Esri International Developer Summit

Palm Springs, CA

Esri’s Geotrigger Service

iOS and Android Tips & Tricks

Ryan Arana, Court Fowler, Josh Yaganeh
Esri R&D Center, Portland

Geotrigger Service

developers.arcgis.com/geotrigger-service

Geotrigger SDK Components

- Manager / Service
  - Wraps native location services on each platform
  - Provides a set of tracking profiles for low, medium, and high power consumption
  - Push Notification Handling
- API Client
  - Provides access to the Geotrigger Service API, allowing you to manage devices, triggers, and tags

Geotrigger SDK for iOS

- Get the SDK from Github or CocoaPods
  - `pod 'GeotriggerSDK'`
- Make sure your app has 'Location Updates' and 'Background Fetch' background modes enabled
- Call one of the `setupWithClientId:` methods

Start the Geotrigger Manager

```
[AGSGTGeotriggerManager setupWithClientId:kClientId
                                isProduction:NO
                                        tags:@[@"test"]
                                  completion:^(NSError *error) {
    if (error == nil) {
        // Turn on location tracking in adaptive mode
        [AGSGTGeotriggerManager sharedManager].trackingProfile =
                kAGSGTTrackingProfileAdaptive;
    } else {
        NSLog(@"Error setting up GeotriggerManager: %@", error);
    }
}];
```

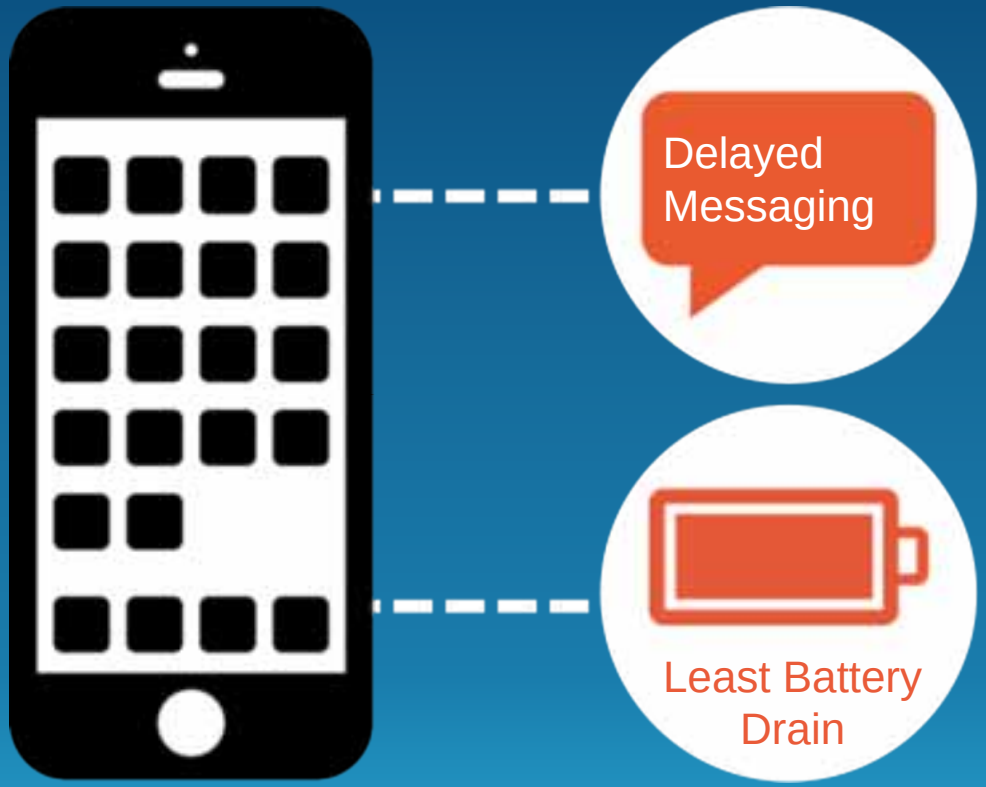
Geotrigger SDK for Android

- Get the Sample App and SDK from GitHub:
  - `github.com/esri/geotrigger-sdk-android`
- Make sure to change the package name in the sample or add the required permissions to your existing `AndroidManifest.xml`
- Use the `GeotriggerHelper` to start the Geotrigger Service

