ArcGIS Runtime SDK for Java: A Beginner’s Guide

Vijay Gandhi

Eric Bader @ECBader

Outline

- Intro to ArcGIS Runtime SDKs
- Get started: download and install the SDK
- Tour of the functionality of the API
- Basics of building a map application
- Online workflow: services, ArcGIS Online, web maps
- Offline workflows: local data, create and update
- Deployment and licensing

ArcGIS Runtime

Runtime built using C++

EXPLOITS THE CAPABILITIES OF THE DEVICE

Functionality exposed to developers via an API native to the platform

INTUITIVE TO LEARN

Common functionality set and conceptual model

EASES MULTI PLATFORM DEVELOPMENT

Device Platforms

Java SE

ArcGIS Runtime SDK for Java

- Integrates with the ArcGIS Platform

- Build native apps for Windows and Linux
  - Windows 8 / 7
  - Ubuntu, RedHat
  - 32 and 64 bit Windows, 32 and 64 bit Linux

- Java SE API, Swing and JavaFX (Beta)
- Eclipse plugin
- Developed alongside Runtime SDK for Android

ArcGIS Runtime SDK for Java

- **Get it:** free download on developers.arcgis.com/java
- What you get:
  - Set of jars to code against
  - Open-source toolkit (mainly UI components)
  - Eclipse plugin, includes map application template
  - Runtime tools: deploy / debug
  - Documentation: Guide, API reference
  - Tons of samples

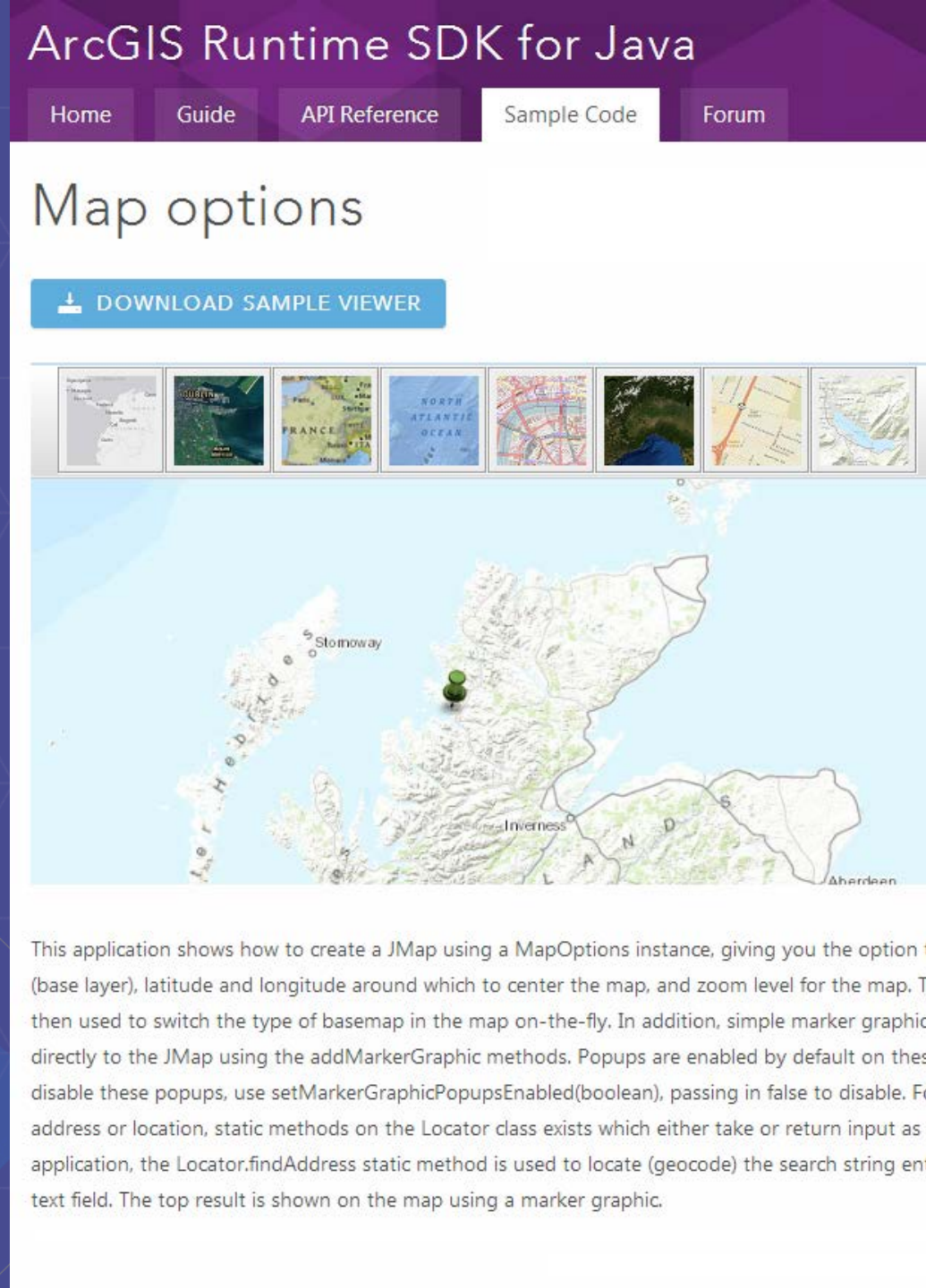
- **Get help:** Guide, API Reference, Forum
- **Give feedback:** website pages, sessions, Forum

