



# Introduction to ArcGIS Maps SDK for JavaScript

Web Beginner Course

# ArcGIS Maps SDK for JavaScript

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 4.26

# การติดตั้ง ArcGIS Maps SDK บน Angular project

ติดตั้ง ArcGIS Maps SDK ผ่าน npm ด้วยคำสั่งต่อไปนี้

```
npm install --save @arcgis/core
```

# การติดตั้ง ArcGIS Maps SDK บน Angular project (ต่อ)

จากนั้น เพิ่ม stylesheet ที่มากับ SDK (มี theme light และ dark ให้เลือกใช้) ไปที่ไฟล์ angular.json  
ภายใต้ section projects > architect > build > options > styles

```
{
  ...
  "projects": {
    "architect": {
      "build": {
        ...
        "options": {
          "styles": [
            ...
            "node_modules/@arcgis/core/assets/esri/themes/light/main.css"
          ]
        }
      }
    }
  }
  ...
}
```

# การแสดงผลที่

ในการแสดงผลที่บนหน้าเว็บ จำเป็นต้อง import class Map และ MapView เข้ามาที่ Component ที่ต้องการใช้งานก่อน

เนื่องจาก MapView ต้องการ container ที่เป็น HTML element จึงจำเป็นต้อง import ViewChild และ ElementRef ของ Angular มาใช้เพื่ออ้างอิงถึง HTML element ในหน้า template ด้วย แต่มีข้อควรระวังอีกอย่างคือ เนื่องจาก HTML element ที่มาจาก ViewChild จะสามารถเข้าถึงได้เมื่อเพิ่งทำการโหลดเรียบร้อยแล้ว จึงแนะนำให้ implement OnInit และทำการ initialize map ภายใน ngOnInit

```
import { Component, OnInit, ViewChild, ElementRef } from '@angular/core';
import Map from '@arcgis/core/Map';
import MapView from '@arcgis/core/views/MapView';

@Component({
  selector: 'map',
  templateUrl: './map.component.html',
  styleUrls: ['./map.component.scss'],
})
...
```

