



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

Gavin Rehkemper

Slides: [esriurl.com/exb-for-dev-23](https://esriurl.com/exb-for-dev-23)

# Agenda

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

# Intro to Experience Builder



# What is ArcGIS Experience Builder

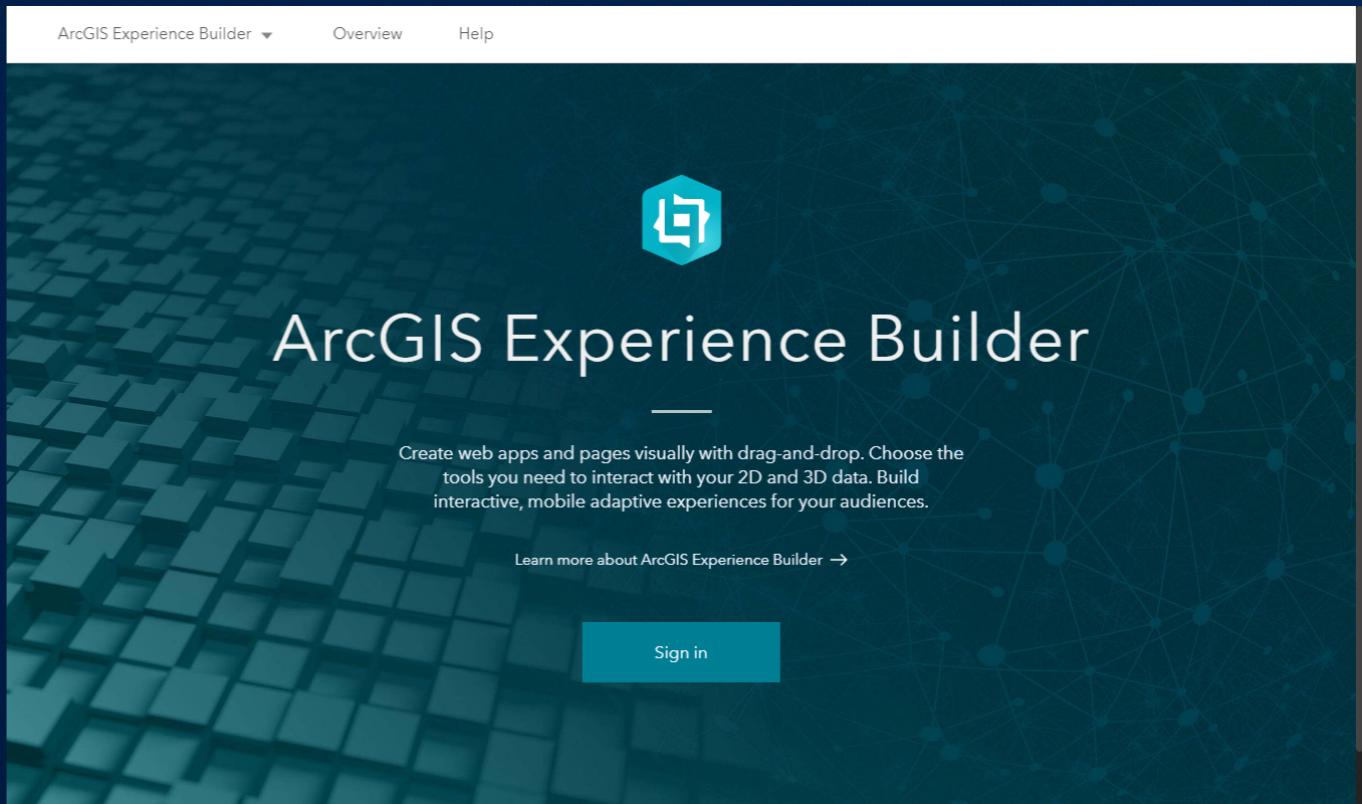
ArcGIS Experience Builder empowers you to quickly transform your data into compelling web apps and pages