Feedback on this topic?

DEMO

The SDK

Vijay



What you can do

- Mapping
- Searching
  - query, find, identify, address finding, locating addresses by coordinates
- Editing
- Geometry operations
- GPS
- Network Analysis (route finding, drive times, closest facility)
- Spatial Analysis (Geoprocessing)
- Advanced Symbology

→ Online and offline

DEMO

Functionality Tour

Vijay Gandhi

Time slider

Shows how to display a time-aware layer from a map package (.mpk) as a local dynamic layer.

Local tiled layer

Tiled map service layer

Loads an ArcGIS Server tiled map service from its URL.

OpenStreetMap layer

OpenStreetMap custom layer

Shows how to display a custom tiled layer adhering to the OpenStreetMap tile naming conventions using the OpenStreetMapLayer.

'No Data' tiles

Download tile cache

Shows how to download a tile cache from an online service which supports the 'exportTiles' operation.

Tiled image service layer

Map and layers

- JMap : Swing component (`JComponent`)
- FXMap (BETA): JavaFX component
- Spatial Reference (SR)
  - First layer in map (basemap) sets SR
  - Subsequent layers reprojected to map SR
- Extent
  - Envelope in SR coordinates
  - Set initial extent to area of interest
- Layers
  - Collection of layer classes with different behaviours
  - All inherit from `Layer` class
  - Order in map is order in which they are added
  - Add a layer to map's layer list:

```
jMap.getLayers().add(Layer);
```

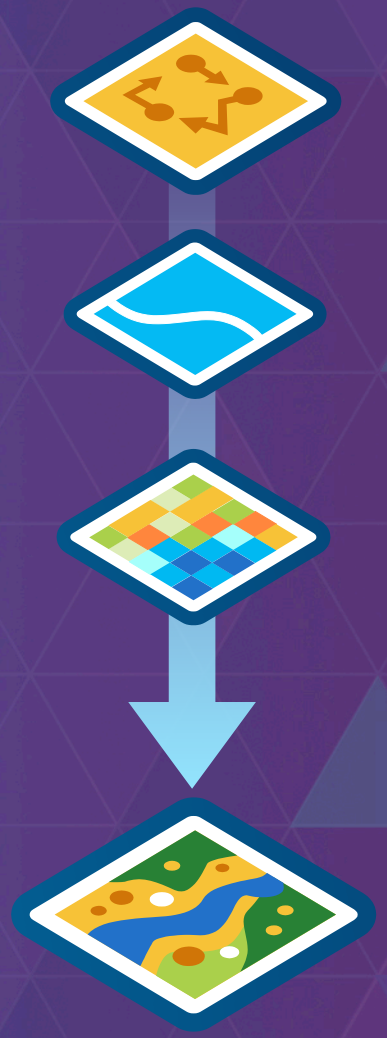
Build a map

- “Live” Data
  - Graphics layers
- Operational Data
  - Dynamic layers / Feature layers
- Basemap
  - Tiled layers
- Map

