "sell_price": 19.459220000000002,
      "links": [
        {
          "rel": "self",
```

Response Status

200 (OK)

Response Headers

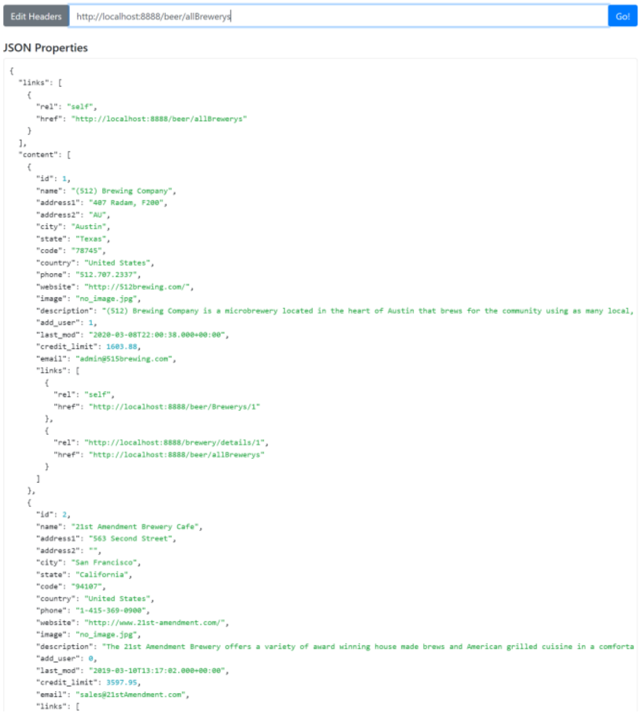
connection	keep-alive
content-type	application/prs.hal-forms+json
date	Mon, 04 Apr 2022 13:36:55 GMT
keep-alive	timeout=60
transfer-encoding	chunked

Response Body

```
{
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8888/beer/allbeers"
    }
  ],
  "content": [
    {
      "id": 1,
      "brewery_id": 812,
      "name": "Hocus Pocus",
      "cat_id": 11,
      "style_id": 116,
      "abv": 4.6,
      "ibu": 0,
      "srm": 0,
      "description": "Our take on a classic summer ale.  A toast to weeds, rays, and summer haze.  A light, crisp ale for mowing lawns, hittin
      "add_user": 0,
      "last_mod": "2019-03-09T13:16:58.000+00:00",
      "image": "1.jpg",
      "buy_price": 14.93,
      "sell_price": 13.124,
      "links": [
        {
          "rel": "self",
          "href": "http://localhost:8888/beer/beer/1"
        },
        {
          "rel": "http://localhost:8888/beer/details/1",
