

# ArcGIS API for JavaScript 4.x

Safouane JGUIRIM et Fabrice GRATALOUP



@safjg



@fgrataloup



# Plan de la présentation

- Introduction
- Nouveautés
- Migrer de l'API JS Google Maps vers l'ArcGIS API for JS
- Fondamentaux ArcGIS API for JS 4.x
  - Layers
  - Pattern
- ArcGIS API for JS et Arcade
- Widgets
- Roadmap

# Introduction

# API JS 4.x

## Introduction

**Une nouvelle API 4.x repensée :**

- **Développement en 2D ou 3D**
- **L'API 3.x est issue d'une architecture de 2008**
- **Plus simple à coder, plus homogène**
- **Accès aux ressources de la plateforme ArcGIS**
- **Amélioration de l'UX**

# API JS 4.x

## Introduction

**4 mises à jour par an :**

**Mars 2017 : 4.3 / 3.20**

**Juillet 2017 : 4.4 / 3.21**

**Septembre 2017 : 4.5 / 3.22**

**Décembre 2017 : 4.6 / 3.24**

**Avril 2018 : 4.7 et 3.24**

**Juillet 2018 : 4.8 et 3.25**

**Septembre 2018 : 4.9 et 3.26**

# Nouveautés

# API JS 4.x

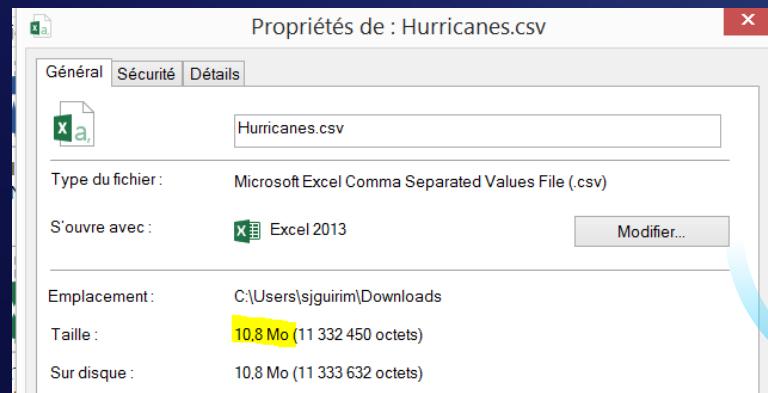
## Nouveautés

Nouveautés de la 4.8	Nouveautés de la 4.9
<p><a href="#"><b>Labeling</b></a> Large point datasets in a webscene</p> <p><a href="#"><b>Heatmap renderer</b></a> Custom background in a webscene Underground navigation and visualization</p> <p><a href="#"><b>FeatureLayer performance improvements</b></a></p> <p><a href="#"><b>Full client-side support for LayerView queries</b></a></p> <p>Widget updates</p> <ul style="list-style-type: none"><li>Bookmarks widget</li><li><a href="#"><b>Sketch updates</b></a></li><li>CSS widget updates</li></ul> <p>Button styling</p> <p>Better WebMap Support</p> <p>Request Modification</p> <p>Promote secured services</p>	<p><a href="#"><b>WebGL enabled by default in 2D</b></a> <a href="#"><b>Gamepad and 3D mouse navigation</b></a> FeatureForm widget (Beta release)</p> <p><a href="#"><b>Feature Layer updates</b></a> Highlight support when searching SceneView Take a screenshot of the SceneView Smart Mapping updates</p> <p><a href="#"><b>StreamLayer updates</b></a> CORS (Cross Origin Resource Sharing) updates Support for AbortSignal URL upgrade behavior</p> <p>Widget updates</p> <ul style="list-style-type: none"><li>BasemapGallery widget</li><li><a href="#"><b>Sketch updates</b></a></li><li>Feature widget sample</li><li>Search widget documentation</li></ul>

# API JS 4.x

## WebGL-Powered Feature Layers

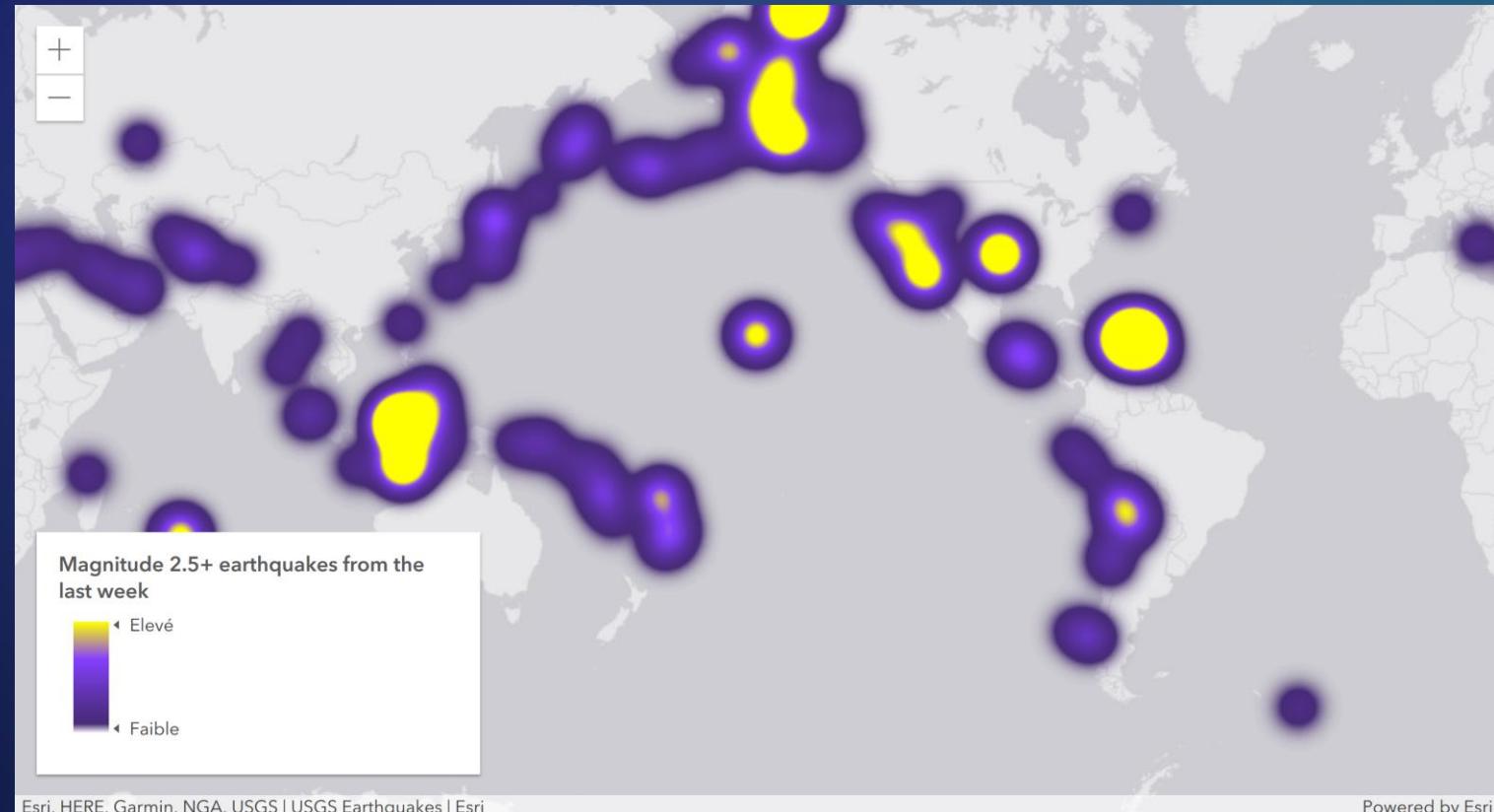
- A partir de la version 4.9, **WebGL est activé par défaut** à la place de SVG dans la vue 2D
- WebGL permet aux développeurs d'ajouter des couches d'entités ponctuelles avec des **centaines de milliers d'objets**.
- Explorer des larges volumes de données côté client, offre des nouvelles possibilités (Changer dynamiquement l'affichage des couches côté client, des requêtes plus efficaces, effectuer des traitements côté client ...)