DEMO

Build a map

Vijay Gandhi



Work with graphics

- API classes: `GraphicsLayer, Graphic`
- A graphics layer contains graphics (you guessed it!)
- Graphic class is immutable: so don't hold references to Graphic objects
- Update / move / remove graphics using methods on `GraphicsLayer`
- Work with graphics via the layer using their unique ID

```
id = addGraphic(Graphic)
graphic = getGraphic(id)
...
updateGraphic(id, Graphic)
updateGraphic(id, Symbol)
updateGraphic(id, Geometry)
...
removeGraphic(id)
setGraphicVisible(id, visible)
select(id)
...
```

Interact with the map

- Extend MapOverlay class, implement the methods you need:
  - onMouseClicked, onMouseMoved, onMouseDragged, ...
  - override onPaint to draw onto map

- Use the toolkit overlays!
  - Including overlays for:
    - editing
    - popups
    - scale bar & navigator
    - hit tests (responding to graphics being clicked)

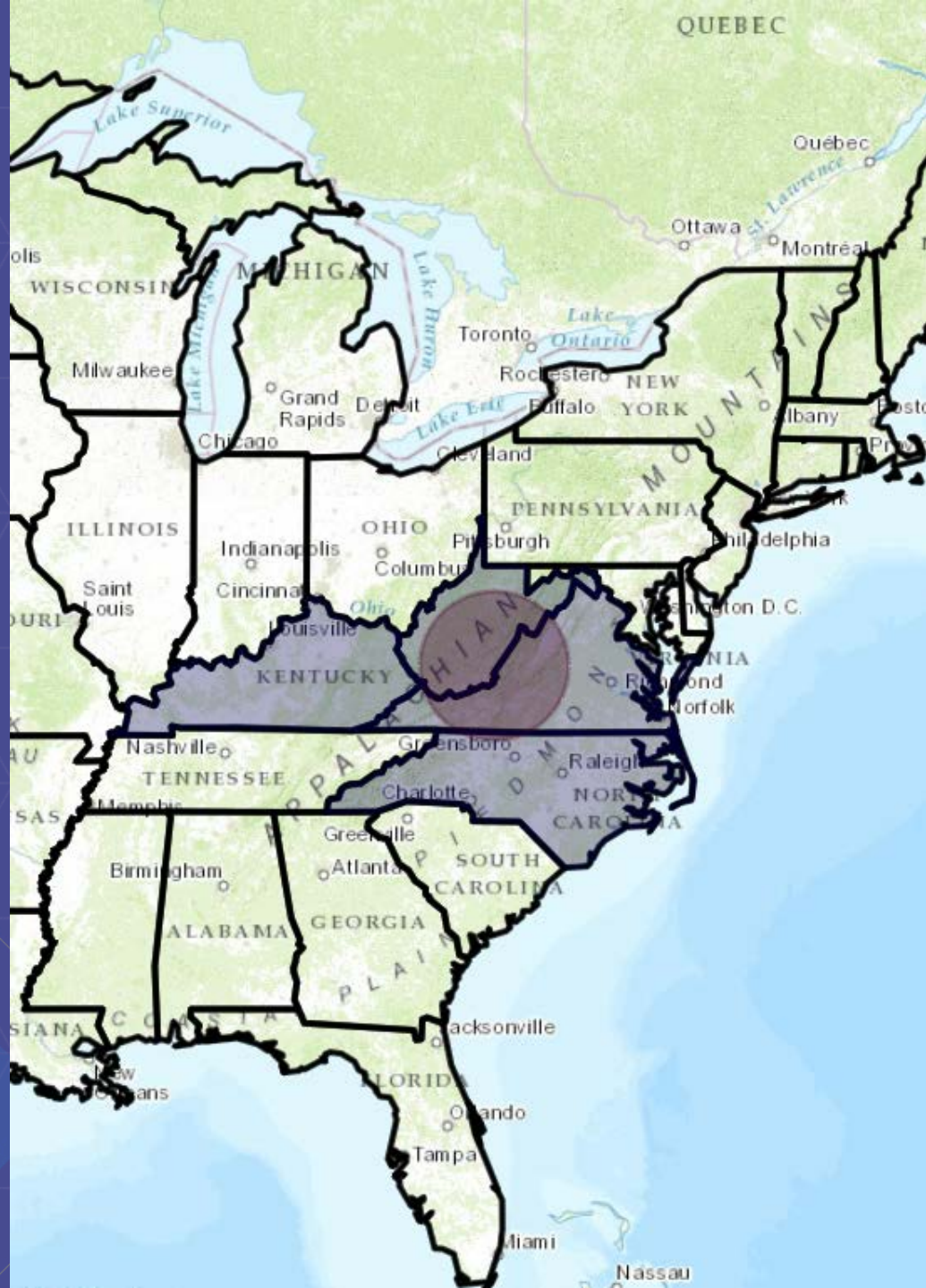
- Add an overlay to the JMap:

```
jMap.addMapOverlay(...);
```

