

당근마켓 동네인증 API

1. 인증키 발급이후 script문 작성

```
<script type="text/javascript"
src="//dapi.kakao.com/v2/maps/sdk.js?appkey=c56a5ac8208747818bdae7eb60e05ea&libraries=services"></script>
```

2. 동네 인증 form 및 modal 형태 출력

```
<!-- 동네 인증 -->
// 해당 폼을 이용해서 현대 db에 저장된 각 카테고리를 찾아가게 함
<form id="location_form" method="post" action="location_form">
  <!-- 모달요소 -->
  <div id="myModal" class="modal">
    <div class="modal-content">
      <span class="close" onclick="closeModal()">&times;</span>

      <div id="modal-content"></div>
      <input class="localsubBtn" type="submit" value="동네 인증"
        id="submit-btn" disabled />
    </div>
  </div>

  <button id="find-me">동네인증</button>
  <br />
  <p id="status"></p>
  <a id="map-link" target="_blank"></a>
  <div id="map" style="width: 500px; height: 500px; margin-left: 50px"></div>
  <input type="hidden" id="location-input" name="newLocation" value="{detail_loc}" />
</form>
```

3. Geolocation API를 사용하여 현재 위치 출력 (성공 시)

```
function success(position) {

  // 현재 위치의 위도와 경도 추출
  const latitude = position.coords.latitude;
  const longitude = position.coords.longitude;

  // 현재 위치 상태 표시
  const status = document.querySelector("#status");
  status.textContent = "동네 인증";

  // 맵 링크 초기화
  const mapLink = document.querySelector("#map-link");
  mapLink.href = "";
  mapLink.textContent = "";

  // Kakao Maps API를 사용하여 현재 위치를 지도에 표시
  const mapContainer = document.getElementById("map");
```

```

const mapOption = {
  center: new kakao.maps.LatLng(latitude, longitude),

  // 지도 크기 레벨 지정
  level: 3,
};
const map = new kakao.maps.Map(mapContainer, mapOption);

const markerPosition = new kakao.maps.LatLng(latitude, longitude);
const marker = new kakao.maps.Marker({
  position: markerPosition,
});

marker.setMap(map);

```

4. Geolocation API를 사용하여 클릭 이벤트 발생 시 현재 위치 정보 출력

```

// 지도 클릭 시 위치 정보 확인
kakao.maps.event.addListener(map, "click", function (mouseEvent) {
  searchDetailAddrFromCoords(mouseEvent.latLng, function (result, status) {
    if (status === kakao.maps.services.Status.OK) {

      // 클릭한 위치의 주소 정보 가져오기
      var detailAddr = !!result[0].road_address
      ? "<div> 도로명주소 : " + result[0].road_address.address_name + "</div>"
      : "";
      detailAddr +=
        "<div> 지번 주소 : " + result[0].address.address_name + "</div>";

      // 정보를 마커와 인포윈도우를 통해 지도 상에 표시
      var content =
        '<div class="bAddr">' +
        '<span class="title">현재 주소정보</span>' +
        detailAddr +
        "</div>";

      marker.setPosition(mouseEvent.latLng);
      marker.setMap(map);

      infowindow.setContent(content);
      infowindow.open(map, marker);
    }
  });
});

```

5. 모달 출력 시 위치 정보 표시 및 로깅

```

// 모달에 위치 정보 표시
kakao.maps.event.addListener(map, "click", function (mouseEvent) {
  const newLatitude = mouseEvent.latLng.getLat();
  const newLongitude = mouseEvent.latLng.getLng();

  const newLocation = checkLocation(newLatitude, newLongitude);

```

```

// 콘솔에 새로운 위치 정보 로깅
console.log("New location:", newLocation);

const modalContent = document.getElementById("modal-content");
let newLocationElement = modalContent.querySelector("#new-location");

if (!newLocationElement) {
  // 새로운 위치 정보 엘리먼트 생성
  newLocationElement = document.createElement("div");
  newLocationElement.id = "new-location";
  modalContent.appendChild(newLocationElement);
}

if (typeof newLocation === "object" && Object.keys(newLocation).length !== 0) {
  // 새로운 위치가 확인되면 모달에 표시
  const detailLoc = newLocation.detail_loc;
  newLocationElement.textContent =
    "인증되었습니다. 현재위치:" +
    newLocation.locCode +
    " - " +
    newLocation.detail_loc +
    ". 클릭한 위치의 위도는 " +
    newLatitude +
    "이고, 경도는 " +
    newLongitude +
    "입니다.";

  // 새로운 위치 값을 입력란에 할당
  document.getElementById("location-input").value = newLocation.locCode + " - " + newLocation.detail_loc;
} else {
  // 새로운 위치가 확인되지 않으면 콘솔에 로그 출력
  console.log("New location:", newLocation);
  newLocationElement.textContent = newLocation;

  // 새로운 위치 값을 입력란에 할당
  document.getElementById("location-input").value = newLocation;
}
});

