

Readme

Important:

Thank you for giving me this opportunity . I have tried my best to develop this project. There are some things that I request you consider.

1. The accuracy of the results depend on the formulas. For example the formula I used to calculate the discount diff is $((\text{basket_price} - \text{similar_product_basket_price}) / ((\text{basket_price} + \text{similar_product_basket_price}) / 2)) * 100$. I have taken the actual value and not the absolute value. If you expect different formula kindly try changing and then testing.
2. For expensive_list query_type I have compared a product to all the similar products and if the basket price of a product is greater than anyone of the similar products, then it gets added to result.
3. For discount_diff and competition filter type , I have compared a product to all the similar products for that particular competition and if the discount_diff is (greater, equal or lesser) based on operator specified in the filter, than the specified value in the filter, for even a single similar product, it gets added to the result.

API URL : <https://netaporterapikanishk.herokuapp.com>

The application contains 7 files:

- run.py
- load_json.py
- processing.py
- routes.py
- Procfile
- Netaporter_gb_similar.json

run.py

This module is the entry point of the flask app. This module imports the Flask module and starts the application

netaporter_gb_similar.json

This is a json file that contains the json data.

load_json.py

This module is responsible for loading the json data from the file "netaporter_gb_similar.json".

routes.py

This module handles the request received at the url <https://netaporterapikanishk.herokuapp.com/.lt> contains a function main() that handles the request. It loads the data from netaporter_gb_similar.json file using the load_json

module.when the data is loaded, it extracts the query_type and the filters from the request.Then it checks the query_type and calls the functions present in processing.py module based on the type of request.

Processing.py

This module is responsible for extracting the data from json.file based on the query and filters.It consists of the following functions:

- check_discount
- check_brand_name
- check_discount_dif_and_competition
- get_discounted_products_list
- get_discounted_products_count_and_avg_discount
- get_expensive_list
- get_competition_discount_diff_list
- check_filters

get_query_type :

This function is used for extracting the query_type from the request.

get_filters:

This function is used for extracting the filters from the request

check_discount:

This function is used for applying the discount filter on the products

This function takes in three parameters:

- product
- operator
- discount_value

It calculates the discount on the product based on the formula $[100 * (\text{difference of regular_price and offer_price}) / \text{regular_price}]$.

after calculating the discount it compares the it with the discount_value parameter based on the operator parameter and returns true or false accordingly.

check_brand_name:

This function used for applying the brand.name filter.

It takes 2 parameters:

- product
- brand

it compares the brand name of the product with the parameter brand.if they are equal, it returns true else false.

Check_discount_dif_and_competition:

This function is used to apply competition and discount_dif filter on the product.

It takes 4 paremeters :

- product
- competiton : the competition in the filter
- discount_dif_operand2: it is the operand2 in discount_dif filter
- discount_diff_operator : it is the operator in discount_dif filter

first it extracts the basket price of the product.

Then it goes through all the similar products of the given competiton

Extracts basket price of each similar product

Calculates discount diff based on formula : $((\text{basket_price} - \text{similar_product_basket_price}) / ((\text{basket_price} + \text{similar_product_basket_price}) / 2)) * 100$

Then it checks if the discount diff is >,<=,< discount_dif_operand2 based on the operator.

If the condition is true for even one of the similar products it return true.

If none of the similar products is satisfies the condition it returns false.

check_filters:

this function applies all the filters to a product and returns true if all the filters are passed else false

get_discounted_products_list:

This function is used when the query_type is discounted_products_list.

It accepts 2 parameters:

- products_json: json data
- filters

It traverses through all the products in the data.For each products it applies all the filters in the requests using the above functions.If a product passes all the filters its id is added to a list named result.Then the result list is returned in json format.

get_discounted_products_count_and_avg_discount():

This function is used when the query_type is discounted_products_count|avg_discount

It accepts 2 parameters:

- products_json: json data
- filters

A variable products_count is used as a counter to count the products that satisfy all the filters.Another variable discount_sum is used to store the sum of discounts of all the products that follow the filters

It traverses through all the products in the data. For each product it calculates discount and applies all the filters in the requests using the above functions. If a product passes all the filters the product_count counter is increased by one and the discount is added to the variable discount_sum. After all the products have been traversed the product count and average discount is returned in json format.

get_expensive_list:

This function is used when the query_type is expensive_list.

It traverses through all the products in the data. For each product it applies all the filters. If the product passes all the filters then it checks if its basket price is greater than any of its competitions. If it is so the product id is added to the result list.

get_competition_discount_diff_list :

This function is used when the query_type is competition_discount_diff_list.

It traverses through all the products in the data. For each product it applies all the filters. If the product passes all the filters then its product id is added to the result list. Then the result list is returned in json format.