

ArcGIS Runtime SDK for Java: A Beginner's Guide

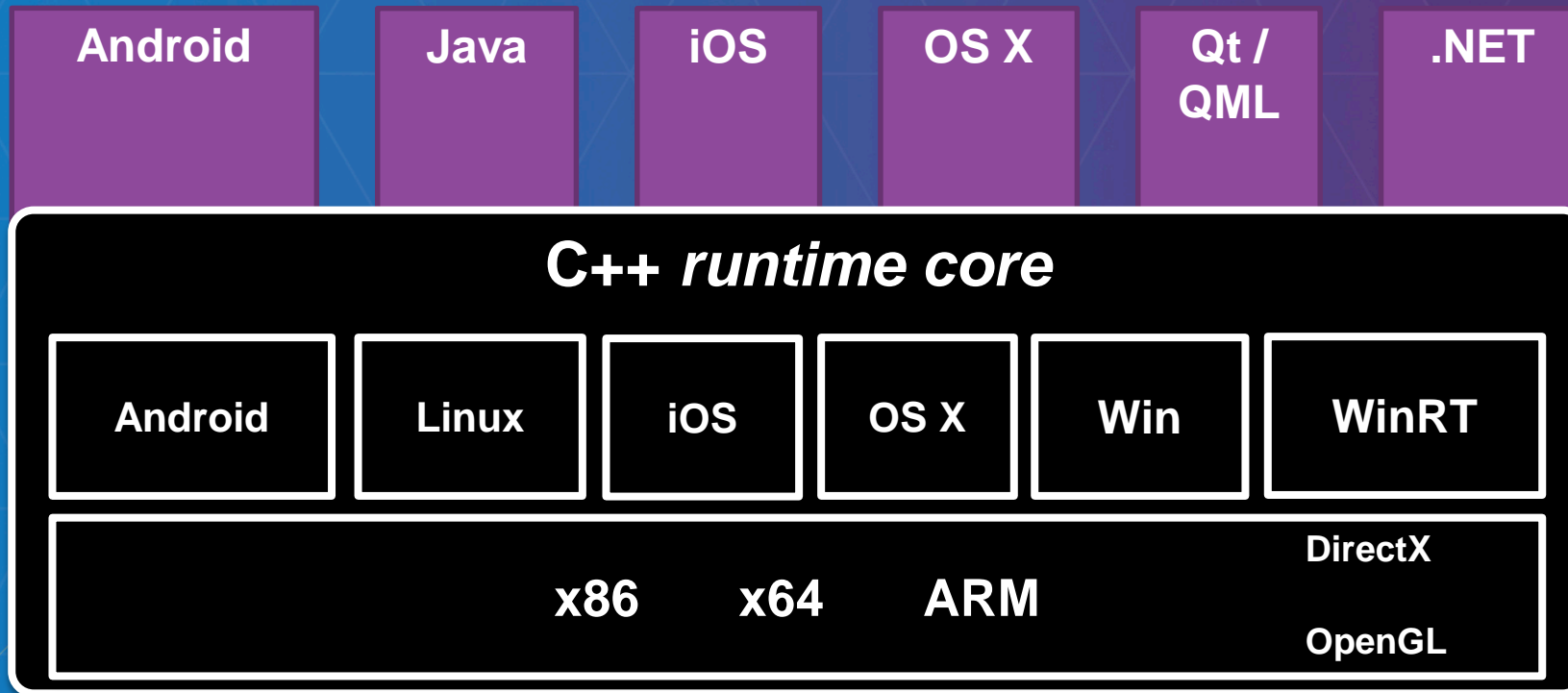
Mark Baird

JC Malott

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 SDKs



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**

Device Platforms



PHONE



TABLET



LAPTOP



DESKTOP



EMBEDDED

Java SE

ArcGIS Runtime SDK for Java

- Integrates with the ArcGIS Platform
- Build native apps for Windows and Linux
 - Windows 7, 8 and 10
 - Ubuntu, RedHat
 - 32 and 64 bit Windows, 32 and 64 bit Linux
- Java SE API, Swing (JavaFX in Quartz)
- 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