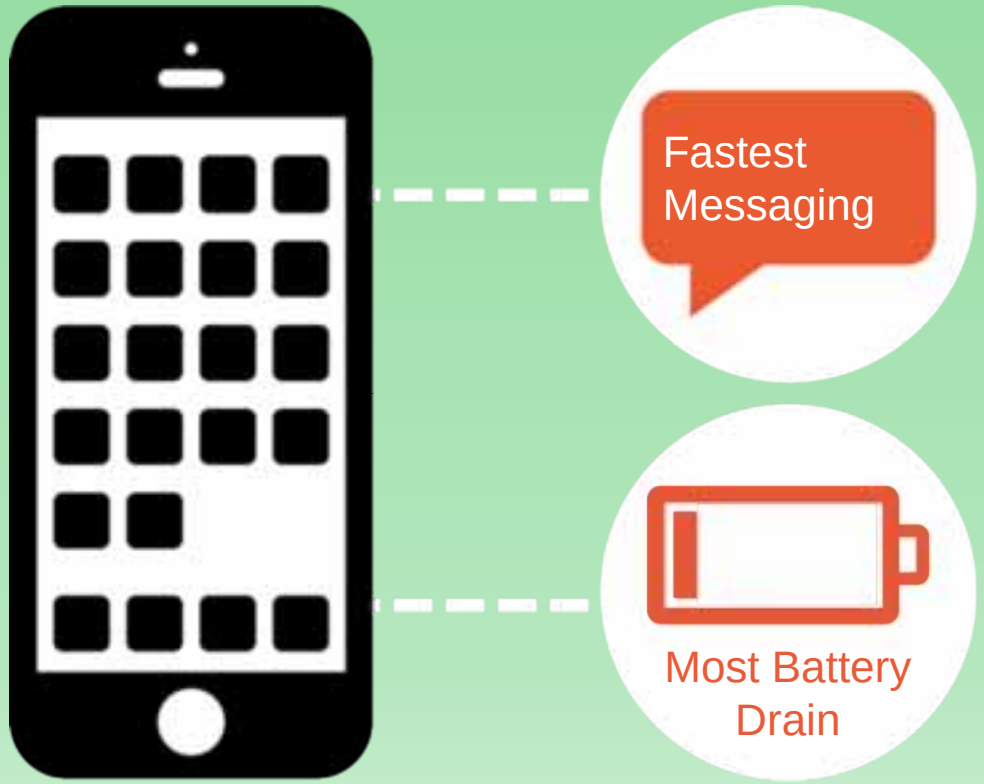
Start the Geotrigger Service

```
GeotriggerHelper.startGeotriggerService(
                    context,
                    AGO_CLIENT_ID,
                    GCM_SENDER_ID,
                    TAGS,
                    GeotriggerService.TRACKING_PROFILE_ADAPTIVE);
```

