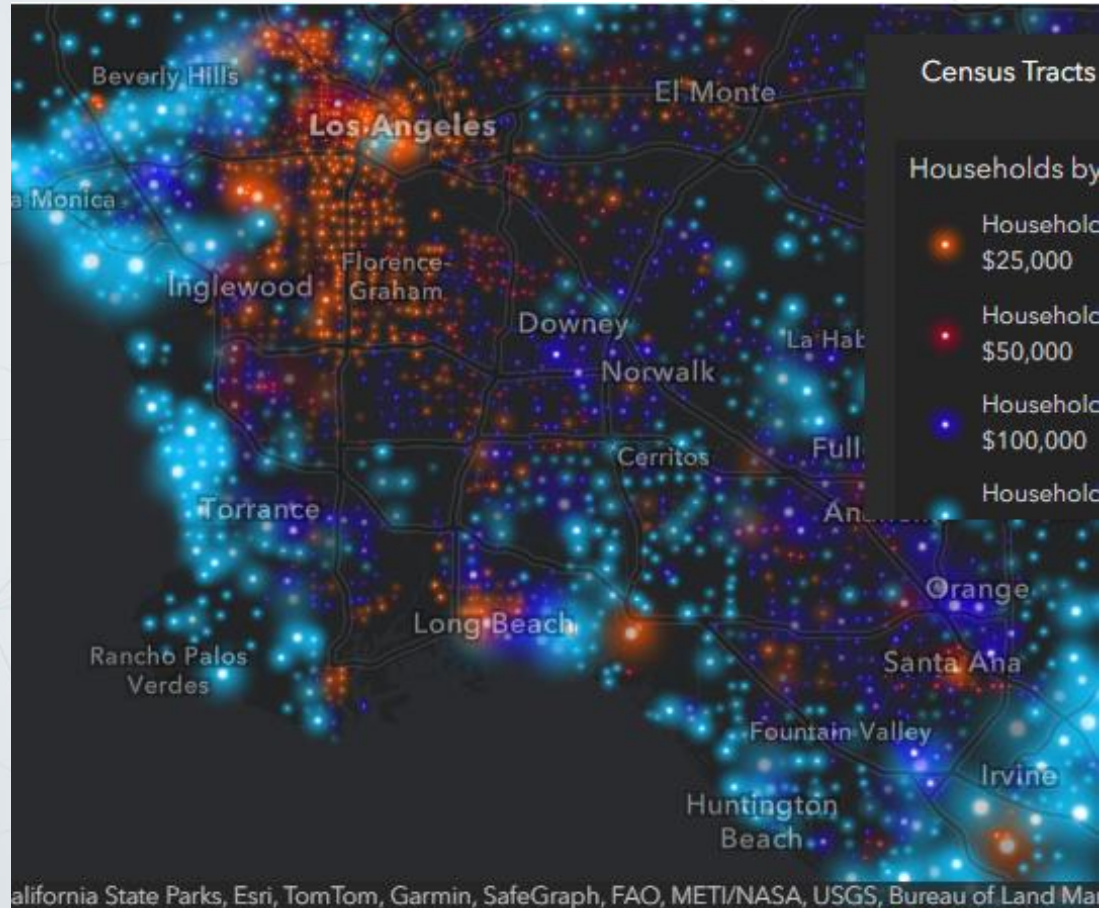




# Introduction to ArcGIS Maps SDK for JavaScript

Web Beginner

# Overview



ArcGIS Maps SDK for JavaScript คือ library สำหรับพัฒนา Web Application ที่เกี่ยวข้องกับ GIS ในรูปแบบต่างๆ เช่น แสดงแผนที่และชั้นข้อมูลต่างๆ แบบ 2D และ 3D รวมไปถึงการดึงข้อมูลจาก Map Service มาแสดงหรือประมวลผลเพิ่มเติม เป็นต้น

\* บทเรียนนี้อ้างอิงจาก ArcGIS Maps SDK for JavaScript เวอร์ชัน 5.0 \*

# Installation for Angular

ที่ Terminal พิมพ์คำสั่งต่อไปนี้เพื่อติดตั้ง @arcgis/core @arcgis/map-components และ @esri/calcite-components

```
npm install --save @arcgis/core @arcgis/map-components @esri/calcite-components
```

