Angular and the ArcGIS API for JavaScript

ANDY GUP   @agup
SEAN OLSON  seanolson-io
JACOB WASILKOWSKI  @JWasilGeo
Angular + Esri JSAPI
Angular? Yes. Angular.

Not "AngularJS", "Angular v1", "Angular.js", etc.

https://github.com/Esri/angular-esri-map
Upcoming, related presentations

1. Using Frameworks with the ArcGIS API for JavaScript
   - Thursday, March 08 at 2:30pm, Primrose A
   - Rene Rubalcava and Tom Wayson

2. Kick Starter: Rapid Application Development using Angular CLI
   - Friday, March 09 at 8:30am, Pasadena/Sierra/Ventura
   - Sean Olson
What's the plan today?

Assuming some familiarity with Angular 2+:

- Walk through sample Angular apps from http://esriurl.com/13900
- Learn how to use Esri JSAPI inside of Angular apps
- Consume Esri JSAPI from CDN
Getting started

1. Download our repo at http://esriurl.com/13900
   - Github "sean-olson-e/Angular-and-the-ArcGIS-API-for-JavaScript"

2. Follow along as we work with Angular sample apps
   - ./sample_apps/1-app-scaffolding
   - ./sample_apps/2-more-app-scaffolding
   - ./sample_apps/3-the-esri-loader
   - ...and more...
Quick start instructions for this demo

# only once on your computer
npm install -g @angular/cli

# go to the first sample app
cd ./sample_apps/1-app-scaffolding

# install dependencies
npm install

# serve the app for development
ng serve

# or, if you want to build for production
ng build --base-href ./
1-app-scaffolding

- Building blocks:
  - Angular CLI
  - "esri-loader"
  - custom demo files:
    - esri-map.component.ts
    - esri-map.component.html
    - esri-map.component.css
Extending the map component
- Pass data from parent to child map component with input binding
- Listen in parent for child map event
The glue that holds everything together is "Esri/esri-loader". 

"A tiny library to help load ArcGIS API for JavaScript modules in non-Dojo applications"

```javascript
/* ANGULAR GALAXY AND COLD VACUUM OF SPACE OUT HERE! */

import { loadModules } from 'esri-loader';

loadModules([  'esri/Map', 'esri/views/MapView', 'esri/Graphic' ]).then(([  EsriMap, EsriMapView, Graphic  ]) => {
  /* ESRI PLANET WITH ATMOSPHERE INSIDE HERE! */
  // for example you could recreate any demo from https://js.arcgis.com

  /* ANGULAR GALAXY AND COLD VACUUM OF SPACE OUT HERE! */
```
3-the-esri-loader

- This brings Esri **WITHIN** Angular
- Which ultimately means Dojo **INSIDE OF** Angular
- How?
  - `loadModules` method that you import as needed
  ```javascript
  import { loadModules } from 'esri-loader';
  ```
  - `loadModules` method dynamically injects an Esri `<script>` tag onto the page
3-the-esri-loader

- Why?
  - Provides a reliable way to load Esri modules using Dojo's AMD loader
- You benefit from getting to:
  - use Angular tooling
  - improve initial app load performance
  - control which Esri modules you want, à la carte
4-types-for-arcgis-api

- TypeScript is used throughout Angular's documentation
- It is "a typed superset of JavaScript that compiles to plain JavaScript"
- It allows for compile-time type checking for JavaScript

```javascript
// without types
let fullName = 'Bob Bobbington';
let age = 37;

// with types
let fullName: string = 'Bob Bobbington';
let age: number = 37;
```

- https://www.typescriptlang.org/docs/handbook/basic-types.html
4-types-for-arcgis-api

- Esri provides type definitions for the ArcGIS API for JavaScript
- For Esri v4.x, install them with
  ```shell
npm install --save @types/arcgis-js-api
  ```
  and also add to tsconfig.app.json.
  ```json
  "types": ["arcgis-js-api"]
  ```
- Types are available through global "__esri" namespace for Esri v4.x
  - We recommend renaming to "esri" with
    ```javascript
    import esri = __esri;
    ```
4-types-for-arcgis-api

- More info is available at https://github.com/Esri/jsapi-resources/

```javascript
// without
const map = new EsriMap({ /* basemap, layers, etc. */ });

// with
const mapProperties: esri.MapProperties = {
  basemap: 'streets'
};
const map: esri.Map = new EsriMap(mapProperties);

// without
const arrayOfGraphics = [];
const myGraphic = new Graphic({ /* geometry, symbol, etc. */ });
arrayOfGraphics.push(myGraphic);

// with
const arrayOfGraphics: esri.Graphic[] = [];
const myGraphic: esri.Graphic = new Graphic({ /* geometry, symbol, etc. */ });
arrayOfGraphics.push(myGraphic);
```
Example of IntelliSense in VS Code

```javascript
const mapOptions: esri.MapProperties = {
  basemap:
};
const map: esri.Map = new EsriMap(mapOptions);

Specifies a basemap for the map. The basemap is a set of tile layers that give geographic context to the MapView or SceneView and the other operational layers in the map. This value can be an instance of
```

