# ArcGIS Integrations with Alexa and Salesforce

# Overview:

- Alexa: What's near me?
- Salesforce Field Service Lightning: custom geofencing for work orders using ArcGIS REST API
- Salesforce: Integrating with ArcGIS API for JavaScript with Salesforce

# Alexa, what is the radius of the earth?

BRIAN YARSAWITCH

# Field Service Lightning location based workorder checkout

BRANDON FITCH

# Embedding a ArcGIS JavaScript Application in Salesforce

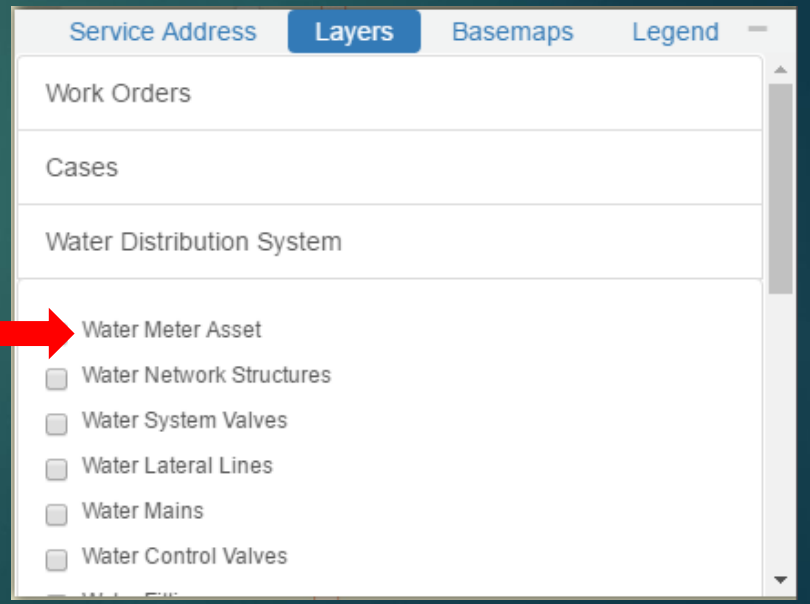
TYLER WARING

# Custom Metadata Types or JSON config Files?

- We started using custom metadata types exclusively.
  - Time consuming for map layer configuration
  - Unable to view and edit all layer configurations at once
- Ended up using JSON to manage map layer configuration ✓
  - Easy to read and edit
  - Copy and paste configuration blocks

# Pulling ArcGIS Server data into a Salesforce Application

- This is pretty standard with ArcGIS JavaScript API 3.24
- Iterate over layers in configuration file and add to map as ArcGISDynamicMapServiceLayers
- Did not use LayerList digit
  - Created Bootstrap widget using collapsible list-group
  - Group Salesforce layers with similar map service layers



# Making graphicsLayers for Salesforce Objects

- Salesforce JSON configuration:
  - ```
    "Cases" : {
        "sfTableName" : "Case",
        "queryFields" : "Id, Subject, Asset.Name, CaseNumber, Description, Status, Type, Owner.Name, Reason, Contact.Name, ContactPhone, ContactMobile, Incident_Geolocation__c",
        "queryDist_KM" : 10,
        "queryConditions" : "WHERE Incident_Geolocation__latitude__s != null",
        "parentLabel" : "Cases",
        "minScale" : 25000,
        "locationField" : "Incident_Geolocation__c",
        "linkId" : null,
        "linkAddress" : false,
        "layerStyle" : {"style" :"circle","color" :{"b" :2,"g" :255,"r" :2,"a" :1},"size" :8,"outline" :{"width" :1.3333333333333333,"color" :{"r" :0,"g" :0,"b" :0,"a" :1},"style" :"solid"}},
        "layerName" : "Cases",
        "layerLabel" : "Cases",
        "layerFieldsLabels" : "Subject, Asset Name, Case Number, Description, Status, Type, Owner Name, Reason, Contact Name, Contact Phone, Contact Mobile",
        "layerFields" : "Subject, Asset.Name, CaseNumber, Description, Status, Type, Owner.Name, Reason, Contact.Name, ContactPhone, ContactMobile",
        "assetAccountName" : null
    },
    ```
- Create graphicsLayers for later use:
  - ```
    var graphicsLayer = new GraphicsLayer({
        id: "GraphicsLayer_" + layerName
    });
    var symbol = new SimpleMarkerSymbol(symbolJSON);
    var renderer = new SimpleRenderer(symbol);
    graphicsLayer.setRenderer(renderer);
    ```

# Add Salesforce data to graphicsLayer

- AJAX Toolkit (connection.js): sforce object
- Salesforce DISTANCE query using the map centerPoint:

```
var query = "SELECT " + salesforceLayerConfigMap[layerName].queryFields + " FROM " +
salesforceLayerConfigMap[layerName].sfTableName + " " + salesforceLayerConfigMap[layerName].queryConditions +
" AND DISTANCE(" + salesforceLayerConfigMap[layerName].locationField+
", GEOLOCATION(" + centerPoint.y + "," + centerPoint.x + "), 'km')<" + salesforceLayerConfigMap[layerName].queryDist_KM;
var result = sforce.connection.query(query);
var records = result.getArray("records");
```

- Loop over the records returned by the distance query and add graphics to the graphics layer.
- Create a map.on “extent-change” event to update the graphics in the graphics layer using the DISTANCE query again.

# Linking assets to a case from the web map

- Query the Asset table using sforce.connection:
  - ```
    var query = "SELECT Id, Name FROM Asset WHERE Id = '" +assetId+"' LIMIT 1";
    var result = sforce.connection.query(query);
    var recordsString = JSON.stringify(result.getArray("records")[0]);
    var records = JSON.parse(recordsString)
    ```
- Message from map to the Map_iFrame.cmp:
  - ```
    var msg = '{"instanceId": "Map_Asset_Link", "caseAssetId": "'+ records.Id+'", "caseAssetName": "'+ records.Name+'"}';
    window.parent.postMessage(msg,lDomain);
    ```
- Fire an event from the iFrame.cmp to the caseIntakePage.cmp:
  - ```
    var setCaseAssetEvent = $A.get('e.c:SetCaseAssetEvent');
    setCaseAssetEvent.setParams(
    {
        instanceId: instanceId,
        caseAssetId: caseAssetId,
        caseAssetName: caseAssetName
    });
    setCaseAssetEvent.fire();
    ```
- Update caseAsset component with the assetId:
  - ```
    var caseAsset = component.find('caseAsset');
    caseAsset.setLookup(eventParams.caseAssetId);
    ```