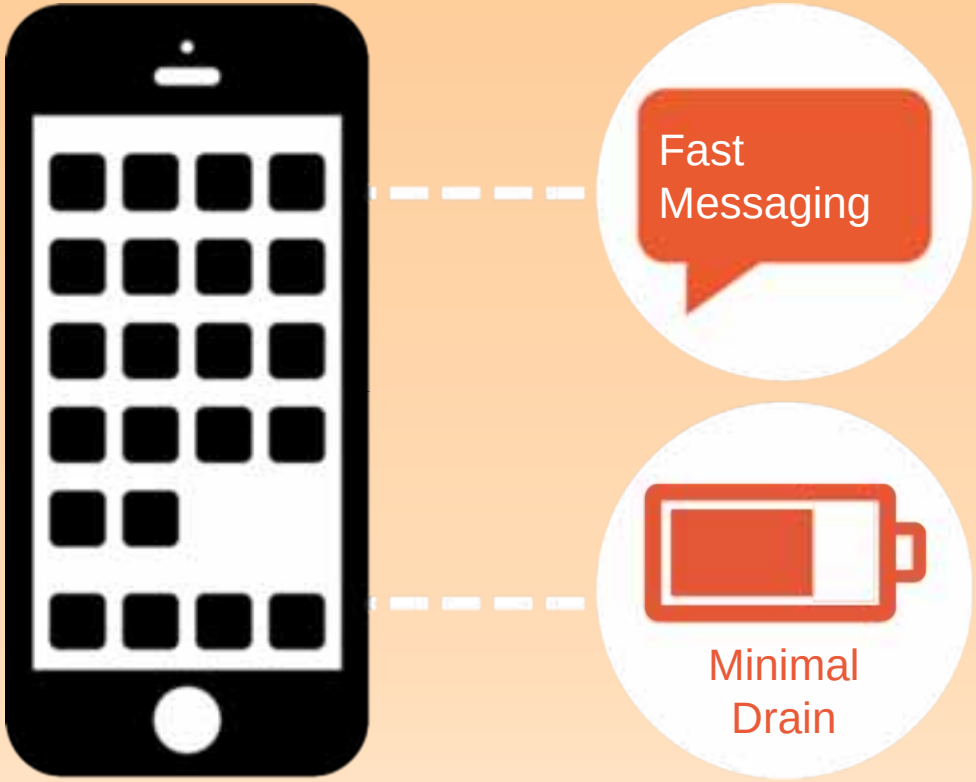
Rough Mode



Fine Mode

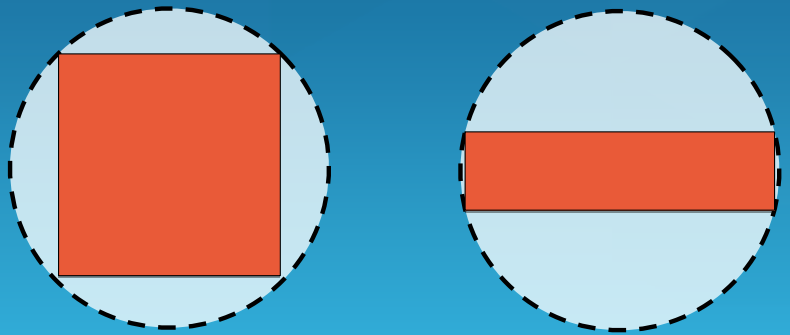


Adaptive Mode



Trigger Size & Shape

- 100m radius is recommended in Adaptive mode
- iOS & Android native geofences are point and radius
- Polygons and Multi-polygons will be approximated by simple circles.



Handling Events

- The Geotrigger Manager / Service can notify your app when one of the following happens:
  - Location updates are received from the OS
  - Locations are uploaded to the Geotrigger API
  - Tracking profile is changed

Geotrigger Broadcast Receiver

- Instantiate the Broadcast Receiver in `onCreate`

```
mGeotriggerBroadcastReceiver = new GeotriggerBroadcastReceiver();
```

- Register the receiver in `onResume`

```
registerReceiver(mGeotriggerBroadcastReceiver,
                 GeotriggerBroadcastReceiver.getDefaultIntentFilter());
```

- Unregister the receiver in `onPause`

```
unregisterReceiver(mGeotriggerBroadcastReceiver);
```

Respond to location updates

```
[AGSGTGeotriggerManager sharedManager].didReceiveLocationUpdates =
    ^(NSArray *locations) {
        NSLog(@"Received location updates: %@", locations);
    };
```

```
public class GeotriggerActivity extends Activity implements
        GeotriggerBroadcastReceiver.LocationUpdateListener {

    @Override
    public void onLocationUpdate(Location loc, boolean isOnDemand) {
       Log.d(TAG, "Location update received!");
    }
}
```

