**Brewery Discovery Application**

**Overview**

The Brewery Discovery Application is a tool that allows users to search for breweries across the world and add their favorites to their personal user account.

**API**: [Open Brewery DB](https://www.openbrewerydb.org/)

**Features**

1. **Get Random Brewery**

* Gets a completely random brewery for the user, no specific type or location. A fun way for users to explore, perhaps they’ll find one that sounds interesting!

1. **View List of Breweries**

* Allows viewers to see a list of multiple breweries. Queryable by country, city, type, distance to location, and more, doing so allows users to refine their search and customize it to their liking.

1. **Get Singular Brewery**

* Gets a singular brewery based on its unique ID. Doing so ‘selects’ this brewery and allows it to be added to the user’s profile for them to easily keep track of it, if it is the most recent singular search.

1. **Add Brewery to Favorites**

* Adds the most recent singular brewery to the user’s favorites, at a specified position. This allows users to have easy access to breweries they are interested in.

1. **Remove Brewery as Favorite**

* Removes/clears the brewery at a specific favorite position in the user’s account. This allows users to remove a saved brewery if they, for example, no longer want to keep track of it.

1. **View All Favorites**

* Allow users to see the 5 breweries they have marked as their favorites on their account. This allows them to easily get information regarding breweries they previously deemed at notable.

1. **View Memory**

* Allows users to see the results of their 10 most recent successful searches for breweries (lists of breweries that were retrieved at the same time only count as 1 search). This allows them to keep track of their previous searches, and allows easier recalling/redisplaying of previous results.

**Route: /create-account**

* **Request type: POST**
* **Purpose:** Creates a new account with a username and password.
* **Request Body:**
  + **username (string):** User’s chosen username.
  + **password (string):** User’s chose password.
* **Response Format: JSON**
  + **Success Response Example**
    - Code: 200
    - Content: {“message”: “Login successful.”}
* **Example Request:**

**{**

**“username”: “testUser”,**

**“password”: “testPassword”**

**}**

* **Example Response:**

**{**

**“message”: “Account created successfully!”**

**},**

**200**

**Route: /login**

* **Request type: POST**
* **Purpose:** Verifies that the password matches the user’s password that is stored in the database.
* **Request Body:**
  + **username (string):** Username in database.
  + **password (string):** Password to match database value against.
* **Response Format: JSON**
  + **Success Response Example**
    - Code: 200
    - Content: {“message”: “Login successful.”}
* **Example Request:**

**{**

**“username”: “testUser”,**

**“password”: “testPassword”**

**}**

* **Example Response:**

**{**

**“message”: “Login successful.”**

**},**

**200**

**Route: /update-password**

* **Request type: POST**
* **Purpose:** Changes the user’s password, given they know the old password.
* **Request Body:**
  + **username (string):** Username in database.
  + **oldPassword (string):** Password to match database value against.
  + **newPassword (string):** New password that will replace the old.
* **Response Format: JSON**
  + **Success Response Example**
    - Code: 200
    - Content: {“message”: “password updated successfully.”}
* **Example Request:**

**{**

**“username”: “testUser”,**

**“oldPassword”: “testPassword”,**

**“newPassword”: “newPassword”**

**}**

* **Example Response:**

**{**

**“message”: “password updated successfully.”**

**},**

**200**

**Route: /delete-user**

* **Request type: DELETE**
* **Purpose:** Deletes a user and all associated fields from the database.
* **Request Body:**
  + **username (string):** Username in database.
  + **password (string):** Password to match database value against.
* **Response Format: JSON**
  + **Success Response Example**
    - Code: 200
    - Content: {“message”: “user successfully deleted.”}
* **Example Request:**

**{**

**“username”: “testUser”,**

**“password”: “testPassword”**

**}**

* **Example Response:**

**{**

**“message”: “user successfully deleted”**

**},**

**200**

**Route: /api-check**

* **Request type: GET**
* **Purpose:** Checks to see if the API is running.
* **Response Format: JSON**
  + **Success Response**
    - Code: 200
    - Content: {“message”: “api is running!”}
* **Response:**

**{**

**“message”: “api is running!”**

**},**

**200**

**Route: /db-check**

* **Request type: GET**
* **Purpose:** Checks to see if the database is connected.
* **Response Format: JSON**
  + **Success Response**
    - Code: 200
    - Content: {“message”: “db is connected!”}
* **Example Response:**

**{**

**“message”: “db is connected!”**

**},**

**200**

**Route: /get-brewery/{…}**

* **Request type: GET**
* **Purpose:** Retrieves the details of a specific brewery based on its id.
* **Path parameter:**
  + **id (string):** The id of the brewery.
* **Example call:**
  + /get-brewery/58144e11-89a4-41bd-b65f-5b33de2b654a
* **Response Format: JSON**
  + **Success Response Example**
    - Code: 200
    - Content: {“id”: …, “name”: …, “brewery\_type”: …, “address\_1”: …, “address\_2”: …, “address\_3”: …, “city”: …, “state\_province”: …, “postal\_code”: …, “country”: …, “longitude”: …, “latitude” …, “phone”: …, “website\_url”: …, “state”: …, “street”: …}
* **Example Response:**

