

# ArcGIS Experience Builder for Developers: Widgets, Development, Deployment, and Automation

Shawn Goulet

Slides: [links.esri.com/2024-fed-uc-arcgis-exb-for-devs](https://links.esri.com/2024-fed-uc-arcgis-exb-for-devs)

# Agenda

- Intro to Experience Builder
- Custom widgets
- Code collaboration
- Deployment
- Deployment automation (dev ops)

# Intro to Experience Builder

# What is ArcGIS Experience Builder

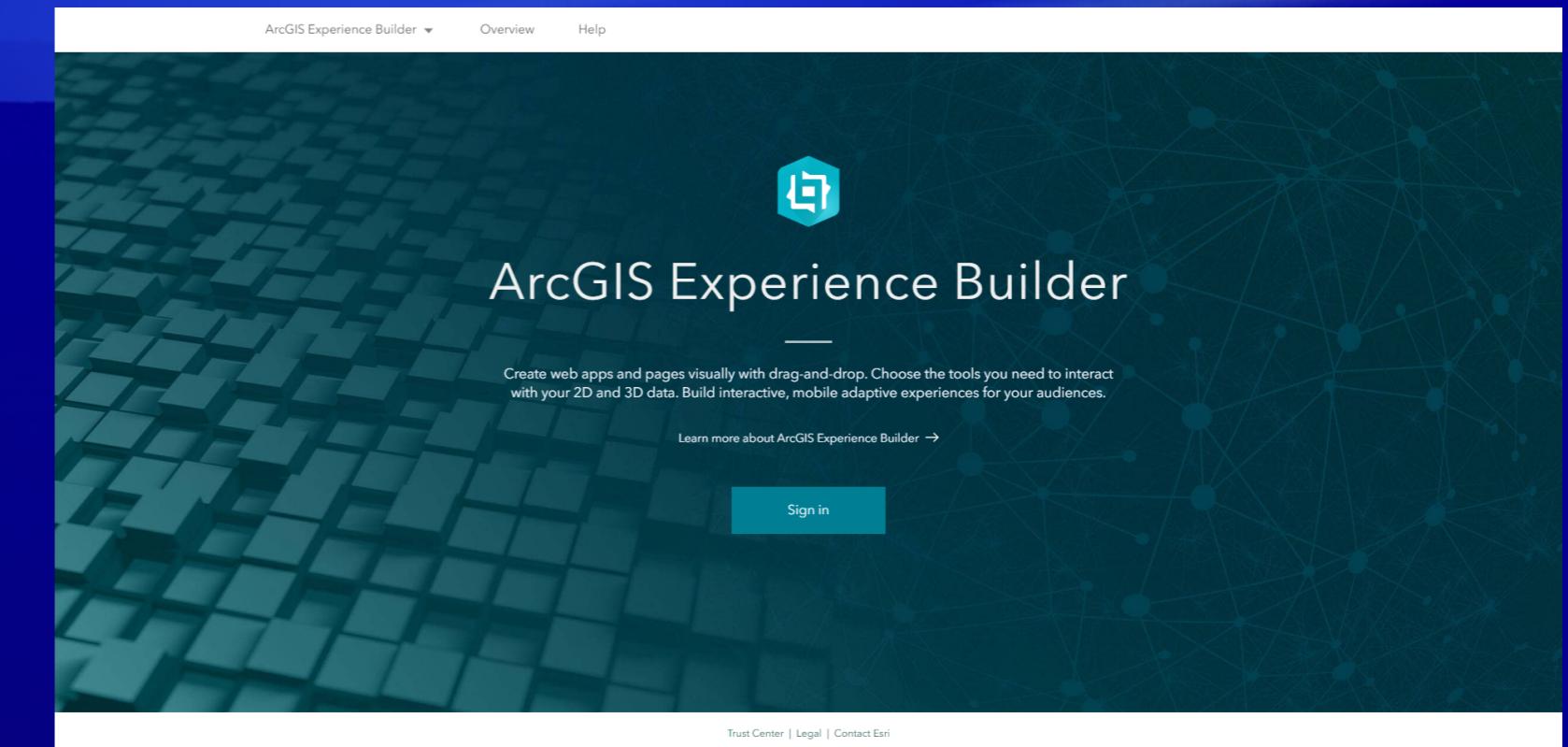
ArcGIS Experience Builder empowers you to quickly transform your data into compelling web apps and pages

