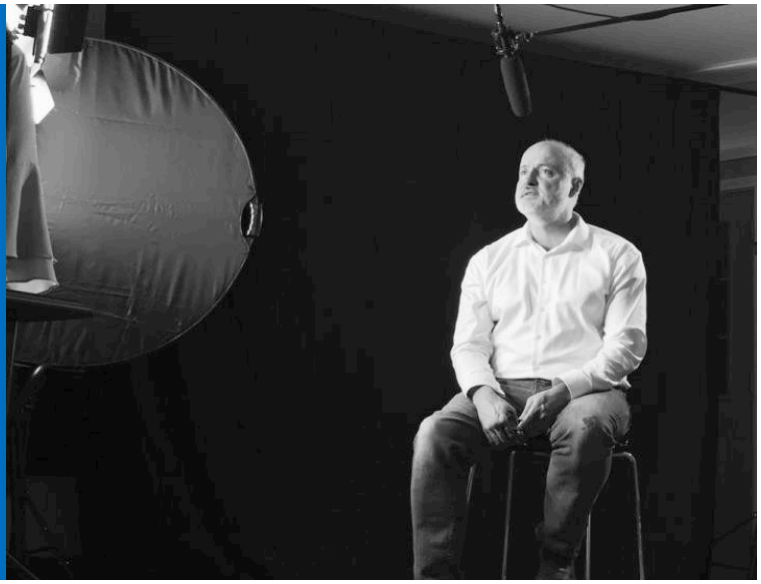
Launch Your Salesforce Career Bootcamp



Module 3: Easy App Pack

Table of Contents

Lesson 1 – Module 3 – Easy App Pack Introduction **3**

Lesson 2 – Creating a Lightning Console Version of the Time Tracker App **4**

Lesson 3 – Creating an Interaction Log **5**

Lesson 4 – Creating Custom Console Components in the Salesforce Service Console **6**

Lesson 5 – Creating a Formula Field, Validation Rule and Rollup Summary **7**

Lesson 6 – Time Tracker Field History Tracking **8**

Lesson 7 – Using Chatter Feed Tracking **9**

Lesson 8 – Creating a Mileage Tracker App **10**

Lesson 9 – Making the Mileage Tracker App Mobile **13**

Lesson 10 – Creating Validation Rules **14**

Lesson 11 – Tracking Trip Field History **15**

Lesson 12 – Creating an Expenses Custom Object **16**

Lesson 13 – Creating an Expenses App **17**

Module Assignments **18**

Mobile Assignment 18

Stretch Assignment 18

Notes and Action Plan **19**

Lesson 1 – Module 3 – Easy App Pack Introduction

In this introductory lecture, I discuss what lies ahead for this module – the Easy App Pack. We are going to be creating several different applications in this module.

The end goal of this module is for you to come away with hands on experience in building apps. This experience will help you expand your knowledge of the Salesforce platform, equip you to speak to in interviews that are specific scenarios you have worked through in building multiple applications, and also to come away with real-world applications that you can use in your own life – whether in an existing job, or helping you land you first or next Salesforce job.

Notes:

Lesson 2 – Creating a Lightning Console Version of the Time Tracker App

In this lecture, we create a Console version of our Time Tracker app.

Notes:

Lesson 3 – Creating an Interaction Log

Be sure you have Sales Console User Permission or Service Cloud User Feature License

Interaction Logs are currently only available for Console Applications in Salesforce Classic and not in Lightning Experience.

Here is a help file https://help.salesforce.com/articleView?id=console_lex_feature_parity.htm&type=5 which details what functionality is available in the Console for Classic and/or Lightning.

Reference: Set Up Interaction Logs for a Console in Salesforce Classic
https://help.salesforce.com/articleView?id=console2_setup_log.htm&type=5

Notes:

Lesson 4 – Creating Custom Console Components in the Salesforce Service Console

In this lesson (which I recorded while enjoying a nice cold that made my voice sound nice and deep, but it really is me) we create a custom console component in the Salesforce Service Console.

