ArcGIS Runtime:
Working with your portal

Divesh Goyal
Nathan Castle
Vijay Gandhi
Agenda

1. Overview
2. Accessing a portal
3. Consuming content in a portal
4. Adding content to a portal
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
ArcGIS Runtime SDKs help you build and deploy native applications to a variety of popular platforms and devices. Add powerful spatial capabilities to your native apps and empower your app users to do all things GIS, even when offline.

**Android**
Develop native mapping apps for Android devices with ArcGIS Runtime SDK for Android.
- Get Started
- Guide

**iOS**
Create mapping apps for Apple iPhone, iPad, and iPod touch devices with ArcGIS Runtime SDK for iOS.
- Get Started
- Guide

**Java**
Build and deploy to Windows, Linux, and macOS platforms with ArcGIS Runtime SDK for Java.
- Get Started
- Guide

**macOS**
Build Cocoa mapping apps for Mac using ArcGIS Runtime SDK for macOS.
- Get Started
- Guide

**.NET**
Deploy apps to Windows, Android, and iOS with ArcGIS Runtime SDK for .NET.
- Get Started
- Guide

**Qt**
Write once and deploy to Windows, Linux, Android, and iOS with ArcGIS Runtime SDK for Qt.
- Get Started
- Guide
Web GIS | Transformation of the ArcGIS Platform
Portals organize your GIS assets

enable discovery
Empower people to create and share maps

- Knowledge Workers
- Executive Access
- Public Engagement
- Work Anywhere
- Enterprise Integration
Available in the Cloud . . .

ArcGIS Online

Server
Online Content and Services

portal

. . . and On-Premises

ArcGIS Enterprise

Server
Content and Services

portal
SaaS or Software
Is it right for me?

- Typical reasons for choosing **ArcGIS Enterprise**
  - Not ready/interested in the cloud
  - Have strict SLAs and governance
  - Have no Internet/WWW connection
  - **You are responsible for maintenance**

- Typical reasons for choosing **ArcGIS Online**
  - Fewer administrative tasks
  - Elasticity
  - **Esri responsible for maintenance**
Portal
Abstracts and organizes all types of geospatial data
Geo Information Model

Organization

Group

Item

User

Application

Web Scene

Web Map

Web Layer

Service

Dataset
Geo Information Model
Accessible through the REST API

Working with users, groups, and items

ArcGIS Online is Esri’s Software-as-a-Service offering that represents GIS as a service. Portal for ArcGIS is software technology from Esri that customers can deploy either on-premises or in the cloud. Both products allow users and organizations to author and share maps and data via a portal that supports users, groups, and sharing.

The ArcGIS REST API offers resources for working with users, groups, and content within a portal for ArcGIS, implemented on ArcGIS Online. This part of the ArcGIS REST API, when used in conjunction with the ArcGIS Web and ArcGIS Desktop Runtime APIs, allows developers to build web, desktop, and enterprise applications that support sharing and collaboration using web maps. Organizational developers can also use these APIs to build custom applications for their users.

Aspects of the portal-specific resources of the ArcGIS REST API are integrated into the object models of the web and device client APIs.

Concepts

Portals allow users and organizations to publish and share content over the web.

A portal can have users who are affiliated with an organization or users who are part of an organization. A portal has users, groups, and content. Users sign in to the portal and create and share contents. The system supports different types of contents, including the following:

- Maps: Web maps that can be displayed on all supported platforms (web, mobile, and desktop)
- Layers: Static, map and image services that can be added as layers in web maps
- Applications: Web and mobile applications whose content is provided by web maps
- Tasks: Geocoding, routing, and other tools based on web services that can be used by applications
- Dictionaries (that can be uploaded, stored, and downloaded) and in some cases, activated to create services (map layers)

Users can keep content private to themselves, shared it with other users, or groups, or make it public and accessible to everyone.

Users can create and join groups. Users can share items with groups. This makes the items visible to and accessible by other members of the group.