- Start with templates and widgets
- Extensibility

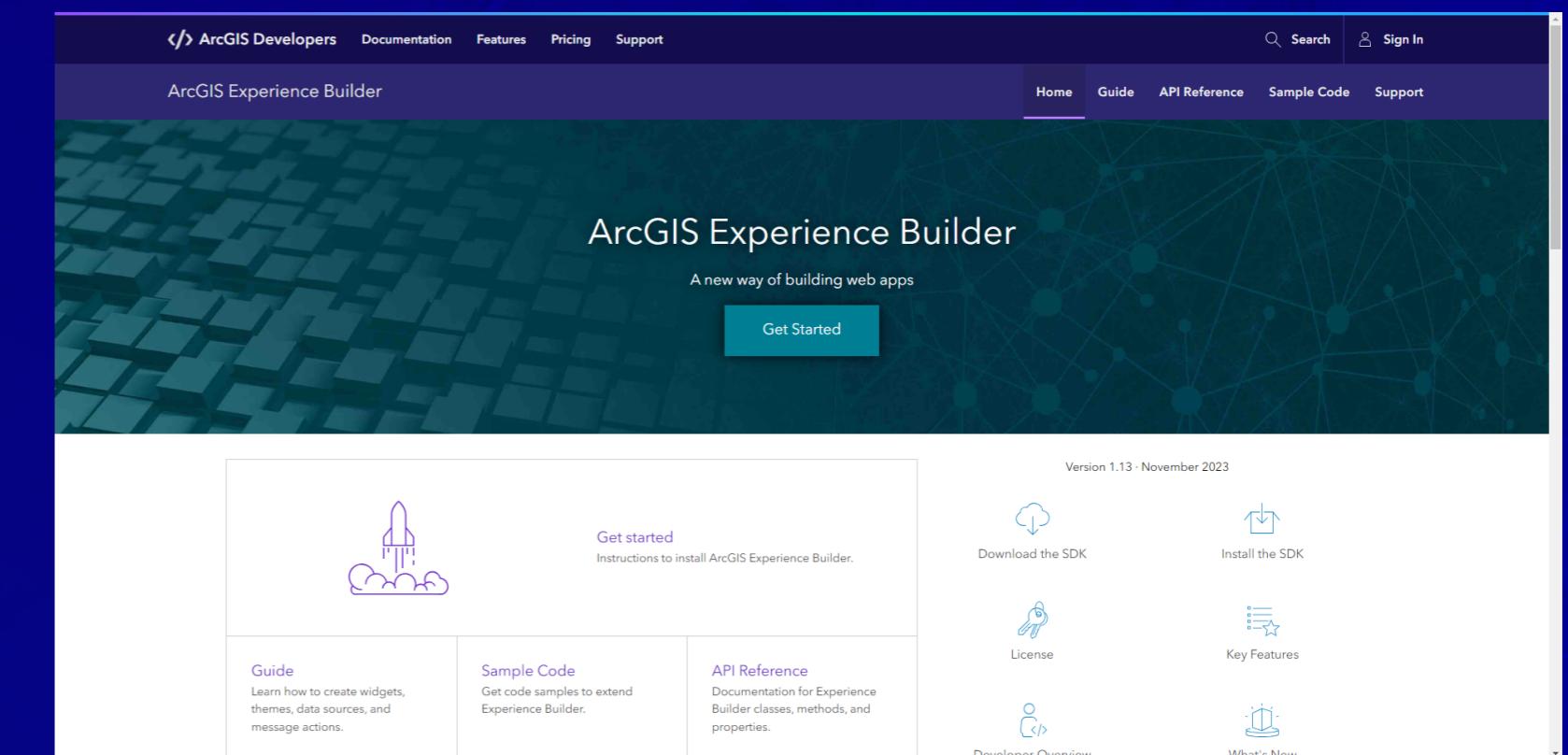


# ArcGIS Experience Builder Editions

- ArcGIS Online and ArcGIS Enterprise
  - URL: [experience.arcgis.com](https://experience.arcgis.com)
  - ArcGIS Enterprise 10.8.1 and up



- Developer Edition
  - Download from the Developers website and install locally
  - Sign in to ArcGIS Online / Enterprise





