

# Brewery Beer Service Docs

Alassane NIANG

# Table of Contents

Get a Beer by its ID .....	2
Using cURL .....	2
Using HTTPIE .....	2
Using HTTP .....	2
Path Parameters .....	2
Response Fields .....	2
HTTP Response Example .....	3
Response Body Example .....	3
Create new beer .....	4
Using cURL .....	4
Using HTTPIE .....	4
Using HTTP .....	4
Request Fields .....	5
Request Body .....	5
HTTP Response Example .....	6
Update an existing Beer .....	7
Using cURL .....	7
Using HTTPIE .....	7
Using HTTP .....	7
Request Fields .....	8
Path Parameters .....	8
Request Body .....	8
HTTP Response Example .....	9
Delete an existing Beer .....	10

**Brewery Beer Manager Service** is a Microservice for managing beer inventory.

The service allows the following operations:

- Create a new beer
- Update an existing beer
- Delete an existing beer
- Get a beer by its id
- Get all beers

# Get a Beer by its ID

## Using cURL

```
$ curl 'http://localhost:8080/api/v1/beer/0a124177-28b5-4741-b9f9-29b728992d16' -i -X GET \
-H 'Accept: application/json'
```

## Using HTTPIE

```
$ http GET 'http://localhost:8080/api/v1/beer/0a124177-28b5-4741-b9f9-29b728992d16' \
'Accept:application/json'
```

## Using HTTP

```
GET /api/v1/beer/0a124177-28b5-4741-b9f9-29b728992d16 HTTP/1.1
Accept: application/json
Host: localhost:8080
```

## Path Parameters

Table 1. `/api/v1/beer/{beerId}`

Parameter	Description
<code>beerId</code>	UUID of desired beer to get

## Response Fields

Path	Type	Description
<code>id</code>	UUID	Id of the beer
<code>version</code>	Long	Version number
<code>createdDate</code>	Timestamp	Date created
<code>lastModifiedDate</code>	Timestamp	Date updated
<code>beerName</code>	String	Beer name
<code>beerStyle</code>	String	Beer style
<code>upc</code>	Long	UPC of beer
<code>price</code>	BigDecimal	Beer price
<code>quantityOnHand</code>	Integer	Quantity on hand

## HTTP Response Example

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 164
```

```
{"id":null,"version":null,"createdDate":null,"lastModifiedDate":null,"beerName":"My Beer","beerStyle":"ALE","upc":123123123123,"price":"2.99","quantityOnHand":null}
```

## Response Body Example

```
{"id":null,"version":null,"createdDate":null,"lastModifiedDate":null,"beerName":"My Beer","beerStyle":"ALE","upc":123123123123,"price":"2.99","quantityOnHand":null}
```

# Create new beer

## Using cURL

```
$ curl 'http://localhost:8080/api/v1/beer/' -i -X POST \  
  -H 'Content-Type: application/json' \  
  -d '{  
    "id" : null,  
    "version" : null,  
    "createdDate" : null,  
    "lastModifiedDate" : null,  
    "beerName" : "My Beer",  
    "beerStyle" : "ALE",  
    "upc" : 123123123123,  
    "price" : "2.99",  
    "quantityOnHand" : null  
  }'
```

## Using HTTPIE

```
$ echo '{  
  "id" : null,  
  "version" : null,  
  "createdDate" : null,  
  "lastModifiedDate" : null,  
  "beerName" : "My Beer",  
  "beerStyle" : "ALE",  
  "upc" : 123123123123,  
  "price" : "2.99",  
  "quantityOnHand" : null  
}' | http POST 'http://localhost:8080/api/v1/beer/' \  
  'Content-Type:application/json'
```

## Using HTTP

```
POST /api/v1/beer/ HTTP/1.1
Content-Type: application/json
Content-Length: 210
Host: localhost:8080
```

```
{
  "id" : null,
  "version" : null,
  "createdDate" : null,
  "lastModifiedDate" : null,
  "beerName" : "My Beer",
  "beerStyle" : "ALE",
  "upc" : 123123123123,
  "price" : "2.99",
  "quantityOnHand" : null
}
```

## Request Fields

Path	Type	Description	Constraints
beerName	String	Beer name	Must not be blank. Size must be between 3 and 100 inclusive
beerStyle	String	Beer style	Must not be null
upc	Number	UPC of beer	Must be positive. Must not be null
price	String	Beer price	Must be positive. Must not be null

## Request Body

```
{
  "id" : null,
  "version" : null,
  "createdDate" : null,
  "lastModifiedDate" : null,
  "beerName" : "My Beer",
  "beerStyle" : "ALE",
  "upc" : 123123123123,
  "price" : "2.99",
  "quantityOnHand" : null
}
```

# HTTP Response Example

HTTP/1.1 201 Created



# Update an existing Beer

## Using cURL

```
$ curl 'http://localhost:8080/api/v1/beer/5d1b50b0-df8f-4924-a9a7-ccbe1583c295' -i -X PUT \
  -H 'Content-Type: application/json' \
  -d '{
    "id" : null,
    "version" : null,
    "createdDate" : null,
    "lastModifiedDate" : null,
    "beerName" : "My Beer",
    "beerStyle" : "ALE",
    "upc" : 123123123123,
    "price" : "2.99",
    "quantityOnHand" : null
  }'
```

## Using HTTPIE

```
$ echo '{
  "id" : null,
  "version" : null,
  "createdDate" : null,
  "lastModifiedDate" : null,
  "beerName" : "My Beer",
  "beerStyle" : "ALE",
  "upc" : 123123123123,
  "price" : "2.99",
  "quantityOnHand" : null
}' | http PUT 'http://localhost:8080/api/v1/beer/5d1b50b0-df8f-4924-a9a7-ccbe1583c295' \
  'Content-Type:application/json'
```

## Using HTTP

```
PUT /api/v1/beer/5d1b50b0-df8f-4924-a9a7-ccbe1583c295 HTTP/1.1
Content-Type: application/json
Content-Length: 210
Host: localhost:8080
```

```
{
  "id" : null,
  "version" : null,
  "createdDate" : null,
  "lastModifiedDate" : null,
  "beerName" : "My Beer",
  "beerStyle" : "ALE",
  "upc" : 123123123123,
  "price" : "2.99",
  "quantityOnHand" : null
}
```

## Request Fields

Path	Type	Description	Constraints
beerName	String	Beer name	Must not be blank. Size must be between 3 and 100 inclusive
beerStyle	String	Beer style	Must not be null
upc	Number	UPC of beer	Must be positive. Must not be null
price	String	Beer price	Must be positive. Must not be null

## Path Parameters

Table 2. `/api/v1/beer/{beerId}`

Parameter	Description
<code>beerId</code>	Id of the beer

## Request Body

```
{
  "id" : null,
  "version" : null,
  "createdDate" : null,
  "lastModifiedDate" : null,
  "beerName" : "My Beer",
  "beerStyle" : "ALE",
  "upc" : 123123123123,
  "price" : "2.99",
  "quantityOnHand" : null
}
```

## HTTP Response Example

HTTP/1.1 204 No Content

# Delete an existing Beer

*to be implemented*