          "href": "http://localhost:8888/beer/allbeers"
        }
      ]
    },
    {
      "id": 2,
      "brewery_id": 264,
      "name": "Grimbergen Blonde",
      "cat_id": -1,
      "style_id": -1,
      "abv": 6.7,
```

Get all Brewerys using HATEAS principles

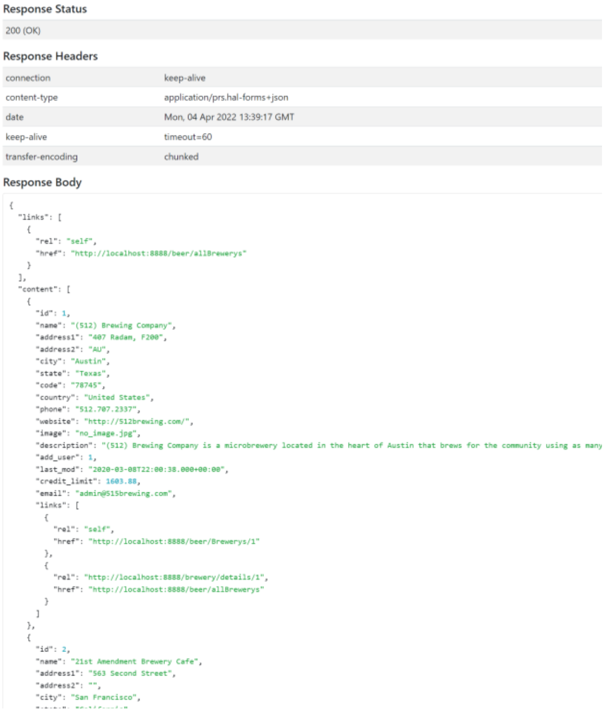
Self-link and HREF link included in the JSON



http://localhost:8888/beer/allBrewerys

JSON Properties

```
{
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8888/beer/allBrewerys"
    }
  ],
  "content": [
    {
      "id": 1,
      "name": "(512) Brewing Company",
      "address1": "407 Radam, F200",
      "address2": "AU",
      "city": "Austin",
      "state": "Texas",
      "code": "78745",
      "country": "United States",
      "phone": "512.707.2337",
      "website": "http://512brewing.com/",
      "image": "no_image.jpg",
      "description": "(512) Brewing Company is a microbrewery located in the heart of Austin that brews for the community using as many local,
      "add_user": 1,
      "last_mod": "2020-03-08T22:00:38.000+00:00",
      "credit_limit": 1603.88,
