

Download All Related Files from a Record Using LWC

Overview

In many service-oriented organizations, agents often need to download and review multiple documents attached to Salesforce records. This guide shows how to implement a Lightning Web Component (LWC) and Apex-based solution that lets users download all related files with just a few clicks.

Use Case

Field Coordinators or support agents frequently receive documents—such as medical faxes or signed forms—attached to records (e.g., Case). These need to be reviewed offline. To simplify this, we've introduced a Quick Action that allows users to:

- View the count of downloadable files
 - Confirm and trigger the download
 - Download multiple files directly to their machine
-

Key Features

- Built with Lightning Web Components (LWC)
 - Apex controller fetches and stages documents
 - Sequential download logic (to avoid browser throttling)
 - Optional cleanup of temporary download artifacts
-

Objective

Enable users to download all related files from a record through a simple button or link in LWC. Files are fetched via Apex and made available as direct download links.

Key Steps and Code Snippets

1. Fetch Related Files

```
@wire(getRelatedFiles, { recordId: '$recordId' })
wiredGetRelatedFiles({ data, error }) {
  if (data) {
    this.contentDocumentIds = data;
    this.isLoading = false;
  } else if (error) {
    this.isLoading = false;
    this.showToast('Error', 'Error fetching files: ' +
error.body.message, 'error');
  }
}
```

2. Download Files

```
downloadFiles() {
  this.downloading = true;
  getTemporaryFiles({ contentDocumentIds: this.contentDocumentIds,
recordId: this.recordId })
    .then(result => {
      result.forEach(file => this.downloadFile(file));
    })
    .catch(error => {
      console.error('Error:', error);
    });
}
```

3. Download Individual File

```
downloadFile(file) {  
    this[NavigationMixin.Navigate]({  
        type: 'standard__webPage',  
        attributes: {  
            url: `/sfc/servlet.shepherd/version/download/${file.Id}`  
        }  
    });  
}
```

4. Display Toast Messages

```
showToast(title, message, variant) {  
    this.dispatchEvent(new ShowToastEvent({ title, message, variant }));  
}
```

Key Apex Code

```
public without sharing class DownloadRelatedFilesController {  
  
    @AuraEnabled(cacheable=true)  
    public static List<Id> getRelatedFiles(Id recordId) {  
        return [SELECT ContentDocumentId FROM ContentDocumentLink WHERE  
LinkedEntityId = :recordId];  
    }  
  
    @AuraEnabled  
    public static List<ContentVersion> getTemporaryFiles(List<Id>  
contentDocumentIds, Id recordId) {  
        return [  
            SELECT Id, Title, FileType  
            FROM ContentVersion  
            WHERE ContentDocumentId IN :contentDocumentIds  
            AND IsLatest = true  
        ];  
    }  
}
```

Explanation

LWC:

- Uses `@wire` to fetch related file IDs (`ContentDocumentId`)
- Calls `getTemporaryFiles` Apex method to retrieve the latest file versions
- Uses `NavigationMixin.Navigate` to trigger direct downloads
- Displays message/error using `ShowToastEvent`

Apex:

- `getRelatedFiles`: Fetches `ContentDocumentLink` records for the target record
 - `getTemporaryFiles`: Retrieves the latest file version (`ContentVersion`) for each document
-

Conclusion

With a few simple Apex methods and JavaScript handlers in LWC, we created a lightweight solution to bulk download all files related to a Salesforce record. This component can be extended with:

- Progress indicators
- Pagination or batching for large file sets
- File access permission checks

This feature is ideal for any team that frequently manages documents—making reviews faster, easier, and more efficient.