Our experiment 1 mainly focuses on choosing the right endpoints for the API and seeing if it fits what we need for our webapp. As we weren't concerned with it working in Phoenix, we made the API calls in python.

In [1]:
```
#Below is the API call to the endpoint which gives recipe information given an
id

import requests

# input: recipe id
# output recipe information

api_key = '88a684f43c204ecea33e2e0b904f7565'

url = f'https://api.spoonacular.com/recipes/716429/information?apiKey={api_ke
y}&includeNutrition=true'


headers = {
    'x-rapidapi-key': api_key,
    'x-rapidapi-host': "spoonacular-recipe-food-nutrition-v1.p.rapidapi.com"
    }

response = requests.request("GET", url, headers=headers)
```

In [2]:
```
import json
dicct = json.loads(response.text)

i = 0

#example of an ingredient that the API call returns
print(dicct['extendedIngredients'][0])
```

```
{'id': 1001, 'aisle': 'Milk, Eggs, Other Dairy', 'image': 'butter-sliced.jp
g', 'consistency': 'solid', 'name': 'butter', 'nameClean': 'butter', 'origina
l': '1 tbsp butter', 'originalString': '1 tbsp butter', 'originalName': 'butt
er', 'amount': 1.0, 'unit': 'tbsp', 'meta': [], 'metaInformation': [], 'measu
res': {'us': {'amount': 1.0, 'unitShort': 'Tbsp', 'unitLong': 'Tbsp'}, 'metri
c': {'amount': 1.0, 'unitShort': 'Tbsp', 'unitLong': 'Tbsp'}}}
```

In [3]:
```
#Below is an API call to the endpoint that takes in a list of ingredients and
 outputs a list of recipes
#input: list of ingredients
#output: list of recipes

ingredients = ['boneless skinless chicken breast', 'onion', 'light soy sauce']
num_results = 5


def ing_to_string(arr):
    query = ''
    for item in arr:
        query += f'{item},+'
    return query[:-2]

url = f'https://api.spoonacular.com/recipes/findByIngredients?apiKey={api_key}
&ingredients={ing_to_string(ingredients)}&ignorePantry=true&number={num_result
s}'


headers = {
    'x-rapidapi-key': api_key,
    'x-rapidapi-host': "spoonacular-recipe-food-nutrition-v1.p.rapidapi.com"
    }

response_recipes = requests.request("GET", url, headers=headers)
```

In [4]:
```
import pandas as pd

df_ing = pd.read_csv('top-1k-ingredients.csv', header=None)
df_ing.columns = ['Ingredients']
df_ing['id'] = 0

for index, row in df_ing.iterrows():
    text = row['Ingredients'].split(';')
    df_ing.at[index, 'Ingredients'] = text[0]
    df_ing.at[index, 'id'] = text[1]
df_ing.head()
```

Out[4]:

	Ingredients	id
0	5 spice powder	1002002
1	acorn squash	11482
2	adobo sauce	6979
3	agave nectar	19912
4	ahi tuna	15117

In [5]:
```
recipes = json.loads(response_recipes.text)

#example of a missed Ingredients in recipe 0
print('Missing Ingredient')
print(recipes[0]['missedIngredients'][0])

#example of an unused ingredient in recipe 0
print('\nUnused Ingredient: ')
print(recipes[0]['unusedIngredients'][0])
```

```
Missing Ingredient
{'id': 11215, 'amount': 3.0, 'unit': 'tsp', 'unitLong': 'teaspoons', 'unitSho
rt': 'tsp', 'aisle': 'Produce', 'name': 'garlic', 'original': '3 tsp minced g
arlic', 'originalString': '3 tsp minced garlic', 'originalName': 'minced garl
ic', 'metaInformation': ['minced'], 'meta': ['minced'], 'image': 'https://spo
onacular.com/cdn/ingredients_100x100/garlic.png'}

Unused Ingredient:
{'id': 11282, 'amount': 1.0, 'unit': 'serving', 'unitLong': 'serving', 'unitS
hort': 'serving', 'aisle': 'Produce', 'name': 'onion', 'original': 'onion',
'originalString': 'onion', 'originalName': 'onion', 'metaInformation': [], 'm
eta': [], 'image': 'https://spoonacular.com/cdn/ingredients_100x100/brown-oni
on.png'}
```

In [6]:
```
#Below is a list of recipes and their ids from the CSV file

for recipe in recipes:
    print(str(recipe['id']) + ' ' + recipe['title'])
```

```
984319 Honey Garlic Chicken
1093574 Cashew chicken
983941 Slow Cooker Asian Glazed Chicken
598510 Gung Bao Chicken
93560 Yakisoba Chicken
```