Esri International Developer Summit

Palm Springs, CA

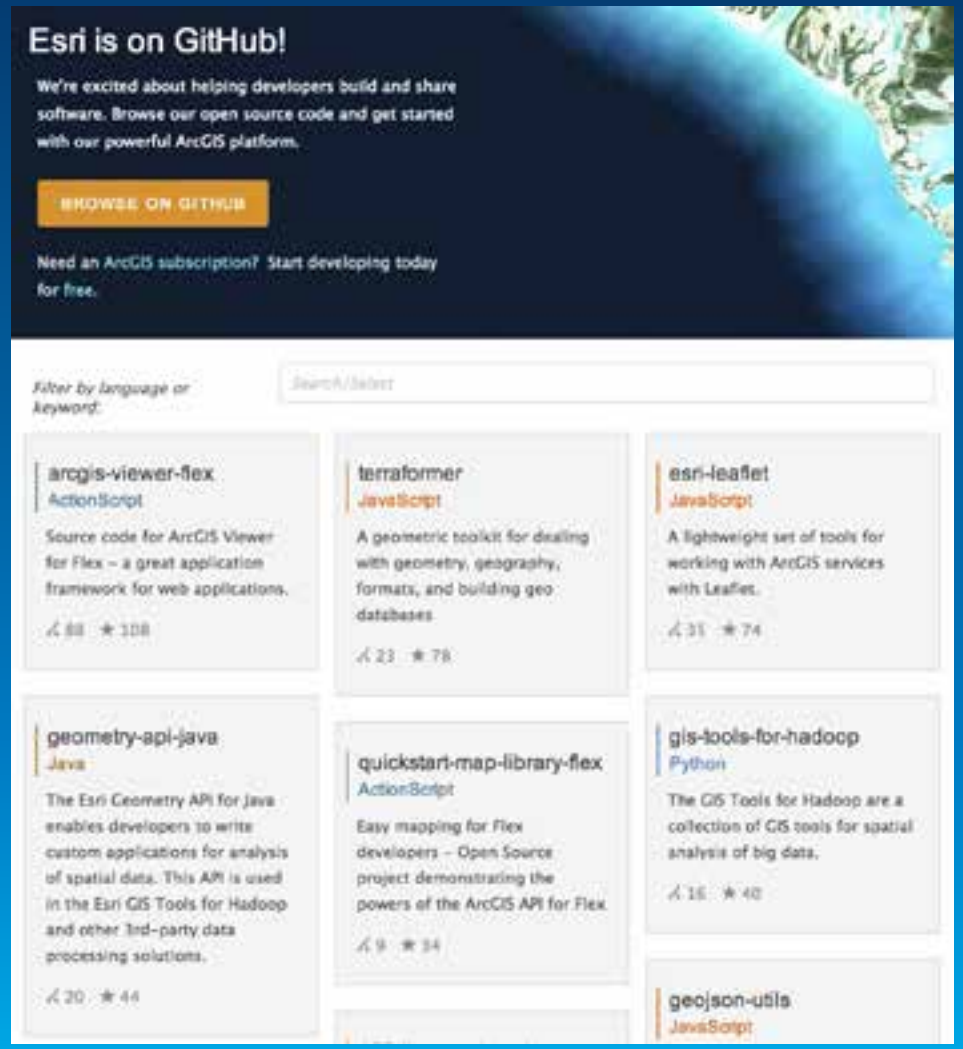
Building Mapping Solutions with Esri Open Source Projects

Allan Laframboise

alaframboise.github.com

@AL_Laframboise

esri.github.com



Esri Open Source Projects

+190 Projects

+450 Esri engineers

+1700 Forks

+1000 Pull Requests

+500 Issues

Quickstart-map-js

Simple mapping examples

Quickstart-map-js

- Simple examples of mapping tasks with the ArcGIS JS API
- Features
  - Basemaps, Geocoding, Directions, Feature Services, Graphics
  - Short, focused, little Dojo
  - Mobile friendly
  - Utils.js – map centering, setting popup…
  - Tips and tricks – CSS styling
- New Esri pins

