Jack Morgan, Pedro De Sousa, Sean Gay, James Philhower
CSE 30332: Programming Paradigms
Fall 2019
24 November 2019

Group Final Project RESTful JSON Specification Table

Resource	Command Type	Input	Output

/food/return	POST	{
		"under":{},
		"equal":{},
		"over":{
		“203”: 5
		}
		}
		}
			{
			"over": {
			"Protein 5g": [
			"Amaranth grain, uncooked",
			"Artichokes, (globe or french), frozen, cooked, boiled, drained, without salt",
			"Asparagus, canned, drained solids",
			"Asparagus, frozen, cooked, boiled, drained, without salt"
			]
			}
			}

Our server only has one resource (/food/return) that does all of the processing for our Web App. It takes in a JSON object that has 3 sub objects that contain specifications for all of the data filters. The sub objects have the following keys: “over”, “equal”, and “under”, and each has the capacity to contain as many filters as the user wants to apply. The filters are applied as key-value pairs where the key is the nutrient (specified by a number- 203, 204, 205, or 269), and the value is the amount that you want to apply for that filter. In the above example, we are filtering for all foods that have Protein over 5g. The response comes over as a JSON object, that contains up to 3 sub objects, depending on how many filters were applied in the request. The sub objects contain key value pairs where the key is a string that has two words: the first is the nutrient, and the second is the amount. The value is an array that contains strings of all the food that matches that specific filter.

README.md:

To run the tests, type “python3 test_ws.py. The test file contains the correct URL and port number (which is 51069 in this case). The web service can be used by anyone who has the ability to create HTTP requests formatted with the above format. There is only one resource set up on our server and CORS has been disabled. Our front end will have a GUI for the user that provides them with the ability to apply as many filters as they want. The front end will take the filters, format the correct request object, send over the request, and format the response in a clear way for the user to see.