**{"id":"58144e11-89a4-41bd-b65f-5b33de2b654a","name":"Cheeky Monkey Brewing Co / Lucky Strike Jillians","brewery\_type":"brewpub","address\_1":"3 Lansdowne St","address\_2":null,"address\_3":null,"city":"Boston","state\_province":"Massachusetts","postal\_code":"02215-3412","country":"United States","longitude":"-71.0946215","latitude":"42.34729","phone":"6178590030","website\_url":"http://www.cheekymonkeyboston.com","state":"Massachusetts","street":"3 Lansdowne St"},**

**200**

**Route: /list-breweries?**

* **Request type: GET**
* **Purpose:** Displays the details of multiple breweries in the form of a list.
* **Optional Queries:**
  + by\_city
  + by\_country
  + by\_dist
  + by\_ids
  + by\_name
  + by\_state
  + by\_postal
  + by\_type
  + page
  + per\_page
  + sort
* **Example call:**
  + /list-breweries?by\_country=south%20korea&per\_page=3
* **Response Format: JSON**
  + **Success Response Example:**
    - Code: 200
    - Content: [{*brewery\_1\_details*}, {*brewery\_2\_details*}, …]
* **Example Response:**

**[{"id":"90d1fa86-77d7-4a21-9818-e338fa6fdb96","name":"583 양조장(583 Brewery)","brewery\_type":"brewpub","address\_1":"583-8, Uiam-ro","address\_2":"Janggye-myeon","address\_3":null,"city":"Jangsu-gun","state\_province":"Jeollabukdo","postal\_code":"55621","country":"South Korea","longitude":"127.6205123","latitude":"35.68271721","phone":"063-353-5515","website\_url":"http://www.583beer.modoo.at","state":"Jeollabukdo","street":"583-8, Uiam-ro"},  
{"id":"200b3434-8f97-4a9d-a072-fcb7379b31d5","name":"CRAFTROOT","brewery\_type":"brewpub","address\_1":"418,Gwangwang-ro","address\_2":null,"address\_3":null,"city":"Sokcho-si","state\_province":"Gangwondo","postal\_code":"24859","country":"South Korea","longitude":"128.5377122","latitude":"38.19724237","phone":"070-8872-1001","website\_url":"https://craftroot.co.kr/","state":"Gangwondo","street":"418,Gwangwang-ro"},  
{"id":"4cc60614-30c4-406a-9fec-c5dc01b22863","name":"Goose Island Brewhouse Seoul","brewery\_type":"brewpub","address\_1":"118, Yeoksam-ro","address\_2":null,"address\_3":null,"city":"Gangnam-gu","state\_province":"Seoul","postal\_code":"06251","country":"South Korea","longitude":"127.0323206","latitude":"37.49360307","phone":"02-6205-1785","website\_url":"https://gooseisland.kr","state":"Seoul","street":"118, Yeoksam-ro"}],  
200**

**Route: /get-random**

* **Request type: GET**
* **Purpose:** Retrieves the details of a completely random brewery.
* **Response Format: JSON**
  + **Success Response Example**
    - Code: 200
    - Content: {“id”: …, “name”: …, “brewery\_type”: …, “address\_1”: …, “address\_2”: …, “address\_3”: …, “city”: …, “state\_province”: …, “postal\_code”: …, “country”: …, “longitude”: …, “latitude” …, “phone”: …, “website\_url”: …, “state”: …, “street”: …}
* **Example Response:**

**{"id":"58144e11-89a4-41bd-b65f-5b33de2b654a","name":"Cheeky Monkey Brewing Co / Lucky Strike Jillians","brewery\_type":"brewpub","address\_1":"3 Lansdowne St","address\_2":null,"address\_3":null,"city":"Boston","state\_province":"Massachusetts","postal\_code":"02215-3412","country":"United States","longitude":"-71.0946215","latitude":"42.34729","phone":"6178590030","website\_url":"http://www.cheekymonkeyboston.com","state":"Massachusetts","street":"3 Lansdowne St"},**

**200**

**Route: /view-favorites**

* **Request type: GET**
* **Purpose:** Retrieves all the values in each of a user’s favorite\_brewery\_*i* columns
* **Request Body:**
  + **username (string):** Username in database whose favorites will be retrieved.
* **Response Format: JSON**
  + **Success Response Example:**
    - Code: 200
    - Content: {*favorite\_brew\_1*, *favorite\_brew\_2*, *favorite\_brew\_3, favorite\_brew\_4, favorite\_brew\_5}*
* **Example Request:**

**{**

**“username”: “testUser”**

**}**

* **Example Response:**

**{**

**“username”: {** **[{"id":"29444b11-30a1-4b43-9b9f-541ea67afc94","name":"Butler Brew Works","brewery\_type":"brewpub","address\_1":"101 S Main St","address\_2":null,"address\_3":null,"city":"Butler","state\_province":"Pennsylvania","postal\_code":"16001-5907","country":"United States","longitude":"-79.895238","latitude":"40.860914","phone":"7242645347","website\_url":"http://www.butlerbrewworks.com","state":"Pennsylvania","street":"101 S Main St"}],  
NULL,**