Quickstart-map-js: Example

```
// Create map
var map = new Map("mapDiv", {
    basemap: "streets",
    center: [-122.69, 45.52], //long, lat
    zoom: 3
});

// Use utils
utils.autoRecenter(map);
utils.setPopup(map, "top", 0, 26);

// Show busy
utils.setStyle("progress", "progress visible");

// Create symbol
var sym = utils.createPictureSymbol(”blue-pin.png", 0, 12, 13, 24);
```

github.com/quickstart-map-js

Bootstrap-map-js

Responsive map app

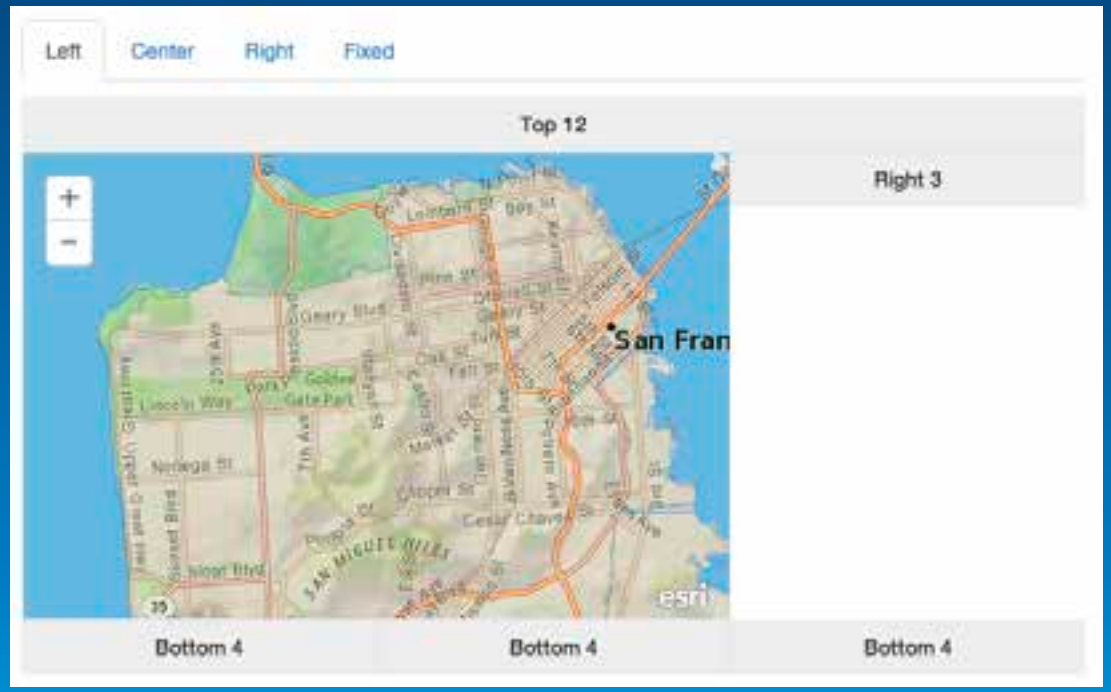
Bootstrap-map-js

ArcGIS JavaScript in a responsive web framework

- Bootstrap ver 3 framework
- Responsive map
- Resize and re-center
- Pop-ups, widgets
- Touch
- CSS Styles

Responsive grid styling

Define row and column behavior



Bootstrap-map-js example

```
<div class="container">
   <div id="mapDiv"></div>
</div>

…

<script>
    <!– Load Bootstrap Map – responsive map -->
     require(["esri/map", "…/src/js/bootstrapmap.js", "dojo/domReady!"],
       function(Map, BootstrapMap) {
           <!-- Get a reference to the ArcGIS Map -->
           var map = BootstrapMap.create("mapDiv",{
             basemap:"national-geographic",
             center:[-122.45,37.77],
             zoom:12
           });
      });
</script>
```