# Installation for Angular (ต่อ)

จากนั้น ที่ไฟล์ styles.scss (หรือ styles.css) ของ Angular project ให้ทำการแก้ไข CSS ดังนี้

```
/* Include calcite, core API and SDK component CSS */
@import "@esri/calcite-components/main.css";
@import "@arcgis/map-components/main.css";

html,
body {
  height: 100%;
  margin: 0;
}
```

# Installation for Angular (ต่อ)

จากนั้นตั้งค่าไฟล์ tsconfig.json ดังนี้

```
{
  "$schema": "https://json.schemastore.org/tsconfig.json",
  // Array of `.ts` files to compile. You can also use glob patterns such as `src/**/*.ts`.
  "include": ["src", "*.ts"],
  "compilerOptions": {
    // When `true`, this allows use of `import` syntax such as `import x from 'xyz'`.
    "esModuleInterop": true,
    // Specify library type definitions to be included in the compilation.
    "lib": ["DOM", "DOM.Iterable", "ES2023"],
    // The module system to use for compilation.
    // Here, ES modules are targeted (ESNext) to enable top-level await and dynamic imports.
    "module": "ES2022",
    // Respects package.json's "exports" conditions.
    "moduleResolution": "Bundler",
    // Allow importing from JSON files
    "resolveJsonModule": true,
    // This sets the output at the minimum version of JavaScript features that will be supported.
    "target": "ES2023",
    // Improves performance by checking only the .ts files you write
    // rather than the .d.ts files from the libraries you are using.
    "skipLibCheck": true,
  },
}
```

# Installation for Angular (ต่อ)

จากนั้นตั้งค่าไฟล์ angular.json เพื่อให้ build แล้วไฟล์ที่จำเป็นไม่หาย ดังนี้

```
{
  "architect": {
    "build": {
      "options": {
        "loader": { ".woff2": "file" },
        "assets": [
          {
            "glob": "**/*",
            "input": "node_modules/@arcgis/core/assets",
            "output": "/assets/"
          },
          {
            "glob": "**/*",
            "input": "node_modules/@esri/calcite-components/dist/cdn/assets",
            "output": "/assets/"
          }
        ],
      }
    }
  }
}
```

# การแสดงผลแผนที่

package @arcgis/map-components มาพร้อม Component แผนที่และ widget ต่างๆ สำหรับงาน GIS ให้เลือกใช้มากมาย ถ้าหากอยากเพิ่มการแสดงผลที่บน Web App คุณสามารถใช้ Component <arcgis-map> กับ Angular component ของคุณได้ตามตัวอย่างนี้

```
gis.component.ts
import { Component, CUSTOM_ELEMENTS_SCHEMA } from '@angular/core';

import '@arcgis/map-components/components/arcgis-map'; // map component

@Component({
  selector: 'app-gis',
  standalone: true,
  templateUrl: './gis.component.html',
  styleUrls: ['./gis.component.scss'],
  schemas: [CUSTOM_ELEMENTS_SCHEMA], // This is required since arcgis-map is non-Angular component
})
export class GisComponent {
  arcgisViewReadyChange(event: CustomEvent) {
    // The view is ready, add additional functionality below
    console.log('Map is ready', event);
  }
}
```