Run developer edition

# Custom widgets

# Technologies

- React
- TypeScript
- Jimu

JIMU-UI

JIMU-LAYOUTS

JIMU FOR BUILDER

JIMU-ARCGIS

## Jimu-core

Widget Manager

Config Manager

DataSource Manager

Extension Manager

Message Manager

TypeScript

React

Redux

Webpack

ArcGIS  
JS API 4.x

# Widget Structure

- Widget
  - dist/
    - compiled code
  - src/
    - runtime/
      - widget.tsx
      - translations/
    - setting/
      - setting.tsx
      - translations/
  - config.ts
  - config.json
  - manifest.json

# MapView and SceneView

- View concept same as the ArcGIS Maps API for JavaScript
  - View is wrapped in JimuMapView
- To create a JimuMapView object: `MapViewManager.createJimuMapView()`
- JimuMapView main properties:

Property	Description
<code>view</code>	MapView/SceneView object (from ArcGIS Maps API for JavaScript)
<code>dataSourceId</code>	Data Source used to create the view
<code>mapWidgetId</code>	the widget that creates the objects
<code>jimuLayerViews</code>	the layer view object wrapper

# Creating a Widget

# Two styles

## Style 1: Class components (pre React 16.8)

- Make use of ES6 class and extend the component class in React.
- Can maintain its own data with state.
- Props accessed via `this.props`.
- Uses the `render()` method.

## Style 2: Functional components (introduced at React 16.8)

- Basic JavaScript function
- Uses React Hooks to use state and other features.
- Can accept and use props.
- Simply return what should be rendered

# Create a widget using class component

```
import { AllWidgetProps, BaseWidget, jsx } from "jimu-core"

export default class Widget extends BaseWidget<AllWidgetProps, any> {
  render() {
    return (
      <div className="widget-start jimu-widget" style={{ overflow: "auto" }}>
        <p>Hello world!</p>
        <p>Widget name: {this.props.label}</p>
      </div>
    )
  }
}
```

# Create a widget using a functional component

```
import { AllWidgetProps, jsx } from "jimu-core"

export default function Widget (props: AllWidgetProps) {
  return <div className="widget-start jimu-widget" style={{ overflow: "auto" }}>
    <p>Hello world!</p>
    <p>Widget name: {props.label}</p>
  </div>
}
```

# Comparison

## Class based

```
import { AllWidgetProps, BaseWidget, jsx } from "jimu-core"

export default class Widget extends BaseWidget<AllWidgetProps, any> {
  render() {
    return (
      <div className="widget-start jimu-widget" style={{ overflow: "auto" }}>
        <p>Hello world!</p>
        <p>Widget name: {this.props.label}</p>
      </div>
    )
  }
}
```