github.com/bootstrap-map-js

AGO Assistant

ArcGIS Online data transfer

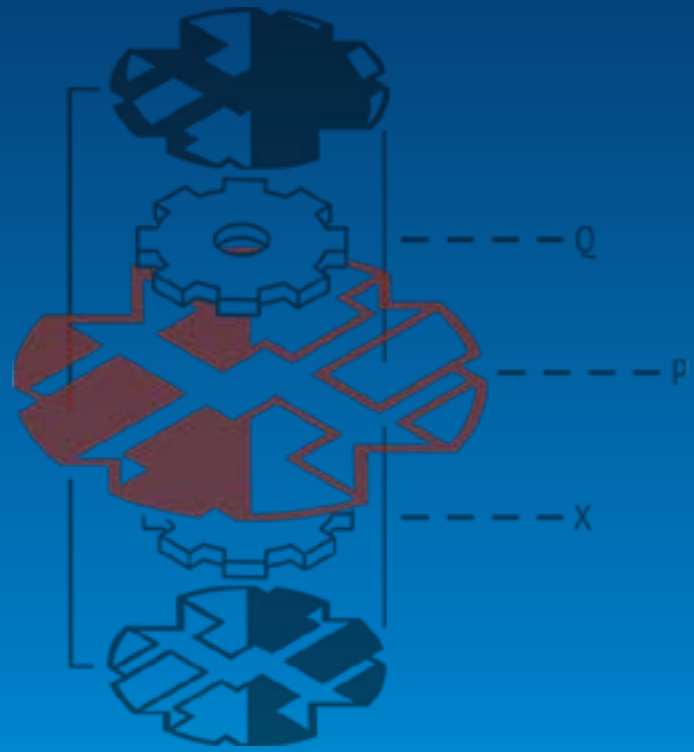
AGO Assistant

- Web tool to copy items from one AGO account to another
- Features
  - Inspect content (view JSON)
  - Update the URLs of services in a web map
  - Update the URL of a registered application or service
  - View user statsCopy
- Uses ArcGIS Online REST API

github.com/ago-assistant

Terraformer

JavaScript Geo Toolkit



Terraformer

- JavaScript Geo toolkit for working with data
- Key features
  - Geometry format conversions (GeoJSON)
  - Geometry operations
  - Coordinate system conversion
  - Store and access geo data
- Node.js and client-side JavaScript

Terraformer Modules

- Terraformer
- terraformer-arcgis-parser
- terraformer-wkt-parser
- terraformer-geostore
- terraformer-geostore-rtree
- terraformer-geostore-memory
- terraformer-geostore-localstorage

Terraformer: Geometry and Features

terraformer.js

```
// create a typed primitive from GeoJSON
var point = new Terraformer.Primitive({ "type": "Point",
"coordinates": [ 100, 1 ] });

// create a Geometry from coordinates or GeoJSON
var point = new Terraformer.Point( [ 10, 10 ] );
var ls = new Terraformer.LineString([ [ 10, 10 ], [ 20, 20 ]]);
var poly = new Terraformer.Polygon([
    [[100.0, 0.0], [101.0, 0.0], [101.0, 1.0], [100.0, 1.0]]]);
var circle = new Terraformer.Circle([-122.6764, 45.5165], 1000);

// creates a feature from a valid GeoJSON Object
var feature = new Terraformer.Feature({"type": "Point",
"coordinates": [ 10, 10 ]}, "properties": {"prop0": "value0"});
```

Terraformer: Geometric Operations

terraformer.js

```
// to Web Mercator and WGS84
primitive.toMercator();
primitive.toGeographic();

var box = poly.bbox;

multi.addPoint([ 10, 10 ]);
multi.insertPoint([ 10, 10 ],1);
multi.removePoint(1);
multi.get(1);

polygon1.within(polygon2);
polygon1.intersects(line);
polygon1.contains(point);
circle.contains(point);
```

Terraformer: WKT Conversion

terraformer-wkt-parser.js

```
// take a WKT representation and convert it into a primative
<script>
  var primitive = Terraformer.WKT.parse('LINESTRING (30 10, 10 30, 40
40)');
</script>

// take a primitive and convert it into a WKT representation
var polygon = Terraformer.WKT.convert(
  {
    "type": "Polygon",
    "coordinates": [
      [ [100.0, 0.0], [101.0, 0.0], [100.0, 1.0], [100.0, 0.0] ],
      [ [100.2, 0.2], [100.8, 0.2], [100.2, 0.8], [100.2, 0.2] ]
    ]
  }
);
```

Terraformer: ArcGIS JSON to GeoJSON

terraformer-arcgis-parser.js

```
<script>
   // take ArcGIS JSON and convert to Primitive or GeoJSON
   var primitive = Terraformer.ArcGIS.parse({
     x:"-122.6764",
     y:"45.5165",
     spatialReference: {
       wkid: 4326
     }
   });

   // take a Primitive or GeoJSON and convert it to ArcGIS JSON
   var point = Terraformer.ArcGIS.convert({
     "type": "Point",
     "coordinates": [45.5165, -122.6764]
   });
</script>
```

