WEB COMPONENTS

Saramma Varghese

CONTENTS

- Web Component fundamentals
- Creation of web components
- Building of Angular Elements

MISSING PART IN ANGULAR COMPONENTS

- Style Encapsulation
- Ways of allowing some styling of these Elements
- Using Elements across teams using different Framework
- Flexibility

“**Web Component** provide a lot of this” - all missing features provided by web component

WEB COMPONENTS

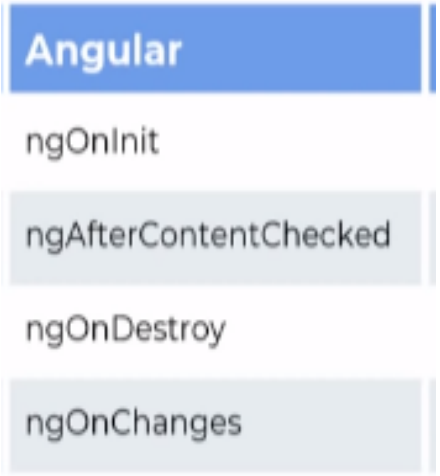
- Platform agnostic
- Goal : encapsulate the code for the components into a nice, reusable package for maximum interoperability.
- It consist of 3 technologies:
  - HTML Template
  - Custom Elements
  - Shadow DOM

HTML TEMPLATE

- It is HTML tag having start and end
- Any in between the tag is template, browser not load it.
- Benefits :
  - Browser parse it once
  - Fast
  - Easy to use

CUSTOM ELEMENTS

- E6 classes
- Inherited from HTML elements
- Can extended from existing HTML element
- Life cycle of custom elements



CONT...

- Steps in creating custom elements-
  - Define custom element

```
<bb-red-strawberry
   img="strawberry.jpg"
   description="Strawberries from Sherry's Garden">
</bb-red-strawberry>
```

```
class BBRedStrawberryElement extends HTMLElement {
  constructor() {
    super();
  }
}

// Define custom element
customElements.define("bb-red-strawberry", BBRedStrawberryElement);
```

CONT...

- Input need to append to template using ConnectedCallback

```
connectedCallback() {
    this.innerHTML = template;
    this._$image = this.querySelector("#element-image");
    this._$description = this.querySelector("#element-description");
    this._render(this);
}
```

- Input to components can change anytime.

  ChangedCallback is used to react to that changes

SHADOW DOM

- Introduced for implementing Style Encapsulation
- It create boundary around custom elements and provide that encapsulation

```
class BBRedStrawberryElement extends HTMLElement {
 constructor() {
  super();
  const template = document.createElement("template");

  // Shadow DOM
  this.attachShadow({ "mode": "open" });
  this.shadowRoot.appendChild(template.content.cloneNode(true));
}
```

CONT...

- **Slots-**
  - flexibility for developers to add html
  - Inside template put an placeholder

```
<img id="shadow-image" class="shadow__image">
<div id="shadow-info" class="shadow__info">
  <slot name="title" id="title" role="header" class="shadow__title"></slot>
  <slot name="description" id="description" class="shadow__description"></slot>
</div>
```

```
<bb-red-strawberry
  img="strawberry.jpg">
  <div slot="title"><i>Strawberries</i></div>
  <div slot="description">Sherry's berries finest strawberries</div>
</bb-red-strawberry>
```

CONT...

- **Custom Properties-**
  - Give flexibility to style the component.
  - Specific to web component
  - In css file,

```
background-color: var(--background-color, #fff);
```

  - In html file,

```
style="--background-color: #A11B38;">
```

ANGULAR ELEMENTS

- Install cli-

  npm i –g @angular/cli

- Create new project-

  ng new projectname

- Add angular elements-

  ng add @angular/elements

- Add custom elements-

  npm i @webcomponent/custom-elements --save

CONT...

- In pollyfills.ts –

  Import ‘@webcomponent/custom-elements/custom-elements.min’;

- In tsconfig.json –

  “target” : “es2015”

- Create component-

  ng g c componentname

CONT...

- ViewEncapsulation is used for introducing shadow DOM in angular

card.component.ts

```
import { Component, OnInit, ViewEncapsulation, Input } from
'@angular/core';

@Component({
  selector: 'bb-card',
  template: ` ... `,
  styleUrls: ['./bb-card.scss'],
  encapsulation: ViewEncapsulation.ShadowDom
})
```

CONT...

- Register component in NgModule-

  entryComponents:[cardComponent]

- Custom element declaration -

```
@NgModule({
  declarations: [CardComponent],
  imports: [BrowserModule],
  entryComponents: [CardComponent],
})

export class AppModule {
 constructor(private injector: Injector) {
   const bbCard = createCustomElement(CardComponent, { injector });
   customElements.define('bb-card', bbCard);
 }
 ngDoBootstrap() {}
}
```

CONT...

- Build using build plus
- Advantages of build plus-
  - Extend cli
  - No eject
  - Build single bundle
  - No need to add angular multiple times
  - Universal module

CONT...

- Add build plus -

  ng add ngx –build –plus

- In angular.json –

  “builder” : “ngx –build –plus:build”

WEBCOMPONENTS IN EXISTING ANGULAR PROJECT

- Install custom elements

  npm install @webcomponent/custom-elements –save

- Add polyfills.ts –

  Import '@webcomponent/custom-elements/custom-elements.min';

CONT...

- Inside app module.ts –

  Schemas : [

  CUSTOM_ELEMENTS_SCHEMA

  ]

ADVANTAGES

- A web component can be used in multiple applications. It provides interoperability between frameworks, developing the web component ecosystem. This makes it **reusable**.

- A web component has a template that can be used to put the entire markup separately, making it more **maintainable.**

CONT...

- As web components are developed using HTML, CSS, and JavaScript, it can run on different browsers. This makes it **platform independent**.

- Shadow DOM provides **encapsulation mechanism** to style, script, and HTML markup**.**

THANK YOU....