Localization

What is it?

- "Localization" / "Translation" / "i18n"
- No "hardcoded strings" within your code

Why?

- Easier at the beginning!

- Not multi-language? May be requirement later

- Cleaner code - no "hardcoded strings" within your code

Localization in Experience Builder

- OOTB widgets are already localized/translated
- Tools are built-in!

manifest.json: translatedLocales

- Within manifest.json

- What languages are supported

```
"translatedLocales": [
    "en",
    "es",
    "zh-cn"
]
```

Translation Files

- "runtime/translations/default.ts"

- "runtime/translations/es.js"

- "runtime/translations/zn-ch.js"

- …

```
export default {
    _widgetLabel: 'My Widget',
    str1: 'String 1',
}
```

- "setting/translations/default.ts"

- "setting/translations/es.js"

- …

```
System.register([], function (_export) {return {execute: function () {_export({
    // the strings
    _widgetLabel: 'Translated Widget Name',
    str1: 'Translated String 1',
})}}});
```

Using the strings

- First:
  ```
  import defaultMessages from "./translations/default";
  ```

- Usually:
  ```
  <FormattedMessage id="str1" defaultMessage={defaultMessages.str1}/>
  ```

- Direct:
  ```
  props.intl.formatMessage({id: 'str1', defaultMessage: defaultMessage.str1})
  ```

Debugging

- URL param Format:

```
&locale=es
```

Walk through

Add localization to the simple widget

Activity

20 Minutes

- Add translated strings to the widget you created in the previous lesson.

- Bonus:
  - Add multiple language support

- Documentation:
https://developers.arcgis.com/experience-builder/guide/extend-base-widget/#i18n-support

</SCRIPT>

esri | THE SCIENCE OF WHERE®

LIVE BY THE CODE 3