Terraformer: GeoStore

terraformer-geostore.js and terraformer-rtree.js

```
// In-memory geostore. Requires id property.
var store = new Terraformer.GeoStore({
  store: new Terraformer.GeoStore.Memory(),
  index: new Terraformer.RTree()
});

store.add(geojson, function(err, resp){
  // callback
});

store.update(geojson, function(err, resp){
  // callback
});

store.contains(geojson, function(err, resp){
  // callback
});
```

github.com/Esri/Terraformer

Geoservices-js

ArcGIS REST Service Library

Geoservices-js

- JavaScript library for accessing Geoservices REST end-points
- Key features
  - Communicate with ArcGIS REST services
  - Access ArcGIS Feature and Geocoding Services
  - Light-weight, pure JavaScript
  - Browser and Node.js
- Built on the Geoservices REST specification

Geoservices-js: Getting Started

geoservices.js

```
// Browser
<script src="browser/geoservices.js"></script>
<script>
  var client = new Geoservices();
</script>


// Node.js
var Geoservices = require('geoservices');
var client = new Geoservices();
```

Geoservices-js: FeatureService Info

geoservices.js

```
// Define parameters for a feature service
var params = {
  catalog: 'http://server6.arcgisonline.com/arcgis/rest/services',
  service: 'Census',
  type: 'MapServer',
  layer: 3
};

// Make request to the service
client.FeatureService( params , function (err, result) {
  if (err) {
    console.error("ERROR: " + err);
  } else {
    console.log("Got the FeatureService Metadata: ", result );
  }
});
```

Geoservices-js: FeatureService Query

geoservices.js

```
// Define query parameters
var query_params = {
  f: 'json',
  returnGeometry: true,
  where: '1=1',
  outSR: '4326'
};

// Request features
var fs = client.FeatureService( params , function(err,
data){
  fs.query( query_params, function( err, result ){
    if (err) {
      console.error("ERROR: " + err);
    } else {
      console.log("Features: ", result );
    }
  });
});
```

Geoservices-js: Geocoding

geoservices.js

```
// Geosearch
client.geocode({ text: "920 SW 3rd Ave, Portland, OR 97201" }, function
(err, result) {
  if (!err) {
    console.log(result.locations[0].feature.geometry.y + ", "
    result.locations[0].feature.geometry.x);
  }
});

// Reverse-geocoding
client.geocode.reverse({ location: "-122.67633,45.51673" },
function (err, result) {
  if (!err){
    console.log(result.address.Address + ", " + result.address.City);
  }
});
```

Geoservices-js: Batch Geocoding

geoservices.js

```
// Simple authentication only!
var client = new Geoservices();
client.authentication.authenticate('username', 'password', {
/* optional options */ }, callback);

// Batch geocoding
var batch = new client.geocode.Batch();

// add addresses to geocode
batch.geocode("123 Fake Street");
batch.geocode("456 Other Street");

// run the batch
batch.run(function (err, results) {
  console.dir(results);
});
```

github.com/Esri/geoservices-js

ArcGIS Services Adaptors

Node.js REST Implementations

Koop/node-geoservices-adaptor

- JavaScript ArcGIS REST service providers
- Features
  - Expose “your” service as an ArcGIS service
  - Easy to implement
  - Consumable by any ArcGIS client or API
    - ArcGIS Online, ArcGIS JS, Map Viewer, ArcMap…
  - Node.js implementation
- Follows the Geoservices REST specification

Koop: GitHub Provider Example

MVC

```
// routes/index.js
module.exports {
  'get /github/:user/:repo/FeatureServer/:layer/:method': {
    controller: 'github',
    action: 'featureservice'
  },
…
// model/github.js
var Geohub = require('geohub');
module.exports = {
  find: function( user, repo, file, options, callback ){
    var key = [ user, repo, file].join('/'),
        type = 'Github';
…
// controller/index.js
module.exports = {
  getRepo: function(req, res){
    var _send = function( err, data ){
…
```

Node-geoservices-adaptor: Example

ExpressJS

```
// citybikes.js
var cityBikesNetworksURL = "http://api.citybik.es/networks.json";

var newMapTemplate =
"http://www.arcgis.com/home/webmap/viewer.html?url=%s&source=sd";
…

idField: {
    value: function(serviceId, layerId) {
      return "id";
    }
  },
fields: {
    value: function(serviceId, layerId) {
return serviceId===allNetworksServiceId?networksFields:cityBikesFields;
    }
  },
```