**NULL,**

**[{"id":"95218728-13e3-4eef-9fa5-1deec394c2f6","name":"Pabst Milwaukee Brewery","brewery\_type":"large","address\_1":"1037 W Juneau Ave","address\_2":null,"address\_3":null,"city":"Milwaukee","state\_province":"Wisconsin","postal\_code":"53233-1405","country":"United States","longitude":"-87.9255251","latitude":"43.04554365","phone":"4149080025","website\_url":null,"state":"Wisconsin","street":"1037 W Juneau Ave"}],**

**NULL}**

**},**

**200**

**Route: /add-favorite/{…}**

* **Request type: PUT**
* **Purpose:** Updates the user’s specified favorite position with the most recent singular search in memory.
* **Path parameter:**
  + **position (int):** The 1-5 position of which *favorite\_brew\_i* value will be updated.
* **Example call:**
  + /add-favorite/5
* **Request Body:**
  + **username (string):** Username in database whose favorites will be modified.
* **Response Format: JSON**
  + **Success Response Example:**
    - Code: 200
    - Content: {“message”: “successfully updated favorite brewery 5”}
* **Example Request:**

**{**

**“username”: “testUser”,**

**}**

* **Example Response:**

**{**

**“message”: “successfully updated favorite brewery 5”**

**},**

**200**

**Route: /clear-favorite/**

* **Request type: PUT**
* **Purpose:** Sets the user’s specified favorite position with null, effectively deleing the previous value.
* **Path parameter:**
  + **position (int):** The 1-5 position of which *favorite\_brew\_i* value will be cleared.
* **Example call:**
  + /clear-favorite/2
* **Request Body:**
  + **Success Response Example:**
    - Code: 200
    - Content: {“message”: “successfully cleared favorite brewery 2”}
* **Response Format: JSON**
* **Example Request:**

**{**

**“username”: “testUser”,**

**}**

* **Example Response:**

**{**

**“message”: “successfully cleared favorite brewery 2”**

**},**

**200**

**Route: /view-memory**

* **Request type: GET**
* **Purpose:** Displays a string representation of the 10 most recent successful searches.
* **Note:** A search can be of any length, but as long as all information was successfully retrieved at the same time, they will all count as 1 successful search.
* **Response Format: JSON**
  + **Success Response Example**
    - Code: 200
    - Content: {“memory”: [[*previous\_serach\_1*], [*previous\_serach\_2*], …]}
* **Example Request:**

**{**

**“username”: “testUser”,**

**}**

* **Example Response:**

**{ "memory": "[[{'id': '092efaaf-02ab-4ac6-a9c3-6269af54ddeb', 'name': 'Backlash Beer Co', 'brewery\_type': 'contract', 'address\_1': None, 'address\_2': None, 'address\_3': None, 'city': 'Boston', 'state\_province': 'Massachusetts', 'postal\_code': '02215', 'country': 'United States', 'longitude': None, 'latitude': None, 'phone': None, 'website\_url': 'http://backlashbeer.com/', 'state': 'Massachusetts', 'street': None}, {'id': '9375dc3a-6315-4ef4-a5e6-35a257deb8a3', 'name': 'Boston Beer Co', 'brewery\_type': 'micro', 'address\_1': '1 Design Center Pl Ste 850', 'address\_2': None, 'address\_3': None, 'city': 'Boston', 'state\_province': 'Massachusetts', 'postal\_code': '02210-2300', 'country': 'United States', 'longitude': None, 'latitude': None, 'phone': '6173685000', 'website\_url': None, 'state': 'Massachusetts', 'street': '1 Design Center Pl Ste 850'}], [{'id': '14ea4e7c-498c-417e-89e9-eabaf9b7051f', 'name': 'Cibolo Creek Brewing Co.', 'brewery\_type': 'brewpub', 'address\_1': '448 S Main St', 'address\_2': None, 'address\_3': None, 'city': 'Boerne', 'state\_province': 'Texas', 'postal\_code': '78006-2338', 'country': 'United States', 'longitude': '-98.72996029', 'latitude': '29.79015427', 'phone': '8308165275', 'website\_url': 'http://www.cibolocreekbrewing.com', 'state': 'Texas', 'street': '448 S Main St'}], [{'id': '9bde954f-21f8-4382-960b-302b52411dbb', 'name': 'Red Dragon Brewery', 'brewery\_type': 'micro', 'address\_1': '1419 Princess Anne St', 'address\_2': None, 'address\_3': None, 'city': 'Fredericksburg', 'state\_province': 'Virginia', 'postal\_code': '22401-3638', 'country': 'United States', 'longitude': '-77.46312137', 'latitude': '38.30737927', 'phone': None, 'website\_url': 'http://www.reddragonbrewery.com', 'state': 'Virginia', 'street': '1419 Princess Anne St'}]]" },**

**200**