

The program uses AJAX (fetch API or XHR object) to download the JSON data from the Stop List API.

o To avoid frequently accessing the API, which results in receiving a 403 status code, the program should use local SessionStorage to keep a copy of the Stop List JSON data. Upon reloading the App, the program should check whether a copy of the JSON data exists in the SessionStorage before issuing the AJAX request.

o Data in the SessionStorage will be cleared after closing the tab/window, thus running the program on a new tab/window should trigger the download of the JSON data again.

  分享 ...

```
javascript ^
// 假设 Stop List API 的 URL
const stopListApiUrl = 'https://data.etabus.gov.hk/v1/transport/kmb/stop';

async function fetchStopListData() {
  // 检查 SessionStorage 中是否存在数据
  const storedData = sessionStorage.getItem('stopListData');
  if (storedData) {
    // 如果存在, 直接解析并返回
    return JSON.parse(storedData);
  }

  try {
    // 不存在则使用 fetch API 发送请求
    const response = await fetch(stopListApiUrl);
    if (!response.ok) {
      // 如果请求失败, 抛出错误
      throw new Error(`HTTP error! status: ${response.status}`);
    }
    const data = await response.json();
    // 将获取到的数据存储到 SessionStorage
    sessionStorage.setItem('stopListData', JSON.stringify(data));
    return data;
  } catch (error) {
    console.error('Error fetching stop list data:', error);
    return null;
  }
}
```

用fetch.then的方法

📄 📄 ➦ 分享 ...

```
// 假设 Stop List API 的 URL
const stopListApiUrl = 'https://data.etabus.gov.hk/v1/transport/kmb/stop';

function fetchStopListData() {
  // 先检查 sessionStorage 中是否有存储的数据
  const storedData = sessionStorage.getItem('stopListData');
  if (storedData) {
    // 如果有存储的数据，解析并返回一个 Promise 对象
    return Promise.resolve(JSON.parse(storedData));
  }

  // 如果没有存储的数据，使用 fetch 发起请求
  return fetch(stopListApiUrl)
    .then(response => {
      if (!response.ok) {
        // 若请求失败，抛出错误
        throw new Error(`HTTP error! status: ${response.status}`);
      }
      // 解析响应为 JSON 格式
      return response.json();
    })
    .then(data => {
      // 将获取到的数据存储到 sessionStorage 中
      sessionStorage.setItem('stopListData', JSON.stringify(data));
      return data;
    })
    .catch(error => {
      console.error('Error fetching stop list data:', error);
      return null;
    });
}
```

种方式，代码能有效避免对 API 的频繁请求，同时在关闭标签页或窗口次打开页面时会重新从 API 获取数据。

Based on search results, the program displays a list of bus stops ordered by the distance from the device's location as shown in [Figure 3]. For each bus stop, it displays the distance and the English name of the bus stop.
o If the program cannot find any bus stops within this search radius, the message – "Cannot locate nearby bus stops" should be displayed instead.

📄 📄 ➦ 分享 ...

面的代码逻辑和当前需求，我们可以在原有的代码基础上进行修改，根的位置和搜索半径展示按距离排序的巴士站列表。以下是完整的代码示

```
81 }
82
83 async function findNearbyStops(lat, lon) {
84   var print = document.getElementById("stopsout");
85   var radius = parseInt(document.getElementById("distance").value, 10);
86   const stopData = await fetchKmbStopData();
87   if (!stopData) {
88     return;
89   }
90   const stops = stopData.data;
91   const nearbyStops = [];
92   for (const stop of stops) {
93     const rawLat = stop.lat;
94     const rawLon = stop.long;
95     const stopLat = parseFloat(rawLat);
96     const stopLon = parseFloat(rawLon);
97     const distance = calculateDistance(lat, lon, stopLat, stopLon);
98     if (distance !== null && distance <= radius) {
99       nearbyStops.push({
100         stop: stop,
101         distance: distance
102       });
103     }
104   }
105 }
```

how to insert map into every summary

```

setTimeout(() => {
  new ol.Map({
    target: mapId,
    layers: [
      new ol.layer.Tile({
        source: new ol.source.OSM()
      })
    ],
    view: new ol.View({
      center: ol.proj.fromLonLat([stopLon, stopLat]),
      zoom: 15
    })
  });
}, 0);

```

how to add image to current position

```

// 创建标记
const marker = new ol.Feature({
  geometry: new ol.geom.Point(ol.proj.fromLonLat([lon, lat])) // 设置标记位置
});

// 设置标记样式（使用图像）
marker.setStyle(new ol.style.Style({
  image: new ol.style.Icon({
    src: 'https://openlayers.org/en/latest/examples/data/icon.png', // 图标路径
    scale: 0.5 // 图标大小
  })
}));

// 创建矢量图层并添加标记
const vectorLayer = new ol.layer.Vector({
  source: new ol.source.Vector({
    features: [marker]
  })
});

// 将矢量图层添加到地图
map.addLayer(vectorLayer);
}

// 其他函数（fetchKmbStopData、ETA、calculateDistance、findNearbyStops 等）保持不变

```

如何让距离显示在web上是按顺序的

```
// 遍历所有站点，计算距离并筛选符合条件的站点
for (const stop of stops) {
  const stopLat = parseFloat(stop.lat);
  const stopLon = parseFloat(stop.long);
  const distance = Math.trunc(calculateDistance(lat, lon, stopLat, stopLon));

  if (distance !== null && distance <= radius) {
    nearbyStops.push({ ...stop, distance }); // 将站点信息和距离一起存储
  }
}

// 按距离升序排序
nearbyStops.sort((a, b) => a.distance - b.distance);
```