JC Malott

Home

Guide

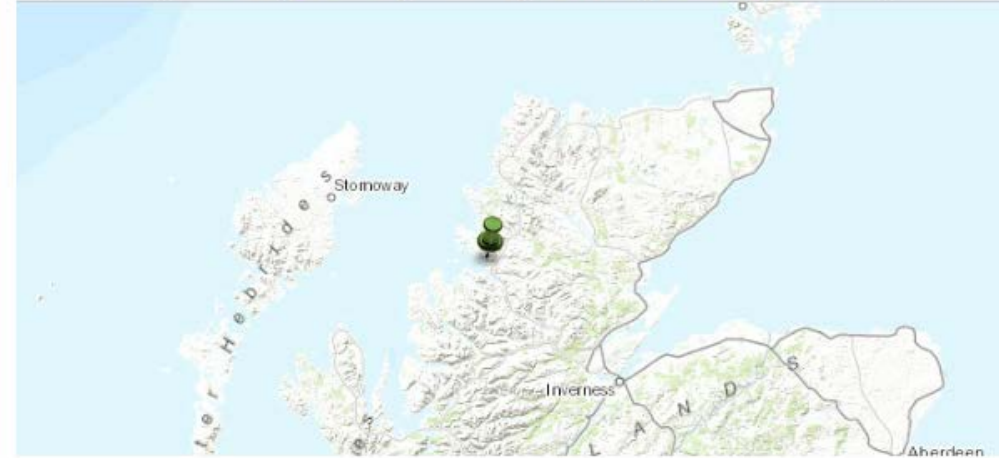
API Reference

Sample Code

Forum

Map options

[↓ DOWNLOAD SAMPLE VIEWER](#)



This application shows how to create a `JMap` using a `MapOptions` instance, giving you the option to specify the basemap (base layer), latitude and longitude around which to center the map, and zoom level for the map. The `MapOptions` instance is then used to switch the type of basemap in the map on-the-fly. In addition, simple marker graphics can be added directly to the `JMap` using the `addMarkerGraphic` methods. Popups are enabled by default on these markers. To disable these popups, use `setMarkerGraphicPopupsEnabled(boolean)`, passing in `false` to disable. For finding an address or location, static methods on the `Locator` class exist which either take or return input as a `String`. In this application, the `Locator.findAddress` static method is used to locate (geocode) the search string entered in the text field. The top result is shown on the map using a marker graphic.

What you can do

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



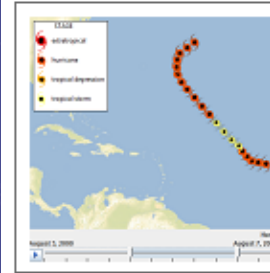
Online and offline

DEMO

Functionality Tour

JC Malott

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)**
- **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