DEMO

Interact with the map

Vijay Gandhi



Online and offline workflows

Online

- ArcGIS for Server services
- ArcGIS Online (web maps)
- Portal for ArcGIS

  - Basemaps: map services
  - Dynamic layers: map services
  - Feature layers: feature services
  - Geocoding: geocode services
  - Route finding: network analyst service
  - Analysis: geoprocessing services

Offline

- ArcGIS for Desktop: prepare data
- Download data from online services
- Local Server (services)

  - Basemaps: local tile cache
  - Dynamic layers: local map services (mpk)
  - Feature layers: local geodatabase or shapefiles
  - Geocoding: local geocoding
  - Route finding: local routing
  - Analysis: geoprocessing services (gpk)

Online workflows

- ArcGIS Services
  - REST API
  - Create via ArcGIS for Desktop, ArcGIS for Server
  - Map services
    - tiled layers, dynamic layers
  - Feature services
    - feature layers, editing, search tasks (query)
  - Image services
    - image service layers
  - Geocode services
    - geocode task
  - Network Analysis services
    - route task, closest facility task, service area task

- Other online data sources: WMS, OpenStreetMap, Bing basemaps, KML

WebMap and Portal

- Open via web map ID, Portal, user credentials if secure
  - get ID from URL
- Retrieve web map via Portal API
  - query for web map items on a Portal
- Create a `WebMap` instance then load into `JMap`:

```
WebMap webmap = new WebMap("webmap_id");
jMap.loadWebMap(webmap);
```

- `JMap` loads all the web map's layers
  - JSON of web map passed to client API, displays the layers according to order, rendering info, popup info, etc.

DEMO

WebMap and Portal

Vijay

Offline workflows

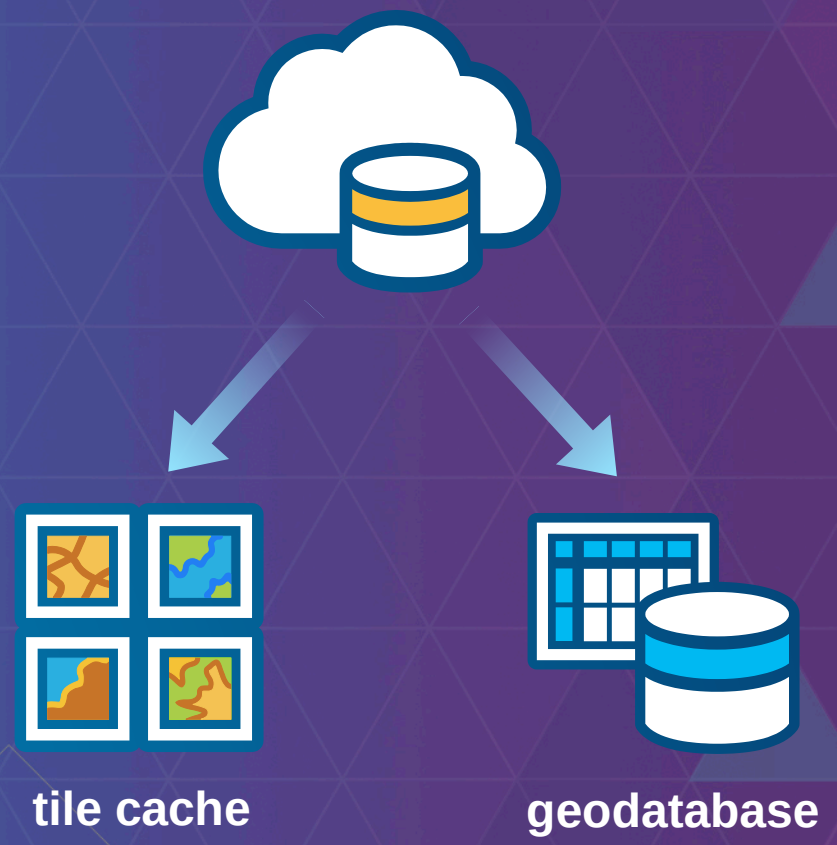
- Create data from services: tile caches, geodatabases
- Offline routing
- Offline geocoding