```

6. 지도 이동 시 중앙 위치 및 현재 위치

```

// 지도 이동 시 중앙 위치 정보 표시
kakao.maps.event.addListener(map, "idle", function () {
  searchAddrFromCoords(map.getCenter(), displayCenterInfo);
});
}

// 현재 위치 찾기 오류 처리
function error() {
  const status = document.querySelector("#status");
  status.textContent = "확인되지 않은 위치입니다.";
}

// Geolocation API를 통해 현재 위치 찾기
function geoFindMe() {
  const status = document.querySelector("#status");
  const mapLink = document.querySelector("#map-link");

  mapLink.href = "";
  mapLink.textContent = "";
}

```

```

if (!navigator.geolocation) {
  // Geolocation API가 지원되지 않는 경우
  status.textContent = "지오로케이션을 지원하지 않는 브라우저입니다.";
} else {
  // Geolocation API를 통해 현재 위치 찾기
  status.textContent = "현재 위치를 찾고 있습니다...";
  navigator.geolocation.getCurrentPosition(success, error);
}
}

```

7. 두 지점간의 거리 계산 (Haversine 공식 사용)

```

// 두 지점 간의 거리 계산 (Haversine 공식 사용)
function getDistanceFromLatLonInKm(lat1, lon1, lat2, lon2) {
  const R = 6371; // 지구의 반지름 (단위: km)
  const dLat = deg2rad(lat2 - lat1);
  const dLon = deg2rad(lon2 - lon1);
  const a =
    Math.sin(dLat / 2) * Math.sin(dLat / 2) +
    Math.cos(deg2rad(lat1)) * Math.cos(deg2rad(lat2)) * Math.sin(dLon / 2) * Math.sin(dLon / 2);
  const c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
  const d = R * c; // 두 지점 간의 거리 (단위: km)
  return d;
}

// 각도를 라디안으로 변환
function deg2rad(deg) {
  return deg * (Math.PI / 180);
}

```

8. 현재 위치 주변의 가까운 위치 찾기 및 사용자에게 선택지 제공

```

// 현재 위치 주변의 가까운 위치 찾기 및 사용자에게 선택지 제공
function checkLocation(latitude, longitude) {
  let minDistance = Infinity;
  let closestLocations = [];
  const locationsWithinRange = [];
  const locations = [
    // ... (미리 정의된 위치 정보 배열)
    // ... (위치 정보 배열 입력시 각 위도, 경도 필요로 )
    // 서울시
    {
      locCode: "서울시",
      detail_loc: "강남구",
      latitude: 37.5172,
      longitude: 127.0473,
    },
    {
      locCode: "서울시",
      detail_loc: "강서구",
      latitude: 37.5509,
      longitude: 126.8497,
    },
    {
      locCode: "서울시",

```

```

        detail_loc: "강동구",
        latitude: 37.5618,
        longitude: 127.0371,
    },
    // ..... 이하 생략 .....
];

const desiredRange = 5; // 변경하고자 하는 거리 범위 (단위: km)

for (let location of locations) {
    const distance = getDistanceFromLatLonInKm(
        latitude,
        longitude,
        location.latitude,
        location.longitude
    );

    if (distance <= desiredRange) {
        locationsWithinRange.push(location);
    }

    if (distance < minDistance) {
        minDistance = distance;
        closestLocations = [location];
    } else if (distance === minDistance) {
        closestLocations.push(location);
    }
}

if (locationsWithinRange.length > 0) {
    if (locationsWithinRange.length === 1) {
        // 주변에 하나의 위치가 있는 경우
        return locationsWithinRange[0].detail_loc;
    } else {
        // 여러 위치가 주변에 있는 경우 사용자에게 선택지 제공
        let userChoice;
        do {
            userChoice = prompt(
                "어디에서 거래할까요? (근방 " +
                desiredRange +
                "km 지역) \n " +
                locationsWithinRange
                    .map((loc, index) => (index + 1) + "." + (loc.locCode + "-" + loc.detail_loc))
                    .join('\n')
            );
        } while (!/^d+$/.test(userChoice) || userChoice <= 0 || userChoice > locationsWithinRange.length);

        userChoice = parseInt(userChoice);

        console.log(locationsWithinRange[userChoice - 1].detail_loc);
        return locationsWithinRange[userChoice - 1].detail_loc;
    }
}

return "";
}

```

9. 모달 부분

```

// 모달 열기
function openModal() {
    document.getElementById("myModal").style.display = "block";
}

// 모달 닫기
function closeModal() {
    document.getElementById("myModal").style.display = "none";
}

// "Show my location" 버튼 클릭 시 모달 열기
document.getElementById("find-me").addEventListener("click", function () {
    const statusContent = document.getElementById("status").outerHTML;
    const mapLinkContent = document.getElementById("map-link").outerHTML;
    const mapContent = document.getElementById("map").outerHTML;

    const content = statusContent + mapLinkContent + mapContent;
    document.getElementById("modal-content").innerHTML = content;
    openModal();
    geoFindMe();
});

// 모달에 위치 정보 표시
document.getElementById("find-me").addEventListener("click", function () {
    const statusContent = document.getElementById("status").outerHTML;
    const mapLinkContent = document.getElementById("map-link").outerHTML;
    const mapContent = document.getElementById("map").outerHTML;

    const content = statusContent + mapLinkContent + mapContent;
    document.getElementById("modal-content").innerHTML = content;
    openModal();
    geoFindMe();
});

```