1. **How to compare two JSON have same properties without order?**
2. **let obj1={name:”Person1”, age:5};**
3. **let obj2={ age:5,name:”Person1” };**

**Ans:**

I have tried to compare those two JSON objects:

<input type="hidden" id="remoteJSON" name="remoteJSON" value="{"allowExternalMembers": "false", "whoCanJoin": "CAN\_REQUEST\_TO\_JOIN"}" /><br />

<input type="hidden" id="localJSON" name="localJSON" value="{"whoCanJoin": "CAN\_REQUEST\_TO\_JOIN", "allowExternalMembers": "false"}" /><br />

I got values with javascript and I tried to compare with : JSON.stringify(remoteJSON) == JSON.stringify(localJSON) but this return false: it seems that the order of the properties is important.

And I even tried deep compare with this solution and always got a false return.

Is there a fast way to do the issue with jQuery (i.e. libraries for comparing JSON) ?

Lodash.isEqual allows you to do that:

var

remoteJSON = {"allowExternalMembers": "false", "whoCanJoin": "CAN\_REQUEST\_TO\_JOIN"},

localJSON = {"whoCanJoin": "CAN\_REQUEST\_TO\_JOIN", "allowExternalMembers": "false"};

console.log( \_.isEqual(remoteJSON, localJSON) );

<script src="https://cdn.jsdelivr.net/npm/lodash@4.17.10/lodash.min.js"></script>

Lodash isEqual() method is the best way to compare two JSON object.

This will not consider the order of the keys in object and check for the equality of object. Example

const object1 = {

name: 'ABC',

address: 'India'

};

const object2 = {

address: 'India',

name: 'ABC'

};

JSON.stringify(object1) === JSON.stringify(object2)

// false

\_.isEqual(object1, object2)

// true

If sequence is not going to change than JSON.stringify() will be fast as compared to Lodash’s isEqual() method.

DeepCompare method to compare two json objects..

deepCompare = (arg1, arg2) => {

if (Object.prototype.toString.call(arg1) === Object.prototype.toString.call(arg2)){

if (Object.prototype.toString.call(arg1) === '[object Object]' || Object.prototype.toString.call(arg1) === '[object Array]' ){

if (Object.keys(arg1).length !== Object.keys(arg2).length ){

return false;

}

return (Object.keys(arg1).every(function(key){

return deepCompare(arg1[key],arg2[key]);

}));

}

return (arg1===arg2);

}

return false;

}

console.log(deepCompare({a:1},{a:'1'})) // false

console.log(deepCompare({a:1},{a:1})) // true

This code will verify the json independently of param object order.

var isEqualsJson = (obj1,obj2)=>{

keys1 = Object.keys(obj1);

keys2 = Object.keys(obj2);

//return true when the two json has same length and all the properties has same value key by key

return keys1.length === keys2.length && Object.keys(obj1).every(key=>obj1[key]==obj2[key]);

}

var obj1 = {a:1,b:2,c:3};

var obj2 = {a:1,b:2,c:3};

console.log("json is equals: "+ isEqualsJson(obj1,obj2));

alert("json is equals: "+ isEqualsJson(obj1,obj2));

Easy way to compare two json string in javascript

var obj1 = {"name":"Sam","class":"MCA"};

var obj2 = {"class":"MCA","name":"Sam"};

var flag=true;

if(Object.keys(obj1).length==Object.keys(obj2).length){

for(key in obj1) {

if(obj1[key] == obj2[key]) {

continue;

}

else {

flag=false;

break;

}

}

}

else {

flag=false;

}

console.log("is object equal"+flag);

I adapted and modified the code to write a function that does a deep comparison of two JS objects.

const isEqual = function(obj1, obj2) {

const obj1Keys = Object.keys(obj1);

const obj2Keys = Object.keys(obj2);

if(obj1Keys.length !== obj2Keys.length) {

return false;

}

for (let objKey of obj1Keys) {

if (obj1[objKey] !== obj2[objKey]) {

if(typeof obj1[objKey] == "object" && typeof obj2[objKey] == "object") {

if(!isEqual(obj1[objKey], obj2[objKey])) {

return false;

}

}

else {

return false;

}

}

}

return true;

};