      "email": "admin@515brewing.com",
      "links": [
        {
          "rel": "self",
          "href": "http://localhost:8888/beer/Brewerys/1"
        },
        {
          "rel": "http://localhost:8888/brewery/details/1",
          "href": "http://localhost:8888/beer/allBrewerys"
        }
      ]
    },
    {
      "id": 2,
      "name": "21st Amendment Brewery Cafe",
      "address1": "563 Second Street",
      "address2": "",
      "city": "San Francisco",
      "state": "California",
      "code": "94107",
      "country": "United States",
      "phone": "1-415-369-0900",
      "website": "http://www.21st-amendment.com/",
      "image": "no_image.jpg",
      "description": "The 21st Amendment Brewery offers a variety of award winning house made brews and American grilled cuisine in a comforta
      "add_user": 0,
      "last_mod": "2019-03-10T13:17:02.000+00:00",
      "credit_limit": 3597.95,
      "email": "sales@21stAmendment.com",
      "links": [
```



Response Status

200 (OK)

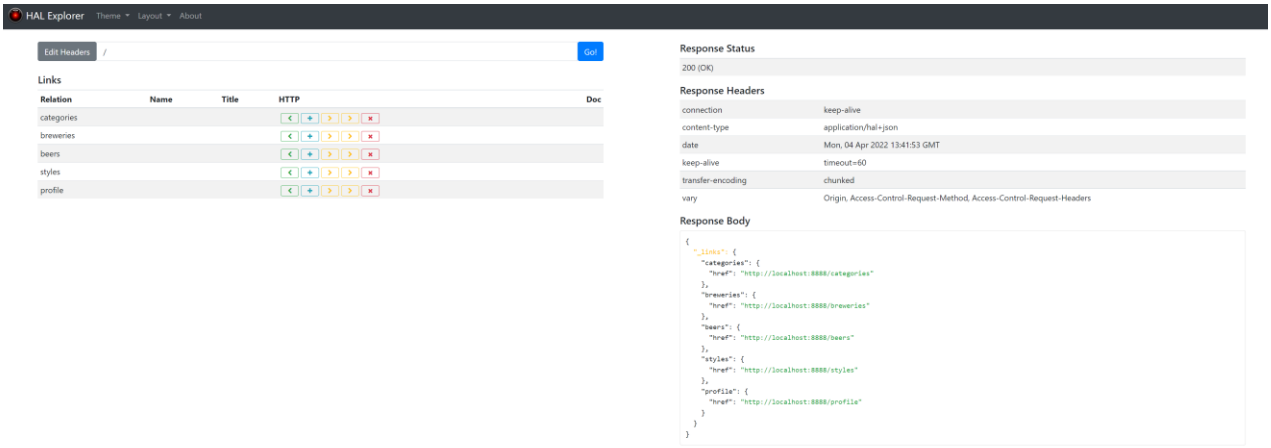
Response Headers

connection	keep-alive
