esri

ArcGIS API for JavaScript:

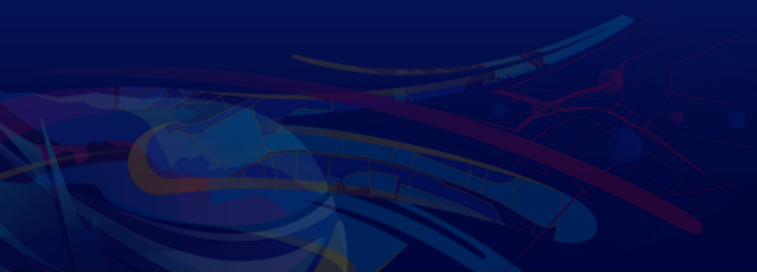
Building apps with ES Modules

René Rubalcava

@odoenet

Agenda

- Why ES Modules?
- ESM CDN
- Assets
- Import Syntax
- Build Tools



What are ES Modules?

Modern way of writing modular JavaScript

```
npm i @arcgis/core
```

```
// old and busted
require(['esri/WebMap'], function(WebMap) {...});

// new hotness
import WebMap from '@arcgis/core/WebMap';
```

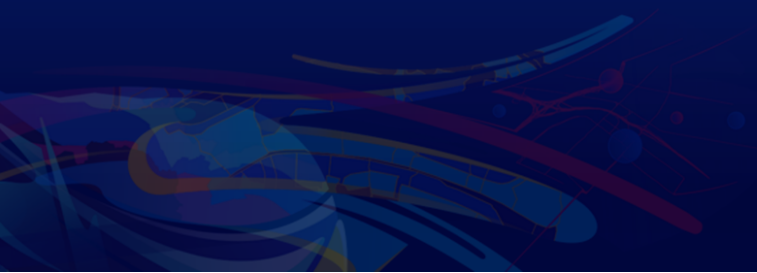
Why does it matter?

- Ideal for modern web development
- Supported in evergreen browsers
- Works with modern build tooling
  - webpack, rollup, snowpack, and more

Getting Started

Prototyping with ESM CDN

- *NOT FOR PRODUCTION*
- Great way to test out some ideas
- Native browser module loading



Prototyping with ESM CDN

```
<script type="module">
  import ArcGISMap from "https://js.arcgis.com/4.18/@arcgis/core/
  import MapView from "https://js.arcgis.com/4.18/@arcgis/core/vi

  const map = new ArcGISMap({
    basemap: 'topo-vector'
  });

  const view = new MapView({
    container: 'viewDiv',
    map,
    zoom: 4,
    center: [-118, 34]
  });
</script>
```

- demo

Assets

you need to copy the `@arcgis/core/assets` folder to your build directory

- for `webpack`, the `@arcgis/webpack-plugin` can do this for you.

Assets

- a few ways to copy assets
- `ncp` - cross-platform copy tool

```
// package.json
{
    "script": {
        "copy": "ncp ./node_modules/@arcgis/core/assets ./public/
        "postinstall": "npm run copy"
    }
}
```

- webpack and rollup have copy plugins

Assets

- Does your app use *routing*?
- App isn't hosted in the root of your domain

```
import config from '@arcgis/core/config';
config.assetsPath = 'https://username.github.io/myrepo/dist/asset
```

Import Syntax

```
// avoid this
import * as geometryEngine from '@arcgis/core/geometry/geometryEn

// only load what you need
import { buffer, intersect } from '@arcgis/core/geometry/geometry
```

Import Syntax

```
// avoid this
import * as watchUtils from '@arcgis/core/core/watchUtils';

// only load what you need
import { whenFalseOnce, whenDefined } from '@arcgis/core/core/wat
```

webpack

webpack builds

```
// webpack.config.js
const ArcGISPlugin = require('@arcgis/webpack-plugin');

module.exports = {
  ...
  plugins: [
    new ArcGISPlugin(),
    ...
  ]
};
```

rollup

rollup builds

```
// rollup.config.js
import commonjs from "@rollup/plugin-commonjs";
import del from "rollup-plugin-delete";
import resolve from "@rollup/plugin-node-resolve";

export default {
  input: "src/main.js",
  output: {
    chunkFileNames: "chunks/[name]-[hash].js",
    dir: "public",
    format: "es"
  },
  plugins: [
    del({ targets: "public/chunks", runOnce: true, verbose: true
    resolve(),
    commonjs()
```

snowpack

snowpack builds

```
// snowpack.config.js
module.exports = {
    mount: {
        // does not copy,
        // but prevents snowpack from trying to parse
        // worker and web assembly files
        "node_modules/@arcgis/core/assets": {
            url: "/assets", static: true,
            resolve: false
        },
        "src": "/",
    },
    plugins: [["@snowpack/plugin-webpack"]]
};
```

webpack module federation

- Dependency sharing
- Dynamically import code from another at runtime

Tip

- You may not want to copy assets folder.
- You can load assets from an external source
- *Make sure versions match*

```
import config from '@arcgis/core/config';
config.assetsPath = 'https://cdn.jsdelivr.net/npm/arcgis-js-api@4
```

Summary

esri

THE SCIENCE OF WHERE

Copyright © 2021 Esri. All rights reserved.

Please provide your feedback for this session by clicking on the session survey link directly below the video.