Un exemplu de JavaScript ce analizează xml cu CD

var xmltoDict = {}; // dictionarul unde va fi inserat datele din xml

var xmlDoc;

// trigger pentru buttonul ascuns alege fisier

document.getElementById('file').addEventListener('click', function() {

document.getElementById('file-browse').click();

// file = document.getElementById("file-browse").files[0];

});

// deschidere si afisarea fisiereului intr-un textarea

// de asemenea inscriere fisiereului in dictionarul xmltoDict

document.getElementById('file-browse').addEventListener('change', function() {

document.getElementById('input-name').value = this.value;

var file = new FileReader();

var text;

file.onload = function() { // functia da incarcare a fisiereului

document.getElementById("data").textContent = this.result; // datele din fiesier sunt adaugate in textarea cu id-ul "data"

text = this.result; // variabilei text i se da valoarea pe care o are textarea

parser = new DOMParser(); // crearea obiectului pentru parsare

xmlDoc = parser.parseFromString(text, 'text/xml'); // variabilei xmlDoc i se ofera valoarea care o returneaza metoda parseFromString din obiectul "parser"

var children\_lv\_1; // variabila pentru copii XML

var children\_lv\_2; // variabila pentru copii XML

if(xmlDoc.children.length === 1) { //aici verificam daca in urma parsarii documentului XML avem vreun copil in caz de true se executa blocul <catalog> .. </catalog>

if(xmlDoc.children[0].childElementCount !== 0) { // verificam daca se mai contin ramuri

children\_lv\_1 = xmlDoc.children[0].childElementCount; // calculam cate ramuri sunt in primul nivel de exemplu <cd> ... </cd>

if(xmlDoc.children[0].children[0].childElementCount !== 0) { // verificam daca mai sunt subramuri intr-un nivel mai jos de <cd> .. </cd>

children\_lv\_2 = xmlDoc.children[0].children[0].childElementCount; // atribuim numarul de subramuri

if(!(xmlDoc.children[0].children[0].children[0].childElementCount !== 0)){

var temp = {}; //variabila temporara

var contor = 0;

// cream un select option cu totii copii din ramura <cd> .. </cd> si optinem <title>, <year> si asa mai departe

for(var i = 0; i < children\_lv\_2; i++){

var s\_option = document.getElementById("select-option");

var option = document.createElement("option");

option.text = xmlDoc.children[0].children[0].children[i].nodeName;

s\_option.add(option);

}

// cream un dictionarul catagalo care o sa fie alcatuit din nodul cd + contor (cd 1) el va contine datele din {title: ..., year: ..., author: ...}

for(var j = 0; j < children\_lv\_1; j++){

contor = j + 1;

for(i = 0; i < children\_lv\_2; i++){

temp[xmlDoc.children[0].children[j].children[i].nodeName] = xmlDoc.children[0].children[j].children[i].textContent;

}

// [xmlDoc.children[0].children[j].nodeName + " " + contor] are valoare de exemplu cd 1 = {title: ..., year: ..., author: ...}

xmltoDict[xmlDoc.children[0].children[j].nodeName + " " + contor] = temp;

temp = {};

}

}

}

}

}

};

file.readAsText(this.files[0]);

});

// in dependenta de primul select option afisam al doilea select option

// care sun complimentare de exemplu daca alegem "year" urmatorul select option

// va fi cu datele despre year "Newest", "Oldest"

// daca alegem "price" va afisa select option cu "lower" sau "higher" s.a.m.d

$('#select-option').on('change', function() {

if(this.value === 'year'){

document.getElementById('price-selection').className = 'hidden';

document.getElementById('year-selection').className = 'shown';

} else if(this.value === 'title' || this.value === 'name'){

var inp\_text = document.createElement("INPUT");

inp\_text.setAttribute("type", "text");

inp\_text.id = 'text-find';

inp\_text.className = 'input-dyn';

document.getElementById('elements').appendChild(inp\_text);

} else if(this.value === 'price'){

document.getElementById('year-selection').className = 'hidden';

document.getElementById('price-selection').className = 'shown';

}

});

//la tastarea butonul se vor lua valorile din select option, dupa care

// in dictionarul nostru se vor cauta valorile

// de exemplu primul select option "year" -> al doilea "newest"

document.getElementById('result').addEventListener('click', function () {

option\_1 = document.getElementById('select-option').value;

if(option\_1 === 'year'){ // optiunea din primul select option se verifica

option\_2 = document.getElementById('year-selection').value;

} else if(option\_1 === 'price'){

option\_2 = document.getElementById('price-selection').value;

}

/\* Can Be a function for Year \*/

var value, final;

if(option\_2 === 'Newest' || option\_2 === 'Expensive'){

value = 0;

} else {

value = 9999999999;

}

// se verifica compatibilitatea optiunilor 1 si 2, daca coincid se cauta

// rezultatul in dictionarul xmltoDict

if(option\_1 !== 'Select Element' && (option\_2 === 'Newest' || option\_2 === 'Oldest')){

result = function () {

var temp = {};

for (var keys in xmltoDict) {

temp = xmltoDict[keys];

// value = temp[option\_1];

if (parseInt(temp[option\_1]) > parseInt(value) && option\_2 === 'Newest') {

value = temp[option\_1];

final = temp;

}

if (parseInt(temp[option\_1]) < parseInt(value) && option\_2 === 'Oldest') {

value = temp[option\_1];

final = temp;

}

}

return final;

};

}

/\* end of the function \*/

/\* Can be a function for the Price

\* aici poate fi adaugata logica pentru pret\*/

if(option\_1 !== 'Select Element' && (option\_2 === 'Expensive' || option\_2 === 'Low-priced')){

result = function () {

var temp = {};

for (var keys in xmltoDict) {

temp = xmltoDict[keys];

// value = temp[option\_1];

if (parseFloat(temp[option\_1]) > parseFloat(value) && option\_2 === 'Expensive') {

value = temp[option\_1];

final = temp;

}

if (parseFloat(temp[option\_1]) < parseFloat(value) && option\_2 === 'Low-priced') {

value = temp[option\_1];

final = temp;

}

}

return final;

};

}

/\* Display Data

in caz ca functia nostra a gasit toate datele in dictionarul nostru

ea va afisa datele in textarea noastra \*/

var finalData = result();

var text = '';

for(var keys in finalData){

if(finalData.hasOwnProperty(keys)){

text += keys.charAt(0).toUpperCase() + keys.slice(1) + " : " + finalData[keys] + "\n";

}

}

console.log(text);

document.getElementById('data').value = text;

});