Create a starter app

### Open the JavaScript Starter App on CodePen (https://codepen.io/tkydo/pen/pMxaqe?editors=1000).

### In CodePen, click Fork and save the pen as ArcGIS JavaScript Tutorials: Display a web scene.

## A - Preview the web scene

### Go to the web scene to view the scene in the Scene Viewer. Make note of the id at the end of the url. (<http://esrisg.maps.arcgis.com/home/item.html?id=0e7e1fbf3ad74803804070aa9c1a6531>)

## B - Display the web scene in your app

### In the require statement, remove the Map and add a reference to the WebScene module.

require([

//\*\*\* ADD \*\*\*//

"esri/WebScene",

"esri/views/SceneView"

], function(WebScene, SceneView) {

### At the beginning of the code in the main function, remove the code to create a Map and SceneView and replace it with code to create a new WebScene and a new SceneView. Set the portalItem id to 0e7e1fbf3ad74803804070aa9c1a6531. Set the webscene to the map.

//\*\*\* ADD \*\*\*//

var webscene = new WebScene({

portalItem: {

id: "0e7e1fbf3ad74803804070aa9c1a6531"

}

});

//\*\*\* ADD \*\*\*//

var view = new SceneView({

container: "viewDiv",

map: webscene,

qualityProfile: "high",

environment: {

lighting: {

directShadowsEnabled: true,

ambientOcclusionEnabled: true

},

atmosphere: {

quality: "high"

}

}

});

### Run your code to view the web scene. Be sure to click on the layers and zoom around!

## C - Add a layer, visualization and styling

### In the require statement, add a reference to the [FeatureLayer](https://developers.arcgis.com/javascript/latest/api-reference/esri-layers-FeatureLayer.html) module.

require([

"esri/WebScene",

"esri/views/SceneView",

//\*\*\* ADD \*\*\*//

"esri/layers/FeatureLayer"

], function(WebScene, SceneView, FeatureLayer) {

### At the end of the code in the main function, create a new FeatureLayer and set the url property to the Child care (points) feature service. Add the feature layer to the Map.

var fLayer = new FeatureLayer({

url: "https://services1.arcgis.com/OMC1KTwzPuqD0WGv/arcgis/rest/services/ChildCare/FeatureServer/0",

elevationInfo: {

mode: "relative-to-ground"

},

returnZ: false,

title: "Childcare centers",

// Set a renderer that will show the points with icon symbols

renderer: {

type: "simple", // autocasts as new SimpleRenderer()

symbol: {

type: "point-3d", // autocasts as new PointSymbol3D()

symbolLayers: [

{

type: "icon", // autocasts as new IconSymbol3DLayer()

resource: {

primitive: "circle"

},

material: {

color: "#F97C5A"

},

size: 4

}

]

}

},

outFields: ["\*"],

// Add labels with callouts of type line to the icons

labelingInfo: [

{

// When using callouts on labels, "above-center" is the only allowed position

labelPlacement: "above-center",

labelExpressionInfo: {

value: "{NAME}"

},

symbol: {

type: "label-3d", // autocasts as new LabelSymbol3D()

symbolLayers: [

{

type: "text", // autocasts as new TextSymbol3DLayer()

material: {

color: "black"

},

halo: {

color: [255, 255, 255, 0.7],

size: 2

},

size: 10

}

],

// Labels need a small vertical offset that will be used by the callout

verticalOffset: {

screenLength: 150,

maxWorldLength: 2000,

minWorldLength: 30

},

// The callout has to have a defined type (currently only line is possible)

// The size, the color and the border color can be customized

callout: {

type: "line", // autocasts as new LineCallout3D()

color: "white",

size: 2,

border: {

color: "#F97C5A"

}

}

}

}

]

});

webscene.add(fLayer);

### Run your code to view the map with the trailheads layer.

## D - Add Search widget

### In the require statement, add a reference to the Search module.

require([

"esri/WebScene",

"esri/views/SceneView",

"esri/layers/FeatureLayer",

//\*\*\* ADD \*\*\*//

"esri/widgets/Search"

], function(WebScene, SceneView, FeatureLayer,Search) {

### At the end of the code in the main function, create a Search widget and set the view. Add the widget to the top right corner of the view.

// Search widget

var search = new Search({

view: view

});

view.ui.add(search, {

position: "top-right",

index: 1

});

### R un the app and try searching for the following

## E - Add legend, layer list widget

### In the require statement, add a reference to the Legend, Expand module.

require([

"esri/WebScene",

"esri/views/SceneView",

"esri/layers/FeatureLayer",

"esri/widgets/Search",

//\*\*\* ADD \*\*\*//

"esri/widgets/Legend",

"esri/widgets/LayerList",

"esri/widgets/Expand" ], function(WebScene, SceneView, FeatureLayer,Search, Legend, LayerList,Expand) {

### At the end of the code in the main function, create a Legend widget and set the view. Add the widget to the bottom right corner of the view.

var legend = new Legend({

view: view

});

var lgExpand = new Expand({

view: view,

content: legend

});

view.ui.add(lgExpand, "top-right");

### At the end of the code in the main function, create a Layerlist widget and set the view. Add the widget to the bottom right corner of the view.

## view.when(function() {

## var layerList = new LayerList({

## view: view

## });

## // Add widget to the top right corner of the view

## // view.ui.add(layerList, "top-right");

## var llExpand = new Expand({

## view: view,

## content: layerList

## });

## // Add the expand instance to the ui

## view.ui.add(llExpand, "top-right");

## });