- Start with templates and widgets
- Extensibility

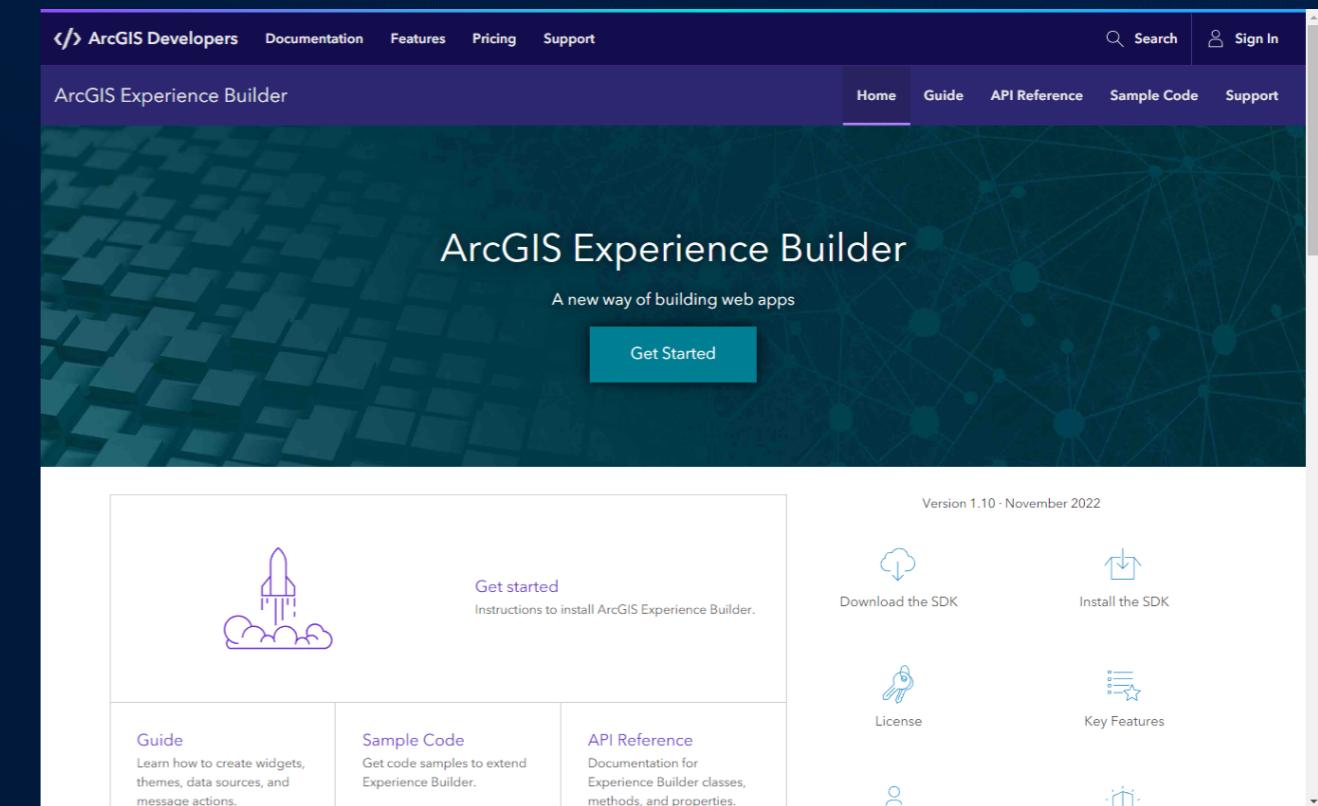


# ArcGIS Experience Builder Editions

- 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-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

# Creating a Widget



# Two styles

## Class components

- 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.

## Function components

- 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 function

```
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>
}
```

The background features a complex, abstract graphic on the left side. It consists of numerous overlapping, curved bands in various colors including teal, yellow, blue, and orange. These bands appear to be stylized maps or topographic charts, with some showing grid patterns and others more organic, flowing shapes. The overall effect is dynamic and layered.

Demo

Creating a custom widget



# 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

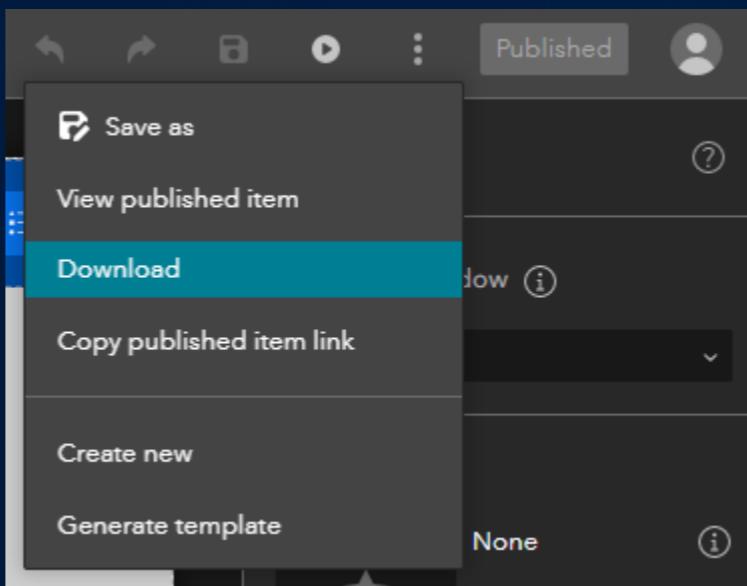
# ArcGIS Enterprise

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

Detailed instructions: [esriurl.com/exb-widget-enterprise](http://esriurl.com/exb-widget-enterprise)

