# Documentation for Cocktails API

Also idk if you'll need it but just so you don't have to go through with stupid process of generating a token here have one I used Token: oat_MQ.NUpRTW42V291VjM5cG82S01CaFVFenBseEdWM3J1azA5VkhGWmV1SDk4MTIwMDE5OQ and the link to the hosted service: Link

# API Routes

## Registration and token generation

**POST: /api/register**
Takes in unique email and password:{"email":"[Unique email]",
"password":"[User password]"}

Returns created user

---

**POST: /api/generatetoken**

Takes in email and password of a user:
{"email":"wasd",
"password":"1234"}

Returns generated token

## Cocktails

**POST: /api/cocktails**

Takes in name,category_id,instruction:
{"name":[name],
"category_id":[category_id],
"instruction":[instruction]}

Returns generated cocktail

---

**PUT: /api/cocktails/update**

Takes in cocktail id,name,instruction,category_id:

{"id":"[Cocktail id]",
"name":[name],
"category_id":[category_id],
"instruction":[instruction]}

Returns updated cocktail

---

**GET: /api/cocktails/**

Returns all cocktails. Allows for parameters filtering/ordering parameters.

Filter: name, category_id, isalcoholic

Order: orderBy, orderDirection  // Default name ascending

---

**GET: /api/cocktails/:id**

Returns cocktail with given id

---

**DELETE: /api/cocktails/:id**

Deletes cocktail with given id and returns success message

---

## Cocktail ingredients

**POST: /api/cocktail_ingredients**

Takes in cocktail_id,ingredient_id,amount:
{"cocktail_id":[Cocktail id],
"ingredient_id":[Category id],
"amount":[Amount]}

Returns generated cocktail ingredient

---

**PUT: /api/cocktail_ingredients/update**

Takes in id, amount:

{"id":"[Cocktail ingredient id]",
"amount":[amount]}

Returns updated cocktail ingredient

---

**GET: /api/cocktail_ingredients/**

Returns all cocktail ingredients

---

**GET: /api/cocktail_ingredients/:id**

Returns cocktail ingredient with given id

---

**DELETE: /api/cocktail_ingredients/:id**

Deletes cocktail ingredient with given id and returns success message

# Categories

**POST: /api/categories**

Takes in name,description:
{"name":[name],
"description":[description]}

Returns generated category

---

**PUT: /api/categories/update**

Takes in id,name,description:

{"id":"[Category id]",
"name":[name],
"description":[description]}

Returns updated category

---

**GET: /api/categories/**

Returns all categories

---

**GET: /api/categories/:id**

Returns category with given id

---

**DELETE: /api/categories/:id**

Deletes category with given id and returns success message

# Ingredients

**POST: /api/ingredients**

Takes in name,description,alcohol,image(if not provided default is set):
{"name":[name],
"description":[description],
“alcohol”:”[Is alcoholic 1/0]”}

Returns generated ingredient

---

**PUT: /api/ingredients/update**

Takes in id,name,description,alcohol:

{"id":"[Category id]",
{"name":[name],

"description":[description],
“alcohol”:”[Is alcoholic 1/0]”}

Returns updated ingredient

---

**GET: /api/ingredients/**

Returns all ingredients

---

**GET: /api/ingredients/:id**

Returns ingredient with given id

---

**DELETE: /api/ingredients/:id**

Deletes ingredient with given id and returns success message

## Images

**Ingredients/images/:id**

Serves an image file of corresponding ingredient. // Doesn’t work for with the hosting I chose 😟