JC Malott



Build a map



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

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)
...
```

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

JC Malott



Online and offline workflows



Online

- ArcGIS for Server services
 - ArcGIS Online (web maps)
 - Portal for ArcGIS
-
- Basemaps: map services
 - Feature layers: feature services
 - Geocoding: geocode services
 - Route finding: network analyst service
 - Analysis: geoprocessing services

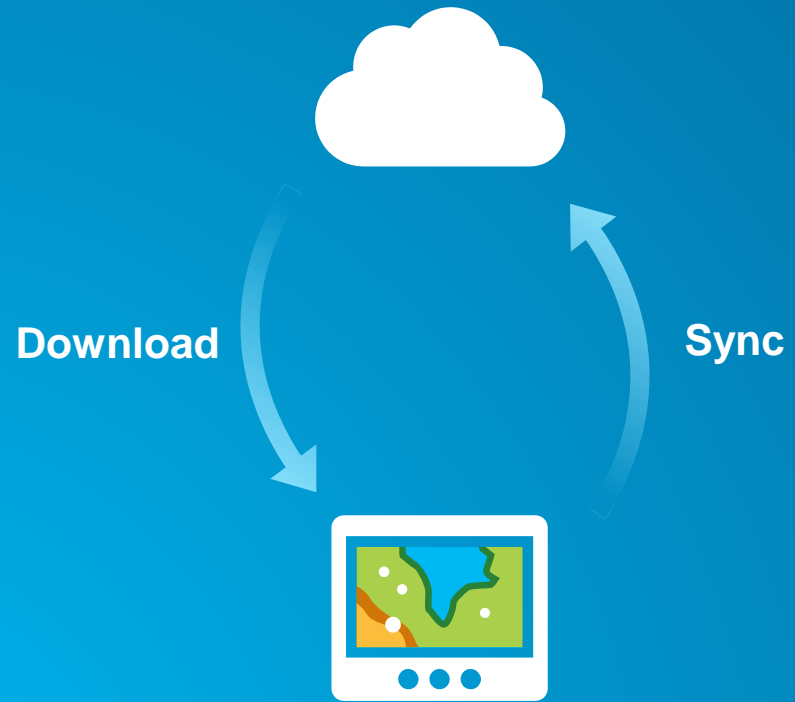


Offline

- ArcGIS for Desktop: prepare data
 - Download data from online services
 - Local Server (services)
-
- Basemaps: local tile cache
 - Feature layers: local geodatabase or shapefiles
 - Geocoding: local geocoding
 - Route finding: local routing
 - Analysis: geoprocessing services (gpk)

Offline

Services Pattern



Desktop Pattern

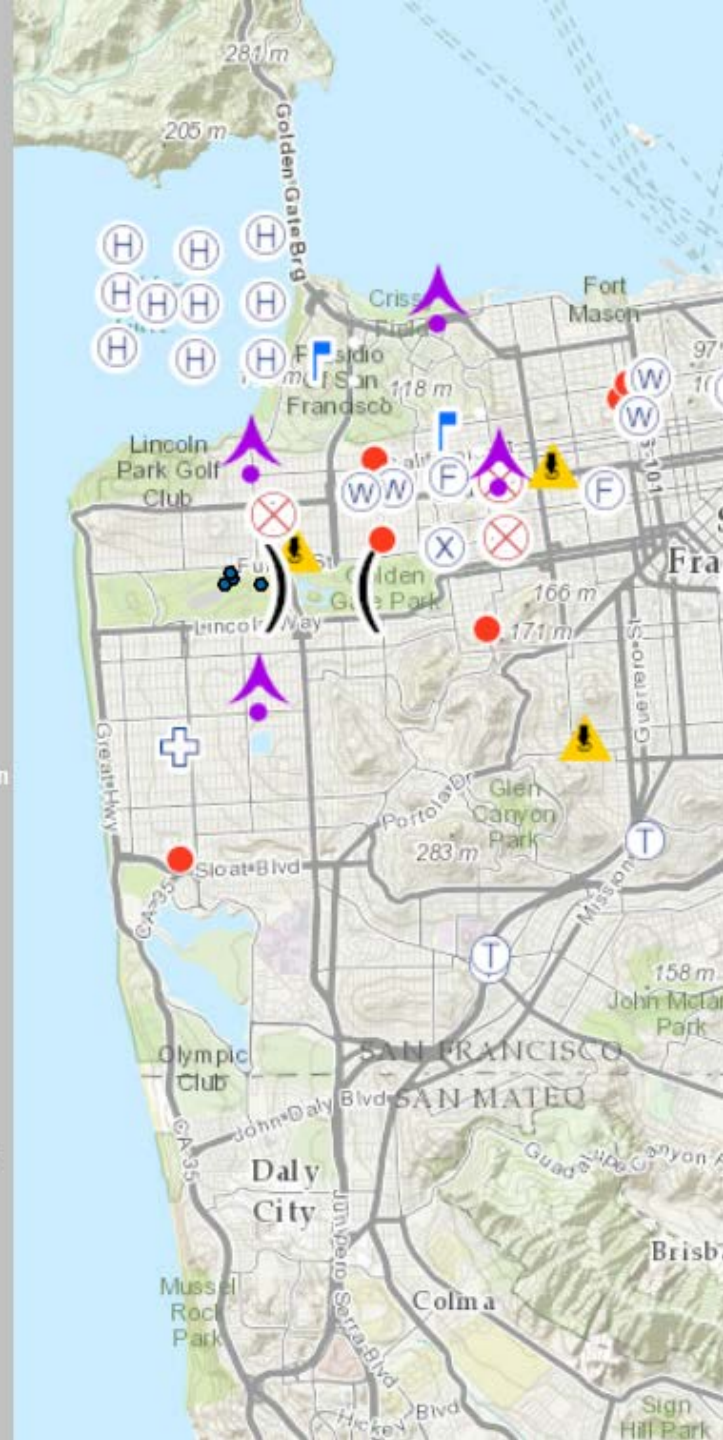


DEMO

Offline editing

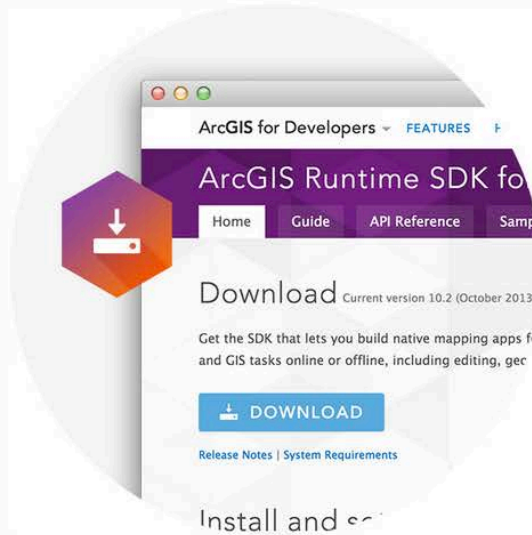
JC Mallot

- Division Break
- Aerial Hazard
- Camp
- Drop Point
- Fire Origin
- Fire Station