# การแสดงผลแผนที่ (ต่อ)

```
...
export class MapComponent implements OnInit {
  // สร้าง ViewChild อ้างอิงถึง #mapPanel ที่เราสร้างไว้ใน template
  @ViewChild('mapPanel', { static: true }) mapPanel!: ElementRef;

  map: Map | null = null;
  mapView: MapView | null = null;

  ngOnInit() {
    // สร้าง instance ของ Map โดยเราสามารถกำหนด basemap รวมถึงเพิ่ม layer ต่างๆ ได้ที่ตรงนี้
    this.map = new Map({
      basemap: 'topo-vector',
    });

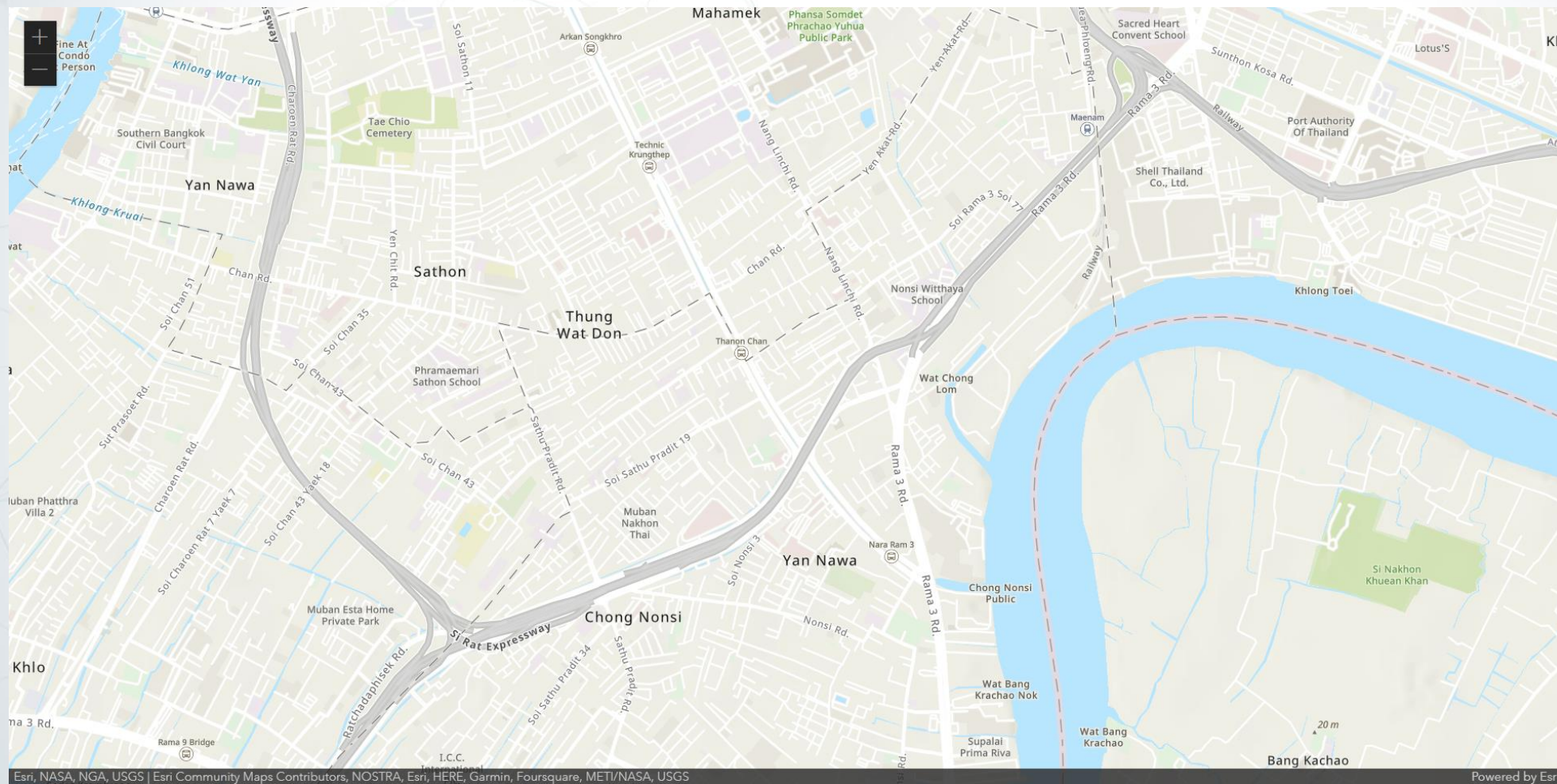
    // สร้าง instance ของ MapView ซึ่งเป็นส่วนที่ใช้ render map บนหน้าเพจ
    this.mapView = new MapView({
      container: this.mapPanel.nativeElement,
      map: this.map,
      center: [100.5408754, 13.7030248], // [longitude, latitude]
      zoom: 15, // zoom level
    });
  }
}
```

# การแสดงผลแผนที่ (ต่อ)

ที่ Template ให้สร้าง <div> ที่มี # ตามด้วยชื่อ reference ไว้เพื่อเป็น container element สำหรับ render map  
ทั้งนี้ อย่าลืมเซตความกว้างและความสูงของ container ให้มีขนาดพอดีกับหน้าจอหรือพื้นที่ที่ต้องการแสดงผลด้วย

```
<div #mapPanel style="width: 100%; height: 100%"></div>
```

# การแสดงผลแผนที่ (ต่อ)



# การแสดงกราฟิกบนแผนที่

ในกรณีที่ต้องการสร้างกราฟิกต่างๆ เช่น หมุด เครื่องหมาย ฯลฯ บนแผนที่ สามารถทำได้โดยการใช้ class Graphic โดยชิ้นส่วนสำคัญในการสร้างกราฟิกบนแผนที่ประกอบด้วย geometry (ตำแหน่งและรูปร่างทางภูมิศาสตร์) กับ symbol (สีและเส้นของสัญลักษณ์ที่จะใช้)

