//NOTE:

//1. This code is to Convert Map entries to Array Element and Array to Map entries

//2. We used Recursive function Method to implement the code

function SampleElement() {

  const NEW\_SONATA\_MAP\_TYPE="MAP"

  //Creating our own Map Object

  let mapelem = new Map();

  mapelem.set(117, 3);

  mapelem.set(261, 3);

  mapelem.set(119, 3);

  //Creating the Second Map Object

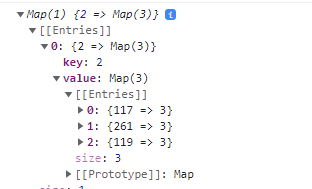
  let mapelem2 = new Map();

  mapelem2.set(2, mapelem); //Placing the first map object to Second map pbject

  console.log(mapelem2)

  //Below is the structure of the mapelem2





//Calling a function which is used for converting Map object entries to Array

let transformmap = transformToArray(mapelem2);

function transformToArray(dataToTransform) {

    if (dataToTransform instanceof Map) {

      const transformedData = new Map();

      dataToTransform.forEach((v, k) => {

        if (

          v instanceof Map ||

          v instanceof Array ||

          v instanceof Set ||

          v instanceof Object

        ) {

            transformedData.set(k,transformToArray(v));

        } else {

          transformedData.set(k, v);

        }

      });

      return Array.from(transformedData);

    }

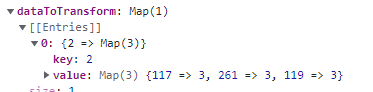
}

// Following are the steps how the functions are running

**STEP 1:**

function transformToArray(dataToTransform) {

//Initial Data of dataToTransform

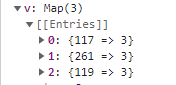


    if (dataToTransform instanceof Map) {//Here we check the condition whether the value it is instance of Map or not

      const transformedData = new Map();//Create a new Map Object named transformedData

      dataToTransform.forEach((value, key) => { //Iterating a Map Object

// value- //key

        if (

          value instanceof Map ||

          value instanceof Array || //Here we check the instance (value is the instanceof Map hence it becomes true)

          value instanceof Set ||

          Value instanceof Object

        ) {

            transformedData.set(key,transformToArray(value)); //Here the recursive function will be called ie.(transformToArray function will be called again) refer next page

        } else {

          transformedData.set(key, value);

        }

      });

      return Array.from(transformedData);

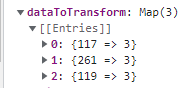
    }}

**STEP 2:**

//Here It is the function which will be running on the if condition statement of first function

function transformToArray(dataToTransform) {

// Now the dataToTransform values are



    if (dataToTransform instanceof Map) { //The condition satisfies because its instance of Map

      const transformedData = new Map(); //Creating the new Map Object

      dataToTransform.forEach((value, key) => { //Iterates untill the 3 entries loop over

        if (

          value instanceof Map ||

          value instanceof Array || //Here the condition doesn’t satisfies

          value instanceof Set ||

          value instanceof Object

        ) {

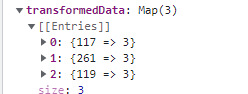
            transformedData.set(key,transformToArray(value));

        } else {

          transformedData.set(key,value);

        }

      }); //After the iteration of 3 entries the transformedData values are   
//NOTE:The TransformedData value is stored for the recursive function ie. Currently running function



      return Array.from(transformedData); // Here we are converting the map Object entries to Array and Return the values for the recursive function (moved back to STEP 1)   
//Converted Map object entries to Array  
 

    }}

**STEP 3:**

function transformToArray(dataToTransform) {

    if (dataToTransform instanceof Map) {   
      const transformedData = new Map();

      dataToTransform.forEach((value, key) => {

        if (

          value instanceof Map ||

          value instanceof Array ||   
          value instanceof Set ||

          Value instanceof Object

        ) {

            transformedData.set(key,transformToArray(value));

        } else {

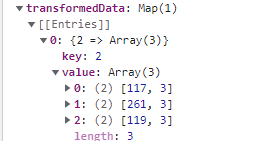
          transformedData.set(key, value);

        }

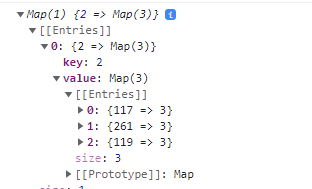
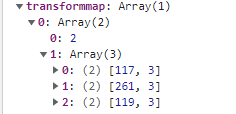
      });

      return Array.from(transformedData); // Returns the transformed array element

}}



let transformmap = transformToArray(mapelem2);

Converted map entries to Object