A user of the portal (and of the REST API) sees the view of the portal that applies to their runtime environment.

Javascript
Python
Find the Right App

Realize greater field operating efficiencies using any of these apps. Most of them come included with ArcGIS.

Collector for ArcGIS
Efficient, secure data collection on smartphones or tablets replaces paper forms. Respond, record, work on- or offline, and sync automatically.

Navigator for ArcGIS
Improve workforce reliability. Navigate directly to assets even when offline. Interact seamlessly with Collector for ArcGIS and other apps to save time.

Operations Dashboard for ArcGIS
Get a real-time view and common operating picture. Add data feeds to events or daily operations. Integrate maps, charts, gauges, and histograms.

Explorer for ArcGIS
Access asset data, search for places and features, or sketch on the map to highlight important aspects. Share useful maps with the public.

Workforce for ArcGIS
Staff and apps can work together. Sync with other apps to get assignments, navigate to them, and collect data in one workflow, on one mobile device.

Survey123 for ArcGIS
Use a form-based survey interface in a lightweight, intuitive app. In just a few clicks you can collect, manage, and analyze survey results.

Drone2Map for ArcGIS
Streamline the creation of professional imagery products from drone-captured imagery for visualization and analysis in ArcGIS.

Industry Specific Apps
Jumpstart your work with pre-configured apps for your industry. Choose from hundreds of focused apps, no coding required, to accomplish your industry-specific workflows.

Premium app. Requires an additional purchase.
Accessing a portal
Accessing a portal

Code Workflow

1. **Instantiate Portal object**
   - URL to portal

2. **Load it**
   - Asynchronous **Loadable pattern**

3. **Authenticate user** (optional)
   - Provide credentials before loading
   - Or, let authentication manager challenge
Accessing a portal

Security mechanisms

- Container based
  - HTTP
  - Integrated Windows Authentication
  - Digital Certificates (PKI)

- ArcGIS Tokens

- Oauth
  - ArcGIS Online Enterprise and Social Logins
    - LDAP, Active Directory
    - Google+, Facebook
Accessing a portal
User Identity

• Details
  - Name, email, bio, thumbnail ...

• Role
  - Admin, Publisher, User

• Privileges
  - Fine grained, task oriented

• Content
  - Items
  - Groups
Demo
Accessing a portal
Consuming content from a portal
Accessing Portal Content

• Content and Types
• Demo – Pretty Portal Browser
• My content
• My organization
• Groups & Searching
• Other notes
Content and Types

- A PortalItem has:
  - Content
  - Type
  - Metadata
  - Thumbnail
  - Sharing
Content and Types

1st class citizens (ArcGIS types):
- MapService
- FeatureService
- WebMap

2nd class citizens:
- Zip archive
- Word document
- PDF

Map myMap = new Map(myPortalItem);
Building a Pretty Portal Browser
// User’s folders and (root-level) items
PortalUserContent myContent = await portal.User.GetContentAsync();
IEnumerable<PortalFolder> folders = myContent.Folders;

// Items in a folder
IEnumerable<PortalItem> items;
items = await portal.User.GetContentAsync("folderId");
Content constructors

- Available for many "1st class" Types

```csharp
// Get the first portal item
PortalItem mapItem = featuredItems
    .First(item => item.Type == PortalItemType.Webmap);

// Construct the map
Map myMap = new Map(mapItem);
```
// Get the portal’s featured items
IEnumerable<PortalItem> featuredItems;
    = await portal.GetFeaturedItemsAsync();

// Get the portal’s default basemaps
IEnumerable<Basemap> baseMaps
    = await portal.GetBasemapsAsync();
// Get this user’s groups

// Query items in a specific group
PortalQueryParameters query = PortalQueryParameters.CreateForItemsInGroup("groupId");
PortalQueryParameters query;
// Find all of my groups matching “fire”
query = PortalQueryParameters.CreateForGroups(
    owner: "nathancastle",
    title: "fire");