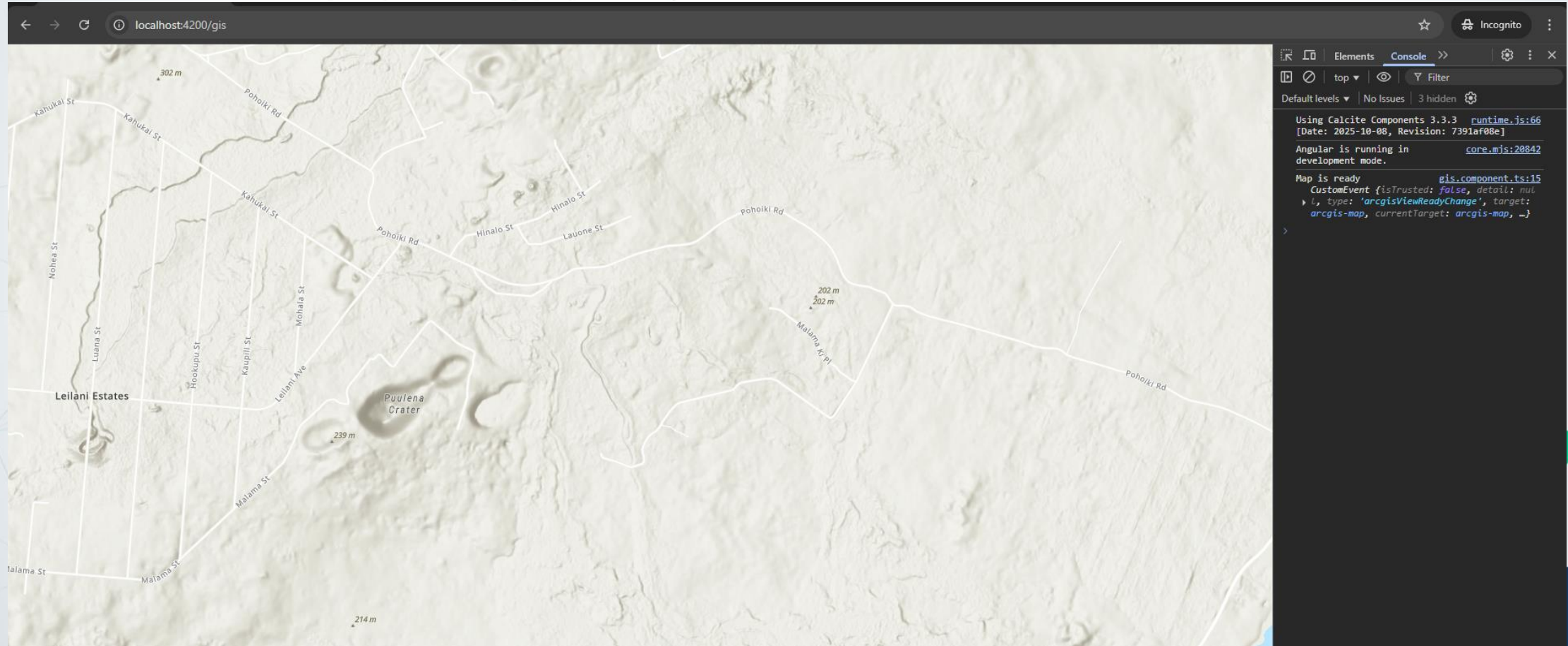
# การแสดงผลแผนที่ (ต่อ)

ที่ฝั่ง template ของ component ให้ทำการเพิ่มโค้ดต่อไปนี้เพื่อแสดงผลแผนที่ และผูก event เมื่อโหลดแผนที่เสร็จกับ method ที่ประกาศไว้

- คุณสามารถเปลี่ยน basemap, center (ใส่เป็น long,lat) และ zoom (1-20) ได้ตามชอบ
- ในกรณีที่ใช้ WebMap ที่เก็บไว้บน ArcGIS Online / Portal ให้ใส่ item-id แทน basemap

```
<arcgis-map  
  basemap="topo-vector"  
  center="-154.88, 19.46"  
  zoom="15"  
  (arcgisViewReadyChange)="arcgisViewReadyChange($event)"  
></arcgis-map>
```

# การแสดงผลแผนที่ (ต่อ)



The screenshot displays a web browser window at `localhost:4200/gis` showing a 3D topographic map. The map features a network of streets including Kahukui St, Pohoiki Rd, Hinalo St, Lauone St, Mōhala St, Lailani Ave, Luana St, Hookupu St, Kaupili St, Malama St, and Ialama St. A prominent crater, Puulena Crater, is visible in the center. Elevation markers are present, such as 302 m, 239 m, 202 m, and 214 m. The map is overlaid with a white street network.

The browser's developer console on the right shows the following logs:

```
Using Calcite Components 3.3.3 runtime.js:66  
[Date: 2025-10-08, Revision: 7391af08e]  
Angular is running in development mode. core.mjs:20842  
Map is ready gis.component.ts:15  
CustomEvent {isTrusted: false, detail: null,  
  type: 'arcgisViewReadyChange', target:  
  arcgis-map, currentTarget: arcgis-map, ...}
```

# การเข้าถึง Component <arcgis-map>

คุณสามารถเข้าถึง component <arcgis-map> เพื่อกระทำการใดๆ ที่ฝั่ง TypeScript (เช่น เพิ่มชั้นข้อมูล, เพิ่มกราฟิก มุมบนแผนที่, pan ไปยังจุดใดๆ, ฯลฯ) โดยการเข้าถึงจาก event.target ที่มาจาก event handler หลังโหลดแผนที่เสร็จ