Thanks, type definitions!
Asynchronous Operations

Three ways to manage out of process, out of sequence operations:

Promises, Custom Events, & Observables
<table>
<thead>
<tr>
<th>Which Async Pattern Can I Use?</th>
<th>Promises</th>
<th>Custom Events</th>
<th>RxJS Observables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent-to-Child Components</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Service to Component</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>One-to-One Communication</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>One-to-Many Communication</td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
Esri Dev Summit 2018: Async Operations Using Promises

Select a world wonder. The selector is then disabled while the panMap method is called, which returns a promise. Once the map has panned, the map component resolves the promise, and the dashboard selector is enabled again.

Select a World Wonder

Colosseum
Angular Component Structure -- Promises

```typescript
import { Component } from '@angular/core';

@Component(
    {
        selector: 'app-component',
        templateUrl: './component.component.html',
        styleUrls: ['./component.component.css']
    }
)
export class AppComponent
{
    title = 'app';

    pointSelected()
    {
        promise = panMap()
        3. promise.then()
    }

    panMap()
    {
        return promise
        promise.resolve();
    }
}
```
Angular Component Structure -- Events

```typescript
// app-component.ts

// header-components.ts

// dashboard-components.ts
1. pointSelected()
   panMap();
}
3. eventHandler()

// map-components.ts
2. panMap()
   event.emit();
```

Managing Asynchronous Operations with Events
2/28/2018
Angular Component Structure -- Observables

```typescript
app.component.ts

header.component.ts

dashboard.component.ts
  control-panel.component.ts
    1. pointSelected(){
        service.requestPan();
    }
    2. requestPan(){
        requestSubscribers.next()
    }
    3. requestSubscription(){
        service.panComplete()
    }
    4. panComplete(){
        completeSubscribers.next()
    }

map.component.ts
```

Managing Asynchronous Operations with Observables
2/28/2016
Maintaining Map State

With Services
Managing Map State with Services: click map to set point graphics persisted in map state
State Management App Schema

deply.component.ts

header.component.ts

navigation.component.ts

dashboard.component.ts

map.component.ts

private mcService: MapStateService

map-state.service.ts

gtPoints()  addPoint()
9-arcgis-api-service

- Refactor sample 2 to move map creation to a service
- Create the map service
- Inject the service into app.component
- Replace the old loadModules() code with the new map service
10-ionic

- Using the ArcGIS API for JavaScript with Ionic.
- Ionic is built on Angular
- Ionic life-cycle: async/await
Where to get help: Esri JSAPI

- https://js.arcgis.com
- GeoNet: ArcGIS API for JavaScript Community
- GIS Stack Exchange
Where to get help: Angular

- Angular docs at https://angular.io/
- Stack Overflow
- Your own favorite places for web dev help
Where to get help: esri-loader

- The README is phenomenal and up-to-date
- New Github issue?
  - must include a live demo that shows the specific problem with esri-loader
Expected behavior

- Describe what you expected or wanted to happen.
- What you are trying to achieve?
- Describe your environment/framework and be specific with version numbers (e.g. React 16.2, react-router 4.2, redux 3.7, node 8.3).

Actual behavior

- Describe what occurs in your code.
- Specifically, what seems to work differently than you intended?
- Provide any error messages you see in the console.

Steps to reproduce the behavior

We can only help you if we're able to easily reproduce the behavior you describe above.

Please provide:

1. Steps to reproduce the behavior.
2. A link to an app where we can carry out those steps (e.g. your publicly facing app, a JSFiddle, a JS Bin, etc.).
3. Relevant code snippet(s) (only if not easily obtained from the above link).
Additional resources

- http://esriurl.com/13900
  - Github "sean-olson-e/Angular-and-the-ArcGIS-API-for-JavaScript"
- https://github.com/Esri/angular-cli-esri-map
- "Esri/esri-loader" is your new friend
- Related presentations this week
- Tom Wayson's blog: "Loader of the Things: One Library to Load Them All"
- And more Angular info at "Esri/jsapi-resources"
Thank you

ANDY GUP  @agup
SEAN OLSON  seanolson-io
JACOB WASILKOWSKI  @JWasilGeo