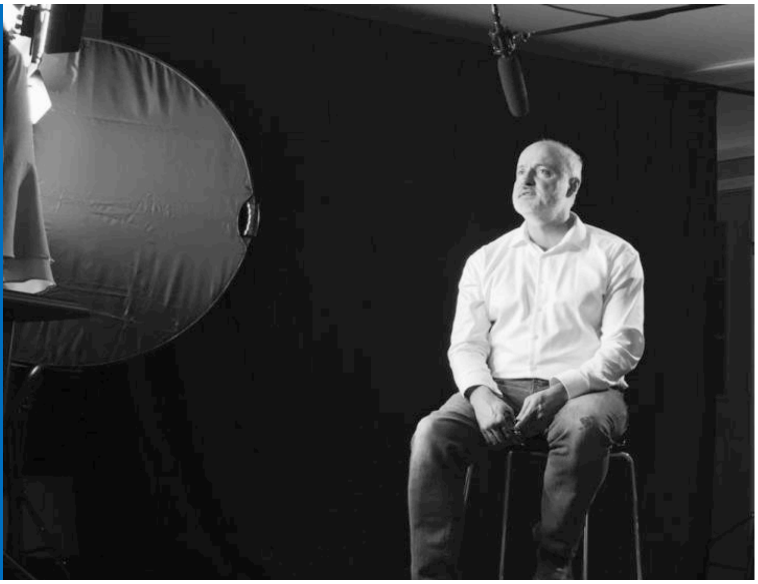
Launch Your Salesforce Career Bootcamp



Module 6: Contract Management App Foundations Pack

Table of Contents

Lesson 1 – Contract Management App Foundations Pack Intro

In this module, the Contract Management App Foundations Pack, we build a more extensive and complex application a Contract Management app. We build out the foundations and security model for a Contract Management application. We then extend and enhance the Contract Management App user interface and user experience.

Notes:

Lesson 2 – Contract Management App Foundations

In this lesson we discuss the foundations of our upcoming Contract Management application. We look at the Fields and Relationships on the Contract standard object.

I also share the Sales Objects ERD from the SOAP API Developer Guide (link below).

We set the Contract Settings to send contract expiration notice emails to Account and Contract Owners.

We then create a sample contract to see the out-of-the-box functionality of Contracts in Salesforce. This establishes a starting point upon which we will start building.

Resource:
Sales Objects Diagram from SOAP API Developer Guide
https://developer.salesforce.com/docs/atlas.en-us.api.meta/api/sforce_api_erd_majors.htm

Notes:

Lesson 3 – Contract Management App Security Model

In this lesson, we explore setting the Contracts object to Private, but demonstrate how Contracts and Accounts are tied together. To make Contracts Private at the Org Wide Default level, you would also be making Accounts private also. This has a cascading effect of having to also make Cases and Opportunities private also.

The next option we explore is closing access to the Contract object at the Profile level.

Notes:

Lesson 4 – Creating Contract Sales Processes

In this lecture we create three different Sales Processes and accompanying Record Types for the Opportunity object related to Contract Management.

Those Sales Processes and Record Types are:

- B2B Widgets
- Call Center Business
- Property Management

Notes:

Lesson 5 – Contract Management App User Interface

In this lecture we create the app and then start addressing some of the user interface limitations. From there, we start improving the UI by adjusting page layouts.

Show a Contract on Opportunity or Quote Details
To show a contract on the opportunity or quote details, use field-level security and field accessibility.

1. From Setup, at the top of the page, select **Object Manager**.
2. In the object management settings for opportunities or quotes, click **Contract** in the fields area.
3. Click **Set Field-Level Security**.
4. Select **Visible** for each profile that you want to see contracts on opportunity or quote details, and then save your changes.
5. Click **View Field Accessibility**.
6. If any profile is marked hidden, click **Hidden**, and then select **Visible** under Page Layout.
7. Save your changes. The profiles are marked editable.

To show an Opportunities or Quotes related list on contracts, use page layouts.

Enable a Contracts Chatter Feed
Add contract fields to be monitored via Chatter.

1. From Setup, enter Feed Tracking in the Quick Find box, then select **Feed Tracking**.
2. From the list of objects, select Contract.
3. Select the **Enable Feed Tracking** checkbox and then the checkboxes for the fields that you want to track.
4. Save your changes.

Resource:
Contract History
https://help.salesforce.com/articleView?id=contract_history.htm&type=5

Notes:

Lesson 6 – Creating a Visualforce Page

In this lesson we create a Visualforce page for our Contract Management application. I also show you how to embed a Visualforce page into a Standard Page Layout.

I also show you how you can embed Visualforce into a Lightning App page as well.

Code Reference for Visualforce Page:

```
<apex:page standardController="Contract" showheader="false" sidebar="false">
 <apex:relatedList list="CombinedAttachments" />
</apex:page>
```

Notes:

Lesson 7 – Creating a Visualforce Page Override

In this lesson we create a Visualforce page override. We also create a tabbed Account Visualforce page.

tabbedAccount Visualforce Page Code:

```
<apex:page standardController="Account" showHeader="true"
      tabStyle="account" >
   <style>
      .activeTab {background-color: #236FBD; color:white;
         background-image:none}
      .inactiveTab { background-color: lightgrey; color:black;
         background-image:none}
   </style>
   <apex:tabPanel switchType="client" selectedTab="tabdetails"
                  id="AccountTabPanel" tabClass='activeTab'
                  inactiveTabClass='inactiveTab'>
      <apex:tab label="Details" name="AccDetails" id="tabdetails">
         <apex:detail relatedList="false" title="true"/>
      </apex:tab>
      <apex:tab label="Contacts" name="Contacts" id="tabContact">
         <apex:relatedList subject="{!account}" list="contacts" />
      </apex:tab>
      <apex:tab label="Opportunities" name="Opportunities"
                id="tabOpp">
         <apex:relatedList subject="{!account}"
                           list="opportunities" />
      </apex:tab>
      <apex:tab label="Open Activities" name="OpenActivities"
                id="tabOpenAct">
         <apex:relatedList subject="{!account}"
                           list="OpenActivities" />
      </apex:tab>
      <apex:tab label="Notes and Attachments"
                name="NotesAndAttachments" id="tabNoteAtt">
         <apex:relatedList subject="{!account}"
                           list="CombinedAttachments" />
      </apex:tab>
   </apex:tabPanel>
</apex:page>
```

Resource:

Overriding an Existing Page with a Visualforce Page
http://developer.salesforce.com/docs/atlas.en-us.pages.meta/pages/pages_quick_start_tabs.htm

Notes:

____

____

____

____

Lesson 8 – Making the Contract Management App Mobile

In this lesson we create a Contract Record Lightning Page and then an App page for the Contract Management App. We then make the Contract Management App available in Salesforce mobile.

We then use the Lightning App Builder to adjust the layouts of the App on Mobile vs. Desktop.

Notes:

Lesson 9 – Adding a Path to the Contract Page Layout

In this lesson, we adjust the order of the **Contract Status** picklist values. We then create a Path for the Contract object, tied to the **Status** field.

Next, we introduce a new picklist field called **Sub-Status**. We then create a Path for this new custom **Sub-Status** field.

Notes:

Lesson 10 –Themes and Branding

In this lesson, I introduce you to a newer feature of Salesforce – Themes and Branding.

Resource:
Themes and Branding – Spring ’18 Release Notes
https://releasenotes.docs.salesforce.com/en-us/spring18/release-notes/rn_lightning_themes.htm

Notes:

Lesson 11 – Removing a Visualforce Page Override

In this lesson, I show you how to remove the Visualforce Page override for viewing account records.

Notes:

Module Review and Assignments

We covered a lot of ground in this module – the Contract Management App Foundations Pack.

We set the Contract Settings to send contract expiration notice emails to Account and Contract Owners. We then explored the implications of setting the Contracts object to Private, along with the cascading effect that has on other objects.

Additionally, we created multiple Sales Processes and corresponding Opportunity Record Types for the Contract Management App, to facilitate the sell of physical products (B2B Widgets) as well as service related product offerings (Call Center Business and Property Management).

Once the foundations of our Contract Management App were set in place, we shifted our focus over to the User Interface and User Experience. We enable a Contracts Chatter Feed and set Field Level Security on certain Contract fields, for example.

We also created a Visualforce page to change the Account Details page, along with setting a Visualforce page override (which we later remove).

We make our app mobile and also added a path to the Contract page layout. A newer feature of Salesforce, Themes and Branding, is also leveraged to adjust the User Interface of our Contract Management app.

Now it’s time to apply the skills that you’ve acquired during this module by completing the assignment(s) below.

Module Assignment

Once a Contract’s **Status** field is set to **Activated**, we don’t want users editing the **Sub-Status** field on a Contract.

You could approach this at least a couple of different ways. Explore your options on how you might tackle this scenario and work through them in your own org.

Stretch Assignment

Now that we have the foundation in place for our Contract Management app, it may be helpful to associate a Contract with a Project. Start thinking through those two objects, Contract (standard object) and Project (custom object) and decide which should be the parent and which should be the child.

What if an enhancement request landed on your desk that dictated that your Contract Management App needed to allow for a single Contract to be able to be associated with one or multiple projects?

How would you establish this relationship between these objects (Contract and Project) to accomplish this?

Whatever your proposed solution is, build it out in your own org.

We will be discussing this further on this week’s group coaching call.

Notes and Action Plan