ตัวอย่างต่อไปนี้ จะสร้างหมุดรูปวงกลมแบบง่ายๆ แปะลงแผนที่ จึงเลือกใช้ geometry แบบ Point (จุดพิกัด 1 จุด) รวมกับ SimpleMarkerSymbol

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

# การแสดงกราฟิกบนแผนที่ (ต่อ)

อันดับแรก เราจะทำการสร้าง instance ของ geometry เพื่อเก็บค่าพิกัดภูมิศาสตร์ของ Graphic ที่ต้องการก่อน โดยในที่นี้จะใช้เป็น Point ซึ่งต้องการ longitude และ latitude ในการสร้าง

```
const point = new Point({  
  longitude: 100.5408754,  
  latitude: 13.7030248,  
});
```

จากนั้น เราจะทำการสร้าง instance ของ SimpleMarkerSymbol ซึ่งจะเป็นตัวกำหนดหน้าตาของหมุดที่จะแสดงบนแผนที่

```
const marker = new SimpleMarkerSymbol({  
  color: [226, 119, 40],  
  outline: {  
    color: [255, 255, 255],  
    width: 2,  
  },  
});
```

# การแสดงกราฟิกบนแผนที่ (ต่อ)

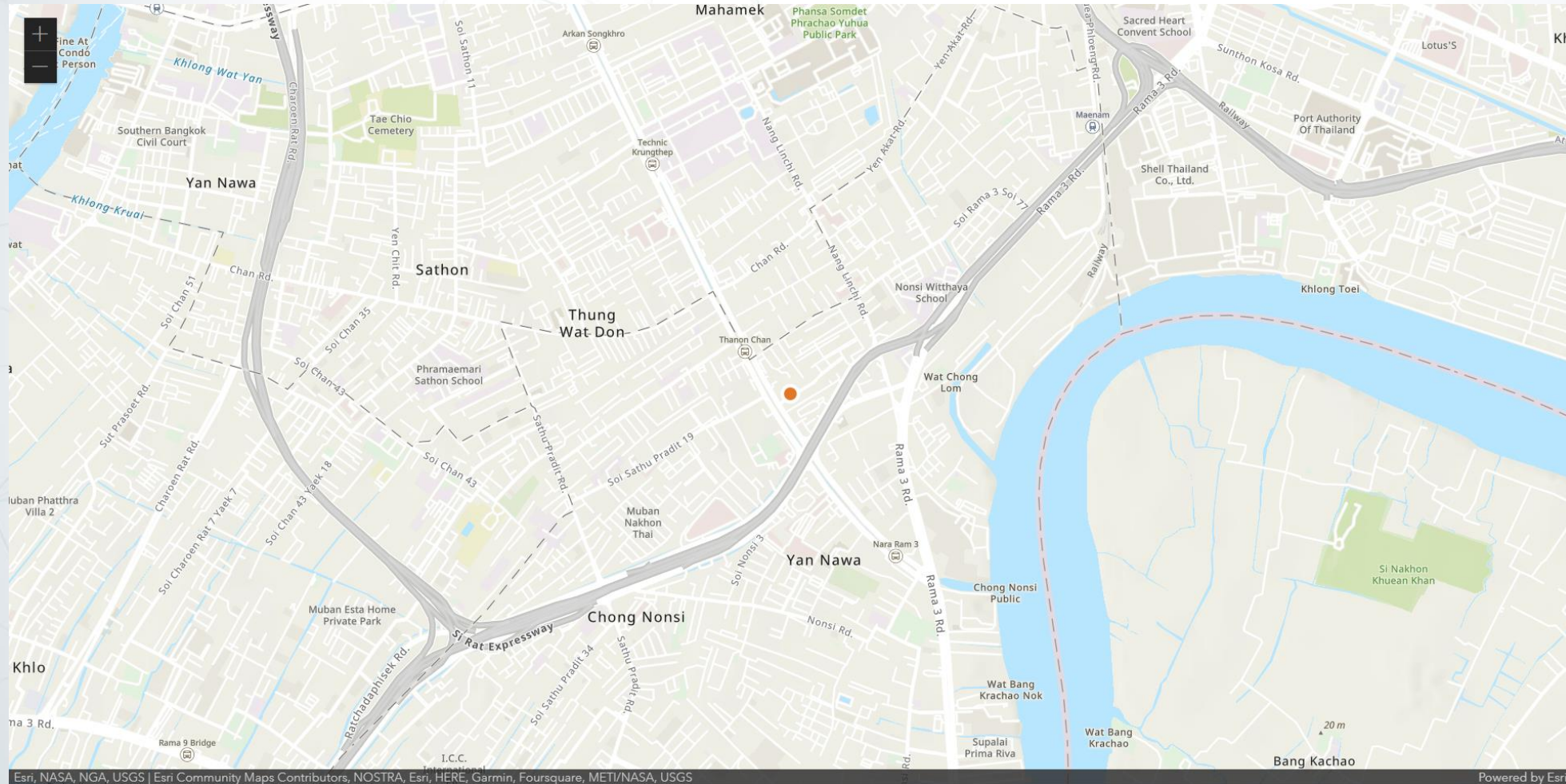
นำ geometry และ symbol ที่สร้างไว้ มาสร้าง instance ของ Graphic

```
const pointGraphic = new Graphic({  
  geometry: point,  
  symbol: marker,  
});
```

