Documentation

Functions

There is only one function that drives this website: runQuery(). runQuery() is a function that is ran onclick of the button "Medication query" this function reads the dropdown selectors, forms a sparql query out of them dispatches the queries and forms the incoming JSON file to match the expected data structure of D3's forcegraph() function.

```
/*runQuery() is a function that is ran onclick of the button "Medication query"
 * this function reads the dropdown selectors, forms a sparql query out of them dispatches the queries and forms
 * the incoming JSON file to match the expected data structure of D3's forcegraph() function.*/
 function runQuery() {
     class SPARQLQueryDispatcher {
         constructor(endpoint) {
             this.endpoint = endpoint;
         }

         query(sparqlQuery) {
             const fullUrl = this.endpoint + '?query=' + encodeURIComponent(sparqlQuery);
             const headers = {'Accept': 'application/sparql-results+json'};

             return fetch(fullUrl, {headers}).then(body => body.json());
         }
     }

     //get the selected dropdown items
     var d1 = document.getElementById("ddDisease1");
     var disease1 = d1.value;
     var d2 = document.getElementById("ddDisease2")
     var disease2 = d2.value;

     //log the diseases ids to the console
     console.log(disease1)
     console.log(disease2)

     //set the endpointURL to wikidata sparql
     const endpointUrl = 'https://query.wikidata.org/sparql';

     //SPARQL Query for getting every drug that treats disease 1
     const sparqlQuery1 = `SELECT DISTINCT ?drug ?drugLabel  ?disease ?diseaseLabel
                            WHERE
                            {
                                wd:${disease1} wdt:P2176 ?drug.

                               BIND(wd:${disease1} as ?disease)
                               SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en" }
                            }`;

     //Dispatch query for disease 1
     const queryDispatcher1 = new SPARQLQueryDispatcher(endpointUrl);

     //add to the previous query every drug that treats disease 2
     queryDispatcher1.query(sparqlQuery1).then(
         function (results1){
             //SPARQL Query for getting every drug of for treating disease 2
             const sparqlQuery2 = `SELECT DISTINCT ?drug ?drugLabel ?disease ?diseaseLabel
                                   WHERE
                                   {
                                     wd:${disease2} wdt:P2176 ?drug.

                                     BIND(wd:${disease2} as ?disease)
                                     SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en" }
                                   }`;

             //Dispatch query for disease 2
             const queryDispatcher2 = new SPARQLQueryDispatcher(endpointUrl);

 /* The next part of the code;
 *  In a loop, sets;
 *  every disease to group 1,
```

```
    *  every drug for disease 1 to group 3,
    *  every drug for disease 2 to group 4
    *  and drugs that treat both as group 5
    *  every group will later be assigned to a different color in the graph.
    *  The nodes are placed in the networkData dictionary
    *
    *  After the nodes are set, the edges between those nodes are extracted from the json file and inserted
    *  into the networkData dictionary, along with the nodes.
    */

                queryDispatcher2.query(sparqlQuery2).then(
                    function (results2){
                        var networkData = {}
                        networkData["nodes"] = [];
                        networkData["links"] = [];
                        var drugSet = new Set();

                        //set both drug nodes to group 1
                        var disease1Node = {"id":results1.results.bindings[0].diseaseLabel.value, "group":1}
                        var disease2Node = {"id":results2.results.bindings[0].diseaseLabel.value, "group":1}
                        networkData["nodes"].push(disease1Node,disease2Node)

                        // for each drug in the results of query 1
                        for(var i =0; i < results1.results.bindings.length;i++){
                            if(!drugSet.has(results1.results.bindings[i].drugLabel.value)){

                                //set node to group 3 if treats disease 1
                                var drugNode = {"id":results1.results.bindings[i].drugLabel.value, "group":3}
                                networkData["nodes"].push(drugNode)
                                drugSet.add(results1.results.bindings[i].drugLabel.value)

                            }
                            //add link of the drug to the dictionary
                            var diseasedrugLink = {"source":results1.results.bindings[0].diseaseLabel.value,
                                "target":results1.results.bindings[i].drugLabel.value, "vaue":1 }
                            networkData["links"].push(diseasedrugLink)
                        }

                        // for each drug in the results of query 2
                        for(var j =0; j < results2.results.bindings.length;j++){
                            if(!drugSet.has(results2.results.bindings[j].drugLabel.value)) {

                                //set node to group 4 if treats disease 2
                                var drug2Node = {"id": results2.results.bindings[j].drugLabel.value, "group": 4}
                                networkData["nodes"].push(drug2Node)
                                drugSet.add(results2.results.bindings[j].drugLabel.value)
                            }else{
                                for(var k =0; k < networkData["nodes"].length; k++){
                                    if(networkData["nodes"][k].id === results2.results.bindings[j].drugLabel.value){
                                        networkData["nodes"][k].group = 5; //set node to group 5 if treats both
                                    }
                                }
                            }
                            //add link of the drug to the dictionary
                            var disease2drugLink = {"source":results2.results.bindings[0].diseaseLabel.value,
                                "target":results2.results.bindings[j].drugLabel.value, "value":1 }
                            networkData["links"].push(disease2drugLink)

                        }
                        // form a ForceGraph with the data in the dictionary
                        const chart = ForceGraph(networkData, {
                            nodeId: d => d.id,
                            nodeGroup: d => d.group,
                            nodeTitle: d => `${d.id}\n${d.group}`,
                            linkStrokeWidth: l => Math.sqrt(l.value),
                            width: 600,
                            height: 600,
                        });

                        // send the ForceGraph after removing the previous chart, if any
                        const div = document.getElementById('chart');
                        div.remove();
                        var elemDiv = document.createElement('div');
                        elemDiv.id = "chart"
                        document.body.appendChild(elemDiv);
                        elemDiv.appendChild(chart);
```

```
                })
            })
    }
```