content-type	application/prs.hal-forms+json
date	Mon, 04 Apr 2022 13:39:17 GMT
keep-alive	timeout=60
transfer-encoding	chunked

Response Body

```
{
  "links": [
    {
      "rel": "self",
      "href": "http://localhost:8888/beer/allBrewerys"
    }
  ],
  "content": [
    {
      "id": 1,
      "name": "(512) Brewing Company",
      "address1": "407 Radam, F200",
      "address2": "AU",
      "city": "Austin",
      "state": "Texas",
      "code": "78745",
      "country": "United States",
      "phone": "512.707.2337",
      "website": "http://512brewing.com/",
      "image": "no_image.jpg",
      "description": "(512) Brewing Company is a microbrewery located in the heart of Austin that brews for the community using as many
      "add_user": 1,
      "last_mod": "2020-03-08T22:00:38.000+00:00",
      "credit_limit": 1603.88,
      "email": "admin@515brewing.com",
      "links": [
        {
          "rel": "self",
          "href": "http://localhost:8888/beer/Brewerys/1"
        },
        {
          "rel": "http://localhost:8888/brewery/details/1",
          "href": "http://localhost:8888/beer/allBrewerys"
        }
      ]
    },
    {
      "id": 2,
      "name": "21st Amendment Brewery Cafe",
      "address1": "563 Second Street",
      "address2": "",
      "city": "San Francisco",
```

Exposing tables like Beer and Brewery



/

Links

Relation	Name	Title	HTTP	Doc
categories			< + > > ×	
breweries			< + > > ×	
beers			< + > > ×	
styles			< + > > ×	
profile			< + > > ×	

Response Status

200 (OK)

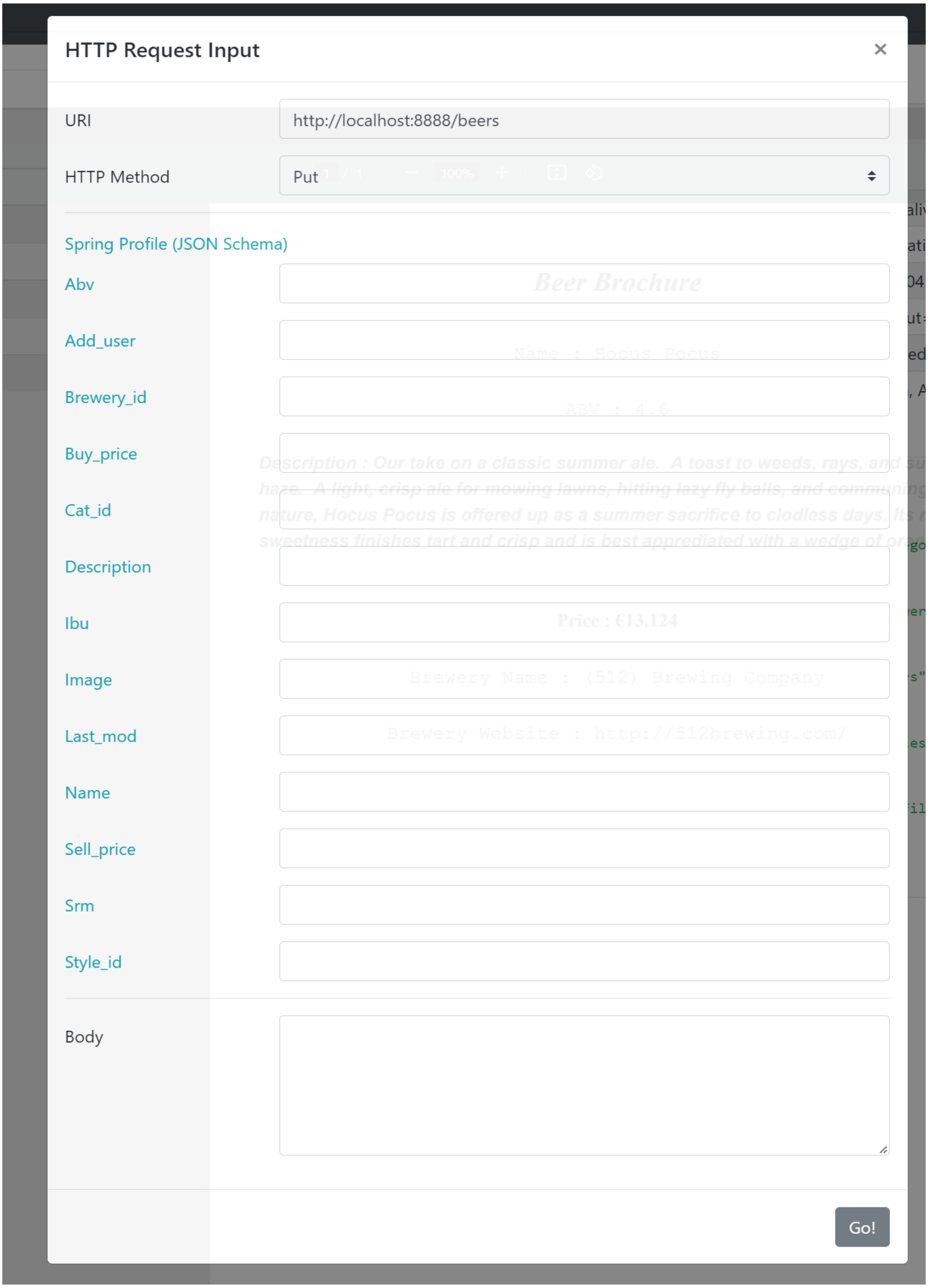
Response Headers

connection	keep-alive
content-type	application/hal+json
date	Mon, 04 Apr 2022 13:41:53 GMT
keep-alive	timeout=60
transfer-encoding	chunked
vary	Origin, Access-Control-Request-Method, Access-Control-Request-Headers

Response Body

```
{
  "_links": {
    "categories": {
      "href": "http://localhost:8888/categories"
    },
    "breweries": {
      "href": "http://localhost:8888/breweries"
    },
    "beers": {
      "href": "http://localhost:8888/beers"
    },
    "styles": {
      "href": "http://localhost:8888/styles"
    },
    "profile": {
      "href": "http://localhost:8888/profile"
    }