สุดท้าย ทำการ add ลงไปที่แผนที่โดยการเรียก MapView.graphics.add()

```
this.mapView.graphics.add(pointGraphic);
```

# การแสดงกราฟิกบนแผนที่ (ต่อ)



# ชนิดของ Geometry

Geometry	คำอธิบาย	ตัวอย่าง
Point	geometry แบบจุดหนึ่งจุดบนแผนที่ ประกอบด้วยข้อมูลพิกัด x, y หรือ longitude, latitude	<pre>{   longitude: 100.5408754,   latitude: 13.7030248, }</pre>
Polygon	geometry แบบพื้นที่ปิดล้อมเป็นรูปหลายเหลี่ยม เช่น ที่ดิน อาคาร หรือขอบเขตต่างๆ ประกอบด้วย rings ซึ่งเป็น Array ของตัวเลขพิกัดทั้งหมดที่อยู่บนเส้นขอบของรูปหลายเหลี่ยม	<pre>{   rings: [     [-64.78, 32.3], [-66.07, 18.45],     [-80.21, 25.78], [-64.78, 32.3]   ] }</pre>
Polyline	geometry ของพิกัดที่เรียงกันเป็นเส้นตรง ตั้งแต่ 1 เส้นขึ้นไป เช่น ถนน ทางรถไฟ แนวท่อก๊าซ ประกอบด้วย paths ซึ่งเป็น Array สำหรับเก็บพิกัดของจุดต่างๆ บนแนวเส้น	<pre>{   paths: [     [-111.3, 52.68], [-98, 49.5],     [-93.94, 29.89]   ] }</pre>

# การเพิ่มชั้นข้อมูลบนแผนที่

ในกรณีที่มีชั้นข้อมูลในรูปแบบของ ArcGIS Map Service ต่างๆ เราสามารถนำ URL ของชั้นข้อมูลเหล่านั้นมาใช้งานร่วมกับ Maps SDK เพื่อแสดงชั้นข้อมูลนั้นบน Web App ที่เราพัฒนาขึ้นได้ โดยการใช้ความสามารถของ class ที่เกี่ยวข้องกับชั้นข้อมูล เช่น MapImageLayer, FeatureLayer, TileLayer เป็นต้น

ในการที่จะทราบว่าชั้นข้อมูลที่ต้องการใช้งานเป็นชนิดใด ให้สังเกตจาก URL และ Description ของ Map Service นั้นๆ

```
import MapImageLayer from "@arcgis/core/layers/MapImageLayer";
```