// Find all groups
query = PortalQueryParameters.CreateForGroups(null, null);
PortalQueryParameters query;

// Find all Web Maps that match "medicare"
query = PortalQueryParameters.CreateForItemsOfType(PortalItemType.WebMap, "medicare");

// Find all Web Maps in group
query = PortalQueryParameters.CreateForItemsOfTypeInGroup(PortalItemType.WebMap, groupId, "medicare");
PortalQueryParameters query;

// Find all Web Maps
query = PortalQueryParameters.CreateForItemsOfType(PortalItemType.WebMap);

// Limit results
Query.Limit = _itemsPerPage;

// Offset into page
Query.StartIndex = (_currentPage - 1) * _itemsPerPage;
Paginating Queries
Querying – convenience methods

- **Build a query**
  - Find groups: `QueryParameters.CreateForGroups("owner", "title");`
  - Find items of type: `QueryParameters.CreateForItemsOfType(PortalItemType.WebMap, "");`
  - Find items in a group: `QueryParameters.CreateForItemsInGroup("groupId")`

- **Execute a query**
  - `IEnumerable<PortalItem> items = await _portal.FindItemsAsync(query);`
  - `IEnumerable<PortalUser> users = await _portal.FindUsersAsync(query);`
  - `IEnumerable<PortalGroup> groups = await _portal.FindGroupsAsync(query);`

- **Convenience methods**
  - Find featured items: `_portal.GetFeaturedItemsAsync();`
  - Find default basemaps: `_portal.GetBasemapsAsync();`
Querying – Rest API

- Sometimes you need a more powerful query
- Customize the query using the REST API
  - PortalQueryParameters wraps a REST API query
  - Get the generated query or set a custom query

```
type: (+"Web Map" -"Web Mapping Application") AND group:"5645fad5ddc64df08424ebf06b5e9bc3" AND (Costs)
```
Hosted Services

- AnalysisService
- GeometryService
- GeocodingService
- TrafficService
- RoutingService
- ElevationService
Adding new content to a portal
Creating

- Add Portal Item
  - Coarse grained API
    - API type for corresponding type of portal item
    - Content and type derived from API types
  - Fine grained API
    - Content and type set by application developer
Creating Coarse grained API

- API
  - using corresponding type in API
  - e.g., ArcGISMap for WebMap
- Content type
  - derived from API type
- Content
  - derived from the API
  - e.g., for a ArcGISMap
    - Basemap (portal's default, pre-configured group of basemaps, custom)
    - Zero or more Layer
- Save
  - using corresponding type in API, e.g., ArcGISMap
  - saveAsAsync(), saveAsync()
Demo

Coarse grained API
Creating a Map
Creating Fine grained API

- **API**
  - using `PortalItem`

- **Content type**
  - set explicitly e.g., FeatureCollection, PDF, CSV, ...
  - set in constructor

- **Content**
  - set explicitly
  - e.g., JSON string for FeatureCollection, binary data for PDF, CSV, ...
  - using `PortalItemContentParameters.createJsonContent()`

- **Save**
  - using `PortalUser`
  - `addPortalItemAsync()`, `updateDataAsync()`, `setThumbnailData()`
Portal Demo

"layers": [  
  {  
    "featureSet": {  
      "features": [  
        {  
          "geometry": {  
            "rings": [  
              [  
                -17819334.3006,  
                2512026.7773999982  
              ],  
              [  
                -17821121.518199999,  
                2501427.3984000012  
              ],  
              [  
                -17838103.058200002,  
                2487923.0260000005  
              ],  
              [  
                -17836314.1855,  
                2498514.5185999982  
              ],  
              [  
                -17819334.3006,  
                2512026.7773999982  
              ]  
            ]  
          }  
        }  
      ]  
    }  
  }  
]
Sharing
Levels

• Consider
  • an organization of health department
  • groups such as Medicare, Health
  • a portal item created by a member of Medicare can be shared as
    - private, default
      - do not share with anyone
    - shared
      - share with members of Medicare group
    - organization
      - share with all members of the health department
    - public
      - share with general public
• Using **PortalItem**