# API JS 4.x

## HeatMapRenderer

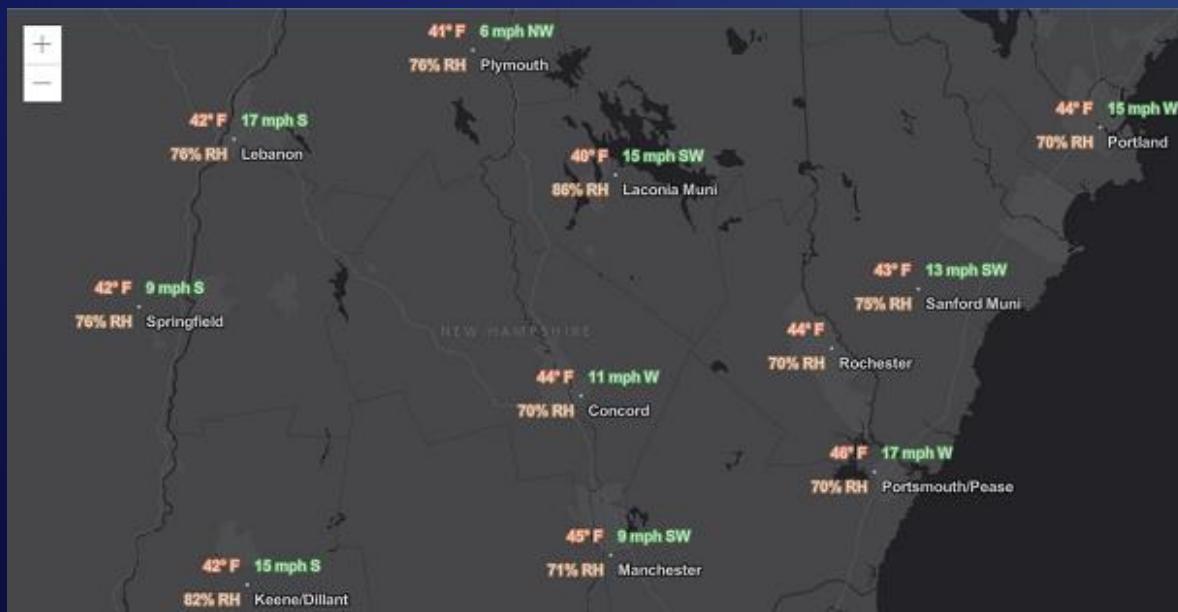
A partir de la version 4.8, prise en charge de [HeatmapRenderer](#), qui vous permet de visualiser des ensembles de données de points volumineux et denses en tant que surface raster afin de mettre en évidence les **zones à forte densité d'entités**.



# API JS 4.x

## Labeling

**Etiquetage de couches WebGL** est maintenant pris en charge pour les couches de point et polygon de type FeatureLayers et CSVLayers dans les MapViews 2D.



Multi-line labels

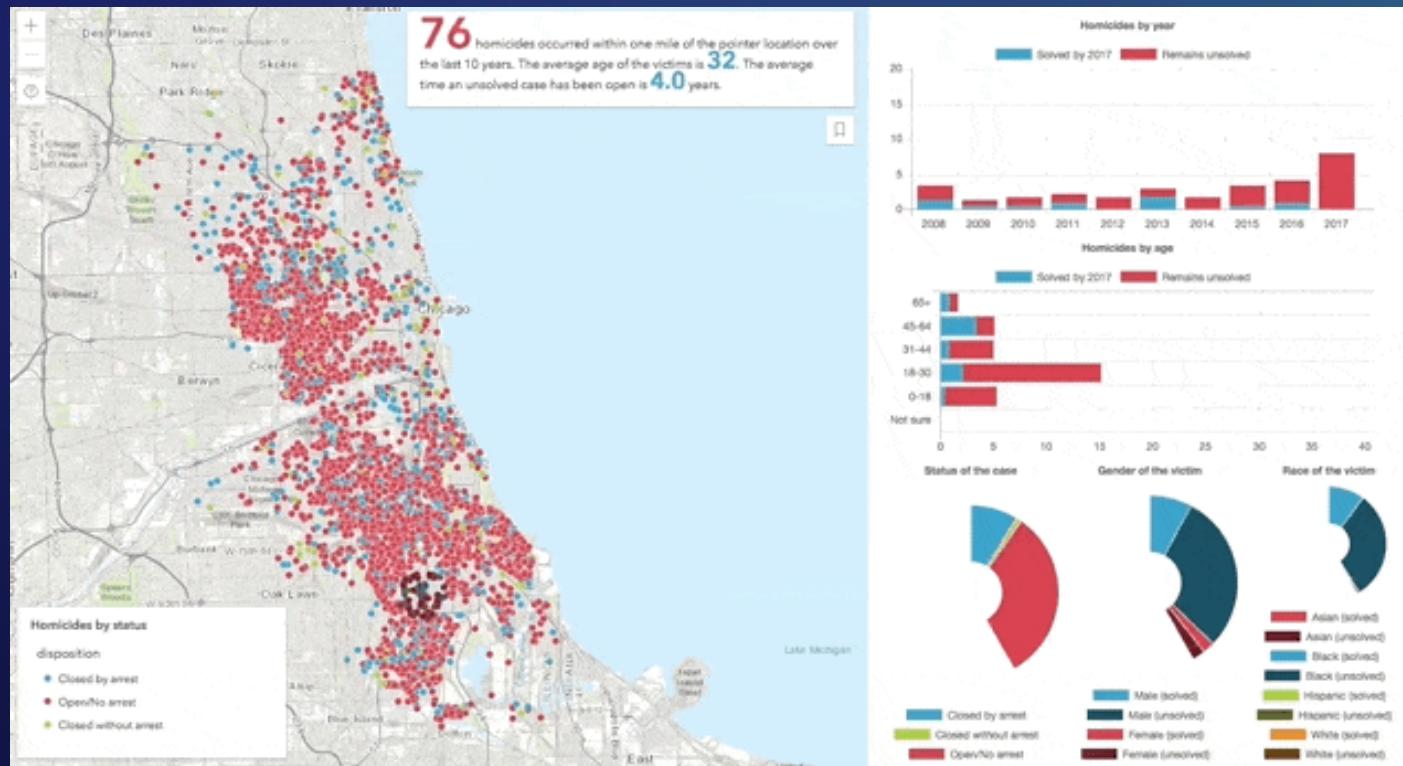


# API JS 4.x

## Prise en charge complète côté client des requêtes

La LayerView qui s'exécute côté client, prend désormais en charge **toutes les types de requêtes** y compris les requêtes spatiales et les requêtes statistiques.

**Les requêtes s'exécutent plus rapidement**, permettant une meilleure interactivité.



# API JS 4.x

## Mise à jour de la Feature Layer

A partir de la version 4.9, il est possible de **requêter les entités reliées**.

Le support de l'ajout, la suppression et la **mise à jour des pièces jointes** a été ajouté.

`applyEdits ()` et `queryFeatures ()` désormais supportés sur des **feature layer de collection de graphics** :

```
var features = [
  {
    geometry: {
      type: "point",
      x: -100,
      y: 38
    },
    attributes: {
      ObjectID: 1,
      DepArpt: "KATL",
      MsgTime: Date.now(),
      FltId: "UAL1"
    }
  },
  {
    geometry: {
      type: "point",
      x: -77,
      y: 35
    },
    attributes: {
      ObjectID: 2,
      DepArpt: "KZBW",
      MsgTime: Date.now(),
      FltId: "SW999"
    }
  }
];  
  
var layer = new FeatureLayer({
  source: features, // autocast as a Collection of esri/Graphic
  ...
});
```

# API JS 4.x

## Widget de dessin

SketchViewModel a ajouté la prise en charge de la **mise à l'échelle et de la rotation des graphiques** existants.

