Web AppBuilder for ArcGIS: Customizing and Extending

Moxie Zhang and Gavin Rehkemper
Let’s talk about App Creation
10 challenges for people building apps

- Quickly turn business requirements into usable apps
- Build apps without dependencies on developer skills
- Easily maintain apps
- Unified UX to build apps that work across multiple form factors and platforms
- Understand how apps are used by end-users
- Understand if the apps are effective at getting the job done
- Secure apps, their content and functionality
- Deploy apps simply and securely
- Monitor and control the use of premium services
How did you create apps?
A Traditional way to Build an App
“Well, that’s not exactly what we wanted...”
Simply, build a better experience for the users
A different way to Build an App

GUI Builder

widgets

Themes

Stem App

An App

config
Access Type

326,175 Total AppBuilder Apps

- 36% Private
- 35% Public
- 14% Shared
- 15% Account
Introduction

Create a Theme

Create a Widget

WAB Communities

Q&A

Agriculture

Invasive Weed

Boone County

Muscatine County
Building Web Apps for Your Organization
Using the ArcGIS API for JavaScript
Web AppBuilder (Developer Edition)
Widget

• Execution at run time
• Configure-in, not cut/paste
• Self sufficient and distributable
• Need container, no coding block
• Has programing framework of container

Theme

• Applied at run time
• Configure-in, not modify css
• Need container
• Self sufficient and distributable
• Has programing framework of container
It’s really just a web app

MyWidget.js

MyWidget.css

MyWidget.html
Inheriting from BaseWidget

```javascript
define(['dojo/base/declare', 'jimu/BaseWidget'],
  function(declare, BaseWidget) {
    var clazz = declare([BaseWidget], {
      ...
    });
    return clazz;
  });
```

A widget derived from the BaseWidget class
Dijit lifecycle

- postCreate
- startup
- ...

Widget events

- onOpen, onClose
- onActive, onDeActive
**BaseWidget**

- App properties (name, icon, localization)
- App config data
- Widget's config data
- Map object
- Widget state (open, closed, active...)
- Events (open/signIn)
- Widget communication

**Your job?**

- Widget UI (HTML/template)
- Widget config file (JSON)
- Widget styles (CSS)
- Localization
- **Your unique business logic / workflows (JavaScript)**
Conventions and Structure

- **MyWidget**
  - css/
  - images/
  - nls/
  - setting/
    - config.json
    - manifest.json
    - Widget.html
    - Widget.js
  - css/ styles.css
  - images/ icon.png
  - nls/ strings.js
  - es-es/ ...
Getting Started

1. Download developer edition
2. Connect to organization or portal
3. Copy widget template
4. Run the builder
5. Create an app with your widget
6. Build your widget in the app
Configure your custom widget inside the builder

- Building a UI for the user:
  - Setting.js
  - Config info
  - getConfig, setConfig
  - Setting.html
  - Usual localization pattern
  - css
Theme
App in style with personality
Theme is you
Because you are special
Your apps deserve to

stand out from the crowd.

via creating your own theme
A menu of tools

Shortcut items

Interactive content

Map, of cause

The “player”
Branding (icon, color, title)

Widgets on screen to form the UI items

Theme Widget: HeaderController

Panel contains widget’s content. It determines how widget is shown.

Placeholder as part of App layout for adding more widgets
Theme Convention and Defaulting

- **images/**
  - icon.jpg

- **layouts/**
  - a-layout-name/
    - icon.jpg, config.json

- **panels/**
  - a-panel-name/
    - images/

- **styles/**
  - a-style-name/
    - Panel.js, Panel.html

- **widgets/**
  - manifest.json
Create a Theme
EMBRACE YOUR OWN STYLE
Community
For you and by you
Online help documentation

Developer Edition help documentation
http://developers.arcgis.com/web-appbuilder
https://geonet.esri.com/community/gis/web-gis/web-appbuilder
https://geonet.esri.com/groups/web-app-builder-custom-widgets
Creating Web Applications Using Templates and Web AppBuilder for ArcGIS

https://training.esri.com
Other Online Resources

• Esri Solutions Widgets:
  - https://github.com/Esri/solutions-webappbuilder-widgets

• Lists of Widgets:
  - http://codesharing.arcgis.com/
  - https://github.com/gavinr/wab-widget-search

• “Awesome ArcGIS” https://github.com/hhkaos/awesome-arcgis

• Example widget shown today: http://esriurl.com/13950
What’s Coming
Evolution with excitements
Web AppBuilder is greatly appreciated by our users. However, we can’t stand still in this fast moving world...
10 challenges for people building apps

1. Quickly turn business requirements into usable apps
2. Build apps without dependencies on developer skills
3. Easily maintain apps
4. Unified UX to build apps that work across multiple form factors and platforms
5. Understand how apps are used by end-users
6. Understand if the apps are effective at getting the job done
7. Secure apps, their content and functionality
8. Deploy apps simply and securely
9. Monitor and control the use of premium services
less coding

Limited needs for grand-up dev works

better UX
cheaper
faster
simpler
managed

added value
What Esri Has

Configurable App simplifies the app creation

- Web AppBuilder (WAB), as a tool, has significantly simplified the app creation process
- Since the first WAB release, in two years, more than 220,000 apps are created and hosted on ArcGIS Online

However, as a tool, WAB can only partially meet the customer’s requirements for apps (in previous slides)
- Esri app creation needs to evolve into being a system to encapsulate the whole app stack and requirements
- Hence ...
“Experience Builder”
What is Experience Builder for ArcGIS

... a web user experience for creating and managing app through the app’s lifecycle

... is the evolution of Web AppBuilder, an integrated and unified web user experience
Anatomy of an ArcGIS app

An App is a living and breathing being made out of...

- Themes/Skins
- Authorization
- Function models (i.e., widgets)
- Captured usage data
- WebScenes
- WebMaps
- Source code and revisions
- Premium Services
- Configuration
- Feedback (comments, rating, issues, etc.)
App Lifecycle

conception → design

design → configure

configure → test

test → program

program → deploy

deploy → use

use → analyze

analyze → track

track → revise

revise → retire
App Lifecycle is complex

Apps are disposable

Apps are data

Apps are a conduit to understand your users
Experience Builder (app centric)

- Resources (maps...)
- Extensions (widgets/themes)
- App Lifecycle Management
- Unified Builder UX
- Assets (images/icons/...)
- Statistics

App (web Item)