github.com/esri/koop

github.com/esri/node-geoservices-adaptor

Esri-Leaflet

ArcGIS Services Plug-in

Leaflet

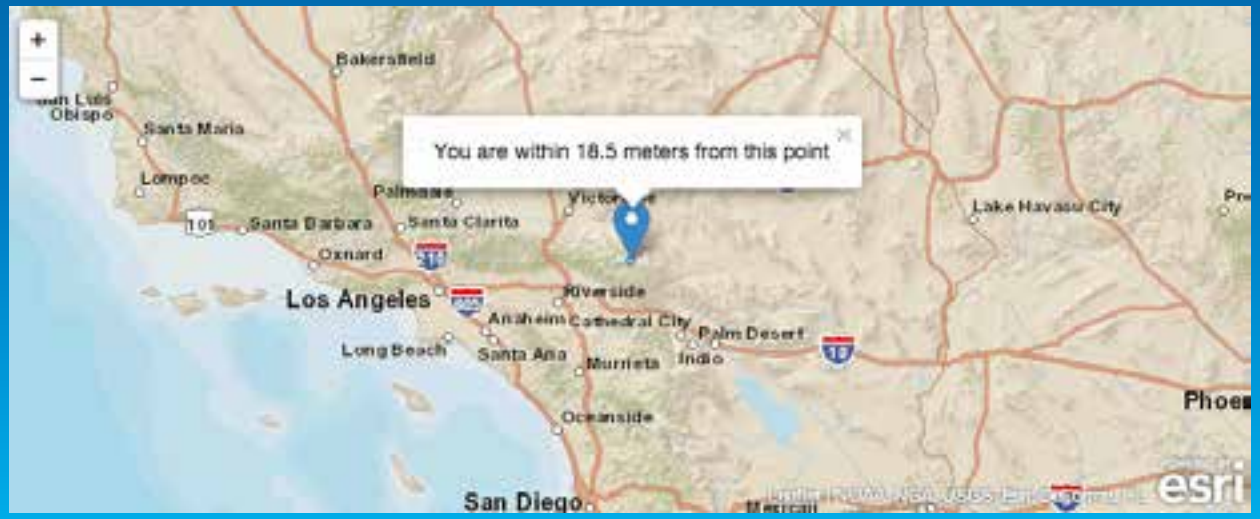
- Open source mapping library
- Pure JavaScript – 33kb
- Simple, easy to use, mobile friendly
- Many plug-ins

www.leafletjs.com

Esri-Leaflet

ArcGIS Online Services Plug-in

- Open source plug-in for ArcGIS Online services
- Extends L.class and namespace = L.esri.xxx
- Core borrows from Terraformer Geo Toolkit
- Samples use Geoservices-js



Esri-Leaflet: Example

Reference L.esri.xxx Library

```
<!DOCTYPE html>
<html>
  <head>
    <title>Esri Leaflet</title>
    <link rel="stylesheet" href="/the/path/to/leaflet.css" />
    <style>
      html, body, #map { width: 100%; height: 100%; }
    </style>
    <script src="/the/path/to/leaflet.js"></script>
    <script src="/the/path/to/esri-leaflet.min.js"></script>
  </head>
  <body>
    <div id="map"></div>
    <script>
        var map = L.map('map');
        L.esri.basemapLayer("Streets").addTo(map);
        map.setView([38.97993, -104.9794], 12);
    </script>
  </body>
</html>
```

Esri-Leaflet: ArcGIS Basemaps

L.esri.BasemapLayer = L.TileLayer.extend({…

```
// Load an ArcGIS basemap
var map = L.map('map').setView([37.75,-122.45], 12);
L.esri.basemapLayer("Topographic").addTo(map);

// Supported basemap types
//L.esri.basemapLayer("Streets").addTo(map);
//L.esri.basemapLayer("Oceans").addTo(map);
//L.esri.basemapLayer("NationalGeographic").addTo(map);
//L.esri.basemapLayer("Gray").addTo(map);
//L.esri.basemapLayer("GrayLabels").addTo(map);
//L.esri.basemapLayer("Imagery").addTo(map);
//L.esri.basemapLayer("ImageryLabels").addTo(map);
```

Esri-Leaflet: ArcGIS FeatureServices

L.esri.FeatureLayer = L.GeoJSON.extend({…

```
// Access ArcGIS FeatureService
var map = L.map('map').setView([45.52963623111275, -
122.67389774322508], 12);
L.esri.basemapLayer("Topographic").addTo(map);

var url =
'http://services.arcgis.com/rOo16HdIMeOBI4Mb/arcgis/rest/se
rvices/stops/FeatureServer/0’

L.esri.featureLayer(url);
```