De plus, vous pouvez également **valider la géométrie pendant la création et / ou la mise à jour** d'un graphique.



```
sketchViewModel.on([
  "update-init",
  "move-start",
  "move",
  "move-complete",
  "scale-start",
  "scale",
  "scale-complete",
  "rotate-start",
  "rotate",
  "rotate-complete",
  "reshape-start",
  "reshape",
  "reshape-complete",
  "update",
  "undo",
  "redo"  ],
  checkGraphicUpdate);
```

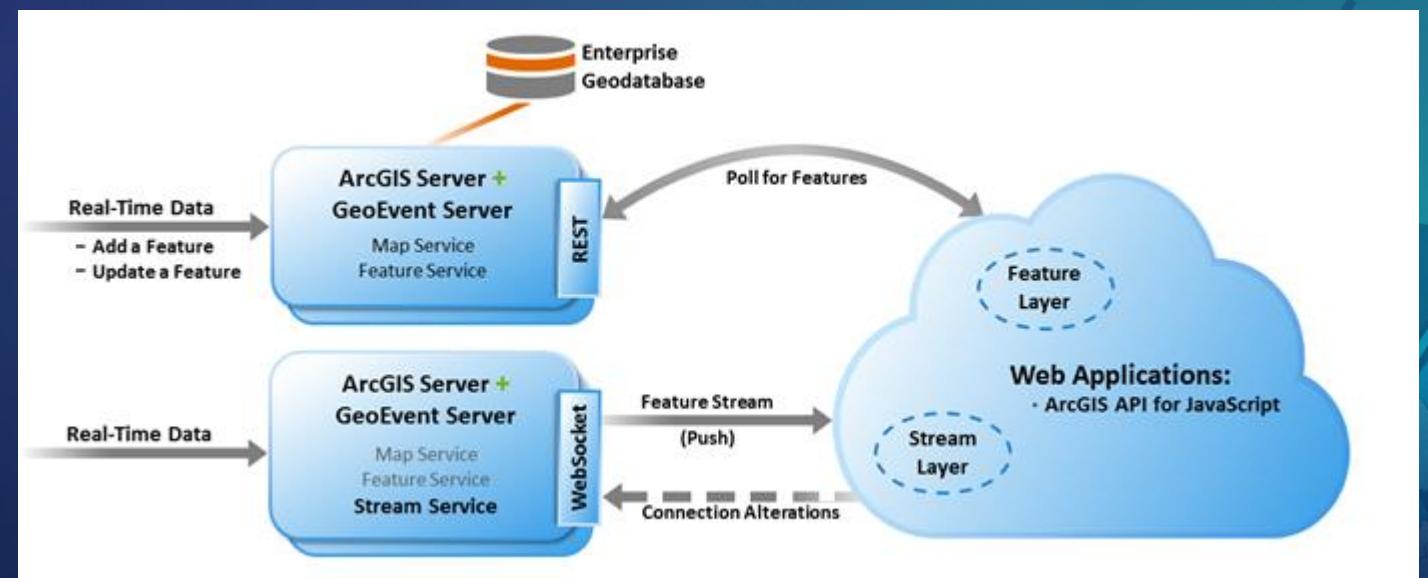
## API JS 4.x

### StreamLayer

Mises à jour de **StreamLayer** Les requêtes côté client sont désormais prises en charge dans StreamLayerView (2D uniquement) , avec une nouvelle méthode queryLatestObservations ajoutée.

StreamLayer va maintenant se **reconnecter automatiquement** lorsque la connexion WebSocket est perdue.

Les entités d'une StreamLayer sont maintenant **automatiquement effacées lorsque la propriété du filtre est modifiée**.



# API JS 4.x

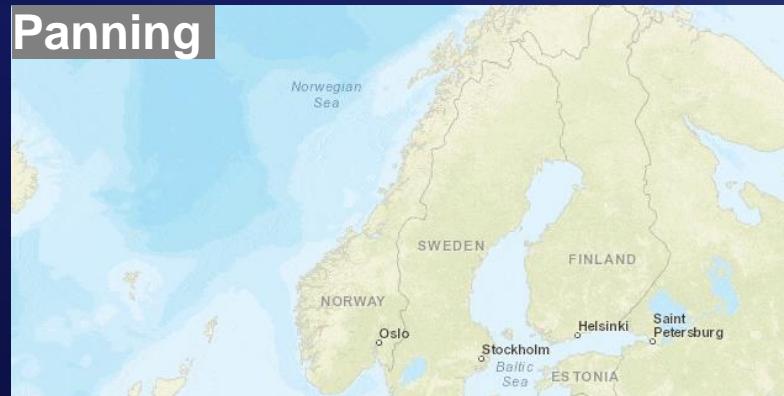
## Usages mobile et gamepad

### Kinetic Panning

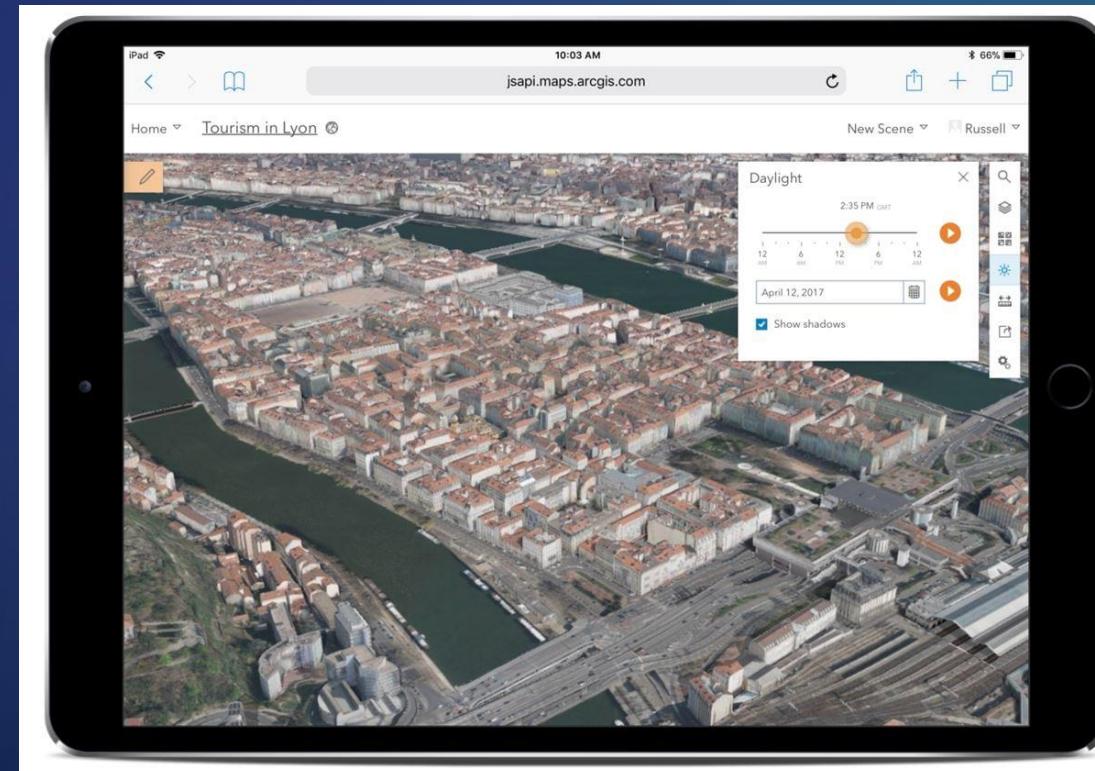
MapView et Webscene sur **support mobile**

Tous les standards **Gamepads et les devices 3Dconnexion** sont maintenant supportés.

#### Panning



#### Kinetic Panning



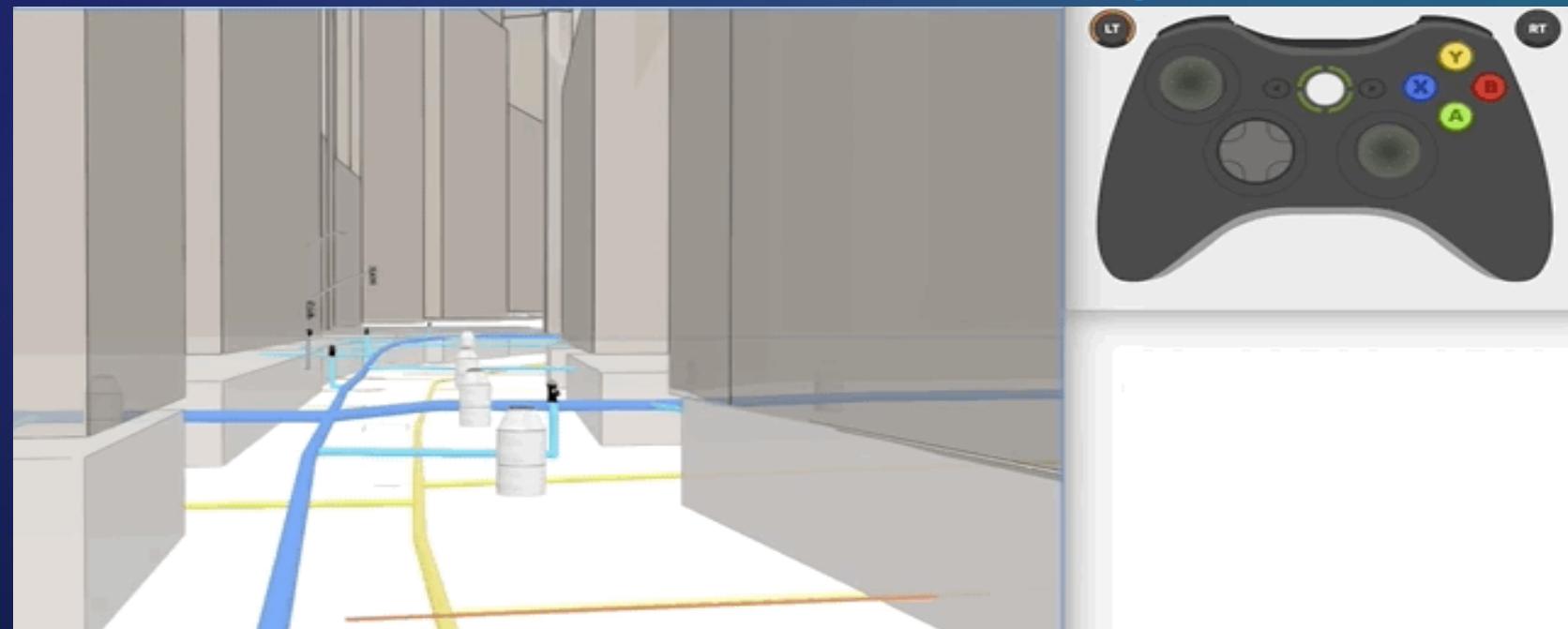
# API JS 4.x

## Navigation et visualisation en sous-sol

La navigation souterraine n'est plus limitée aux scènes locales, dans les scènes globales vous pouvez **afficher et naviguer vers des données de sous-sol.**