  }
}
```

POST Request for beers



HTTP Request Input

URI: http://localhost:8888/beers

HTTP Method: Put

Spring Profile (JSON Schema)

Abv

Add_user

Brewery_id

Buy_price

Cat_id

Description

Ibu

Image

Last_mod

Name

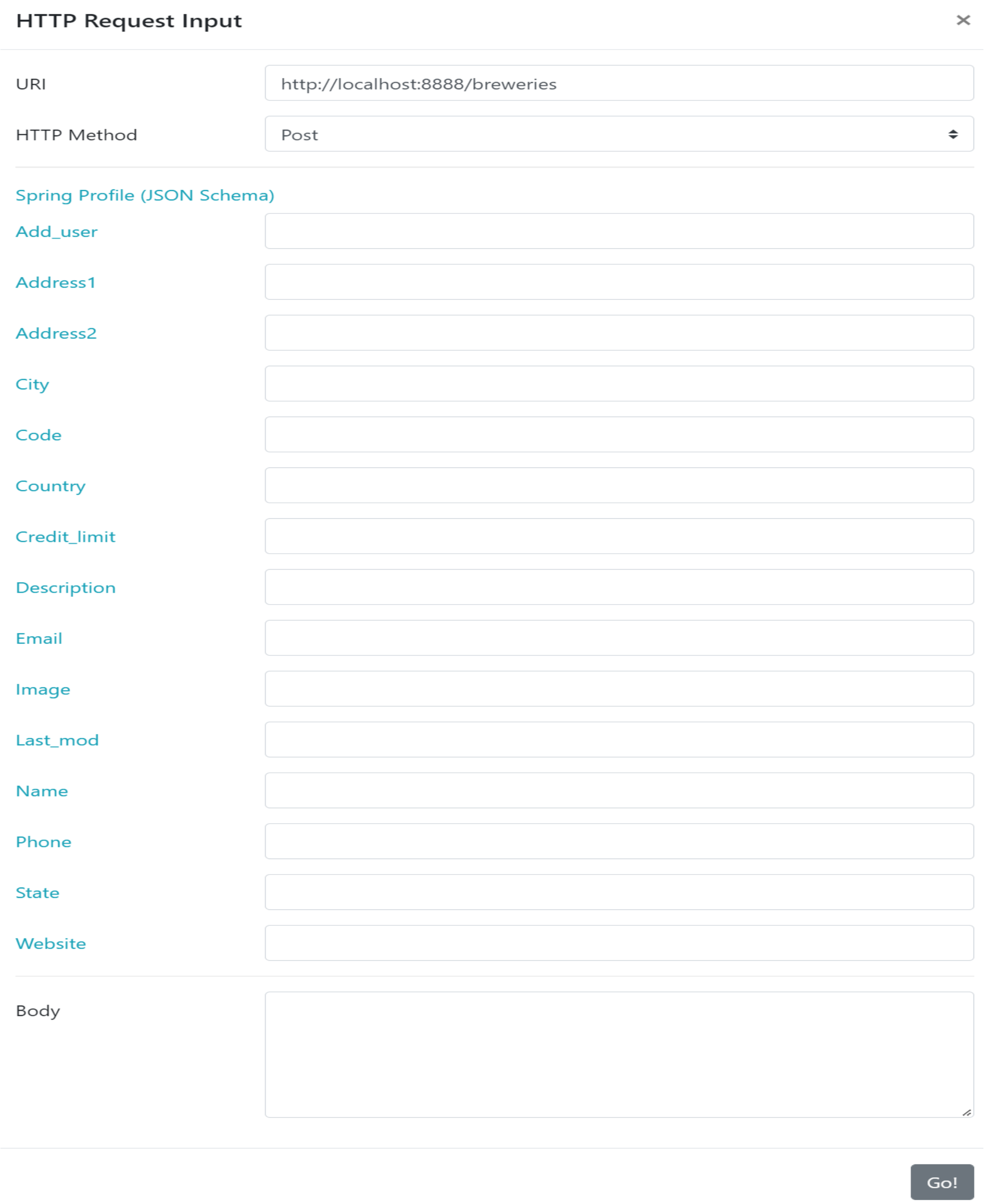
Sell_price

Srm

Style_id

Body

POST Request for Brewery



HTTP Request Input ×

URI	http://localhost:8888/breweries
HTTP Method	Post

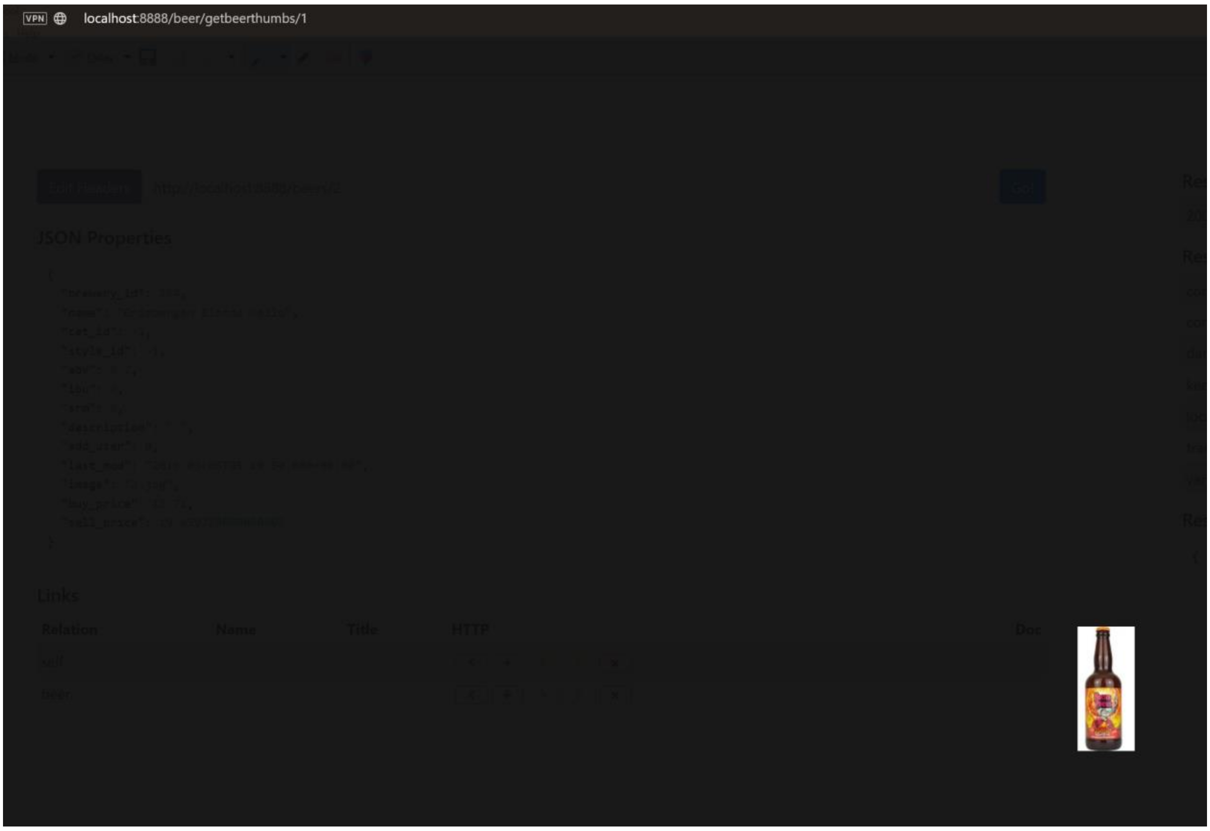
Spring Profile (JSON Schema)

Add_user	
Address1	
Address2	
City	
Code	
Country	
Credit_limit	
Description	
Email	
Image	
Last_mod	
Name	
Phone	
State	
Website	
Body	

Other Functions with no UI Needed

Get a beer thumbnail by ID

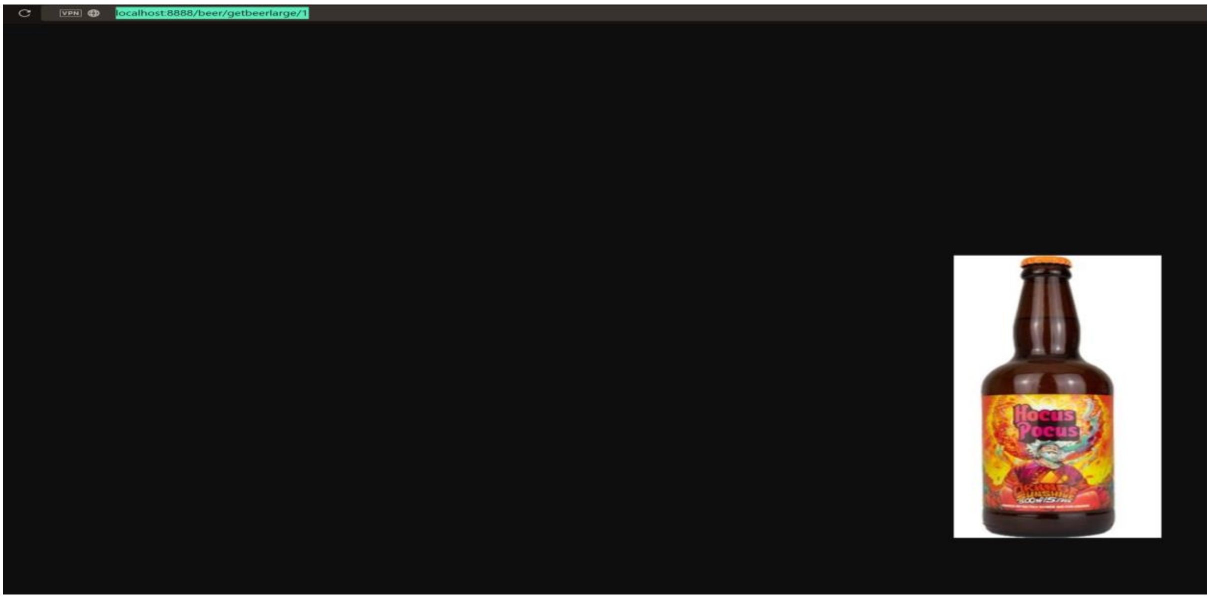
http://localhost:8888/beer/getbeerthumbs/1



localhost:8888/beer/getbeerthumbs/1

Get a beer larger image by ID

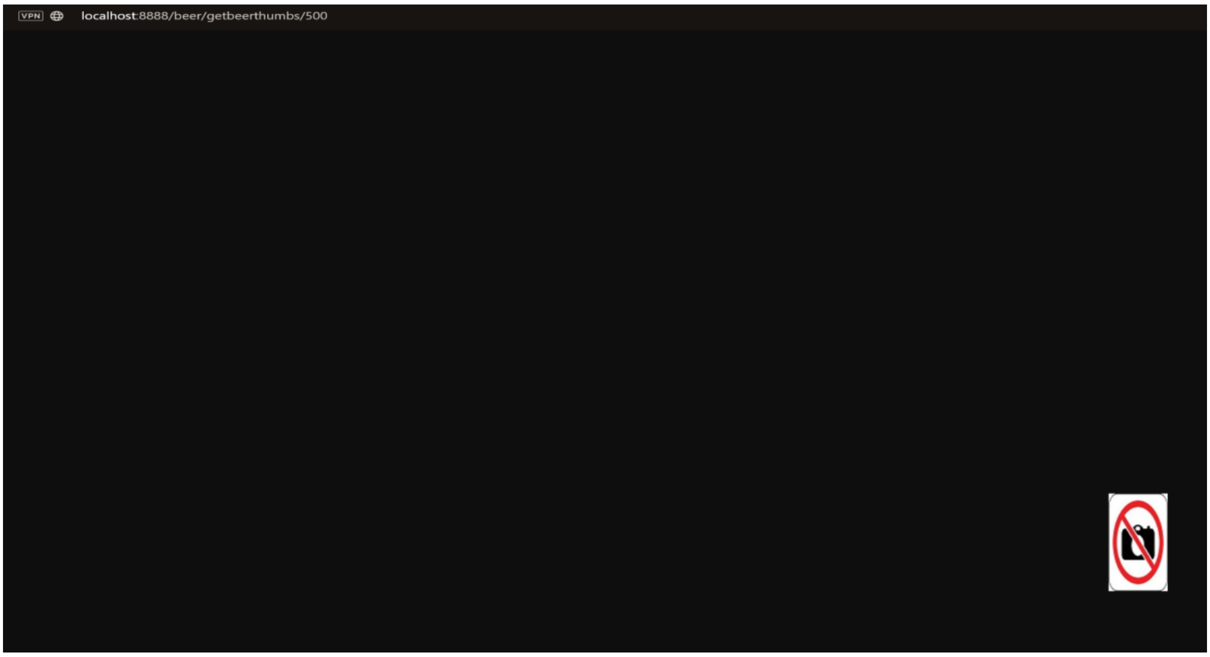
http://localhost:8888/beer/getbeerlarge/1



localhost:8888/beer/getbeerlarge/1

Get a beer thumbnail by ID when the beer has no thumbnail image

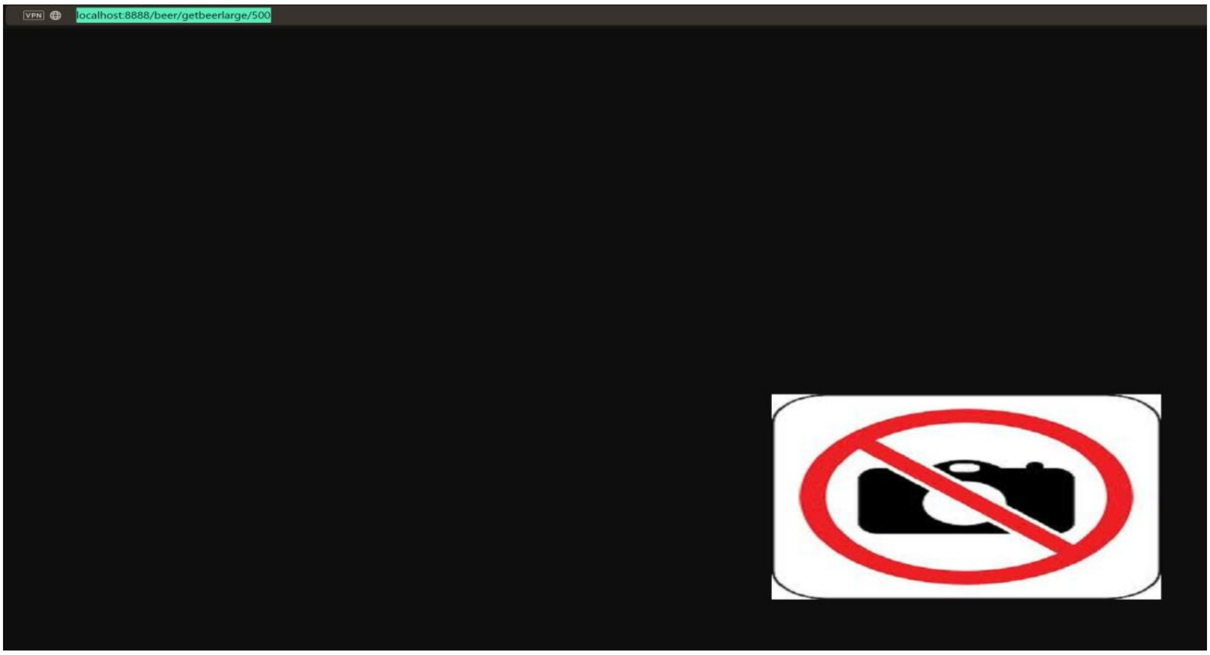
http://localhost:8888/beer/getbeerthumbs/500