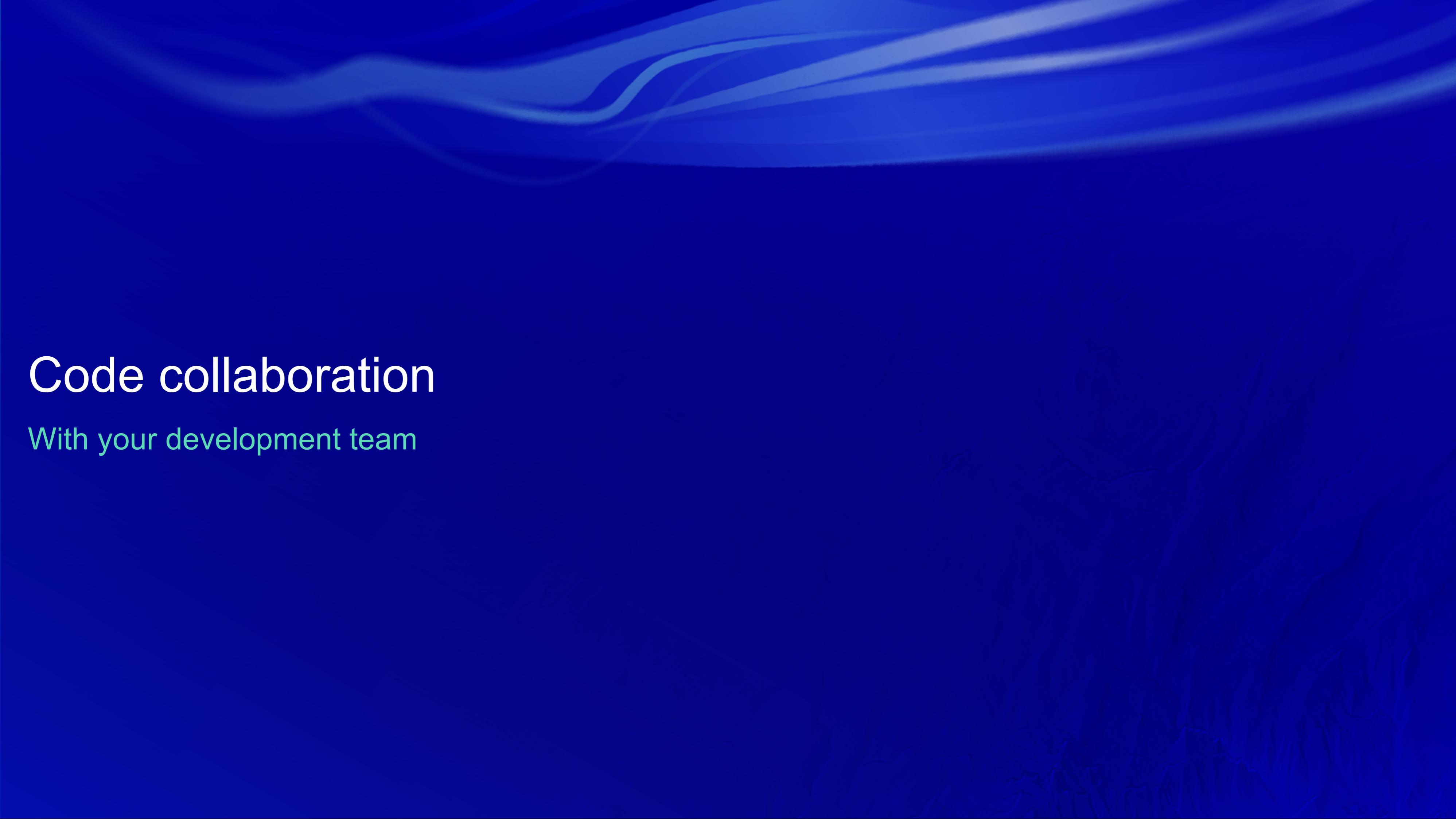
## Function based

```
import { AllWidgetProps, jsx } from "jimu-core"

export default function Widget (props: AllWidgetProps) {
  return <div className="widget-start jimu-widget" style={{ overflow: "auto" }}>
    <p>Hello world!</p>
    <p>Widget name: {props.label}</p>
  </div>
}
```

# Demo

Creating a custom widget

The background of the slide features a dark blue gradient with subtle, flowing white and light blue wavy lines that create a sense of depth and motion.

# Code collaboration

With your development team

# Web extension repo

- Folder
- Within the client/ Folder
- Contains a manifest.json file with the contents:

```
{  
  "name": "my-web-extension-repo",  
  "type": "exb-web-extension-repo",  
  "description": "This is a sample extension repository.",  
  "copyright": "",  
  "license": ""  
}
```

# Web extension repo

- my-web-extension-repo/
  - manifest.json
  - themes/
  - widgets/
    - my-custom-widget/

# The source code repository IS a web extension repo!

- New team members: clone repository into the client/ folder

More info:

[developers.arcgis.com/experience-builder/guide/getting-started-widget](https://developers.arcgis.com/experience-builder/guide/getting-started-widget)

# Deployment

# Two patterns

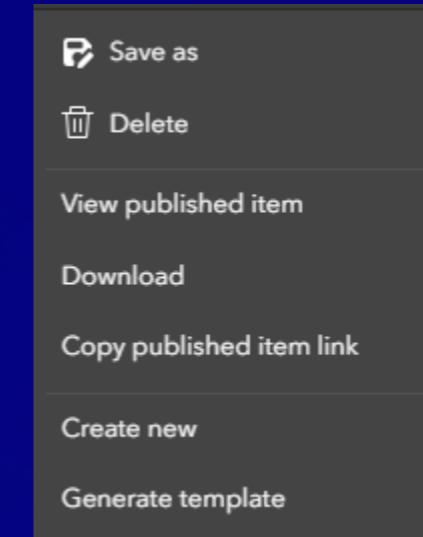
1. ArcGIS Enterprise
2. Developer Edition > host on a web server

