DisplayInvoiceListTable - Full Developer Guide

# 1. Overview

The DisplayInvoiceListTable Lightning Web Component (LWC) allows users to view, sort, and paginate a list of invoices and their associated invoice line items. It also includes functionality to add new invoice line items via a modal form. This guide will walk a developer through setting up the component from scratch.

# 2. Prerequisites

- Salesforce Developer Org or Sandbox  
- Access to create Apex classes, LWC components, and custom objects  
- Understanding of Apex, LWC, and Salesforce Data Model

# 3. Salesforce Setup

Ensure the following custom objects and fields are created in Salesforce:

* Invoice\_\_c (Custom Object):
* • Invoice\_Date\_\_c (Date)
* • Buyer\_Name\_\_c (Lookup to Account)
* • Seller\_Name\_\_c (Lookup to Account)
* • Invoice\_Status\_\_c (Picklist)
* • Pending\_Amount\_\_c (Roll-up Summary)
* Invoice\_Line\_\_c (Custom Object):
* • Product\_Name\_\_c (Lookup to Product)
* • Quantity\_\_c (Number)
* • Price\_\_c (Currency)
* • Taxes\_\_c (Currency)
* • Product\_Total\_\_c (Formula)
* • Grand\_Total\_\_c (Formula)

# 4. Apex Controller

Create an Apex class named DisplayInvoiceListTable.cls with the following methods:

- getInvoiceList()  
- getInvoiceLineById(Id invoiceId)  
- getAllProducts()  
- saveInvoiceLine(List<Invoice\_Line\_\_c>)

# 5. Lightning Web Component Setup

Create the LWC named DisplayInvoiceListTable with the following files:

- displayInvoiceListTable.html  
- displayInvoiceListTable.js  
- displayInvoiceListTable.js-meta.xml

# 6. JavaScript Logic

Handle Apex calls, pagination, modals, sorting, and navigation in displayInvoiceListTable.js.

# 7. HTML Template

Use SLDS and lightning components for the UI layout, such as lightning-datatable, lightning-button, lightning-input, and lightning-combobox.

# 8. Deployment Instructions

1. Deploy Apex class to Salesforce.  
2. Deploy LWC files to Salesforce using VS Code or Developer Console.  
3. Add the component to a Lightning App Page using Lightning App Builder.  
4. Test all functionality end-to-end.

# 9. Testing Instructions

Verify that:  
- Invoice list loads with data and pagination  
- Clicking 'View' loads invoice line items  
- Add Product modal opens and saves records  
- Sorting and navigation work as expected

# 10. Best Practices & Tips

• Use toast messages for success/error feedback  
• Keep logic modular and reusable  
• Use NavigationMixin to navigate to records  
• Always validate user input before saving