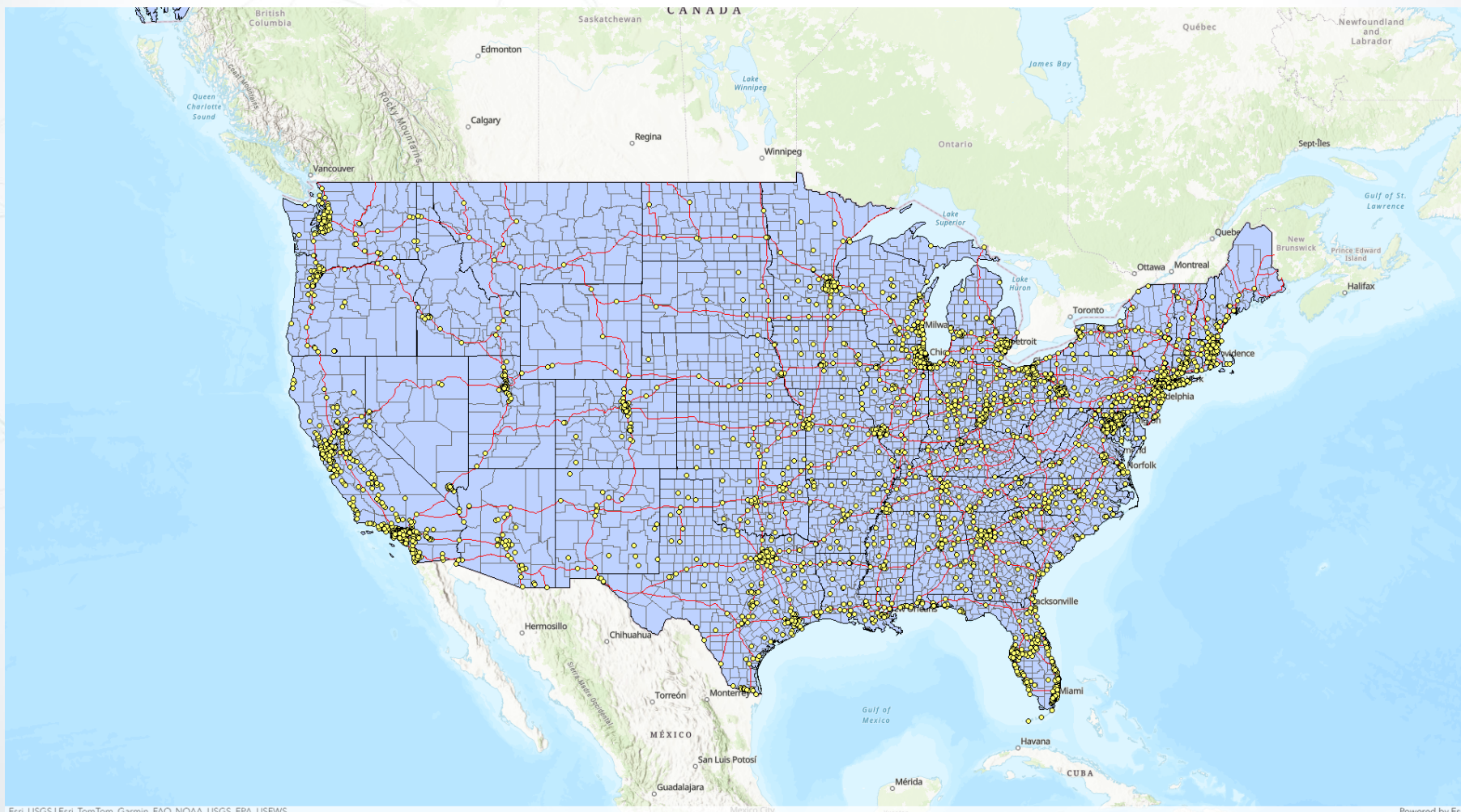
```
import { Component, CUSTOM_ELEMENTS_SCHEMA } from '@angular/core';  
  
import '@arcgis/map-components/components/arcgis-map';  
import { ArcgisMap } from '@arcgis/map-components/components/arcgis-map';  
  
@Component({  
  selector: 'app-gis',  
  standalone: true,  
  templateUrl: './gis.component.html',  
  styleUrls: ['./gis.component.scss'],  
  schemas: [CUSTOM_ELEMENTS_SCHEMA], // Set the schema here  
})  
export class GisComponent {  
  mapComponent!: ArcgisMap;  
  
  arcgisViewReadyChange(event: CustomEvent) {  
    // The view is ready, add additional functionality below  
    console.log('Map is ready');  
    this.mapComponent = event.target as ArcgisMap;  
    // Pan map to specified [lon,lat]  
    this.mapComponent.view.goTo([100, 13]);  
  }  
}
```