**Le sol peut également être rendu transparent** dans les scènes globales et locales

La nouvelle propriété de couleur de fond permet aux utilisateurs de définir **l'affichage de la surface de la terre sous le fond de carte.**



<https://developers.arcgis.com/javascript/latest/sample-code/sceneview-underground/index.html>

<https://github.com/richiecarlmichael/Esri-Gamepad>

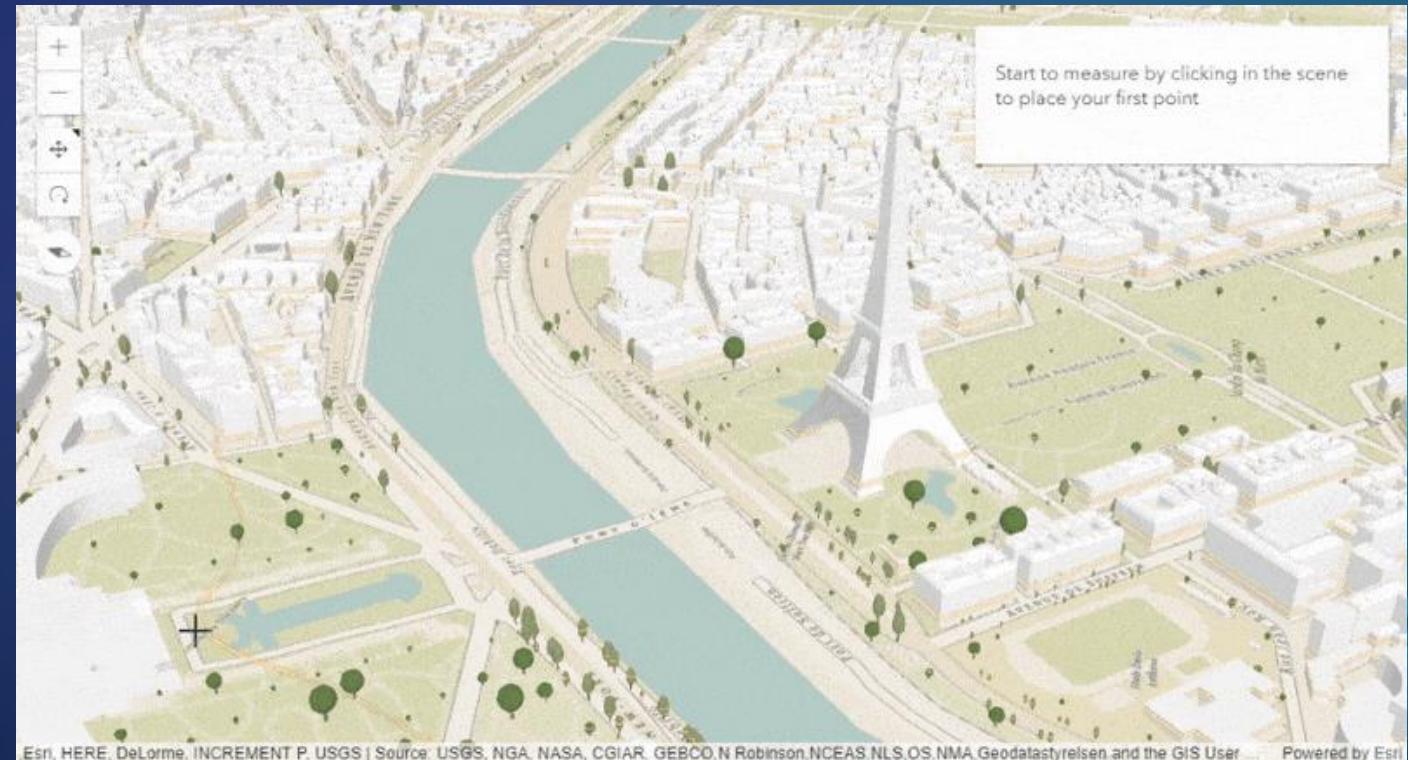
<https://github.com/sjuirim/Esri-Gamepad>

# API JS 4.x

## Nouvelles capacités

### Mesurer en 3D

- Aires
- Longueurs



<https://developers.arcgis.com/javascript/latest/sample-code/widgets-measurement-3d/index.html>

## API JS 4.x

## FeatureForm (Beta)

Enter the incident information

Incident Type

Street sign

Incident Description

Street sign is damaged.