# Developer Edition

1. Build the Experience with your custom widgets
2. Download as ZIP



3. Unzip files
4. Add clientId to cdn/1/config.json
5. Host the files on a web server

# Deployment automation

Dev ops

# Why automation?

- Faster builds
- Consistent builds
- Easier for testers
- The "reference" application

# How to automate

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

# Folder structure

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

# Auto-build

- Create file:

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

- Demo repo:

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

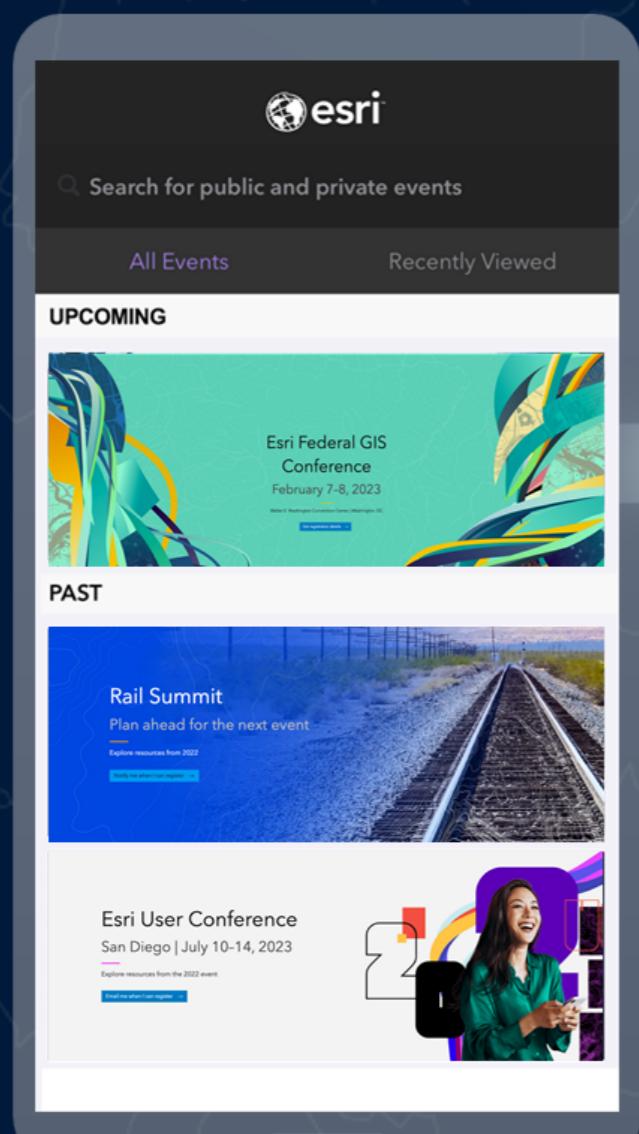
The background features a dark blue gradient with white topographic contour lines on the left. On the right, there are several thick, colorful diagonal stripes in shades of yellow, teal, light blue, and green, which overlap each other.

Thank you!