# การเพิ่มชั้นข้อมูล

คุณสามารถเพิ่มชั้นข้อมูลเพื่อแสดงบนแผนที่ โดยการสร้าง instance ของ map layer ชนิดใดๆ (เช่น MapImageLayer, FeatureLayer, TileLayer, etc.) จากนั้นเพิ่มลงไปยังแผนที่โดยการเรียกใช้ method ArcgisMap.map.add(layer)

```
gis.component.ts
...
import MapImageLayer from '@arcgis/core/layers/MapImageLayer';
...
export class GisComponent {
  mapComponent!: ArcgisMap;
  ...
  arcgisViewReadyChange(event: CustomEvent) {
    ...
    // Add USA Layer
    const USALayer = new MapImageLayer({
      url: 'https://sampleserver6.arcgisonline.com/arcgis/rest/services/USA/MapServer',
    });
    this.mapComponent.map.add(USALayer);
  }
}
```

# การเพิ่มชั้นข้อมูล (ต่อ)



# ประเภทของชั้นข้อมูลที่ใช้ง่าย

Class	คำอธิบาย	ตัวอย่าง URL
MapImageLayer	ชั้นข้อมูลขนาดใหญ่ที่ประกอบด้วยชั้นข้อมูลย่อย (sublayer) หลายอัน	<a href="https://sampleserver6.arcgis.com/arcgis/rest/services/Census/MapServer">https://sampleserver6.arcgis.com/arcgis/rest/services/Census/MapServer</a>
FeatureLayer	ชั้นข้อมูลที่ทำหน้าที่เป็นตารางข้อมูล สามารถ query ข้อมูลมาใช้งานได้	<a href="https://sampleserver6.arcgis.com/arcgis/rest/services/USA/MapServer/2">https://sampleserver6.arcgis.com/arcgis/rest/services/USA/MapServer/2</a>
TileLayer	ชั้นข้อมูลที่ประกอบด้วย Tile image เป็นช่องๆ เหมาะสำหรับการแสดงผล หรือใช้เป็น Basemap	<a href="https://services.arcgis.com/arcgis/rest/services/World_Terrain_Base/MapServer">https://services.arcgis.com/arcgis/rest/services/World_Terrain_Base/MapServer</a>
GraphicsLayer	ชั้นข้อมูลที่เก็บ Graphic เหมาะสำหรับการวาดเขียนต่างๆ เพื่อไม่ให้ปะปนกับชั้นข้อมูลอื่น	-

# การเพิ่มกราฟิก

- หากต้องการเพิ่มกราฟิกต่างๆ เช่น หมุดบนแผนที่ เส้นทาง พื้นที่อาคาร ฯลฯ ให้ทำการสร้าง instance ของคลาส Graphic โดยภายใน instance จะประกอบด้วย symbol ซึ่งเป็นหน้าตาของสัญลักษณ์ที่จะใช้เป็นกราฟิกบนแผนที่ และ geometry คือข้อมูลพิกัดที่จะแสดงบนแผนที่ นอกจากนี้อาจเพิ่มฟิลด์ attributes เพื่อแนบข้อมูลอื่นๆ (เช่น ที่อยู่, หมายเลขโทรศัพท์, อีเมล, ฯลฯ) สำหรับใช้ร่วมกับ Popup ก็ได้
- เมื่อได้ instance ของ Graphic แล้ว ทำการเพิ่มลงไปบนแผนที่โดยใช้ method `ArcgisMap.view.graphics.add(graphic)` (เพิ่มทีละอัน) หรือ `ArcgisMap.view.graphics.addMany([graphic1, graphic2, ...])` (เพิ่มหลายอันพร้อมกัน)
- คุณสามารถลบกราฟิกที่ไม่ต้องการแล้วออกจากแผนที่ โดยการเรียก method `ArcgisMap.view.graphics.remove(graphic)` (ลบทีละอัน) หรือ `ArcgisMap.view.graphics.removeMany([graphic1, graphic2, ...])` (ลบหลายอัน) หรือ `ArcgisMap.view.graphics.removeAll()` (ลบทั้งหมด)
- เพื่อความเป็นระเบียบ และจัดการง่าย ขอแนะนำให้สร้าง instance ของคลาส GraphicsLayer แล้ว add layer นี้ไปที่ `ArcgisMap.map` จากนั้นนำกราฟิกที่สร้างมา add เข้าที่ GraphicsLayer แทน