• Current sharing level
  - `getAccess()`

• Public and/or Organization
  - `shareWithAsync(everyone: boolean, organization: boolean)`

• Shared
  - `shareWithGroupsAsync(groups: Iterable<PortalGroup>)`
  - `unshareWithGroupsAsync(groups: Iterable<PortalGroup>)`

• Private
  - `unshareAsync()`
Demo

Fine grained API
Creating a Feature Collection
Licensing

**Basic**
- Named User – Level 1
- License Key
- **All capabilities of Lite**
  - Edit features in mobile database
  - Synchronize edits
- **Add, update, delete content on Portal**
- Use of ArcGIS Online analysis services

**Standard**
- License Key
- **All capabilities of Basic**
- Access to additional data
- Visual Analysis
- Local Server
  - Map services
  - Feature services
    - Edit file geodatabases
  - GP services
    - Subset of ArcGIS Desktop basic tools

**Advanced**
- License Key
- **All capabilities of Standard**
- Local Server
- Feature services
  - Edit enterprise geodatabases
- GP services
  - Subset of ArcGIS Desktop standard and advanced tools

**Lite**
- Named User – Level 1
- License Key
- **View maps, scenes, layers, packages from the ArcGIS Platform/Portal**
  - Edit features in public feature services
  - Generate mobile geodatabase
  - Download updates from Sync enabled services
  - Routing
  - Place finding
  - Calculate service areas

**Analysis Extension**
- License Key
- Local Server GP tools
  - 3D Analyst
  - Spatial Analyst
  - Network Analyst
<table>
<thead>
<tr>
<th>Related sessions</th>
<th>Date</th>
<th>Time</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile Data Collection with Runtime SDK</td>
<td>Tue</td>
<td>5:30 – 6:00 pm</td>
<td>San Jacinto</td>
</tr>
<tr>
<td>ArcGIS Runtime: Building 3D Applications</td>
<td>Wed</td>
<td>1:00 – 2:00 pm</td>
<td>San Jacinto</td>
</tr>
<tr>
<td>ArcGIS Runtime: Network Analysis</td>
<td>Wed</td>
<td>2:30 – 3:30 pm</td>
<td>San Jacinto</td>
</tr>
<tr>
<td>ArcGIS Runtime: Analysis</td>
<td>Wed</td>
<td>4:00 – 5:00 pm</td>
<td>San Jacinto</td>
</tr>
<tr>
<td>ArcGIS Runtime: An Introduction to the API and Architecture</td>
<td>Thu</td>
<td>9:00 – 10:00 am</td>
<td>Catalina - Madera</td>
</tr>
<tr>
<td>ArcGIS Runtime: Road Ahead</td>
<td>Thu</td>
<td>2:30 – 3:30 pm</td>
<td>Primrose B</td>
</tr>
</tbody>
</table>
### Related sessions

<table>
<thead>
<tr>
<th>ArcGIS Runtime: Building Cross-Platform Apps</th>
<th>Thu 4:00 – 5:00 pm</th>
<th>Primrose C-D</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArcGIS Runtime: Working with Maps Online and Offline</td>
<td>Thu 4:00 – 5:00 pm</td>
<td>Mojave Learning Center</td>
</tr>
<tr>
<td>ArcGIS Runtime: Editing your Data Online and Offline</td>
<td>Thu 5:30 – 6:30 pm</td>
<td>Catalina – Madera</td>
</tr>
<tr>
<td></td>
<td>Fri 1:00 – 2:00 pm</td>
<td>Mesquite C</td>
</tr>
<tr>
<td>ArcGIS Runtime: Everything (or Anything) You Wanted to Know About the ArcGIS Runtime SDKs but Were Afraid to Ask</td>
<td>Fri 8:30 – 9:30 am</td>
<td>Catalina - Madera</td>
</tr>
</tbody>
</table>