# การเพิ่มชั้นข้อมูลบนแผนที่ (ต่อ)

สมมุติว่าเราต้องการนำชั้นข้อมูลของ USA

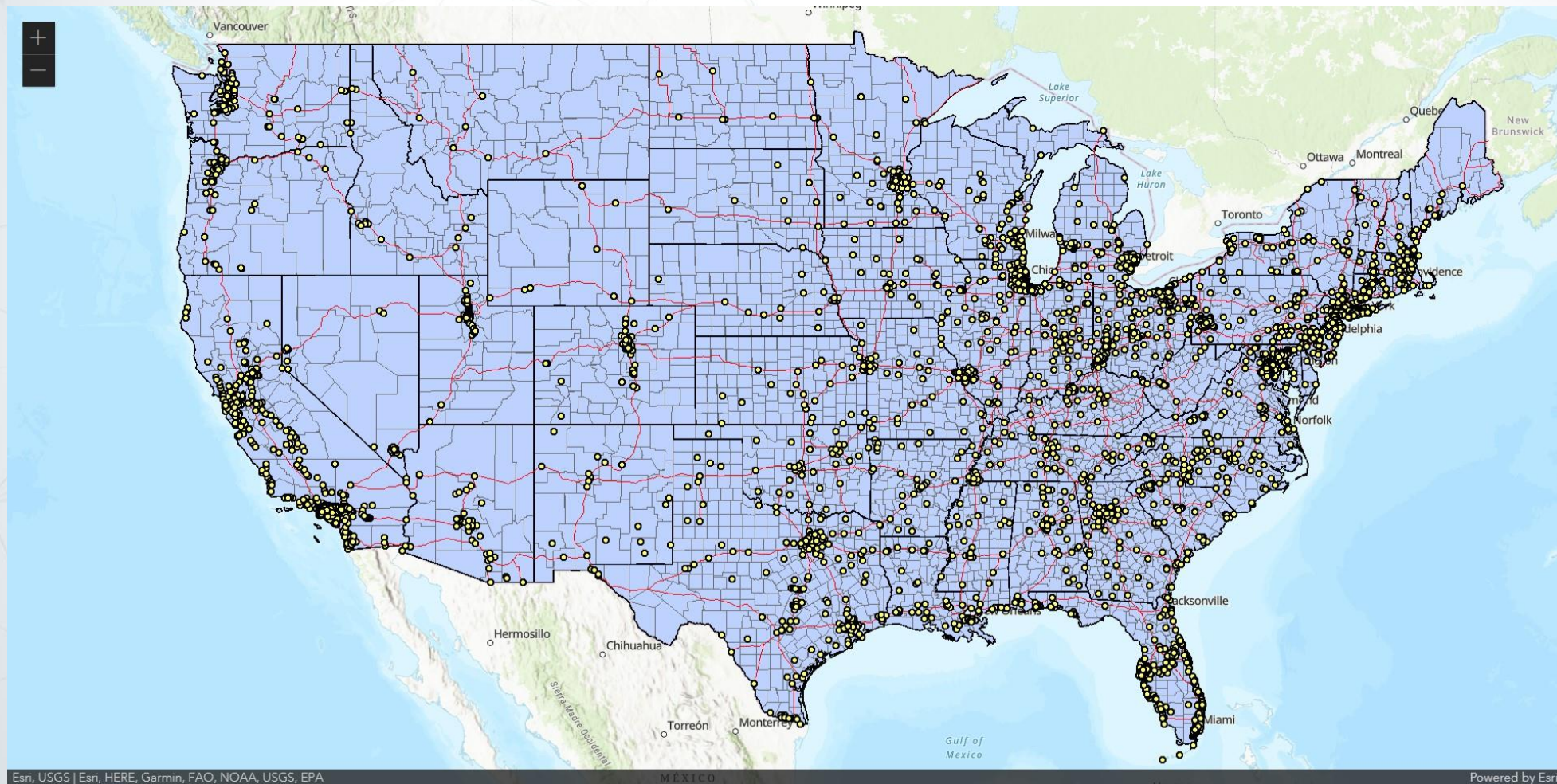
<https://sampleserver6.arcgisonline.com/arcgis/rest/services/USA/MapServer/> เพื่อแสดงกราฟิกของชั้นข้อมูลทุกชั้น ให้เราทำการสร้าง instance ของ MapImageLayer และระบุ URL ของ Service ลงไป

```
const layer = new MapImageLayer({  
  url: 'https://sampleserver6.arcgisonline.com/arcgis/rest/services/USA/MapServer/',  
});
```