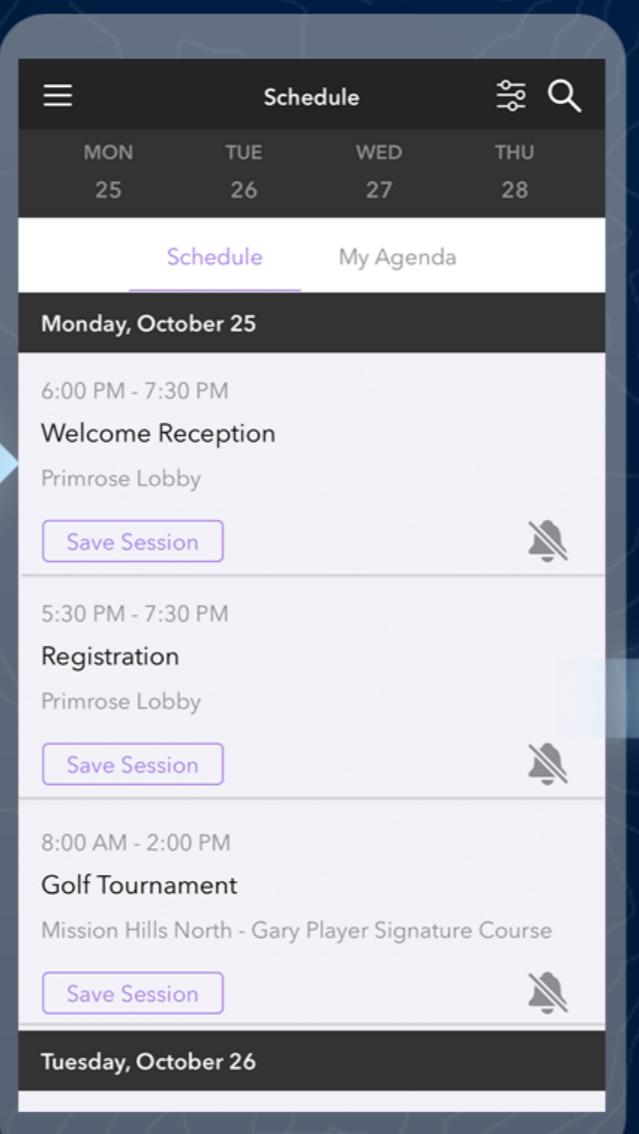
Questions?

# Please Share Your Feedback in the App

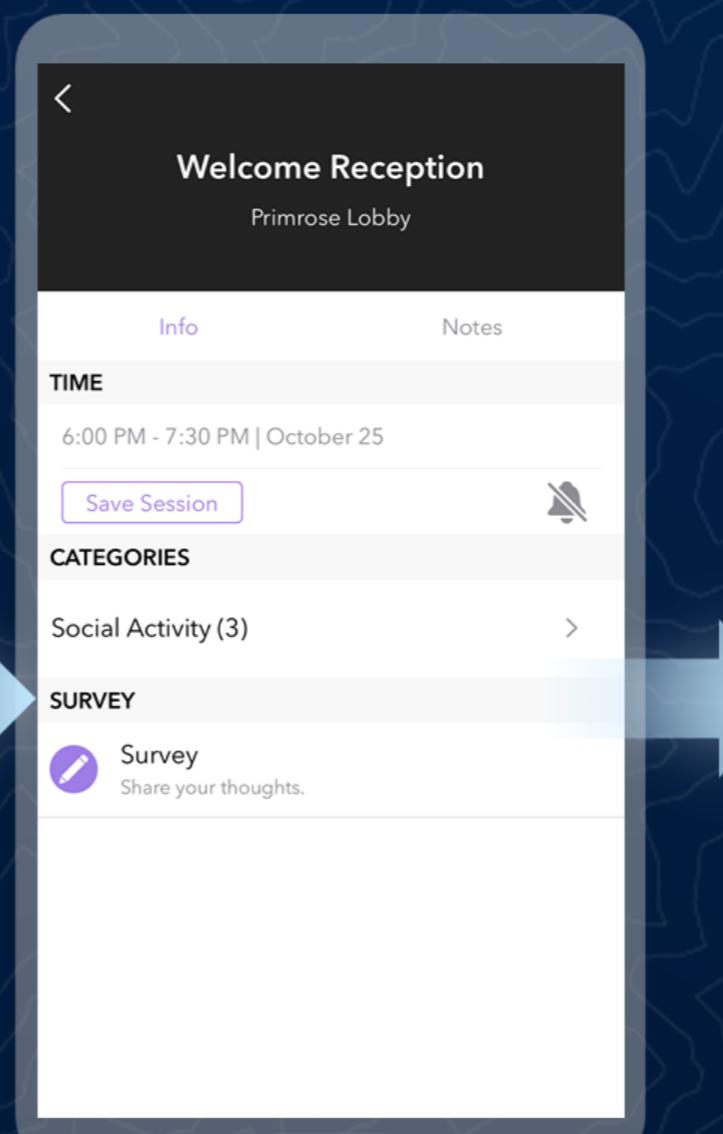
Download the Esri Events app and find your event



Select the session you attended



Scroll down to "Survey"



Log in to access the survey