Incident Location

300 W Pearl St

Reported Date

9/20/2018      11:20:41 AM

12/31/1969      4:00:00 PM

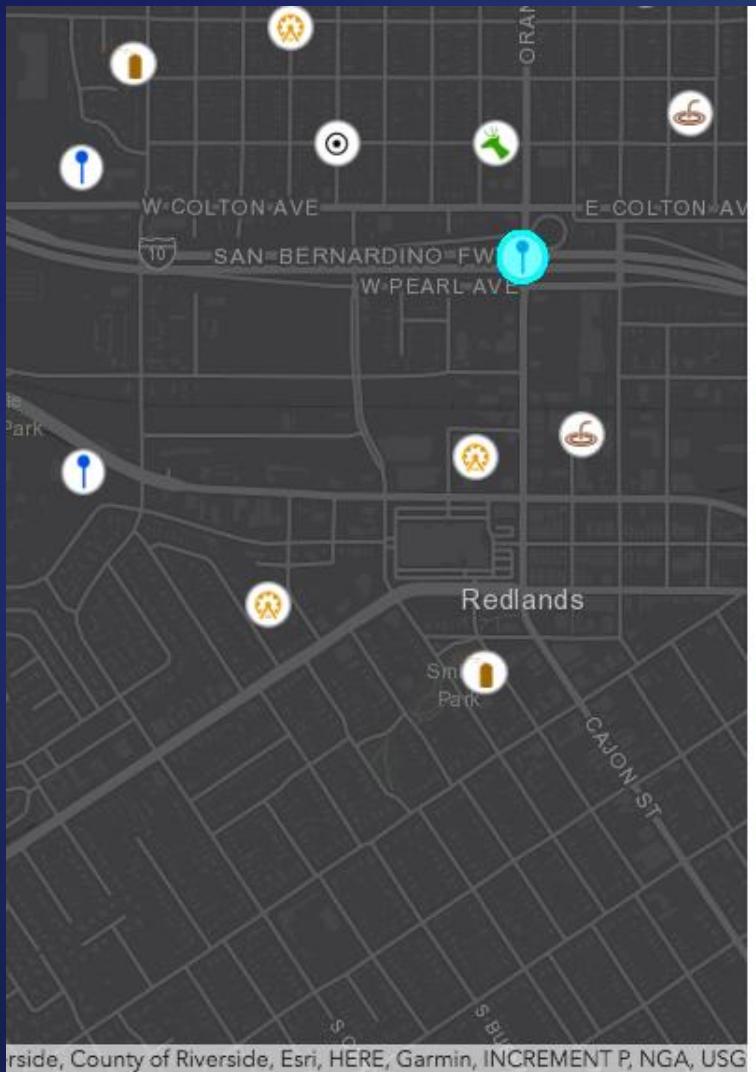
Reported by

John H

User Contact

Status

Update incident info

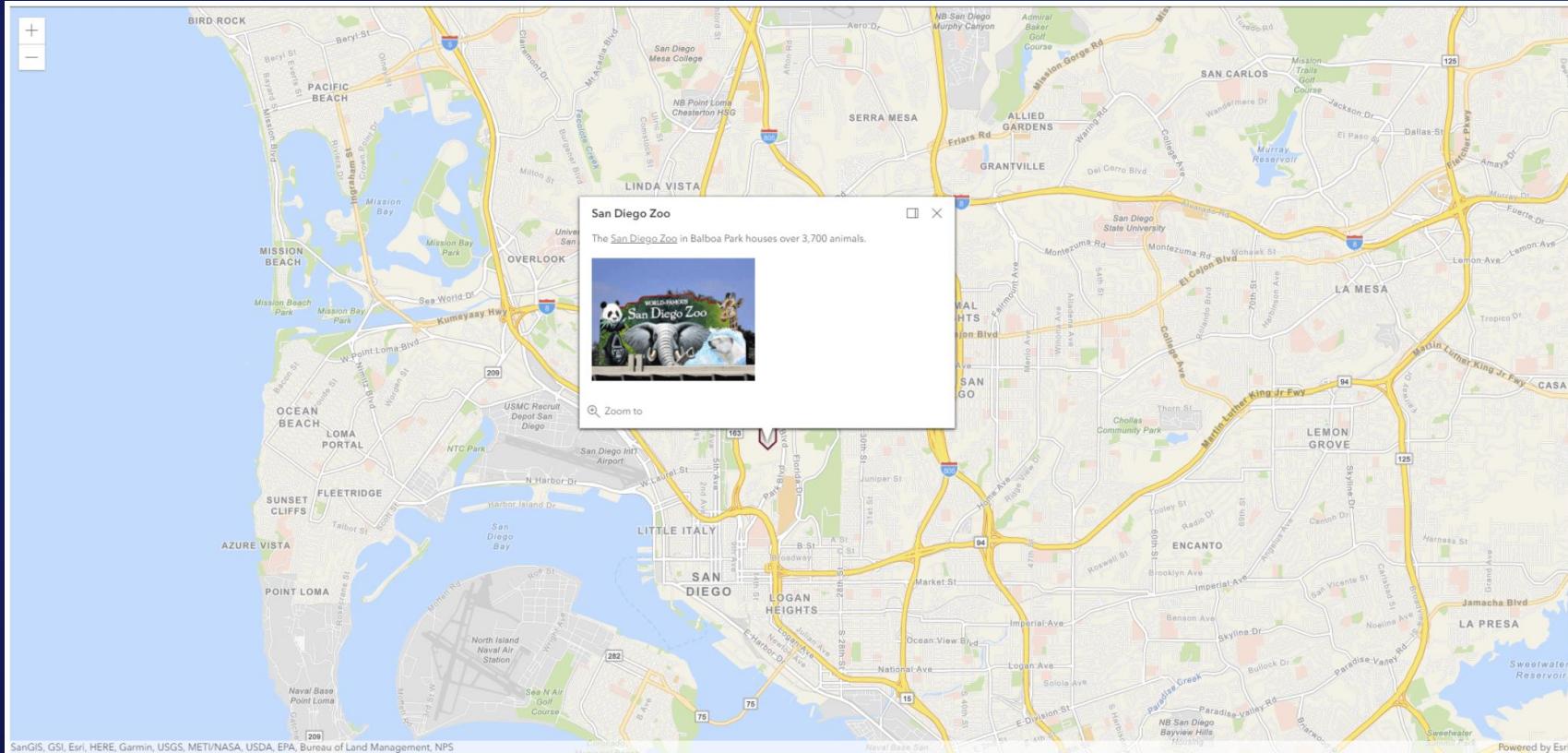


Riverside, County of Riverside, Esri, HERE, Garmin, INCREMENT P, NGA, USG

# Migrer de l'API JS Google Maps vers l'ArcGIS API for JavaScript

# API JS 4.x

## Migrer de l'API JS Google Maps vers l'API JS ArcGIS



<https://www.esri.com/arcgis-blog/products/js-api-arcgis/announcements/migrating-from-google-maps-javascript-api-to-arcgis-api-for-javascript/>

<https://developers.arcgis.com/javascript/latest/guide/migrating-from-google-maps-to-arcgis-javascript-api/>

<https://www.arcorama.fr/2018/08/migrer-de-lapi-javascript-google-maps.html>

# Fondamentaux de l'API 4.x

# API JS 4.x

## Fondamentaux

AMD (Asynchronous Module Definition)

Système de gestion de modules asynchrone :

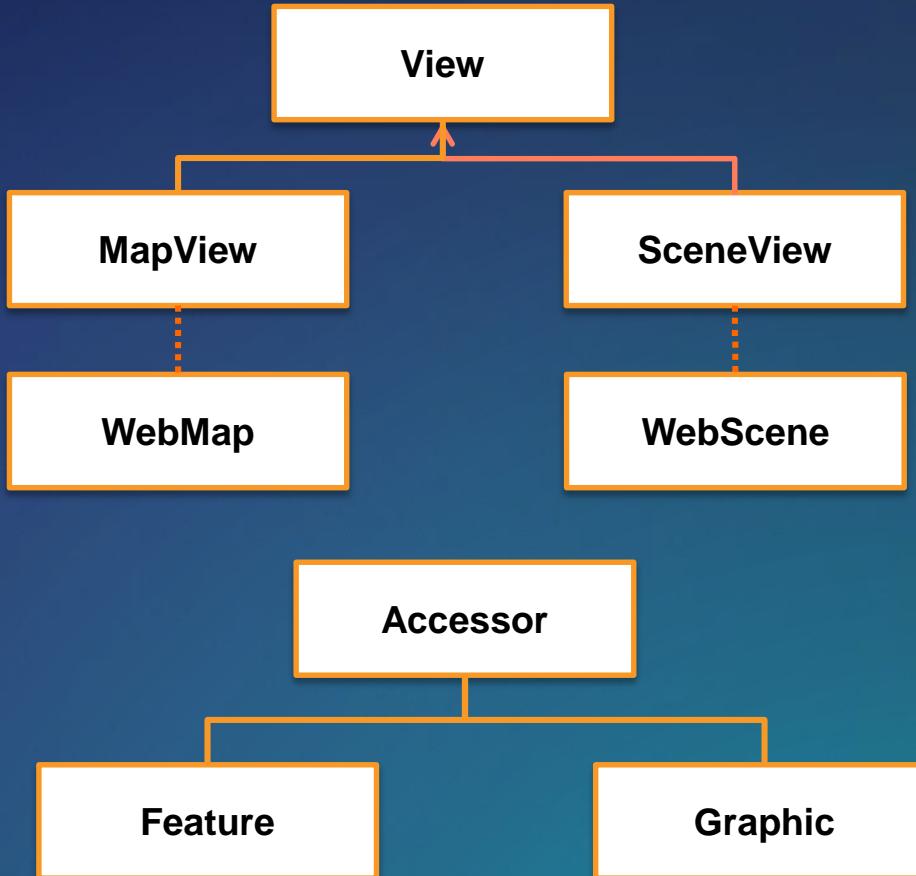
- require : ensemble des modules requis
- define : enregistre un nouveau module et ses dépendances à l'aide de « dojo/\_base/declare »

# API JS 4.x

## Fondamentaux

- Séparation des concepts
  - WebMap et WebScene : description
  - MapView et SceneView : représentation
    - Séparation des moteurs 2D et 3D
- Map / Scene
  - Basemap
  - Operational Layers
  - Ground
  - SpatialReference

Même concepts que SDK Runtime



```
var map = new Map({  
  basemap: "topo"  
});  
  
var view = new MapView({  
  map:map,  
  container: "div"  
});
```

# API JS 4.x

## Fondamentaux

- La map et la scene peuvent être définies à partir d'un item portal

```
*****
 * Creates a new WebMap instance. A WebMap must reference
 * a PortalItem ID that represents a WebMap saved to
 * arcgis.com or an on-premise portal.
*****/
```

```
var webmap = new WebMap({
  portalItem: {
    id: "e691172598f04ea8881cd2a4adaa45ba"
  }
});
```

```
*****
 * Set the WebMap instance to the map property in a MapView.
*****/
```

```
var view = new MapView({
  map: webmap,
  container: "viewDiv"
});
```

```
var scene = new WebScene({
  portalItem: {
    id: "3a9976baef9240ab8645ee25c7e9c096"
  }
});
```

```
*****
 * Set the WebScene instance to the map property in a SceneView.
*****/
```

```
var view = new SceneView({
  map: scene,
  container: "viewDiv"
});
```