Respond to Tracking Profile Changes

```
[AGSGTGeotriggerManager sharedManager].didChangeTrackingProfile =
    ^(NSString *old, NSString *new) {
        NSLog(@"Changed profile from %@ to %@", old, new);
    };
```

```
public class GeotriggerActivity extends Activity implements
        GeotriggerBroadcastReceiver.TrackingProfileChangeListener {

    @Override
    public void onTrackingProfileChange(String previous, String current) {
       Log.d(TAG, String.format(
            "Changed profile from %s to %s", previous, current));
    }
}
```

Making API Requests

- Use cases include:
  - Applying tags to a Device
  - Creating Triggers
  - Getting a list of existing Triggers
- Wait for the ready event before making requests

Geotrigger Broadcast Receiver

```
public class GeotriggerActivity extends Activity implements
        GeotriggerBroadcastReceiver.ReadyListener {

    @Override
    public void onReady() {
        Log.d(TAG, "GeotriggerService Ready!”);

        // Make API requests here (or track state that you’ll check elsewhere)
    }
}
```

Adding a Tag to a Device

```
[[AGSGTApiClient sharedClient] postPath:@"device/update"
                             parameters:@{ @"addTags": @[@"pinball-machines"] }
                                success: ^(NSDictionary *resp) {
                                    NSLog(@"Device updated: %@", resp);
                                }
                                failure: ^(NSError *error) {
                                    NSLog(@"Request failed: %@", error);
                                }];
```

Adding a Tag to a Device

```
JSONObject params = new JSONObject();
try {
    params.put("addTags", "pinball-machines");
} catch (JSONException e) {
    Log.e(TAG, "Error creating device update parameters.", e);
}

GeotriggerApiClient.runRequest(context, "device/update", params,
        new GeotriggerApiListener() {
            public void onSuccess(JSONObject data) {
                Log.d(TAG, "Device updated: " + data.toString());
             }

            public void onFailure(Throwable error) {
                Log.d(TAG, "Request failed.", error);
             }
        });
```

Creating a Trigger

```
AGSGTTriggerBuilder *builder = [AGSGTTriggerBuilder new];
builder.triggerId = @"indiana-jones-the-pinball-adventure";
builder.tags = @[@"pinball-machines"];
builder.direction = @"enter";
[builder setGeoWithLatitude:latitude
                  longitude:longitude
                   distance:distance];
builder.notificationText =
    @"You are near a pinball machine! Indiana Jones: The Pinball Adventure";
builder.notificationUrl = @"http://www.ipdb.org/machine.cgi?gid=1267";
NSDictionary *params = [builder build];

[[AGSGTApiClient sharedClient] postPath:@"trigger/create"
                             parameters:params
                                success:nil
                                failure:nil];
```

Creating a Trigger

```
JSONObject params = new TriggerBuilder()
        .setTriggerId("indiana-jones-the-pinball-adventure")
        .setTags("pinball-machines")
        .setGeo(latitude, longitude, distance)
        .setDirection(TriggerBuilder.DIRECTION_ENTER)
        .setNotificationText("You are near a pinball machine! " +
                             "Indiana Jones: The Pinball Adventure")
        .setNotificationUrl("http://www.ipdb.org/machine.cgi?gid=1267")
        .build();

GeotriggerApiClient.runRequest(context, "trigger/create", params,
    new GeotriggerApiListener() {
        public void onSuccess(JSONObject data) { Log.d(TAG, data.toString()); }

        public void onFailure(Throwable error) { Log.e(TAG, "Error!", e); }
    });
```

Updating Released Apps

- Updates to our SDK will be significant:
  - Battery improvements
  - Bug fixes
  - New features / Events that can handled

Learn More!

Drop by the ArcGIS for Developers Island to ask questions about the Geotrigger Service.

Don’t forget to rate this talk!
www.esri.com/events/devsummit/session-rater

Understanding our world.