Csaba Garaguly - 20137575
Javascript - Chall,enge 3

Recap of the course

For this challenge, we learned about APIs. In the course and in the lectures the main focus was on the Mapbox.gl API, but we also learned about other APIs and how to read data from them and then display them on our site. Furthermore about the Mapbox.gl API we learned how to add controls zoom in zoom out, search bar countrols, etc…
Another big chapter was about openweather API, we learned how to read data from a position. The challenge was to read data from at least 2 API that is connected to each other.

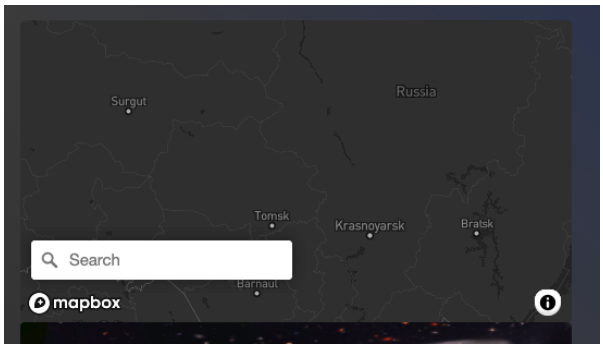
The Challange

What I did for this challenge is, first of all, I wanted to be creative and since I love traveling around the world, I came up with the idea to generate a random position and pull the data from that position. My first problem was that most of the time it landed on the water, which I asked for help from the teacher later on and they provided me a water.io API that can tell if the position is on land or water. With this idea I went further, I decided to display the data on the left side and the map and a random picture on the right side.
The map updates as soon as there is a new position, the picture under it is just the picture of the day for that day from NASA API.
For the data, I'm using a while loop which generates random coordinates till the position is on land. We know if it's on land by the water.io API. I'm using async functions so the while stops till we get an answer from the site.
Furthermore, I'm using airmap and openweather API to pull the data from the positions and using NASA's meteor API to check if in the current week period is there any asteroid that could be dangerous

```
//regenerate if positon is water
async function displaywol(){
  console.log(waterorland)
  decider = false;

  while (decider != true) {
      if (waterorland == true) {
        randomLong = Math.round((Math.random()*360 - 180) * 1000)/1000;
        randomLat = Math.round((Math.random()*160 - 60) * 1000)/1000;

        let response = await fetch(`https://api.onwater.io/api/v1/results/${randomLat},${randomLong}
        let data = await response.json()
        console.log(response.status)
        plocation.textContent = "Loading... (" + response.status + ")"
        //console.log(data.water)
        if (data.water == false) {
          waterorland = false
          openweatherrun();
          mapload();
          distance();
          airmap();
        }
        //document.querySelector("#td").innerHTML = `${inusa}`
      }
      else if (waterorland == false){
        break;
      }
      else if (waterorland = "undefined"){
        console.log(waterorland)
        break;
      }
      else {
        console.log(waterorland)
        break;
      }
```



```
//check if water or land --!regenerate
  //onwater.io //token: Nz9XmvuxxP4RikZvjVpF
  let onwater = {}
  let xhr2 = new XMLHttpRequest()
  xhr2.open('GET', `https://api.onwater.io/api/v1/results/${randomLat},${randomLong}?
  xhr2.responseType = 'text'

  var waterorland;
  xhr2.addEventListener('load', function(){
    if (xhr2.status === 200) {
      //console.log("wateronio success")
      onwater = JSON.parse(xhr2.responseText)
      waterorland = onwater.water
      console.log(waterorland)
      if (waterorland == true) {
        //console.log("inside if runs")
        displaywol();
      }
      //console.log(waterorland)
    }
    else {
      console.log(xhr.status)
      console.log("no success")
    }
  }, false)
  xhr2.send()
```

Always on land

```
//openweather
const plocation = document.querySelector('#location span')
const pweather = document.querySelector('#weather span')
const ptemperature = document.querySelector('#temperature span')
const pwind = document.querySelector('#wind span')
const plocaltime = document.querySelector('#localtime span')

function openweatherrun(){
  let openWeatherData = {}
  let xhr = new XMLHttpRequest()
  xhr.open('GET', `https://api.openweathermap.org/data/2.5/weather?lat=${randomLat}&lon=${randomLong}&
  xhr.responseType = 'text'

  xhr.addEventListener('load', function(){
    if (xhr.status === 200) {
      plocation.textContent = "Loading..."
      openWeatherData = JSON.parse(xhr.responseText)
      populateWeatherInfo()
    }
    else {
      plocation.textContent = "error"
      //p.textContent = "error" + xhr.status
    }
  }, false)

  xhr.send()
  //pull name,,temp,,wind,,time
  function populateWeatherInfo(){
    const location = openWeatherData.name
    const tempk = openWeatherData.main.temp
    //(tempf - 32) * 5/9  //F to C
    const tempc = Math.round(tempk-273.15)
    const wind = Math.round(openWeatherData.wind.speed)
    const country = openWeatherData.sys.country
    const weather = openWeatherData.weather[0].description
```

LOCATION	BELYY YAR, RU
WEATHER	BROKEN CLOUDS
TEMPERATURE	10 C
WIND	2 MPH
LOCAL TIME	09:40

```
//picture of the day
async function nasapotd(){
  let response = await fetch(`https://api.nasa.gov/planetary/apod?api_key=18E77fmBP8xCUkudwtUE3ppQdU44sZMSs
  //console.log(response);
  let data = await response.json()
  //console.log(data)
  document.querySelector("#nasaearth").innerHTML = ""
  document.querySelector("#nasaearth").innerHTML += `<img src="${data.url}" style="width: 44vw; height: 22v
}
nasapotd()
```

```
//Asteroid data
async function nasaasteroids(){
  let response = await fetch(`https://api.nasa.gov/neo/rest/v1/neo/3542519?api_key=18E77fmBP8xCUkuc
  let data = await response.json()
  var asteroid = data.name;
  var noasteroid;
  if (asteroid = "") {
    noasteroid = "None"
  }
  else {noasteroid = "Yes"}
  document.querySelector("#asteroid").innerHTML = `${noasteroid}`
  document.querySelector("#astname").innerHTML = `${data.name}`
  document.querySelector("#astdang").innerHTML = `${data.is_potentially_hazardous_asteroid}`
}
nasaasteroids()
```

ASTEROIDS	YES
NAME	(2010 PK9)
DANGEROUS	TRUE

```
//Elevation //Airmap
async function airmap(){
  let response = await fetch(`https://api.airmap.com/elevation/v1/ele/?points=${randomLat},${randomLong}`);
  let data = await response.json()
  //console.log("line 242")
  console.log(data)
  var elevation = data.data[0];
  //console.log(data.data[0] + "--no");
  if (elevation == null) {
    elevation = "Unknown"
    console.log("this runns")
    document.querySelector("#elevation").innerHTML = ""
  }
  document.querySelector("#elevation").innerHTML = `${data.data[0]} M`
}
airmap()
```


```
// distance calculation
//Kennedy Space Center, Cape Canaveral, Florida
var fllong = -80.6489808; //fl = Florida
var fllat = 28.5728722;

async function distance(){
let response = await fetch(`https://api.distancematrix.ai/maps/api/distancematrix/json?origins=${fllat},${f
&key=D9jGnYskyQ6TqTC0xFZU4CBvkJijt`);
  let data = await response.json()
  console.log(data.rows[0].elements[0].distance.text)
  var inusa = data.rows[0].elements[0].distance.text
  if (inusa === "0 m") {
    inusa = "Out of USA"
  }
  document.querySelector("#td").innerHTML = `${inusa}`
}
distance()
```