



## REFRESH STANDARD COMPONENTS FROM LIGHTNING WEB COMPONENT

January 31, 2021   sanketthoughts   Lightning Aura Component, Lightning Web Component   Leave a comment

Lightning web Component does not have an equivalent functionality as Aura Components to refresh standard components on the Lightning page. Aura Components uses [force:refreshView](#) event reloads data for standard components. Lightning Web Component lacks this kind of feature. As a workaround, We need to borrow help from Aura component while working with Lightning Web Component in scenarios where you need standard component on the lightning record page to reflect the latest changes.

### Use Case – force:refreshView in LWC

Business wants a button to link file to the record which is generated using apex method called onclick. The notes and attachment related list needs to reflect the file immediately once button is clicked.

### Solution – force:refreshView in LWC

We are going to create two components – Lightning Web Component and Aura Component. Aura Component calls Lightning Web Component inside it and listen to event fired from Lightning Web Component and refresh the standard component on the lightning page. Components name are:-

- **fileCreationUsingApex** (LWC)
- **standardComponentRefresher** (LWC)

#### fileCreationUsingApex.html

The html file of the Lightning web component has a button which fires the apex method.

```
<template>
  <lightning-card title="Refresh Standard Component" icon-name="custom:custom19">
    <div class="slds-m-around_medium">
      <lightning-button label="Link File" onclick={handleClick}></lightning-button>
    </div>
  </lightning-card>
</template>
```

#### fileCreationUsingApex.js

The JavaScript file of the component has the public property recordId decorated with @api decorator to fetch the current record id and has handleClick method bind with button on html file which invoke the apex method. Once method executes successfully, it fires an custom event to [communicate with parent Aura Component](#).

```
import { LightningElement, api } from 'lwc';
import linkExistingFile from '@salesforce/apex/fileCreationController.linkExistingFile';
export default class FileCreationUsingApex extends LightningElement {
  @api recordId;
  handleClick() {
    linkExistingFile({ parentId: this.recordId }).then(result => {
      // Creates the event with the record ID data.
      const selectedEvent = new CustomEvent('linked', { detail: this.recordId });
      // Dispatches the event.
      this.dispatchEvent(selectedEvent);
    }).catch(error => {
      console.log(JSON.stringify(error));
    });
  }
}
```

#### fileCreationUsingApex.js-meta.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
  <apiVersion>50.0</apiVersion>
  <isExposed>true</isExposed>
  <targets>
    <target>lightning__RecordPage</target>
  </targets>
</LightningComponentBundle>
```

Now, The lightning Web Component is ready, Let's see the Aura Component.

**standardComponentRefresher.cmp**

The Aura Component embed Lightning Web Component inside it. It passes the current record id to the child Lightning Web Component and has the event handler defined.

```
<aura:component implements="flexipage:availableForAllPageTypes,force:hasRecordId" access="global">
    <c:fileCreationUsingApex onlinked="{!c.refresh}" recordId="{!v.recordId}" />
</aura:component>
```

**standardComponentRefresherController.js**

```
(({
    refresh: function (component, event, helper) {
        $A.get('e.force:refreshView').fire();
    }
}))
```

We also have an Apex controller named **fileCreationController** and have method to insert file and link file to the record.

**fileCreationController**

```
public with sharing class fileCreationController {
    @AuraEnabled
    public static void linkExistingFile(String parentId) {
        ContentVersion cv = new ContentVersion(
            Title='SampleTitle',
            PathOnClient ='SampleTitle.jpg',
            VersionData = Blob.valueOf('Unit Test ContentVersion Body to be insert in test class for testing the'),
            FirstPublishLocationId = parentId
        );
        insert cv;
    }
}
```

**Demo – Refresh Standard Component from LWC**

Once you have deployed all the components and apex class into your org, go to desired lightning record page and edit the page with Lightning App Builder to place your component on the page. You should place Aura Component on the lightning page.