จากนั้น ทำการเพิ่มลงไป instance ของ Map ที่เราสร้างไว้

```
this.map.add(layer);
```

# การเพิ่มชั้นข้อมูลบนแผนที่ (ต่อ)



# ชนิดของชั้นข้อมูล

ชนิดของชั้นข้อมูล	คำอธิบาย	ตัวอย่าง URL
MapImageLayer	ชั้นข้อมูลที่ประกอบด้วย sublayer หลายอัน ลงท้ายด้วย /MapServer	<a href="https://sampleserver6.arcgisonline.com/arcgis/rest/services/USA/MapServer">https://sampleserver6.arcgisonline.com/arcgis/rest/services/USA/MapServer</a>
TileLayer	ชั้นข้อมูลที่ประกอบด้วย cache tile image หรือแผนที่ที่ตัดแบ่งเป็นช่องสี่เหลี่ยมหลายๆ ช่อง ลงท้ายด้วย /MapServer เหมือนกัน แต่รายละเอียดจะต่างกัน	<a href="https://services.arcgisonline.com/arcgis/rest/services/World_Terrain_Base/MapServer">https://services.arcgisonline.com/arcgis/rest/services/World_Terrain_Base/MapServer</a>
FeatureLayer	ชั้นข้อมูล sublayer ที่ทำหน้าที่เป็น ตารางข้อมูลให้ query ค่าออกมาได้ด้วย ลงท้ายด้วย /MapServer/<ตัวเลข>	<a href="https://sampleserver6.arcgisonline.com/arcgis/rest/services/USA/MapServer/0">https://sampleserver6.arcgisonline.com/arcgis/rest/services/USA/MapServer/0</a>

# Event Handler

ในกรณีที่เราต้องการเพิ่ม Event Handler เพื่อรองรับ action บางอย่าง เช่น เมื่อคลิกไปบนแผนที่ เราสามารถทำได้ ที่ instance ของ MapView ซึ่งมี method on() ให้ใช้งาน โดยข้อมูลต่างๆ ที่เกี่ยวข้องกับ action ที่ทำลงไป เช่น ตำแหน่งที่คลิกลงไป จะถูกส่งเข้ามาอยู่ในตัวแปร event ซึ่งเราสามารถนำไปใช้ต่อได้ใน handler function ที่เราจะเขียนเอง

```
this.mapView.on('click', (event) => { console.log(event) });
```

# References

- <https://developers.arcgis.com/javascript/latest/get-started/>
- <https://developers.arcgis.com/javascript/latest/api-reference/esri-Map.html>
- <https://developers.arcgis.com/javascript/latest/api-reference/esri-views-MapView.html>
- <https://developers.arcgis.com/javascript/latest/sample-code/intro-graphics/>
- <https://developers.arcgis.com/javascript/latest/api-reference/esri-Graphic.html>
- <https://developers.arcgis.com/javascript/latest/api-reference/esri-geometry-Point.html>
- <https://developers.arcgis.com/javascript/latest/api-reference/esri-symbols-SimpleMarkerSymbol.html>
- <https://developers.arcgis.com/javascript/latest/sample-code/intro-layers/>
- <https://developers.arcgis.com/javascript/latest/api-reference/esri-layers-MapImageLayer.html>

***End of Chapter***