Esri-Leaflet: Symbols

```
// Create FeatureLayer and define styles
L.esri.featureLayer(url, {
   style: function (feature) {
      return getStyle(feature);
   }).addTo(map);

function getStyle(feature) {
   var c,o = 0.5;
   switch (feature.properties.BIKEMODE) {
      case "Low traffic through street":
         c = "#007D7D";
         break;
      case "Bike boulevard":
         c = "#00FF3C";
         break;
      …
   }
 return {color: c, opacity: o};
}
```

Esri-Leaflet: Popups

```
// Create FeatureLayer and bind to popup
L.esri.featureLayer(featureServiceUrl, {
          onEachFeature: createPopup
        }).addTo(map);

// Define popup content - show all fields and values
function createPopup(geojson,layer) {
  if (geojson.properties) {
    var popupText =  "<div style='max-height:200px;'>";
    for (prop in geojson.properties) {
      var val = geojson.properties[prop];
      if (val) {
        popupText += "<b>" + prop + "</b>: " + val + "<br>";
      }
    }
    popupText += "</div>";
    layer.bindPopup(popupText);
  }
}
```

Esri-Leaflet: DynamicMapLayer

```
// ArcGIS Server Dynamic Map Service - Hurricane Tracks
dynLayer =
L.esri.dynamicMapLayer("http://tmservices1.esri.com/arcgis/re
st/services/LiveFeeds/Hurricane_Recent/MapServer", {
layers:[0,1] });

// Identifying Dynamic Map Service Features
map.on("click", function(e) {
    dynLayer.identify(e.latlng, {
        layerDefs: {
            0: "STORMNAME='ANDREA'",
            1: "STORMNAME='ANDREA'"
          }
        }, function(data) {
            popupText =  "<center><b>" +
         data.results[0].attributes.STORMNAME + "</b><br>" +
         data.results[0].attributes.STORMTYPE + "</center>”;
       L.popup().setLatLng(e.latlng).setContent
    (popupText).openOn(map);
        }
    });
});
```

Esri-Leaflet: ClusterFeatureLayer

esri-leaflet.js + clustered-feature-layer.js

```
// Reference cluster plug-in and esri feature layer
<script
src="lib/markercluster/leaflet.markercluster.js"></script>

<script src="lib/esri-leaflet/extras/clustered-feature-
layer.js"></script>

// Create and add a new feature cluster layer
var fl =
L.esri.clusteredFeatureLayer("http://services.arcgis.com/rO
o16HdIMeOBI4Mb/arcgis/rest/services/stops/FeatureServer/0",
{
    cluster: new L.MarkerClusterGroup(),
        onEachMarker: function(geojson, marker) {
          marker.bindPopup("<h3>"+
             geojson.properties.stop_name+"</h3><p>
             Stop ID: "+geojson.properties.stop_id+"</p><p>”
        +geojson.properties.stop_desc+"</p>")
         }
}).addTo(map);
```

Esri-Leaflet: FeatureService Query

esri-leaflet.js + geoservices.js

```
// Access feature service directly and query (geoservices.js)
var fs = new
GeoServices.FeatureService({url:featureServiceUrl}, function
(err, results) {
    var queryOptions = document.getElementById("query");
    var query = queryOptions.text;
    var queryEnvelope =

JSON.stringify(L.esri.Util.boundsToExtent(map.getBounds()));
    // Build query parameters
    var params = {
            f:"json”, where: query,
        geometry: queryEnvelope,
        spatialRel: "esriSpatialRelIntersects”,
            returnGeometry:true, outSR: 4326, outFields:"*"
     };
     // Query the feature service
     fs.query(params, function (err, results) {
       addFeaturesToMap(results);  // Manual
    }
}
```

Esri-Leaflet: Geocoding

esri-leaflet.js + geoservices.js

```
// Reference geoservices.js
<script src="lib/geoservices/geoservices.js"></script>
…
var GeoServices = new Geoservices.Geoservices({});
var options = {
     text:searchString,
     outFields: "Loc_name,Place_addr",
     bbox: mapBounds }

// Add geocodes to map
GeoServices.geocode(options, function (err,result) {
    for (var i = 0; i < result.locations.length; i++) {
         var place = result.locations[i];
         var pt = new L.LatLng(place.feature.geometry.y,
place.feature.geometry.x);
       var marker = L.marker(pt).bindPopup(place.name + "</br>" +
place.feature.attributes.Place_addr);
         layerPlaces.addLayer(marker);
    }
}
```

Other stuff

- Retina support (sort of…)
- Layers
  - TiledMapLayers
  - HeatmapLayer
  - DemographicLayer
- Controls
  - esri-leaflet-geocoder

esri-leaflet

Esri-Leaflet: Holly Grail?

