

Lightning Web Components

- Nithesh Nekkanti

Agenda

- Why Lightning Web Components (LWC) ?
- Web Components
- Lightning Web Components
 - Lifecycle Methods
 - Call Apex methods
 - Lightning Data Services
 - Events
 - LWC and Aura Co-existence and Communication between them.
 - Misc.

Why LWC?

Back in 2014

Data services
UI components
Component model
Templates
Rendering optimization
Modules
Language extensions

Frameworks

ECMAScript 5
Events
Standard Elements
Rendering

Web Standards

In 2018-19

Specialized services
Data services
UI components

Frameworks

Web Components
Templates
Custom elements
Shadow DOM
Modules
ECMAScript 7
Events
Standard Elements
Rendering

Web Standards

Salesforce's LWC

Security
Lightning Data Service
Base Lightning Components

Lightning
Web Components

Web Components
Templates
Custom elements
Shadow DOM
Modules
ECMAScript 7
Events
Standard Elements
Rendering

Web Standards

Web Components

Pillars of Web Components

- Import
- Template
- Custom Element
- Shadow DOM



Demo of a UserCard Web Component
@
<https://pentagonal-yard.glitch.me/>

Lightning Web Components

LifeCycle Methods

- `constructor()`
- `connectedCallback()`
- `disconnectedCallback()`
- `render()`
- `renderedCallback() *`
- `errorCallback() *`

Lightning Web Components

Anatomy

Call Apex from LWC

Import the Apex Method Name from the controller as a module

```
import findContacts from '@salesforce/apex/ContactController.findContacts';  
import <method-name> from '@salesforce/apex/<controller-name>.<method-name>'
```

Actual Server Call

```
@wire(findContacts, { searchKey : <param value> })  
  results({error, data }) {  
    if(data) {  
      this.contacts = data;  
    } else if(error) {  
      console.error(error);  
    }  
  }  
}
```

(or) @wire(findContacts) contacts;

Lightning Data Services in LWC

FEATURE	LIGHTNING-RECORD-FORM	LIGHTNING-RECORD-VIEW-FORM	LIGHTNING-RECORD-EDIT-FORM
Create Records	✓		✓
Edit Records	✓		✓
View Records	✓	✓	
Read-Only Mode	✓	✓	
Layout Types	✓		
Multi Column Layout	✓		
Custom Layout for Fields		✓	✓
Custom Rendering of Record Data		✓	✓

Record Creation in JS :

```
import { createRecord } from 'lightning/uiRecordApi';
```

Lightning Data Services in LWC

Record Creation in JS

```
import { createRecord } from 'lightning/uiRecordApi';

const fields = {};
fields.Name = this.name;
const recordInput = { apiName: Account, fields };
createRecord(recordInput)
    .then(account => {
        console.log(account.id);
    })
    .catch(error => {
        console.error(error.body.message);
    });
```

Events (Dispatch)

* No Application Events in Lightning Web Components

Simple Event Dispatch:

```
this.dispatchEvent(new CustomEvent('event-name'));
```

Advanced One:

```
new CustomEvent('event-name', {  
    bubbles: true,  
    composed: false,  
    detail: { <payload> } }  
));
```

Events (Handling)

In simpleContactCard component's JS

```
this.dispatchEvent(new CustomEvent('notify'));
```

In The Container App

```
<template>
  <c-my-component onnotify={handleNotify}></c-my-component>
</template>
```

In the Container App's JS

```
function handleNotify(e) {
  console.log(e.detail);
}
```

Events (Handling)

In simpleContactCard component's JS

```
this.dispatchEvent(new CustomEvent('notify'));
```

In The Container App

```
<template>
  <c-my-component onnotify={handleNotify}></c-my-component>
</template>
```

In the Container App's JS

```
function handleNotify(e) {
  console.log(e.detail);
}
```

Aura & LWC Coexistence

```
<aura:component>  
    <!-- This is an LWC component -->  
    <c:lwcApp onfilterchange="{!c.handleFilterChange}"/>  
</aura:component>
```


Misc.

- `refreshApex(<reactive-property>);`
- No `Aura:Ids` && `ids`
- **Salesforce Modules** - https://developer.salesforce.com/docs/component-library/documentation/lwc/reference_salesforce_modules
- **Template Directives** - https://developer.salesforce.com/docs/component-library/documentation/lwc/lwc.reference_directives
- **Share JS Between LWC & Aura**

Misc.

Aura components

Attributes
aura:if
~~Application events~~
Component events
Public functions
~~Expressions~~
Marker interfaces
aura:id + find()
global/public/private
~~valueChange event~~
Composition markup
Facet (Aura.Component[])
aura:iteration
myEvent="{!c.doStuff}"
aura:handler name="myEvent"
component.set('count', 12)

Lightning web components

slot
for:each directive
querySelectorAll
Module properties
@api decorator
this.count = 12
Metadata file "targets"
Template file
@salesforce/schema imports
CustomEvent
addEventListener('myEvent')
if:true / if:false directives
onmyevent={runMyHandler}
isExposed=true

Thanks!

References:

<https://developer.salesforce.com/docs/component-library/documentation/lwc/>

<https://www.html5rocks.com/en/tutorials/webcomponents/imports/>

https://developer.mozilla.org/en-US/docs/Web/Web_Components

<https://www.pluralsight.com/courses/play-by-play-understanding-lightning-web-components-salesforce-developers>