# API JS 4.x

## Fondamentaux

- **Couches supportées :**

### 2D Layers :

**BaseDynamicLayer (4.4)**  
GeoRSSLayer  
**KMLLayer (4.5)**

### 3D Layers

**BaseElevationLayer(4.4)**  
ElevationLayer  
IntegratedMeshLayer  
PointCloudLayer  
SceneLayer

### **2D/3D Layers**

**BaseTileLayer (4.4)**  
**BingMapsLayer (4.8)**  
WebTileLayer  
OpenStreetMapLayer  
CSVLayer  
FeatureLayer  
GraphicsLayer  
GroupLayer  
ImageryLayer  
MapImageLayer  
StreamLayer  
TileLayer  
VectorTileLayer  
**MapNotesLayer (4.4)**  
**WMSLayer (4.4)**  
**WMTSLayer (4.4)**

# API JS 4.x

## Fondamentaux

- Basemap et Ground donnent le **contexte des couches Opérationnelles**
- Basemap et Ground défini par :
  - Well-Know ID
  - Un item de la plateforme

```
var map = new Map({
  /**streets, satellite, hybrid, topo, gray,
  dark-gray, oceans, national-geographic, terrain, osm,
  dark-gray-vector, gray-vector, streets-vector, streets-night-vector,
  streets-relief-vector, streets-navigation-vector*/
  basemap: "topo",
  ground : "world-elevation"
});

var view = new MapView({
  map:map,
  container: "div"
});
```

# API JS 4.x

## Fondamentaux

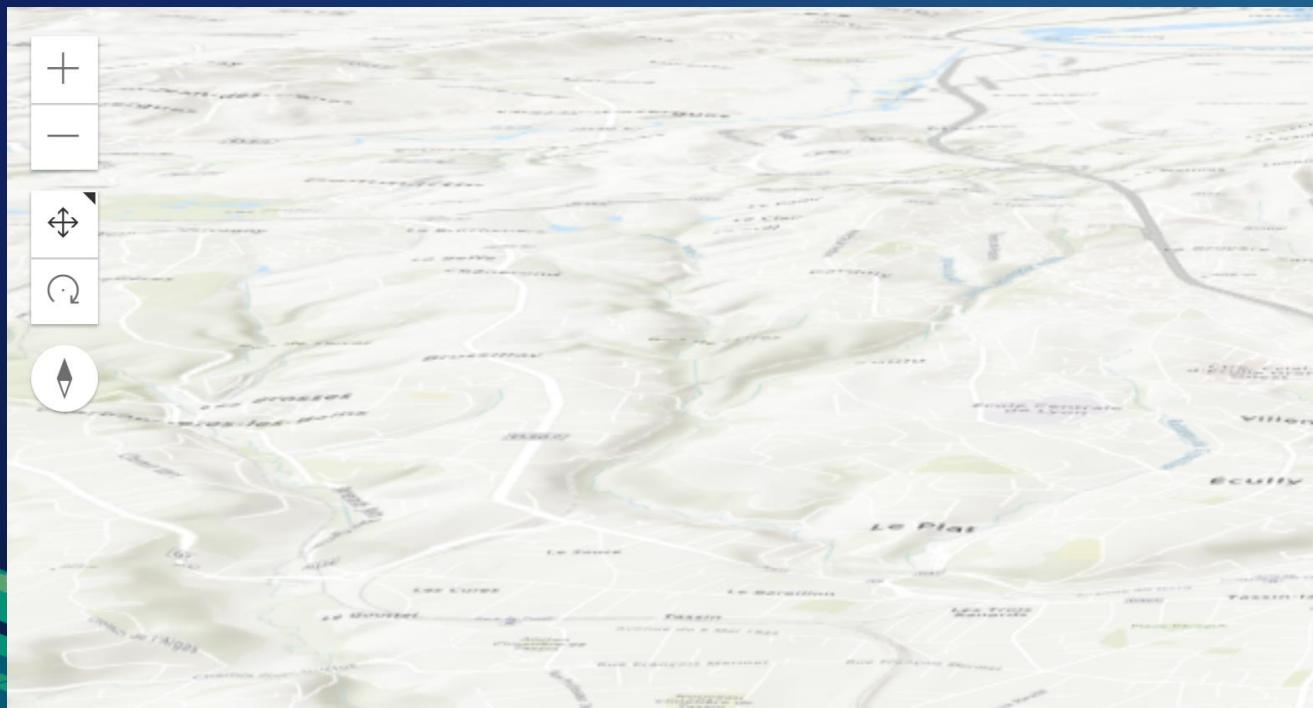
- Accessor:
- **Socle de l'API**, la plupart des objets héritent d'accessor
- Permet d'accéder aux **propriétés d'un élément** rapidement
- **Alternative aux méthodes get/set**

```
map.set("basemap", "topo");
// is equivalent to accessor
map.basemap = "topo";
```

### Autocasting :

- Moyen pratique de **définir des propriétés typées sans appeler des modules AMD** supplémentaires via la fonction require ()

```
var diamondSymbol = {
  type: "simple-marker", // autocasts as new SimpleMarkerSymbol()
};
```



## Démo: Environnement

TileLayer  
VectorTileLayer  
ElevationLayer  
Environnement  
Camera  
Autocast

# API JS 4.x

## Fondamentaux

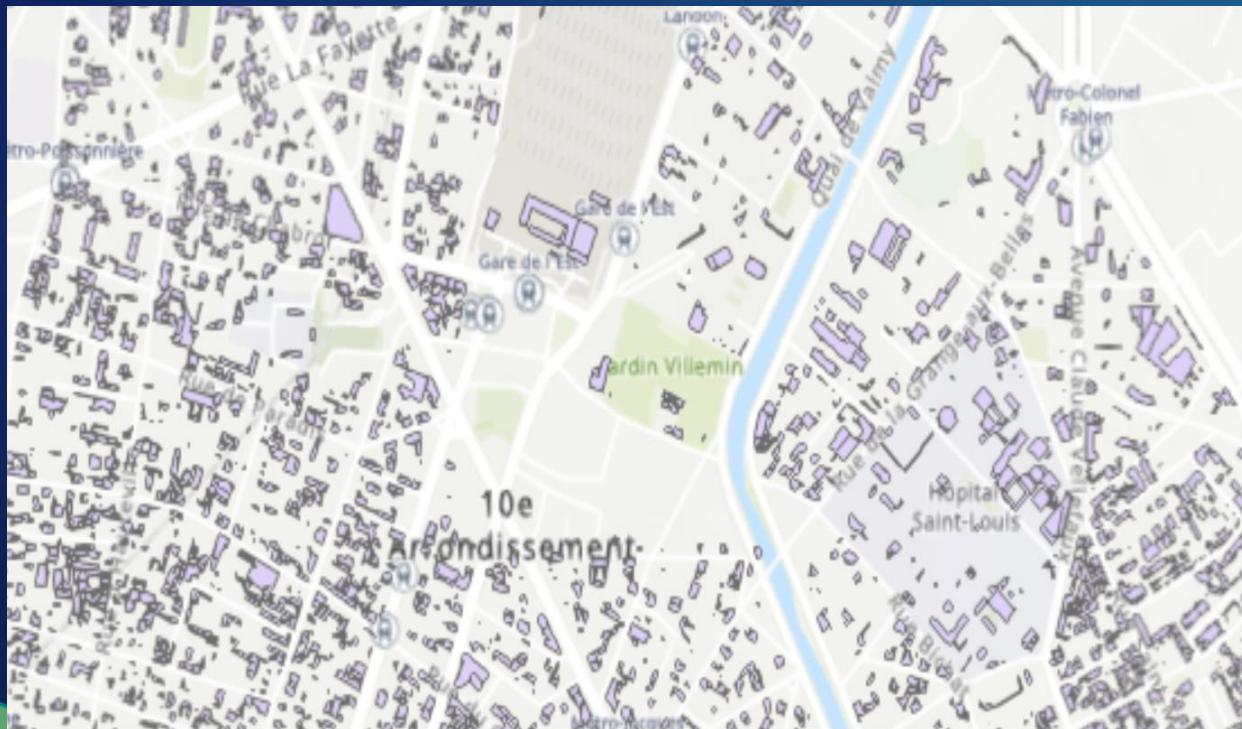
- OperationalLayer :