# การเพิ่มกราฟิก (ต่อ)

gis.component.ts

```
...
import Graphic from '@arcgis/core/Graphic';
import Point from '@arcgis/core/geometry/Point';
import SimpleMarkerSymbol from '@arcgis/core/symbols/SimpleMarkerSymbol';

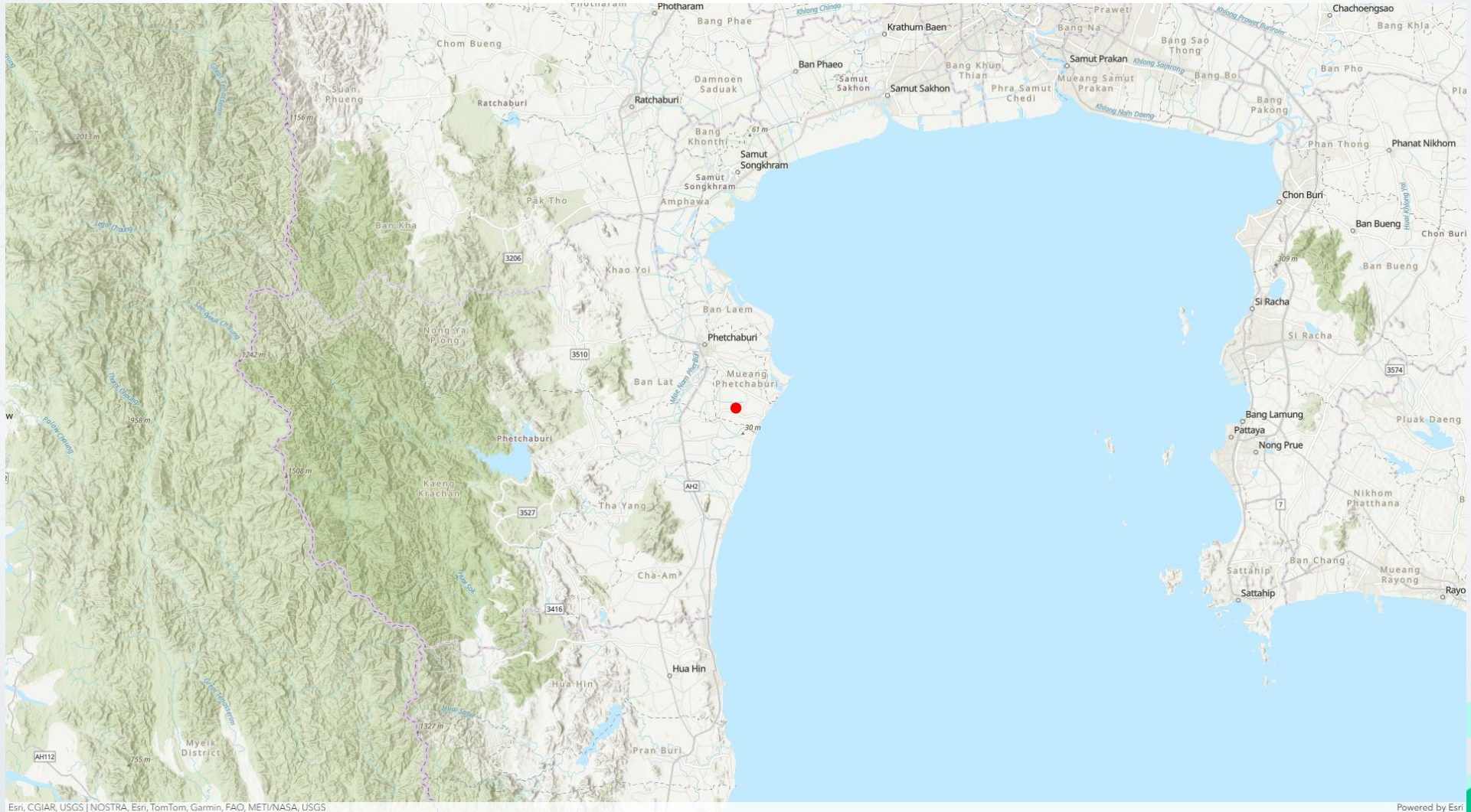
...
export class GisComponent {
  ...
  arcgisViewReadyChange(event: CustomEvent) {
    ...
    // Add point graphic
    const point = new Point({
      longitude: 100,
      latitude: 13,
    });

    const marker = new SimpleMarkerSymbol({
      color: [255, 0, 0],
      outline: {
        color: 'white',
        width: 1,
      },
    });

    const pointGraphic = new Graphic({
      geometry: point,
      symbol: marker,
      attributes: {
        title: 'My Point',
      },
    });

    this.mapComponent.view.graphics.add(pointGraphic);
  }
}
```

