

WIRE VS IMPERATIVE

Salesforce Lightning Web Components (LWC) provides two ways to call Apex methods – wire and imperative

By Deepa

WIRE-@WIRE DECORATOR

Import Wire in LWC

```
import { LightningElement, wire } from 'lwc';
```

Automatically updates when the tracked property changes, keeping the component in sync with the server.

The Apex method calling using wire can be annotated as @Auraenabled(catchable=true) so that the data is stored in temporary client side cache memory.

DML Operation cannot be performed. only we can read the salesforce org data.

IMPERATIVE

Explicitly invoked by the developer when needed (e.g., on button click, page load, etc.).

No automatic updates. Data must be manually updated when needed by the developer.

Stores data in a regular variable, and the developer must manually update it after fetching new data.

Data is fetched explicitly by the developer, based on when the developer chooses to trigger the call.

WIRE-@WIRE DECORATOR

Stores data in a reactive variable, which updates automatically when the tracked property changes.

Error handling is automatic and managed by the wire service. Errors are returned through the wire result.

Cannot be used with objects not supported by the User Interface API, such as Task and Event.

Might be challenging to debug in complex scenarios

IMPERATIVE

Full control over when and how data is fetched, allowing for more precise timing and custom logic.

DML Operations like insert, update and delete can be performed

The Apex method calling using imperative must be annotated as @Auraenabled.

Requires manual error handling and custom logic to handle exceptions and responses.

More flexible for debugging, as the developer has control over the entire flow and can add custom logs.



THANK YOU