- Geometry operations done locally via API
- Local Server for offline geoprocessing
- Local Server for offline dynamic map services

Offline

- Services pattern: create from ArcGIS services
- Desktop pattern: create in ArcGIS for Desktop

- Create local tile caches (basemaps)
- Create local geodatabases
  - geodatabase for storing feature data locally
  - edit offline
  - query offline
  - sync edits back with service



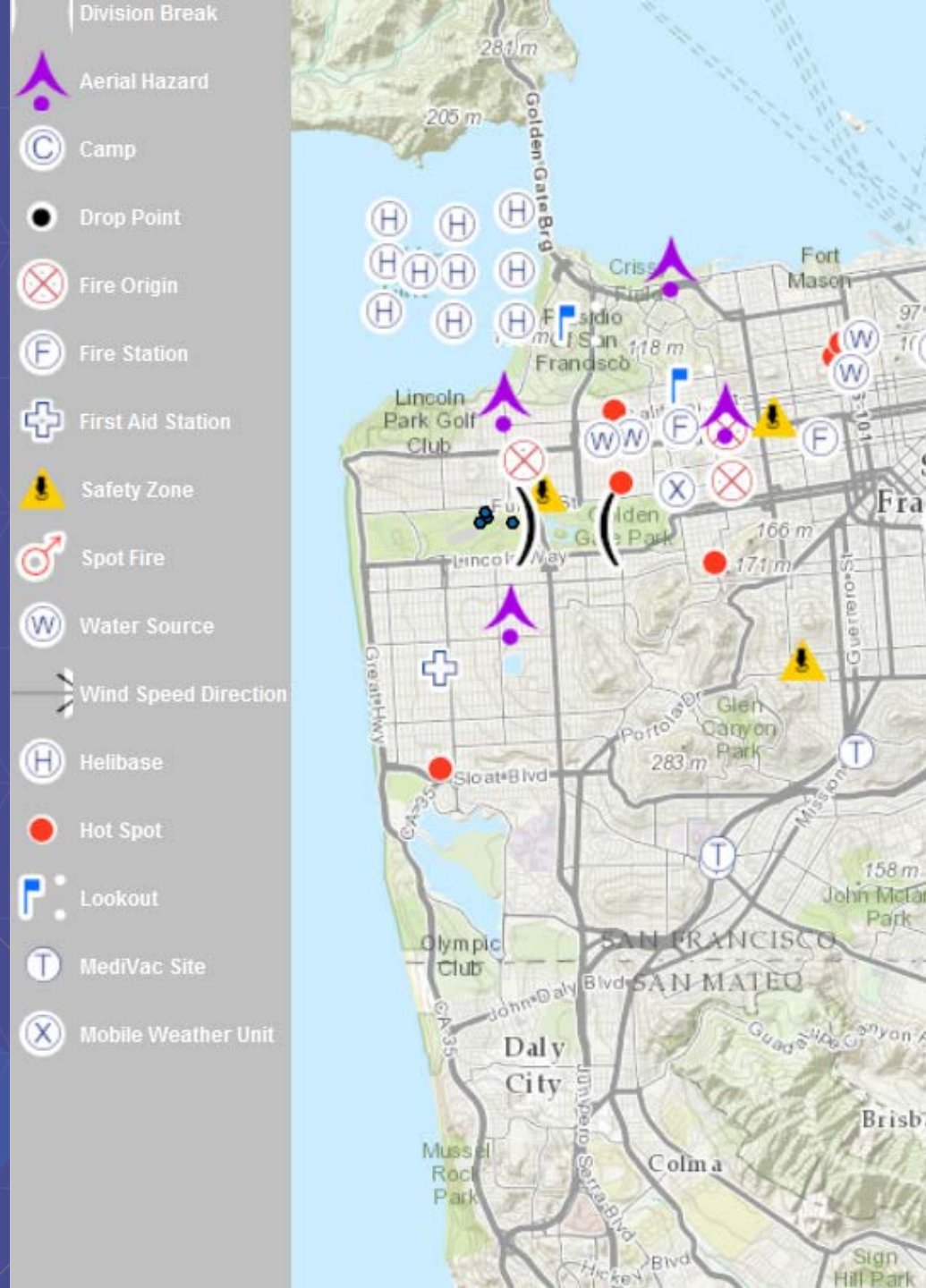
DEMO

Offline analysis

DEMO

Offline editing

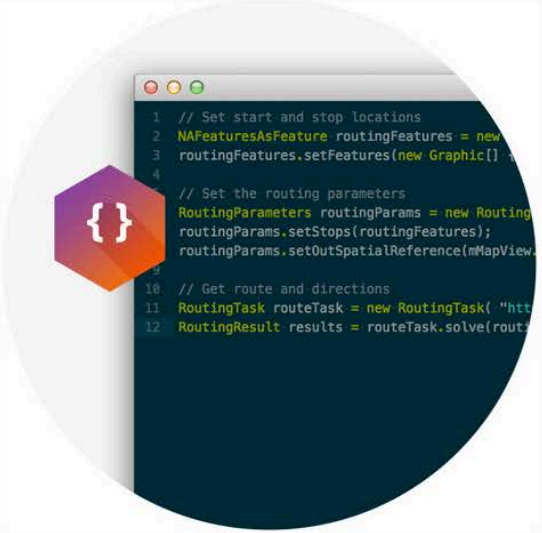
Vijay Gandhi



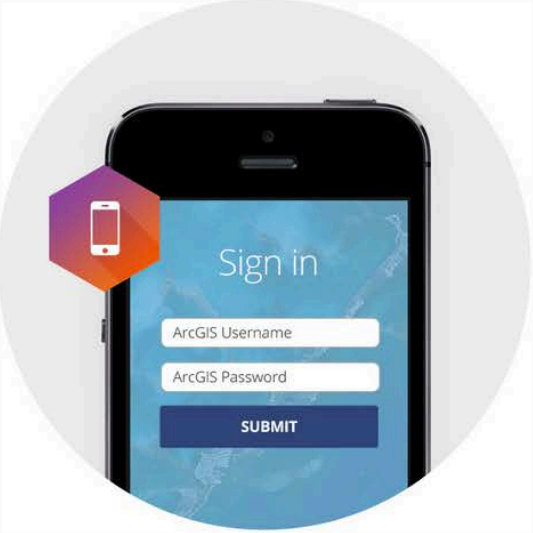
Runtime Licensing

Development and Deployment Workflow

1. Download and Install



2. Develop and Test

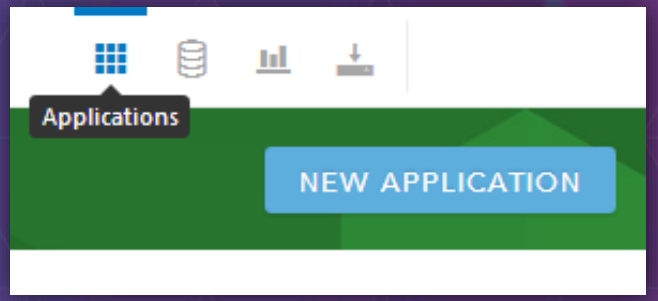