# การเพิ่มกราฟิก (ต่อ)



# การเลือกใช้ Geometry และ Symbol

Geometry	Symbol	คำอธิบาย	ตัวอย่างข้อมูล
Point	SimpleMarkerSymbol	จุดบนแผนที่ ประกอบด้วยค่า longitude/latitude หรือ x/y เหมาะสำหรับแสดงสถานที่สำคัญ, ตำแหน่งของวัตถุ, ฯลฯ	<pre>{   "longitude": 100,   "latitude": 13 }</pre>
Polyline	SimpleLineSymbol	เส้นตรงอย่างน้อย 1 เส้น ประกอบด้วยค่า paths ซึ่งเป็น array เก็บจุดปลายของเส้นตรงแต่ละเส้น เหมาะสำหรับแสดงเส้นทาง, ถนน, ท่อประปา, สายไฟฟ้า, ฯลฯ	<pre>{   "paths": [     [-111.3, 52.68], [[-98, 49.5],     [-93.94, 29.89]   ] }</pre>
Polygon	SimpleFillSymbol	รูปหลายเหลี่ยม ประกอบด้วยค่า rings ซึ่งเป็น array เก็บจุดทุกจุดที่อยู่บนเส้นขอบของรูปหลายเหลี่ยม เหมาะสำหรับพื้นที่ปิด เช่น อาคาร, ที่ดิน, แหล่งน้ำ, ฯลฯ	<pre>{   "rings": [     [-64.78, 32.3], [-66.07, 18.45],     [-80.21, 25.78], [-64.78, 32.3]   ] }</pre>

# การเพิ่ม Event Handler

กรณีที่คุณต้องการเพิ่ม event handler เมื่อทำการใดๆ กับแผนที่ เช่น เมื่อคลิกบนจุดใดๆ บนแผนที่ คุณสามารถทำได้ โดยการใช้ method `ArcgisMap.view.on("event-name", function)` ดังตัวอย่างต่อไปนี้

คุณสามารถนำจุดที่คลิกไปใช้งานต่อได้ (เช่น pan map โดยใช้ `ArcgisMap.view.goTo(point)` เป็นต้น) โดยการเข้าถึง ฟิลด์ `event.mapPoint` ซึ่งจะมี longitude/latitude และ x,yให้นำไปใช้งาน

```
...  
export class GisComponent {  
  mapComponent!: ArcgisMap;  
  ...  
  arcgisViewReadyChange(event: CustomEvent) {  
    ...  
    // Add event on click  
    this.mapComponent.view.on('click', (event) => {  
      console.log('Clicked', event);  
      this.mapComponent.view.goTo(event.mapPoint);  
    });  
  }  
}
```

gis.component.ts

# การเพิ่ม Map Widget ต่างๆ

นอกจาก `<arcgis-map>` แล้ว ใน package `@arcgis/map-components` ยังมี component อื่นๆ ที่เป็น map widgetให้นำมาใช้งานด้วย เช่น ปุ่มย่อ-ขยายแผนที่ ปุ่มแสดงพิกัดปัจจุบัน หน้าต่างเปิด-ปิดชั้นข้อมูล ฯลฯ โดยสามารถใช้งานได้โดยการ import เข้ามาที่ฝั่ง TypeScript จากนั้นเพิ่ม tag ของ component นั้นใน template โดยเพิ่มไว้เป็น child node ของ `<arcgis-map>` นั้น และระบุตำแหน่งของ widget บนแผนที่ด้วย attribute `slot`