- No webmap support
- No dijits (Legend, Swipe, Popup, Gauge, AttributeInspector…)
- No toolbars (Draw, Edit, Nav…)
- Symbol mapping
- Renderers
- Editing and Updates
- Geometry operations and types
- Tasks
- …

Authenticating Apps

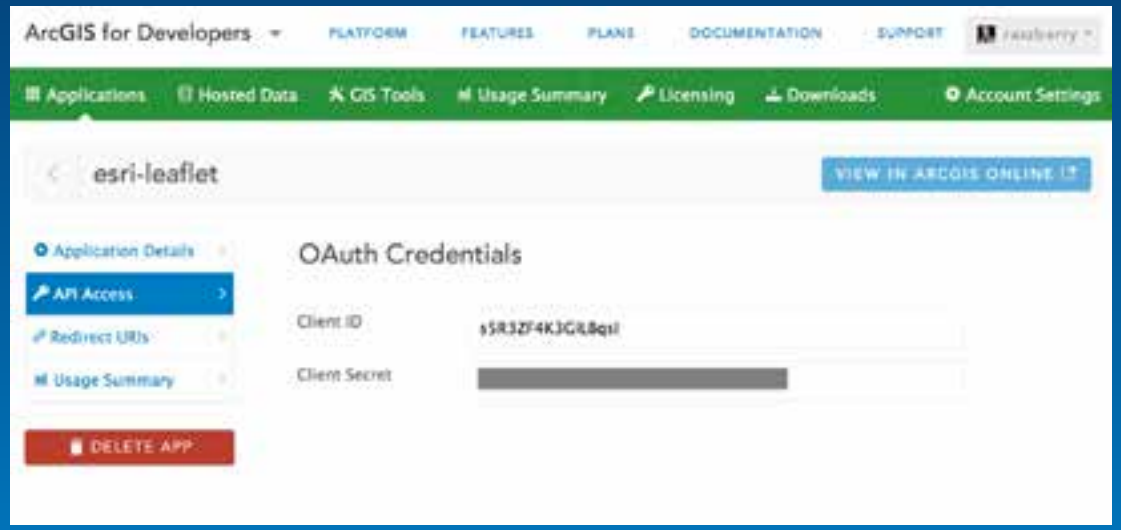
Accessing Secure Services

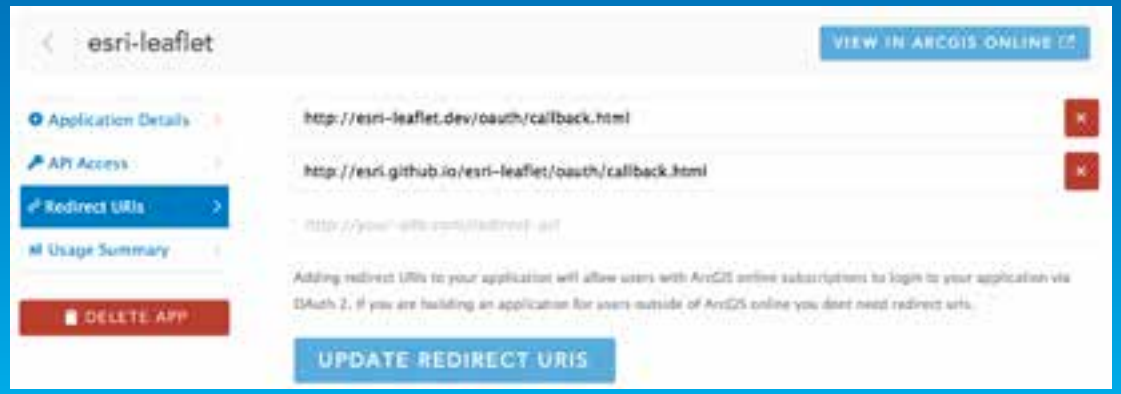
OAuth 2.0

- Allows apps to securely access data on behalf of users
- Generate an access token
- Browser app makes API calls to secure services with token
- No server-side code required

Step 1: Register App on ArcGIS for Developers

- ArcGIS Developer subscription
- Create a new app
- Get Client ID
- Add re-direct URIs