3. Deploy and Distribute

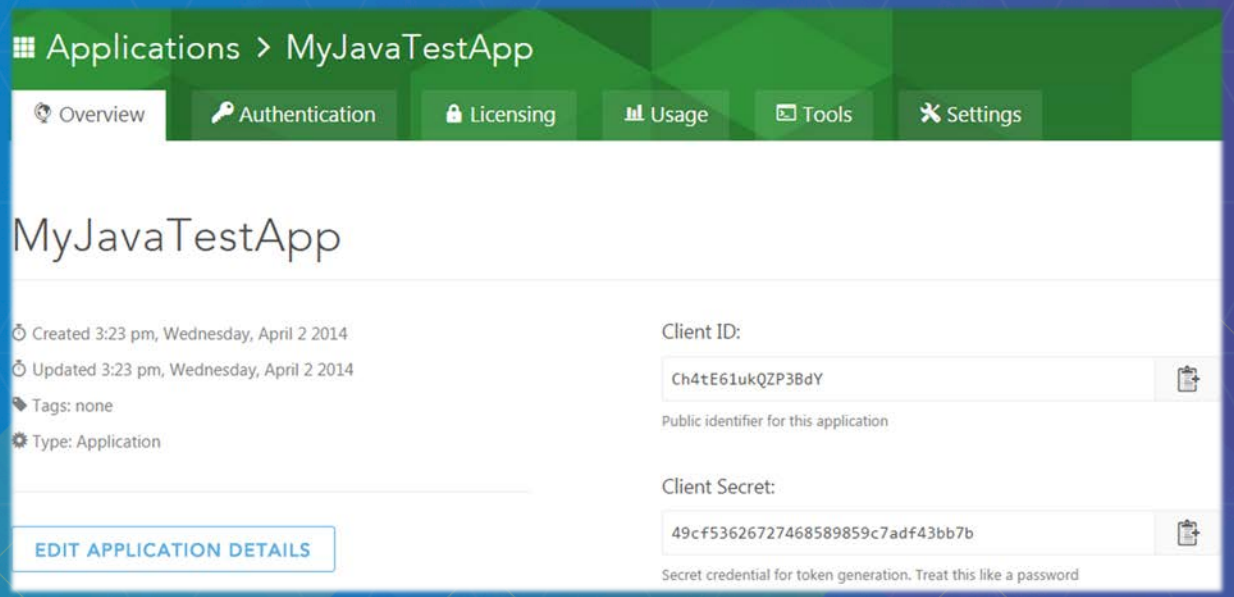
License your app at Basic level

1. Go to developers.arcgis.com and log in (or create a developer account)
2. Create a New Application (or select existing)
3. Click on Runtime SDK Licensing
4. Copy the Client ID and set it in your app

2.



3.



4.

```
// set the client ID
ArcGISRuntime.setClientID("myClientID");
```

License your app at Standard level

2 ways:

1. Use an organization account (ArcGIS Online or Portal for ArcGIS)
   - Requires users of your app to log in with their account

2. Use a license string obtained from Customer Service or your international distributor
   - License burned into the app
   - Extensions can also be added with this option (e.g. Local Server geoprocessing)

** You must use workflow 2 if you want to license any extensions **

Next sessions

- Tuesday, 4:30pm – 5:00pm, Demo Theatre 1 – Oasis 1

ArcGIS Runtime SDK for Java: Let's Build a JavaFX

- Wednesday, 10:30am – 11:30am, Smoketree A - E

ArcGIS Runtime SDK for Java: Advanced Topics

- Thursday 2:30pm – 3:30pm, Mojava Learning Center

ArcGIS Runtime SDK for Java: Animating Thousands of Graphics and Features

- Thursday 2:30pm – 3:30pm, Mojava Learning Center

Building Native Apps that Target Multiple Platforms

- Friday 8:30am – 9:30am, Primrose A

The Road Ahead: ArcGIS Runtime

Rate this session at
http://www.esri.com/RateMyDevSummitSession

esri

Understanding our world.