

Custom Pagination in LWC

The Context

- Paginate your large data-tables using effective pagination solution in LWC.

Approach

- Clone or copy pagination component from source [here](#)
- Add pagination component to your parent component where data table is implemented.

Parentcomponent.html

```
<div class="slds-float_right slds-size_1-of-1">
  <template if:true={totalAccounts}>
    <c-pagination totalrecords={totalAccounts} page-size={pageSize} onpagination={handlePagination}></c-pagination>
  </template>
</div>
```

- Parameters need to be passed from Parent component
 - pageSize** (number of records to be displayed per page- here pageSize is 10)
 - Totalrecords** (total number of records)]
- Events - **pagination**
 - Implement 'onpagination' event action handler in your parent component as below. This event returns clicked page number.

Parentcomponent.js

```
handlePagination(event){
  const start = (event.detail-1)*this.pageSize;
  const end = this.pageSize*event.detail;
  this.accounts = this.allaccounts.slice(start, end);
}
```

Code

Parentcomponent.html

```
<template>
  <div class="slds-p-around_medium">
    <lightning-datatable key-field="id" data={accounts} columns={columns}>
      </lightning-datatable>

  </div>

  <div class="slds-float_right slds-size_1-of-1">
    <template if:true={totalAccounts}>
      <c-pagination totalrecords={totalAccounts} page-size={pageSize}
        onpagination={handlePagination}></c-pagination>
    </template>
  </div>
</template>
```

Parentcomponent.js

```
import { LightningElement } from 'lwc';
import getAccounts from '@salesforce/apex/AccountDataController.getAccounts';

export default class AccountDataTable extends LightningElement {
  accounts;
  allaccounts;
  totalAccounts = 0;
  pageSize = 10 ;
  columns = [
    { 'label' : 'Name', 'fieldName' : 'name'},
    { 'label' : 'Account Number', 'fieldName' : 'accountnumber'},
    { 'label' : 'Account Source', 'fieldName' : 'source'},
    { 'label' : 'Active', 'fieldName' : 'active'},
    { 'label' : 'Phone', 'fieldName' : 'phone'}
  ];
};
```

```
connectedCallback(){
  this.init();
}

init(){
  getAccounts().then(res=>{
    let accounts = res.lstAcc;
    let acc_a = [];
    if(accounts){
      this.totalAccounts = accounts.length;
      accounts.forEach(acc => {
        let a ={};
        a.Id = acc.Id;
        a.name = acc.Name;
        a.accountnumber = acc.AccountNumber;
        a.source = acc.AccountSource;
        a.active = acc.Active__c;
        a.phone = acc.Phone;
        acc_a.push(a);
      });
      this.allaccounts = acc_a;
      console.log(this.accounts);
      this.accounts = this.allaccounts.slice(0,this.pageSize);
    }
  });
}

handlePagation(event){
  const start = (event.detail-1)*this.pageSize;
  const end = this.pageSize*event.detail;
  console.log(start,end);
  this.accounts = this.allaccounts.slice(start, end);
}
}
```

Output

<input type="checkbox"/> Name	Account Number	Account Source	Active	Phone
<input type="checkbox"/> Lightning Web Components Specialist S...				
<input type="checkbox"/> Test				
<input type="checkbox"/> test39	39	email	Yes	7755433
<input type="checkbox"/> test40	40	email	Yes	7755433
<input type="checkbox"/> test41	41	email	Yes	7755433
<input type="checkbox"/> test42	42	email	Yes	7755433
<input type="checkbox"/> test43	43	email	Yes	7755433
<input type="checkbox"/> test44	44	email	Yes	7755433
<input type="checkbox"/> test45	45	email	Yes	7755433
<input type="checkbox"/> test46	46	email	Yes	7755433

Viewing 1-10 of 202

< 1 2 3 4 5 6 7 8 ... 20 >

References:

1. [JS Slice](#)
2. [Lightning datatable](#)

