Developing Your Own Widget with the ArcGIS API for JavaScript

Matt Driscoll – @driskull
JC Franco – @arfncode
Agenda

- Widgets
- Theming
- Widget Framework
- Putting it all together
Widgets
About

- What?
  - Encapsulated UI components
  - Cohesive (integrated, unified)
  - Single-purpose pieces of functionality
- Why?
  - Reusable
  - Interchangeable
- How?
  - Different frameworks are available
Widget Theming
About

- Why?
  - Consistency
    - Visual
    - Functional
  - User options Out-of-the-box
    - of-the-box
    - Custom

- How?
  - Sass
  - BEM
Out-of-the-box themes

Theme Switcher
Theming with Sass

- CSS preprocessor
- Powered-up CSS
  - Nesting
  - Variables
  - Functions
  - Mixins
  - Inheritance
Sass makes it easier to...

- Restyle
- Organize
- Write less code :}

9
Naming CSS classes: BEM

Block Element Modifier

- Semantic
- Low specificity
- Scopes styles to blocks

```javascript
// block
d.example-widget {}

// block__element
d.example-widget__input {}

// block--modifier
d.example-widget--loading {}

// block__element--modifier
d.example-widget__input--disabled {}
```
Recap

- Consistency
- User options
- Authoring
Widget Framework
About

• Views + ViewModels
  ▪ Separation of concerns
  ▪ Reusable
    ○ UI replacement
    ○ Easier integration
• Built with TypeScript
Views

- Extend esri/widgets/Widget
- Rely on ViewModel
- Focus on UI
ViewModels

- Extend esri/core/Accessor
- Provide APIs to support view
- Focus on business logic
esri/widgets/Widget

- Lifecycle
- API consistency
  - Unified object constructor
  - Properties
  - Watching
Lifecycle

- constructor
- postInitialize
- render
- destroy
render

- Defines UI
- Reacts to state
- Uses JSX
TypeScript

- Typed JavaScript

type PresenterName = "Matt" | "JC";

interface Person {
    name: string;
}

interface Presenter extends Person {
    name: PresenterName;
}

// TS2322: Type '{ name: "Alan" }'; is not assignable to type 'Presenter'.
const myPresenter: Presenter = { name: "Alan" };
// const
const numbers = [1, 2, 3];

// fat arrow functions
letters.forEach(letter => console.log(letter));

// template literals
const myString = `last number: ${numbers[numbers.length - 1]}`;

// decorators
class Example {
  @log
  stringify(item: object): string {
    /* ... */
  }
}
TypeScript

- IDE support
  - Visual Studio, WebStorm, Sublime, and more!
import { declared, subclass } from "esri/core/accessorSupport/decorators";

import Base = require("my/class/base");

@subclass("MyClass")
class MyClass extends declared(Base) {
}

export = MyClass;
Defining a variable

```java
@subclass("MyClass")
class MyClass extends declared(Base) {

    // adds variable `_foo`
    _foo: Foo = new Foo();

}
```

// ...

// ...
Defining a property

```javascript
// ...

@subclass("MyClass")
class MyClass extends declared(Base) {

    // adds property `foo`
    @property()
    foo: Foo = new Foo();

}

// ...
```
Defining a method

```java
// ...  

@subclass("MyClass")
class MyClass extends declared(Base) {

    // adds method `bar`
    bar(): string {
        return this._getText();
    }

    private _getText(): string {
        // gets text
    }
}
```
Widget decorators

- @subclass + declared
- @property
  - autocast
  - computed
  - read-only
  - aliased
- @aliasOf
- @renderable
- @accessibleHandler
Recap

- Views + ViewModels
- esri/widgets/Widget
- TypeScript
Let's build a widget!

Magnifier
Let's build a widget!

- Demo Start HTML
- Steps ViewModel
- Steps View Steps
- Sass Steps
- 
Let's Recap

- Widgets are single functionality UI components. We use them for reusability/interchangeability.
- Themes
  - SASS
- Widget Framework
- Constructing a widget
  - ViewModels
  - Views
Suggested Session

- ArcGIS API for JavaScript: Customizing Widgets
Additional Resources

- Styling
- Implementing Accessor
- Setting up TypeScript
- Widget Development JS
- API SDK
Use the source, Luke

esriurl.com/buildwidgetsuc2017
Please Take Our Survey on the **Esri Events App**!

**Download the Esri Events app and find your event**

**Select the session you attended**

**Scroll down to find the survey**

**Complete Answers and Select “Submit”**
Questions?
Thank you!