# ArcGIS Enterprise

1. Deploy custom widgets to Experience Builder within ArcGIS Enterprise
2. Users within ArcGIS Enterprise can build Experiences with those custom widgets
3. ArcGIS Enterprise 11.0 or higher
  - Documentation: [links.esri.com/exb-share-custom-widget](https://links.esri.com/exb-share-custom-widget)
  - Detailed instructions (blog post): [links.esri.com/exb-add-custom-widgets-blog](https://links.esri.com/exb-add-custom-widgets-blog)

# Developer Edition

1. Build the Experience with your custom widgets
2. Download as ZIP (image below)
3. Unzip files
4. If there is any private content, add clientId to cdn/<app-string>/config.json
5. Host the files on a web server



# Deployment automation

Dev ops

# Why automation?

- Faster builds
- Consistent builds
- Test integration
- The "reference" application
- Branch deployments

# How to automate

- Store the reference version of each app
- Auto-build using your CI/CD env of choice:
  - GitHub Actions\*
  - Azure DevOps
  - AWS CodePipeline
  - GitLab Actions
  - Jenkins
  - Etc.

# Folder structure

- manifest.json
- apps/
  - 0/
  - 1/
  - zero/
  - one/
  - ...
- widgets/
  - custom-widget-1/
  - custom-widget-2/
  - ...

# Auto-build

- Create file:

`.github/workflows/build-app.yml`

- Demo repo:

[github.com/gavindr/experience-builder-devops-example](https://github.com/gavindr/experience-builder-devops-example)

# Auto-build demo

<https://github.com/gavinr/experience-builder-devops-example>

# Auto-build outcomes

- Consistent outputs
- Easier version management (via Git branching)
- Quicker bug investigations (is it happening on the reference app?)
- More organized management of app config versions
- Quickly/easily share app both within & outside of team/organization

# Resources

---

Code from presentation

[code](#)

Experience Builder developer documentation

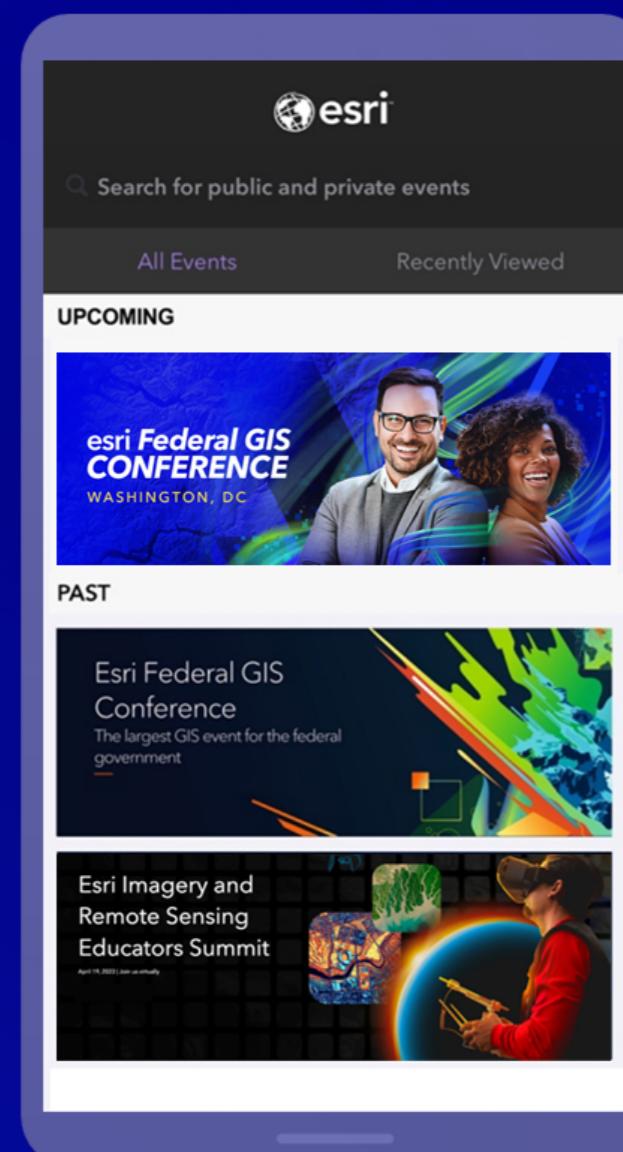
[developers.arcgis.com/experience-builder](https://developers.arcgis.com/experience-builder)