- [CSV Layer \(CSV\)](#)
- [Feature Layer \(ArcGISFeatureLayer\)](#)
- [GeoRSS Layer \(GeoRSS\)](#)
- [Image Service Layer \(ArcGISImageServiceLayer\)](#)
- [Image Service Vector Layer \(ArcGISImageServiceVectorLayer\)](#)
- [KML Layer \(KML\)](#)
- [Map Service Layer \(ArcGisMapServiceLayer\)](#)
- [ArcGISStreamLayer](#)
- [Tiled Image Service Layer \(ArcGISTiledImageServiceLayer\)](#)
- [Tiled Map Service Layer \(ArcGISTiledMapServiceLayer\)](#)
- [Vector Tile Layer \(VectorTileLayer\)](#)
- [WebTiledLayer](#)
- [Web Feature Service Layer \(WFS\)](#)
- [WMS Layer \(WMS\)](#)

Layer implémentation :

- Chargement du module AMD
- Création de la layer
- Mise à jour des propriétés de la couche
- Ajout sur la map (ou scene)

```
require(["esri/layers/FeatureLayer"], function(FeatureLayer) {  
  
    var featureLayer = new FeatureLayer(  
    {  
        url:"https://sampleserver6.arcgisonline.com/arcgis/rest/services/Hurricanes/MapServer/0"  
    });  
  
    var map = new Map({  
        layers:[featureLayer]  
    });  
});
```



## Démo: FeatureLayer

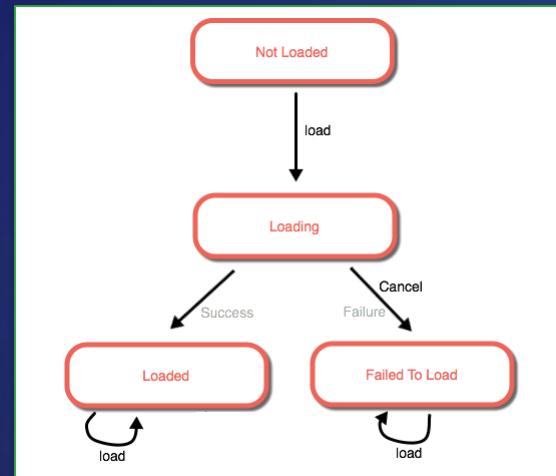
FeatureLayer  
AMD  
DefinitionExpression

# API JS 4.x

## Fondamentaux

Loadable :

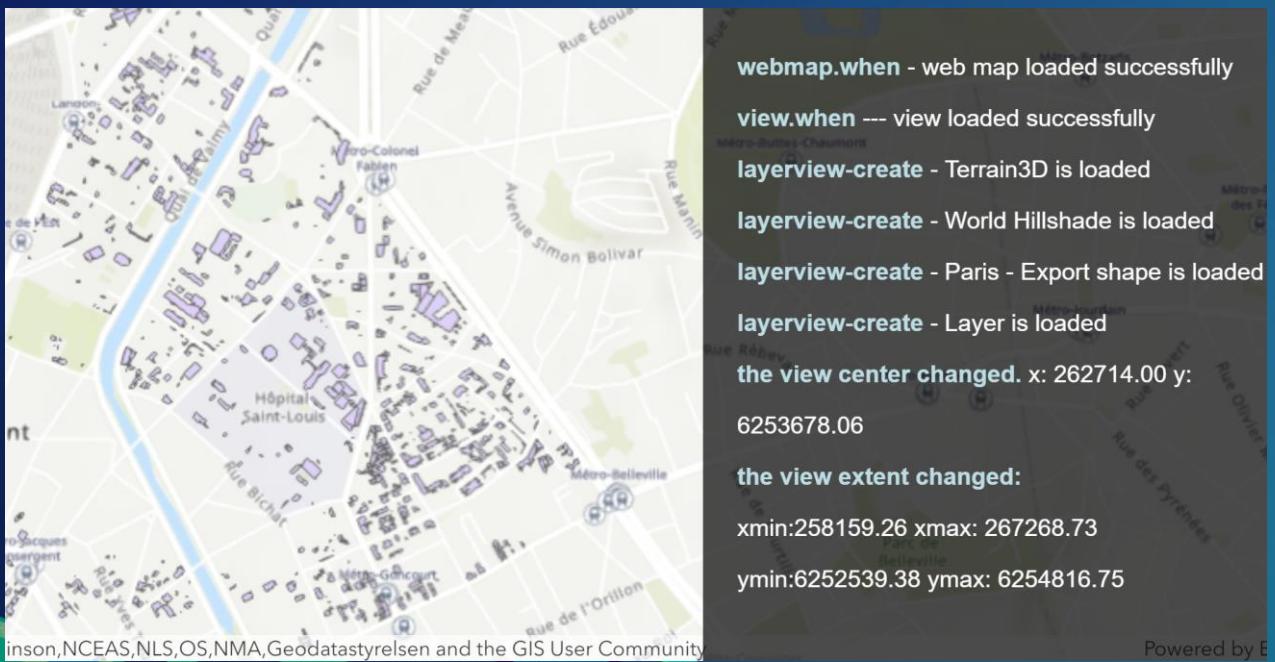
- Les objets cartographiques ont un cycle de vie qui peut être testé



Watch:

- On peut tester le changement de propriétés d'un objet
- La classe esri/core/watchUtils est un utilitaire permettant d'aider au test de ces propriétés

```
layer.watch("loadStatus", function(status) {/**/});
```



## Démo: Loadable & Watch

Loadable  
Watch  
WatchUtils

# API JS 4.x

## Fondamentaux

Les **renderer** définissent l'aspect visual de chaque entité dans une layer en utilisant un **symbole**.

Il existe **trois types de renderer** :

- SimpleRenderer
- UniqueValueRenderer
- ClassBreaksRenderer

Des **variables visuelles** basés sur des champs:

- Variation de couleur, taille, opacité
- Extrusion

Type	Object Specification	Legend Example
color	color object specification	
size	size object specification	
opacity	opacity object specification	
rotation	rotation object specification	-

Symbol3DLayer Type	Symbol3D Type	Flat/Volumetric	Size Units	Example
IconSymbol3DLayer	PointSymbol3D, PolygonSymbol3D	flat	points	
ObjectSymbol3DLayer	PointSymbol3D, PolygonSymbol3D	volumetric	meters	
LineSymbol3DLayer	LineSymbol3D, PolygonSymbol3D	flat	points	
PathSymbol3DLayer	LineSymbol3D	volumetric	meters	
FillSymbol3DLayer	PolygonSymbol3D, MeshSymbol3D	flat	-	
ExtrudeSymbol3DLayer	PolygonSymbol3D	volumetric	meters	
TextSymbol3DLayer	PointSymbol3D, LineSymbol3D, PolygonSymbol3D, LabelSymbol3D	flat	points	Text



## Démo: Symbole & Renderer & Autocasting

SimpleFillRenderer  
UniqueValueRenderer  
PolygonSymbol3D  
ExtrudeSymbol3DLayer  
VisualVariables  
Renderer  
Autocasting

# API JS 4.x

## Fondamentaux

### View Events

Utilisation des événements pour interagir avec la View

- click
- double-click
- drag
- hold

```
view.on("drag", function(event) {  
});
```