localhost:8888/beer/getbeerthumbs/500

Get a beer large image by ID when the beer has no large image

http://localhost:8888/beer/getbeerlarge/500

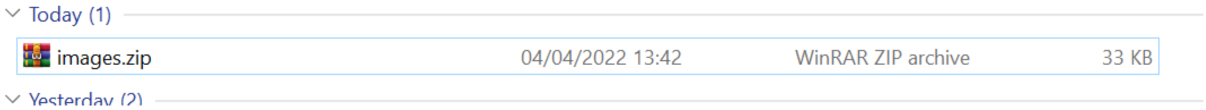


localhost:8888/beer/getbeerlarge/500

Downloads all beer images

http://localhost:8888/beer/zip

Output



Today (1)

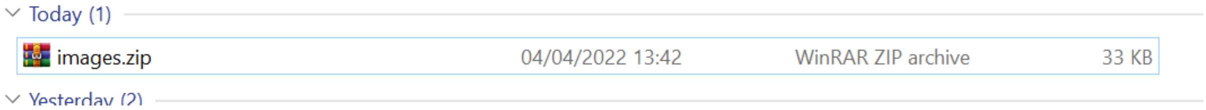
images.zip 04/04/2022 13:42 WinRAR ZIP archive 33 KB

Yesterday (2)

Can also be done for using a beer ID to download a specific image

http://localhost:8888/beer/zip/1

Output



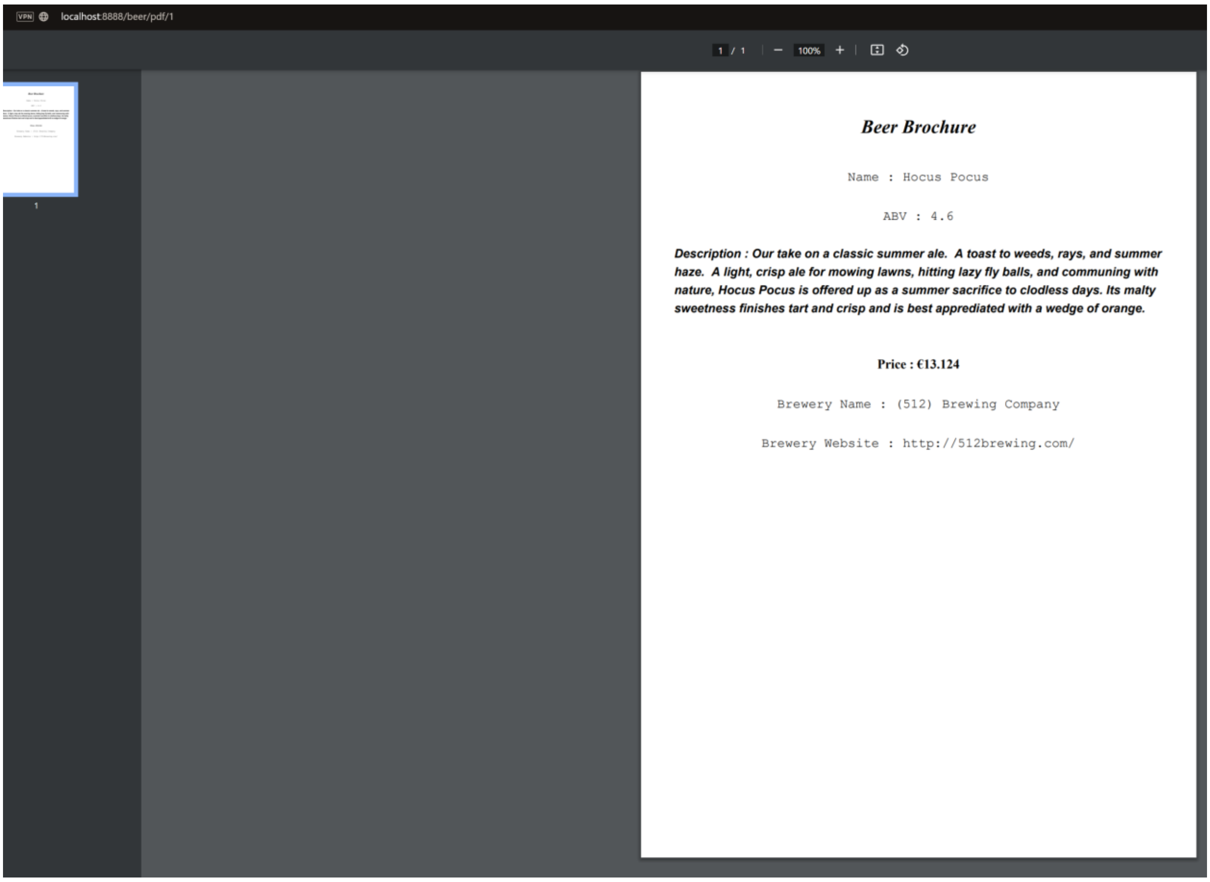
Today (1)

images.zip 04/04/2022 13:42 WinRAR ZIP archive 33 KB

Yesterday (2)

PDF Generator

http://localhost:8888/beer/pdf/1



Beer Brochure

Name : Hocus Pocus

ABV : 4.6

Description : Our take on a classic summer ale. A toast to weeds, rays, and summer haze. A light, crisp ale for mowing lawns, hitting lazy fly balls, and communing with nature, Hocus Pocus is offered up as a summer sacrifice to clodless days. Its malty sweetness finishes tart and crisp and is best apprediated with a wedge of orange.