Esri Community: Experience Builder

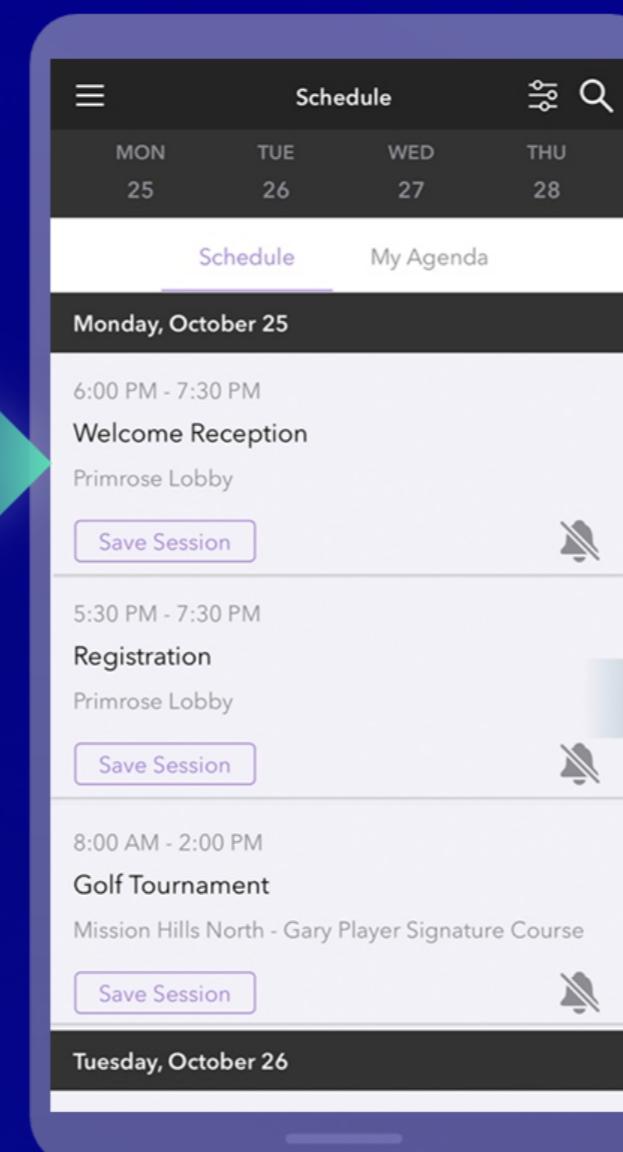
[community.esri.com/t5/arcgis-experience-builder/ct-p/arcgis-experience-builder](https://community.esri.com/t5/arcgis-experience-builder/ct-p/arcgis-experience-builder)

# Please Share Your Feedback in the App

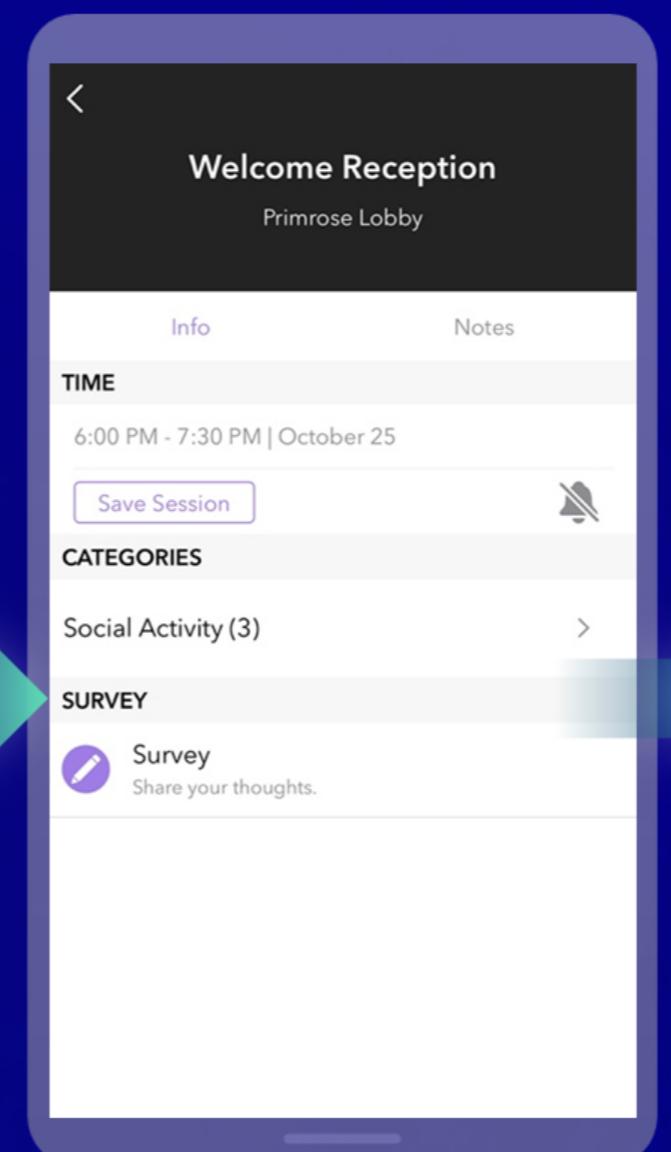
Download the Esri Events app and find your event