Step 2: Create a login page

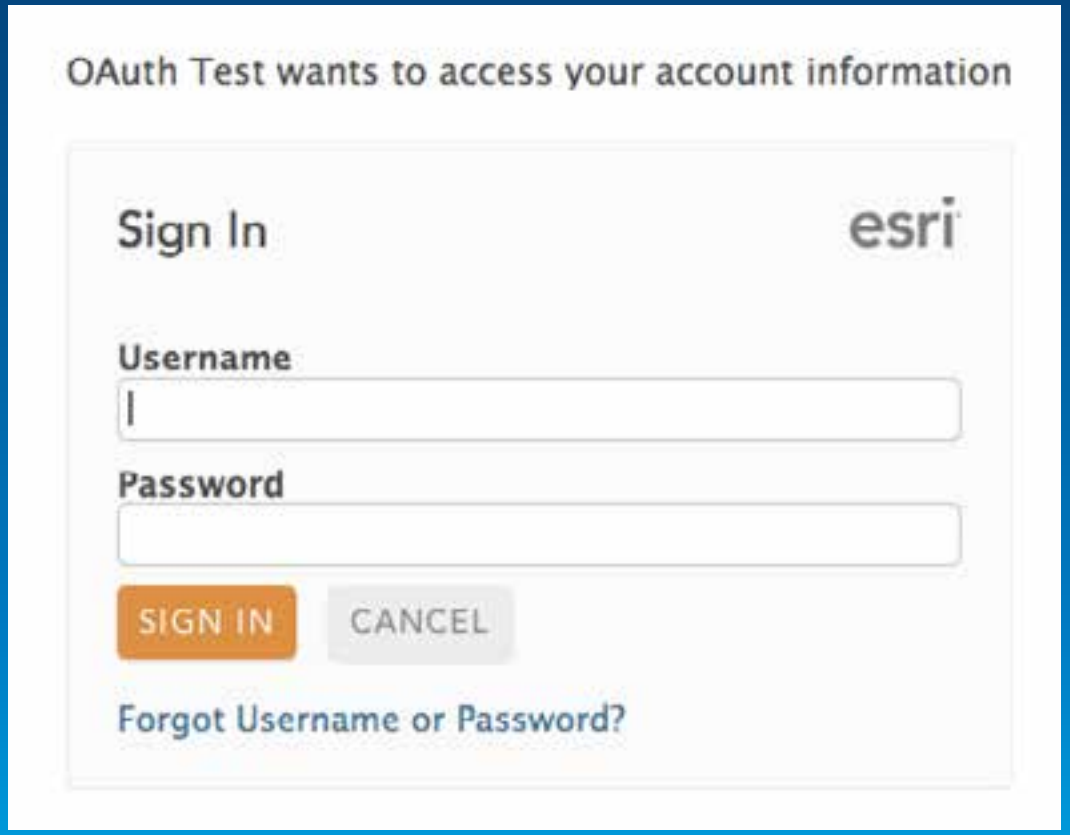
```
// Client ID
var clientID = 's5R3ZF4K3GILBqsI';

// URL
var url = "https://www.arcgis.com/sharing/oauth2/authorize?client_id=";

// URI
var uri = encodeURIComponent(window.location.origin)+"%2Foauth%2Fcallback.html”;

function startAGOOAuth() {
  window.open(url + clientID
    &redirect_uri=” + uri,
    +"&response_type=token&expiration=20160
    "oauth-window", "height=400,width=600");
}
```

The authorization page



Step 3: Make Requests with Token

```
var accessToken;

window.oauthCallback = function(token) {
  accessToken = token;
}

// Access services with token
L.esri.get("http://route.arcgis.com/arcgis/rest/services
  World/Route/NAServer/Route_World/solve", {
    token: accessToken,
    stops: s.lng+","+s.lat+"; "+e.lng+","+e.lat,
    outputLines: 'esriNAOutputLineTrueShape'
    }, function(response){
…
```

esri-leaflet/directions

Licensing

ArcGIS Developer Subscriptions

Licensing

- **Free ArcGIS Developer Subscription**
  - Testing and development
  - Public deployments (non-commercial, non-government)
  - 50 credits/mo.
- **Paid ArcGIS Developer or ArcGIS Organization Subscription**
  - Private deployments
  - Public deployments
  - Commercial (generate revenue) and government
  - 200 credits/mo.

alaframboise.github.io

Summary

- Quickstart-map-js
- Bootstrap-map-js
- AGO-assistant
- Terraformer
- Geoservices-js
- Koop/node-geoservices-adaptor
- Esri-Leaflet
- …

“Building Geo Apps better together”