Javascript - Day -9 : MRF - array method:

Solving problems using array functions on rest countries data.

* 1. Get all the countries from Asia continent /region using Filter function

CODE:  
var newRequest = new XMLHttpRequest();

//we will send a instruction by applying on load

newRequest.onload = function () {

  if (newRequest.status >= 200 && newRequest.status <= 400) {

    var x = JSON.parse(newRequest.response);

    const countriesInAsia = x.filter((x) => x.region == "Asia"); //filter function used to create an array of json countries whose region is in asia.

    console.log(countriesInAsia); //displaying resultant array

  } else {

    console.log(this.responseText);

  }

};

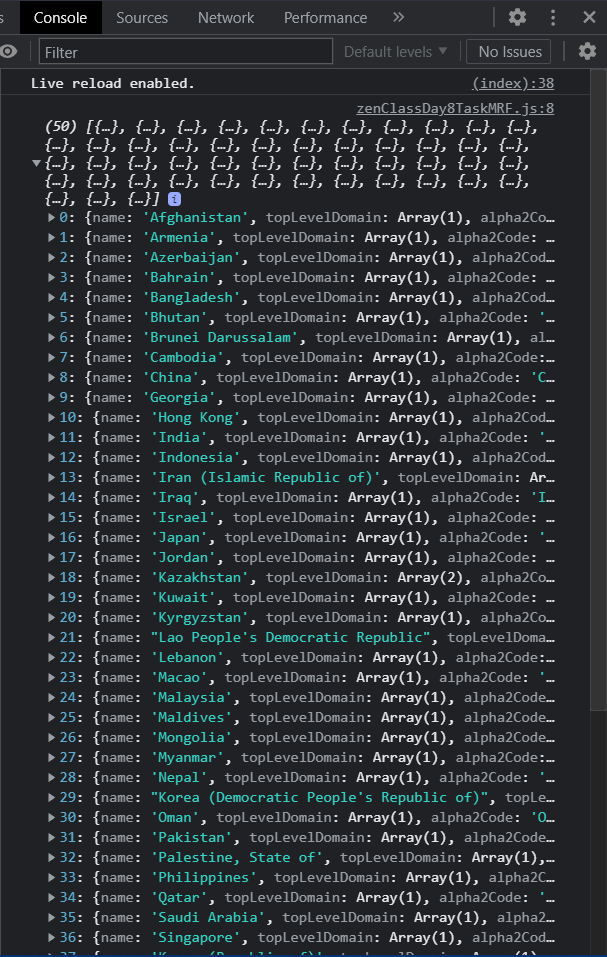
newRequest.open(

  "GET",

  "https://raw.githubusercontent.com/rvsp/restcountries-json-data/master/res-countries.json"

);

newRequest.send();

OUTPUT:  


1. Get all the countries with a population of less than 2 lakhs using Filter function

CODE:  
 var newRequest = new XMLHttpRequest();

//we will send a instruction by applying on load

newRequest.onload = function () {

  if (newRequest.status >= 200 && newRequest.status <= 400) {

    var x = JSON.parse(newRequest.response);

    const populationLess2L = x.filter((x) => x.population < 200000); //filter function used to create an array of json countries whose population is less than 2 lakhs

    console.log(populationLess2L); //displaying resultant array

  } else {

    console.log(this.responseText);

  }

};

newRequest.open(

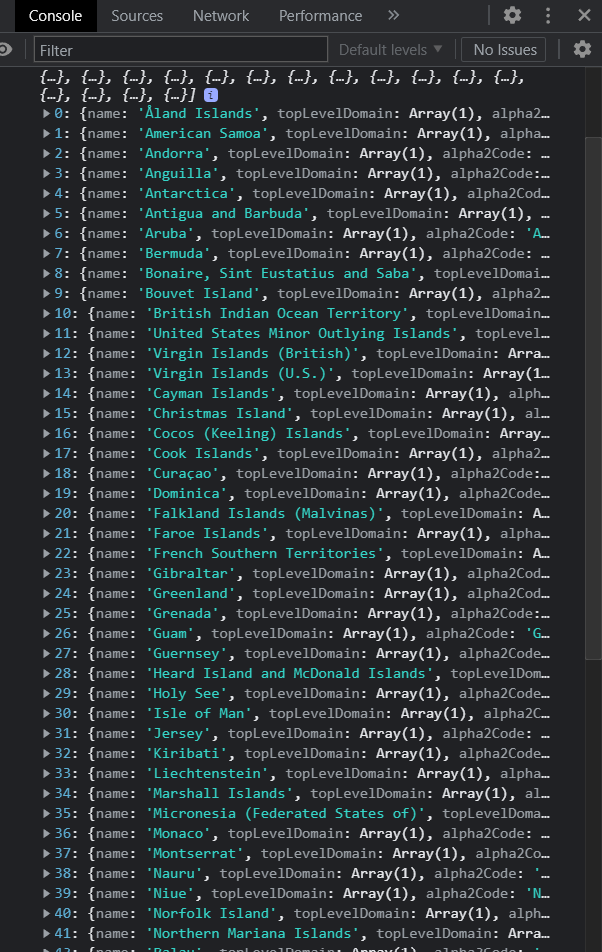
  "GET",

  "https://raw.githubusercontent.com/rvsp/restcountries-json-data/master/res-countries.json"