Price : €13.124

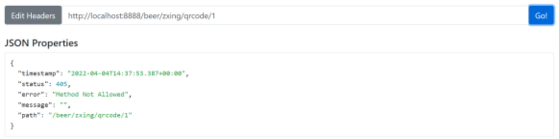
Brewery Name : (512) Brewing Company

Brewery Website : http://512brewing.com/

PDF Generator

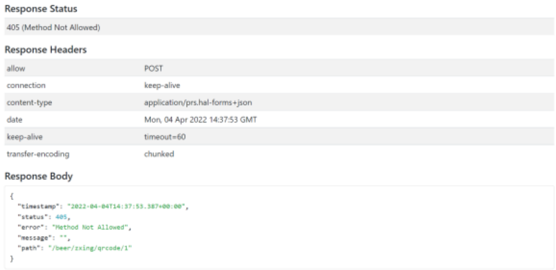
http://localhost:8888/explorer/index.html#uri=http://localhost:8888/beer/zxing/qrcode/1

An error occurs where the method is not allowed



JSON Properties

```
{
  "timestamp": "2022-04-04T14:37:53.387+00:00",
  "status": 405,
  "error": "Method Not Allowed",
  "message": "",
  "path": "/beer/zxing/qrcode/1"
}
```



Response Status

405 (Method Not Allowed)

Response Headers

allow	POST
connection	keep-alive
content-type	application/prs.hal-forms+json
date	Mon, 04 Apr 2022 14:37:53 GMT
keep-alive	timeout=60
transfer-encoding	chunked

Response Body

```
{
  "timestamp": "2022-04-04T14:37:53.387+00:00",
  "status": 405,
  "error": "Method Not Allowed",
  "message": "",
  "path": "/beer/zxing/qrcode/1"
}
```

Self-evaluation

While creating the REST API for this assignment I uncovered some strengths and weaknesses along the way. Below are a few of the main strengths and weaknesses I have discovered. The strength of my API would be the exception handling. I have implemented what I believe to be a well-structured exception handler that clearly states what the exception is. I found that having a well-structured exception handler lets me debug the programme faster as when my code fails I got errors explaining where the problem was.

The API I created has a lot of functionality and is not just a basic API with delete, update and add functionality. I can generate PDF files, download zip files and display images of certain beers in the browser. I also created functions that use the HATEAS principles. The API can automatically create a PDF file with related information on a given beer. This is a feature that I think would be of particular use in the real world. Customers in the real world will be able to view information about the beer. The API also lets the user download images of the beers which allows customers to see images of the beers they want.

The weakness of my REST API is the inability to perform more than one operation at a time. For Example, for delete operations, I must perform each delete on a specific beer so if I had to delete 100 beers this could take a very long time. In the future when creating a REST API, I would think about making CRUD operations less tedious.

My REST API has an issue with generating a QR Code which I have not managed to figure out. I also did not implement the Google Maps location for each brewery. The PDF file does not show all the data that was asked for the assignment. I have not gotten these parts of the assignment finished as I struggled with the functionality of these questions. I also ran out of time to finish these questions. I need to do more research to get these functions to work.

My API currently does not have much security implemented. There is no proper protection of the data and from a security standpoint, I would have to say my API is vulnerable. I have not implemented any security system in the API this could be something implemented in the future.

Overall, I believe I have learned about the general strengths and weaknesses when it comes to the creation of API. I think this assignment has helped me understand the potential uses of REST API and how I can use REST API in the future.

Benchmarking and Enhancements.

A RESTful API is an architectural style for an application program interface (API) that uses HTTP requests to access and use data. That data can be used to GET, PUT, POST and DELETE data types, which refers to the reading, updating, creating and deleting of operations concerning resources.

Using an API or Rest API brings a lot of benefits to a software project.

Why I am Using Rest

Scalability. This protocol stands out due to its scalability. Thanks to the separation between client and server, a product may be scaled by a development team without much difficulty.

Independence. With the separation between client and server, the protocol makes it easy for developments across a project to take place independently. In addition, the **REST API** always adapts to the working syntax and platform. This offers the opportunity to use multiple environments while developing.

Flexibility and portability. With the indispensable requirement for data from one of the requests to be properly sent, it is possible to perform a migration from one server to another or carry out changes on the database at any time. Front and back can therefore be hosted on different servers, which is a significant management advantage.

Advantages of Rest API Development over SOAP

REST overcomes many of the disadvantages of **SOAP**, such as the need for clients to know the operation semantics as a pre-requisite for its use, or the use of different ports for different types of notifications. In addition, **REST** can handle many resources, while **SOAP** needs many operations to accomplish that.

These are some of the **advantages of REST**:

- It is usually simple to build and adapt.
- Low use of resources.
- Process instances are created explicitly.
- With the initial URI, the client does not require routing information.
- Clients can have a generic ‘listener’ interface for notifications