### Promise

Représente le résultat d'un traitement asynchrone

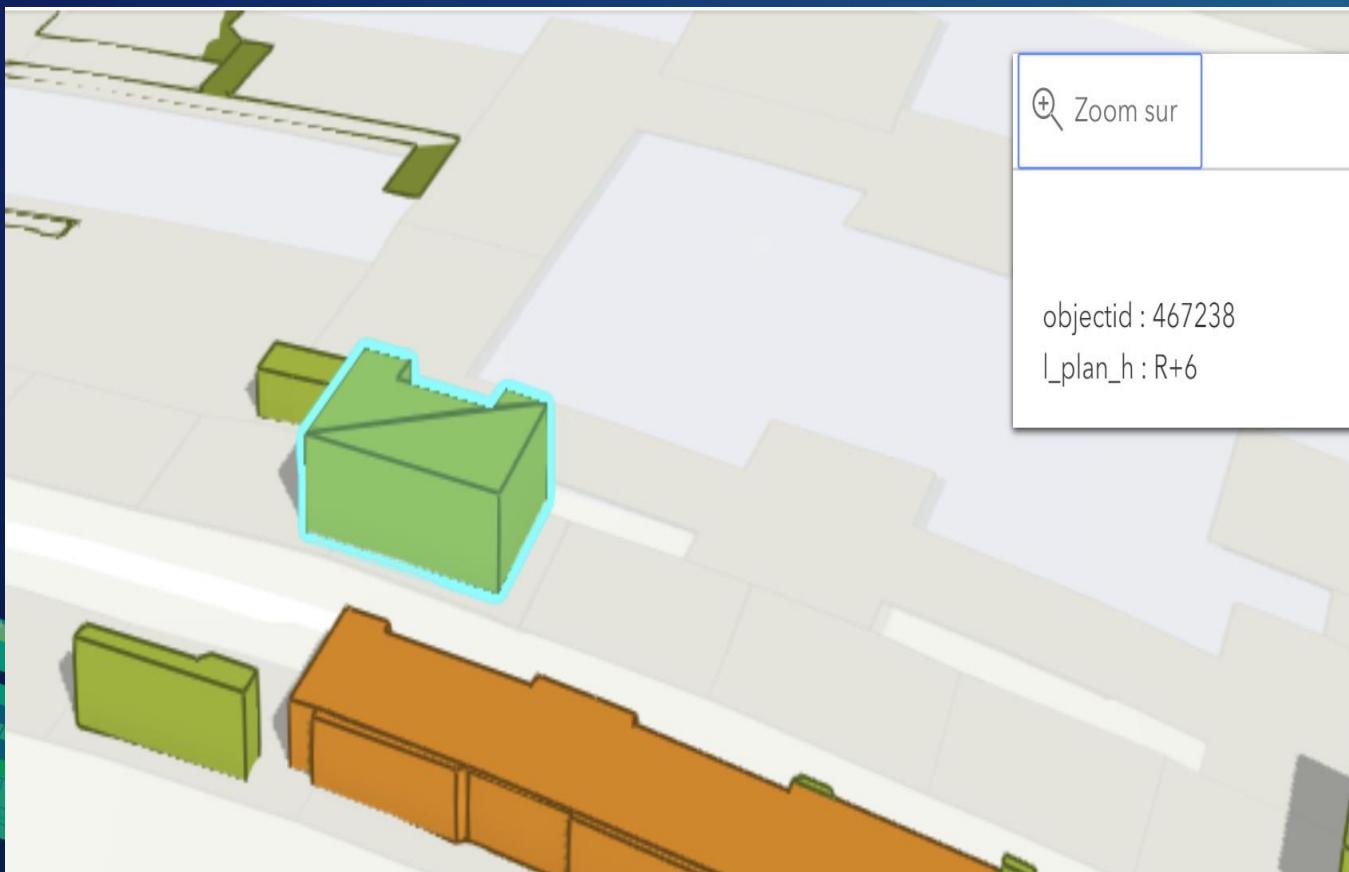
3 état:

- pending
- resolved
- rejected

De nombreuses classes héritent de Promise:

- En particulier : view.then(...) remplace on(map, "load", ...)

```
someAsyncFunction().then(function(resolvedVar) {  
});
```



## Démo: Promise & Popup

Promise  
Popup  
Event

# API JS 4.x

## Fondamentaux

### Collections

Les collections sont des objets qui contiennent un tableau d'éléments du même type elles produisent des méthodes très utiles pour travailler avec ces entités :

- filter()
- find()
- reduce()

Objets collection notables :

- map.Layers
- queryEvent.FeatureSet
- graphicsLayer.graphics
- featureLayer.source

```
// filteredLayers is a Collection of all the non-visible layers in the map
var filteredLayers = map.layers.filter(function(layer) {
  return !layer.visible;
});
```

# Arcade

# API JS et Arcade

## Pourquoi?

- Portable :
  - Arcgis API JS 3 & 4
  - AGOL & Portal for Arcgis
  - Arcgis Pro
  - Arcgis Runtime
- Sécurisé
  - Expressions de scripts ne compromettent pas la sécurité de l'application
- Léger
  - Expressions courtes, langage léger et sources non volumineuses
- Géospatial
  - Classes orientés géospatiales



## Démo: Query & Arcade

Query  
Arcade expression  
Collection

# Widgets

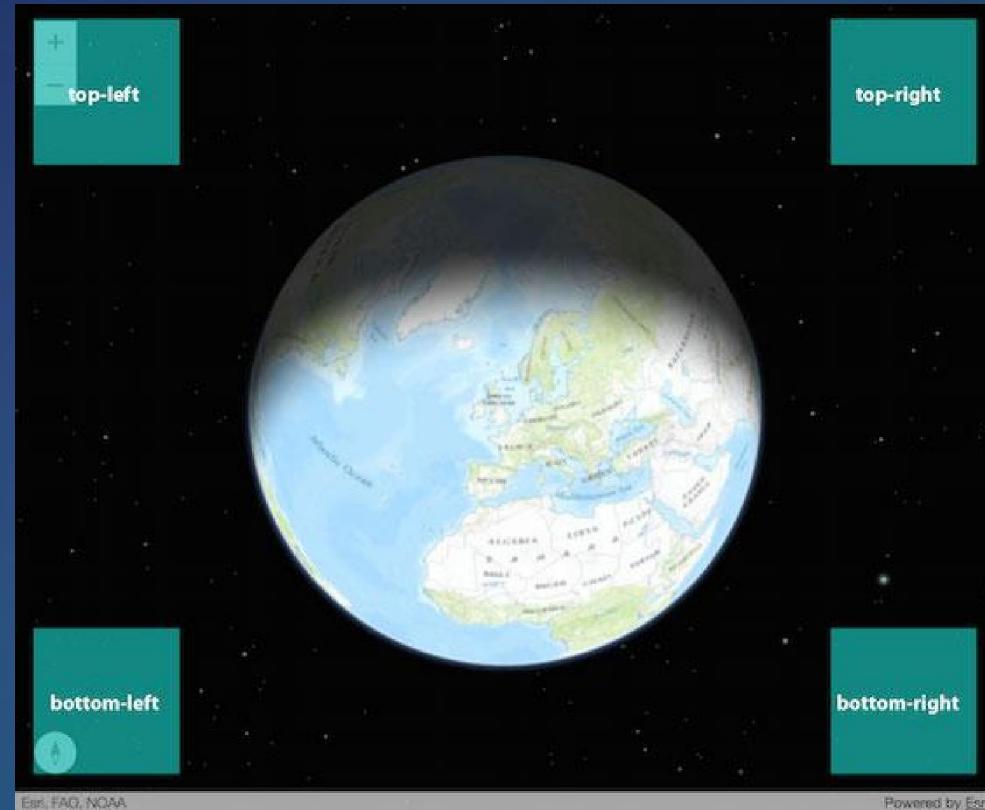
# Widget Composition

Les widgets sont composés de view et viewModels

- Séparation des responsabilités
- Creation, extensibilité
- Intégration de framework
- Facilité de maintenance

esri/widgets/Widget

- Cycle de vie
  - constructor
  - postInitialize
  - render
  - destroy
- API



# Widget

## Composition

Custom/New widgets : TypeScript

- Javascript du futur
- TypeScript est un langage transcompilé en JavaScript
- Développé par Microsoft en OpenSource
- But : Améliorer et de sécuriser la production de code JavaScript.

Avantages TypeScript:

- Typage optionnel des variables et des fonctions
- Création de classes et d'interface
- Attention : Plutôt adapté à des développeurs assez expérimentés



A screenshot of a code editor showing a tooltip for the variable 'item'. The tooltip lists several properties and methods, each preceded by a small blue icon indicating its type: basemap (property), declaredClass, destroy, destroyed, error, get, initialized, set, state, view, and watch. The code snippet above the tooltip starts with 'const' and ends with a closing brace '}'.



## Démo: Custom widget avec TS

TypeScript  
NodeJS  
TypeScript compiler  
Widget

# Roadmap

# API JS 4.x

## Roadmap

4.10

### Nouvelles fonctions d'éditions pour aller vers une édition complète 2D et 3D:

Ajout, suppression de vertex sur mobile

Amélioration de l'outil de dessin

#### Rendering :

Rendu complet des layers en webGL

Tout comme la SceneView, la MapView utilisera un seul canvas HTML et WebGL pour visualiser l'ensemble des layers.

Pour le moment en 4.9 seuls le VectorTileLayer, FeatureLayer et CSVLayer peuvent tirer parti de la carte graphique.

Cela devrait se traduire par une MapView plus performante, plus rapide à démarrer.

#### Widget :

Widget de mesure pour la 2D

Widget TemplatePicker

#### Label :

Label des entités polylines



# API JS 4.x

## Fondamentaux

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

Documentation et ressources des différents API et SDKs ArcGIS

- <https://developers.arcgis.com/labs/>

Tutoriaux ArcGIS DevLabs pour apprendre à créer des applications cartographiques

- <https://github.com/Esri>

Projets Esri sur GitHub

- <https://github.com/Esri/esri-loader>

Une petite bibliothèque pour charger des modules à partir des versions 4.x ou 3.x de l'API ArcGIS pour JavaScript dans des applications non-dojo (Angular, React ...)

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

# Questions / Réponses

<https://github.com/sjguirim>

<https://github.com/fgrataloup>