);

newRequest.send();

OUTPUT:





1. Print the following details name, capital, flag using forEach function

CODE:  
var newRequest = new XMLHttpRequest();

//we will send a instruction by applying on load

newRequest.onload = function () {

  if (newRequest.status >= 200 && newRequest.status <= 400) {

    var x = JSON.parse(newRequest.response);

    x.forEach(

      (x) => console.log(`name:${x.name}, capital:${x.capital}, flag:${x.flag}`) //to display respective countries' name, capital, flag in the console

    );

  } else {

    console.log(this.responseText);

  }

};

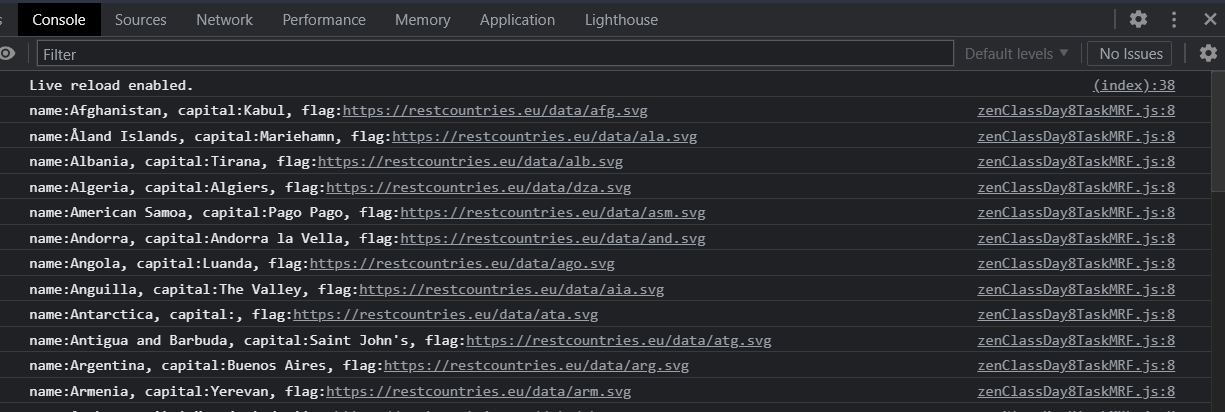
newRequest.open(

  "GET",

  "https://raw.githubusercontent.com/rvsp/restcountries-json-data/master/res-countries.json"

);

newRequest.send();

OUTPUT:  


1. Print the total population of countries using reduce function

CODE:  
var newRequest = new XMLHttpRequest();

//we will send a instruction by applying on load

newRequest.onload = function () {

  if (newRequest.status >= 200 && newRequest.status <= 400) {

    var x = JSON.parse(newRequest.response);

    const totalPopulation = x

      .map((populationArray) => populationArray.population)

      .reduce((x, y) => x + y);

    //first mapping json population data into an array of population

    //reduce function is used to add population from array created from mapping

    console.log(totalPopulation); //displaying resultant array

  } else {

    console.log(this.responseText);

  }

};

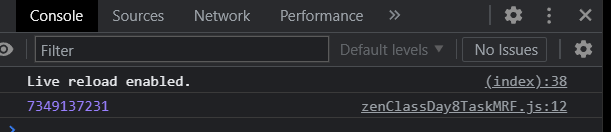
newRequest.open(

  "GET",

  "https://raw.githubusercontent.com/rvsp/restcountries-json-data/master/res-countries.json"

);

newRequest.send();

OUTPUT:  


1. Print the country which uses US Dollars as currency.

var newRequest = new XMLHttpRequest();

//we will send a instruction by applying on load

newRequest.onload = function () {

  if (newRequest.status >= 200 && newRequest.status <= 400) {

    var x = JSON.parse(newRequest.response);

    const currencyArray = x.map((x) => x.currencies);

    //creating an array of currencies related json data

    const nameArray = x.map((x) => x.name);

    //creating an array of country names later to be mapped with currencies

    //from the api we can observe that currencies in itself is an array of json thus this is combination of array-json-array

    const newArray = [];

    // //creating new array to remove json out of inner array

    for (key in currencyArray) {

      currencyArray[key][0].country = nameArray[key];

      newArray.push(currencyArray[key][0]);

      //logic to push inner currency json into newly created array so that filter function can be easily applied

      //in the final array to recognize to which country the USD belongs to, we add new element called country in to the currency json.

    }

    // console.log(newArray);

    const finalArray = newArray.filter((x) => x.code == "USD");

    // //using normal filter function on array of json with comparing code = USD to identify currency as requested in question as dollars

    console.log(finalArray);

  } else {

    console.log(this.responseText);

  }

};

newRequest.open(

  "GET",

  "https://raw.githubusercontent.com/rvsp/restcountries-json-data/master/res-countries.json"

);

newRequest.send();

OUTPUT:

