Lightning component development using LWC\_Day 3

Agenda

1. Publishing of lightning components
2. Salesforce lightning Case Study ( Initiation )

Implementing data connectivity in salesforce using life cycle hook methods of LWC

In components, we need to access data sometimes when the component loads into web browser or mobile app without using event (like clicking a button)

1. Constructor() – which will call before the component loads into browser or app
2. connectedCallback() – which invokes after the component loads into browser and it will be called after “constructor()” in the call stack.
3. renderedCallback() – will invoke after the focus returned from the child component into the parent component while processing the data.
4. disconnectedCallback() – will be called when the components got removed from the browser or app at runtime.
5. errorCallback() – will be executed while error rendering happens the component.

Activity:

Create a component to display data from the “Account” object while loading component in the web browser.

**Publishing Lightning Components**

Lightning components can be surfaced into different channels of salesforce platform to be used by the end-users of applications.

Platforms into which lightning components can be published are:

1. Salesforce Lightning Tab

Lightning Tabs are used to create “Object Tabs” in salesforce. The Tab is made up of single component and it will be spanned across the visible screen size of the device. Lightning Tabs are also called as “Standalone Tabs” where the entire functionality related to specific object can be implemented by single component loaded into the Tab.

To publish component into lightning Tab:

AURA components: interface (force:appHostable)

LWC component: target(lightning\_\_Tab)

Activity:

Create component and publish as standalone Tab in application.

1. Salesforce Lightning Page

Page is a container made of multiple components. Developer can load standard, custom or custom managed components in the page.

Lightning pages support “Single Page Application” architecture where the page remains static and individual components in the page will be refreshed with data from the objects in Async approach.

Developers can create applications that work as single page UI by using templates of lightning pages and these templates have form factor alignment for desktop, mobile or tablet devices.

AURA components: interface (flexipage:availableForAllPageTypes)

LWC component: target (lightning\_\_AppPage)

Activity:

Create a component to demonstrate “Single Page Application” architecture from lightning page.

1. Lightning Record Page

When developer wants to customize record pages of salesforce object, it can be done as complete customization where the entire page is redesigned using custom components or partial customization where in the existing page, custom components can be embedded for additional UI use cases.

Custom components are used for getting the data to be shown on the record by accepting salesforce record ID for that record and logic can be applied based on current record.

AURA components: interfaces (force:hasRecordId, flexipage:availableForRecordHome)

LWC components: target (lightning\_\_RecordPage), @api recordId

Activity:

Create a component to show additional information of a record (Opportunity) as embedded data in the current record.

1. Digital Experience Pages (formally called as “Community” pages)

Components can be added to Custom Digital Experience pages for creating public facing sites in salesforce

AURA components: interface (forceCommunity:availableForAllPageTypes)

LWC components: target (lightningCommunity\_\_Page)

1. Global Actions

Components can be used for creating “Global actions” which can be launched from the salesforce header in browser or bottom toolbar in mobile app for providing functionality irrespective of applications.

AURA components: interface (force:lightningQuickAction)

LWC components: Wrap LWC component into AURA component.

1. Standard Button Overrides

Components can be used to change the functionality of “standard buttons” on salesforce objects to provide action customization.

AURA components: interface (lightning:actionOverride)

LWC components: Wrap LWC component into AURA component.

Lightning components are used for publishing using “marker” interfaces in AURA framework and through XML targets in LWC framework.

**Salesforce Lightning Case Study**

Salesforce got custom applications that need to be reviewed and prepare “mobile” app to be accessed from web browser and mobile app.

Create a simpler app for mobile access based on the use cases of “custom” application available in the training org of salesforce.

Custom Applications in Training Org:

Recruitment:

Typical HR application for Hiring process in a company called “AW computing”

Objects in Recruiting App:

Position, Candidate, Job Application, Interviewers, Reviews, Offers, Job Posting Site and Job Postings

Certification:

Training App in company for getting employees certified on Internal Certifications of company “AW computing”

Objects in Certification App:

Certification, Course, Course Delivery, Course Attendee, Certification Attempt, Certified Resource

**Certification Mobile App**

There is demand from training coordinators in AW computing for working on-the-go using mobile for training operations in company. They must work with certifications, courses and course deliveries using mobile as they frequently do actions on those objects.

Prepare a mobile version of “Certification” App and give access to training coordinators for handling operations from phone or tablet.

Steps:

1. Understand the “Certification App” data model to deal with objects.

Certification\_\_c

* Name
* Certification\_Description\_\_c
* Status\_\_c

Course\_\_c

* Name
* Course\_Description\_\_c
* Status\_\_c
* Duration\_\_c
* Certification\_\_c (lookup)

Course\_Delivery\_\_c

* Name (Auto number)
* Region\_\_c
* Location\_\_c
* Start\_Date\_\_c
* Status\_\_c
* Course\_\_c (master-detail)
* Instructor\_\_c (lookup of USER)

Pre-Requisite for building Mobile app in salesforce

1. Install “Salesforce Mobile App” into phone / tablet for testing.
2. Create custom application with accessibility to Desktop and Phone for users in the company
3. Design the requirements using custom lighting components and publish into application for testing through the devices.

**Requirements for creating “mobile app” for Certification App**

R1. Prepare a custom application with name “Certification-Mobile” and provide access to the Training team in AW computing.

Solution: Tool – App Manager

R2: Prepare a custom component that will be added as Certification Tab in the application using which all other objects will be handled.

The component should have “tabbed GUI” and alignment of tabs should vary according to the devices from which the application is launched.

Wire frame as below:

MHome

Details

Trainings

MHome

Details

Trainings

R3: create component for displaying in “MHome” tab of application and the component should have below wire frame.