

HOW TO USE REFRESH APEX IN LIGHTNING WEB COMPONENTS

Swipe next →

@TrailheadIQ

Introduction to refreshApex in LWC

- refreshApex is a function used in Lightning Web Components (LWC) to refresh data retrieved via @wire after certain actions, such as record updates or user-triggered events.
- It's essential when you need to fetch the latest server-side data without reloading the entire page.

Swipe next →

Why Use refreshApex?

- **Refresh Data on Demand:** It lets you manually re-fetch data from the server, ensuring that your component stays up-to-date with changes made to Salesforce records.
- **Improves User Experience:** By avoiding full-page reloads, you provide a smoother, faster user experience.
- **Control Over Data:** You decide when to fetch fresh data based on user actions like saving a record, switching filters, or navigating between components.

How refreshApex Works

- refreshApex works by refreshing the cache associated with a @wire method.
- It triggers the wire service to retrieve the latest data from the backend.

Basic Workflow:

- Data is initially fetched using @wire.
- After a user action (like saving a record), call refreshApex to refresh the cached data.
- The UI is automatically updated with the latest data after refresh.

Example Code - refreshApex in Action

Here's an example of how to use refreshApex with @wire to refresh a list of accounts when a new account is created.

```
Js

import { LightningElement, wire } from 'lwc';
import { refreshApex } from '@salesforce/apex';
import getAccounts from '@salesforce/apex/AccountController.getAccounts';
import createAccount from '@salesforce/apex/AccountController.createAccount';

export default class RefreshApexExample extends LightningElement {
  accounts;
  wiredAccounts;

  @wire(getAccounts)
  wiredAccountsResult(result) {
    this.wiredAccounts = result;
    if (result.data) {
      this.accounts = result.data;
    } else if (result.error) {
      this.error = result.error;
    }
  }

  handleCreateAccount() {
    // Call an Apex method to create an account
    createAccount()
      .then(() => {
        // Refresh the list of accounts once the account is created
        return refreshApex(this.wiredAccounts);
      })
      .catch(error => {
        this.error = error;
      });
  }
}
```

Key Points to Remember

- **Only Refresh Wired Data:** refreshApex only works with data retrieved via @wire. It does not refresh imperative Apex calls.
- **Manual Control:** It gives you control over when to refresh the data, making it useful for cases where data changes based on user interaction.
- **Efficient Data Handling:** By refreshing only the necessary parts of the UI, it helps improve the performance and user experience of your application.

Swipe next →

When to Use refreshApex

- After creating, updating, or deleting a record to ensure the component reflects the most recent data.
- When users change filters or perform actions that modify the underlying data set.
- In dashboards or reporting components where data needs to be regularly updated without reloading the whole page.

Swipe next →

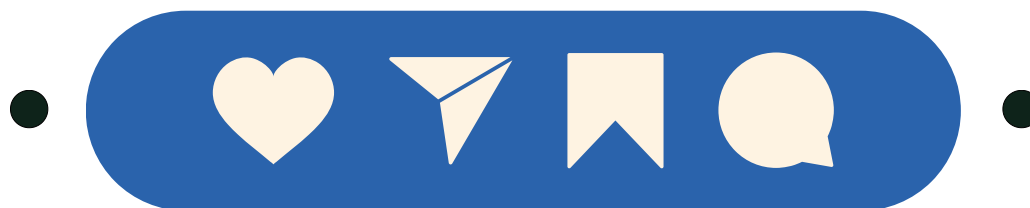
Key Takeaways

- refreshApex is a valuable tool for maintaining up-to-date data in your Lightning Web Components.
- It provides control over data refresh, improves user experience, and boosts application performance.
- By using refreshApex, you ensure your LWC components are always synced with Salesforce records without relying on page reloads.

Swipe next →

THANK YOU

Remember, every like, share, and comment helps us reach more people and make a bigger impact. Together, we can make the Salesforce community stronger and more informed. #TrailheadIQ #SalesforceCommunity



www.trailheadiq.com



TRAILHEAD IQ