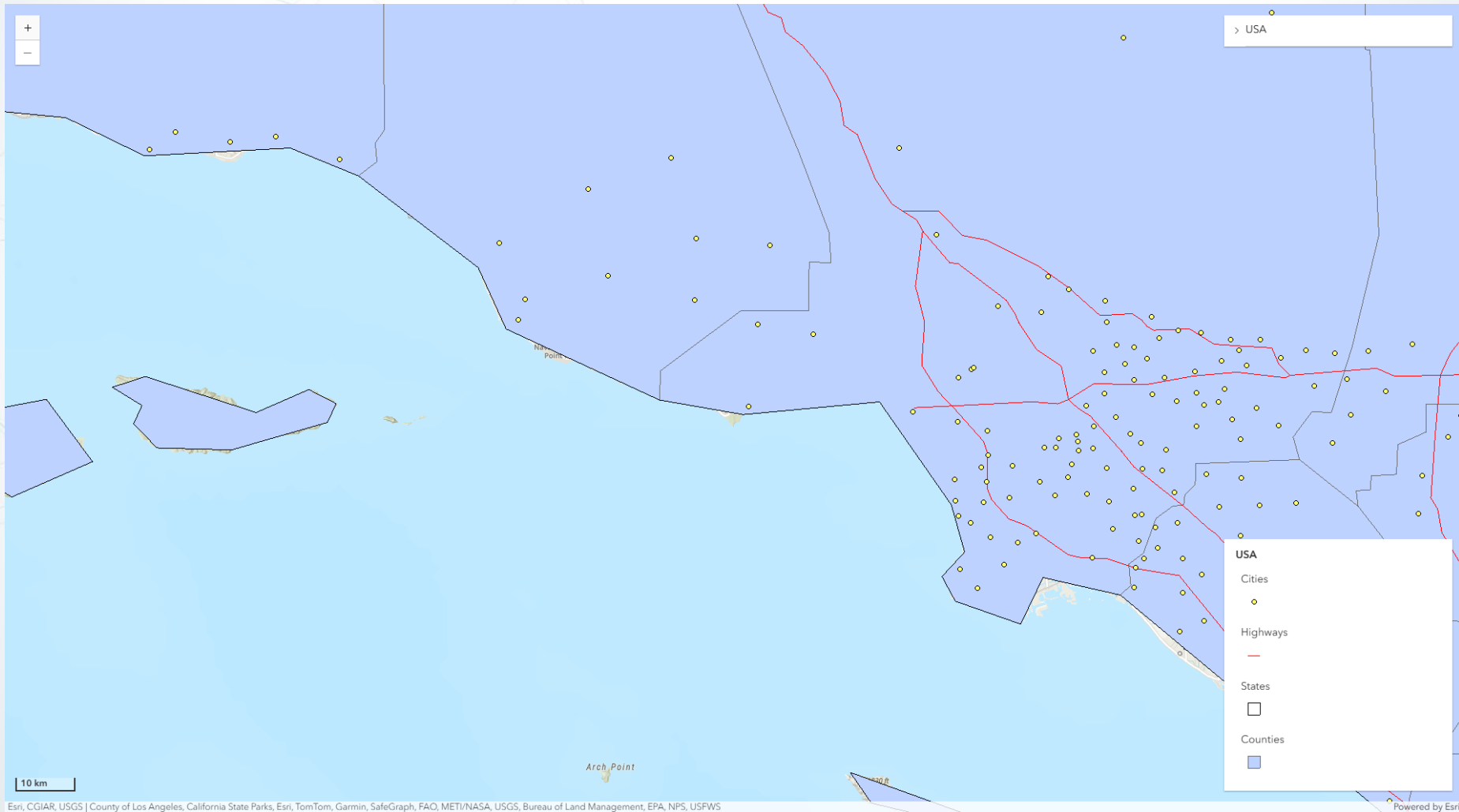
```
...
import '@arcgis/map-components/components/arcgis-zoom';
import '@arcgis/map-components/components/arcgis-layer-list';
import '@arcgis/map-components/components/arcgis-legend';
import '@arcgis/map-components/components/arcgis-scale-bar';
...
export class GisComponent {
  ...
}
```

gis.component.ts

```
<arcgis-map basemap="topo-vector" center="-118.805, 34.027" zoom="10" (arcgisViewReadyChange)="arcgisViewReadyChange($event)">
  <arcgis-zoom></arcgis-zoom>
  <arcgis-layer-list slot="top-right"></arcgis-layer-list>
  <arcgis-legend slot="bottom-right"></arcgis-legend>
  <arcgis-scale-bar slot="bottom-left"></arcgis-scale-bar>
</arcgis-map>
```

gis.component.html

# การเพิ่ม Map Widget ต่างๆ (ต่อ)



# Reference

- <https://developers.arcgis.com/javascript/latest/>

***End of Presentation***