- First Aid Station
- Safety Zone
- Spot Fire
- Water Source
- Wind Speed Direction
- Helibase
- Hot Spot
- Lookout
- MediVac Site
- Mobile Weather Unit

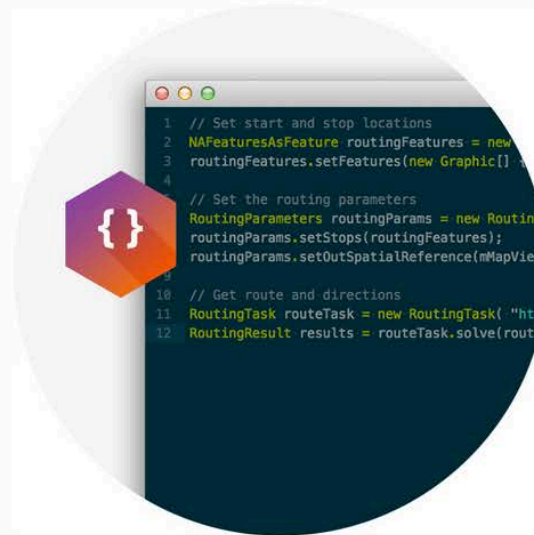


Runtime Licensing

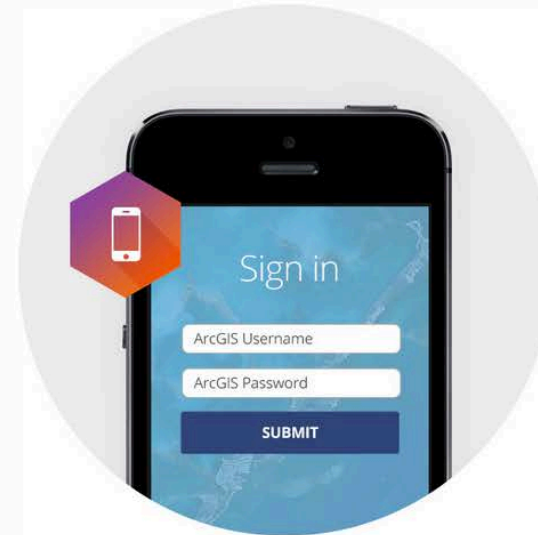
Development and Deployment Workflow



1. Download and Install



2. Develop and Test



3. Deploy and Distribute

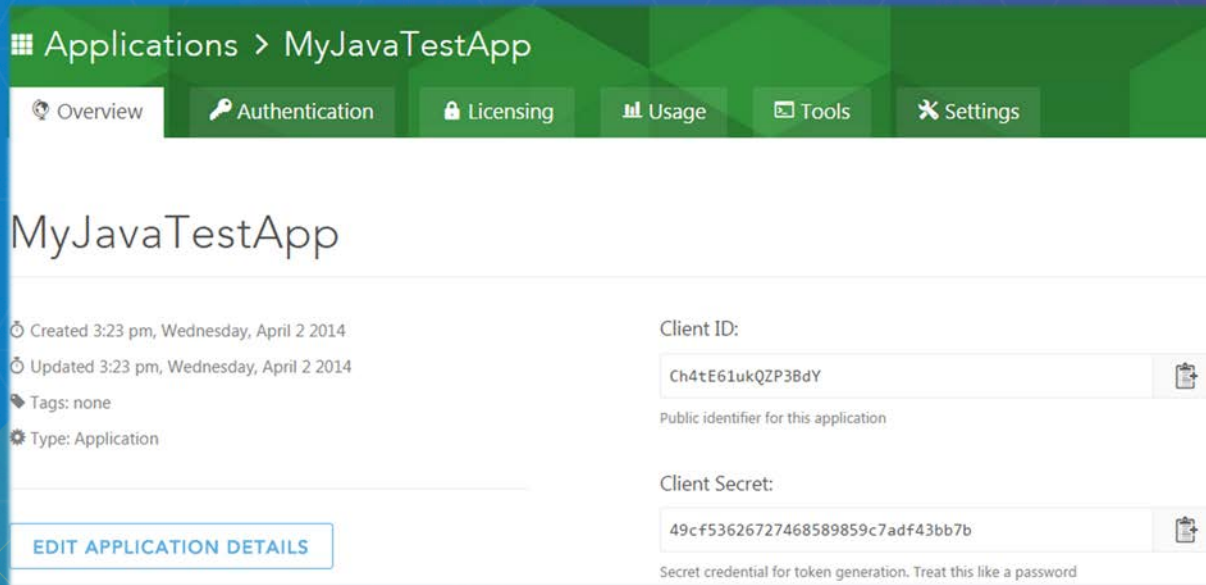
License levels and functionality

License Level	Available functionality
Developer (development and testing only)	All functionality including Local Server extensions (watermarks and debug messages will be generated, nag screens with Local Server)
Basic	All functionality, except : <ul style="list-style-type: none">• Local locators (geocoding)• Local routing• Local geodatabase editing• Local geodatabase sync operations with an upload• Local Server
Standard	All functionality Local Server extension licenses are additional and require the Standard license.

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

3.

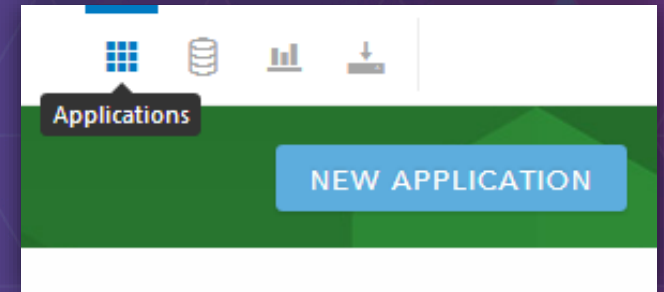


The screenshot shows the 'Applications > MyJavaTestApp' page in the ArcGIS Developers console. The 'Licensing' tab is selected, displaying the following information:

- Created: 3:23 pm, Wednesday, April 2 2014
- Updated: 3:23 pm, Wednesday, April 2 2014
- Tags: none
- Type: Application
- Client ID: Ch4tE61ukQZP38dY (Public identifier for this application)
- Client Secret: 49cf53626727468589859c7adf43bb7b (Secret credential for token generation. Treat this like a password)

An 'EDIT APPLICATION DETAILS' button is visible at the bottom left.

2.



The screenshot shows the 'Applications' page in the ArcGIS Developers console. A blue button labeled 'NEW APPLICATION' is prominently displayed on the right side of the page.

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

- Wednesday 1:00pm – 2:00pm

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

- Wednesday 2:30pm – 3:30pm

Building Native Apps that Target Multiple Platforms

- Thursday 5:30pm – 6:30pm

The Road Ahead: ArcGIS Runtime

- Friday, 10:00am – 11:00am

ArcGIS Runtime SDK for Java: Advanced Topics



Understanding our world.