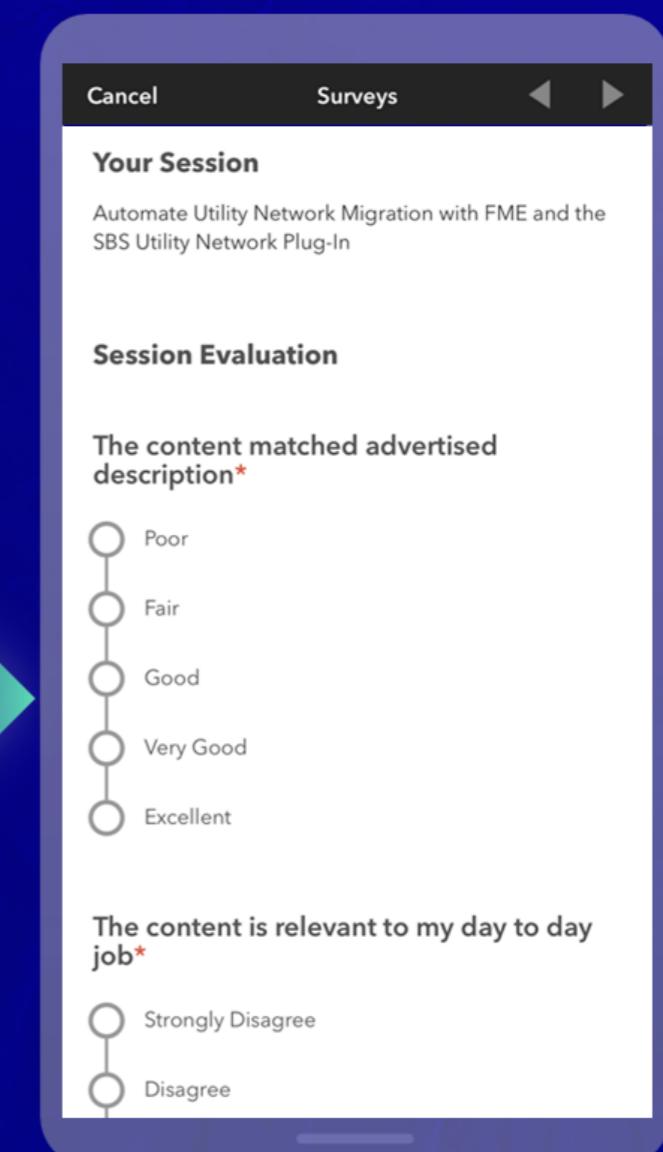
Select the session you attended



Scroll down to "Survey"



Log in to access the survey



# Print Your Certificate of Attendance

Print Station Located on the Concourse by Rooms 151A/B

- **Tuesday February 13, 2024**

- Expo

12:30pm – 6:30pm

Exhibit Hall B, *Lower Level*

- Expo Social

5:51pm – 6:30pm

Exhibit Hall B, *Lower Level*

- **Wednesday February 14, 2024**

- Expo

10:45am – 5:15pm

Exhibit Hall B, *Lower Level*

- Networking Reception

6:30pm – 9:30pm

Smithsonian National Museum of Natural History

*Shuttle Bus Pick Up – L Street North Lobby*



THE  
SCIENCE  
OF  
WHERE®

Copyright © 2024 Esri. All rights reserved.