Notes:

Lesson 5 – Creating a Formula Field, Validation Rule and Rollup Summary

In this lesson, we further enhance the Time Tracker application by creating a custom formula field on the Time Entry object to calculate the Hours Worked on each Time Entry. We use the following formula for our Hours Worked field:

(Time_Out_c-Time_In_c)*24

We then create a Validation Rule on Time Entries, to prevent users from inadvertently going back in time by entering a Time Out value that is before Time In.

Once the Validation Rule is in place to prevent negative hours in the Hours Worked custom formula field, we next create a Rollup Summary field on the Time Card object to total up the hours worked.

Notes:

Lesson 6 – Time Tracker Field History Tracking

In this lesson we track field history on the Time Entry object. We then look at a scenario where we need to track history on more than 20 fields by going to the Account object. We enable field history tracking on the Account object and then specify the 20 fields we wish to track. In the next lesson we will look at ways to track more than 20 accounts in Field History Tracking by leveraging the Chatter feed.

Notes:

Lesson 7 – Using Chatter Feed Tracking

In this lesson, I leverage the Feed Tracking of Chatter. This enables us to go beyond the 20 field limit of Field History Tracking, to in-turn track an additional 20 fields on the Account object. We also enable Chatter Feed Tracking on the Time Entry object. I then encounter a Lighting Glitch where the Chatter Feed fails to display on the Time Entry Lightning Page until I go into the Lightning App Builder.

Notes:

Lesson 8 – Creating a Mileage Tracker App

For this next app that we are going to build, I am going to leverage a YouTube video I previously recorded. Follow along in your own org to create the Mileage Tracker application.

You can also subscribe to my YouTube Channel here: https://www.youtube.com/c/mikewheelermedia

This is a useful utility app that you can use today to track your mileage for any work-related travel. For example, if you are traveling to a client site, log your mileage using this app, because you can count that mileage as a business expense for your taxes.

To create the Mileage Tracker app, first you want to make a custom object

1. Go to **Setup > Object Manager**.
2. Click the **Create** dropdown then click **Custom Object**.
3. **Label** the object "**Trip**", and make the plural label, "**Trips**" the object name will default to what you set as the label.
4. Make the **Data Type** as an **Auto Number**.
5. Make the display format **T-{00000}**.
6. Set the **Starting Number** to **1**.
7. Select the **Allow Reports**, **Allow Activities**, **Tracking Field History**, **Allow Search**, **Add Notes and Attachments Related List to the Default Page Layout**, and **Launch New Custom Tab Wizard After Saving this Custom Object** checkboxes.
8. Click **Save**.
9. Set the **Tab Style** to a car.
10. Click **Next**.
11. Click **Next**.
12. Check the **Include Tab** check box at the top of the page to include the tab in none of the applications.
13. Click **Save**.

Now we need to create new fields on the object.

1. Go back to the **Trip** object.
2. Click on **Fields and Relationships** in the left column.
3. Click **New**.
4. Select **Date** as the data type.
5. Click **Next**.
6. **Label** the field **Date**.
7. Click Show Formula Editor.
8. Add the formula **Today ()** to auto populate the date field.
9. Click **Check Syntax** to check for any errors in your formula.
10. Click **Next**.
11. Click **Next**.
12. Click **Save and New**.

Now were going to make a two number fields.

1. Select **Number** as the data type.
2. Click **Next**.
3. Label the field "Starting Mileage".
4. Set the **Length** to **7** digits long.
5. Set the **Decimal Places** to **1**.
6. Click **Next**.

7. Click **Next**.
8. Click **Save and New**.
9. Select **Number** as the data type.
10. Click **Next**.
11. Label the field "**Ending Mileage**".
12. Set the **Length** to **7** digits long.
13. Set the **Decimal Places** to **1**.
14. Click **Next**.
15. Click **Next**.
16. Click **Save**.

Now we're going to make a formula field to subtract miles, so we can calculate the mileage for each trip.

1. Click **New** to create a new field.
2. Select **Formula** as the data type.
3. Click **Next**.
4. Label the field "**Distance Traveled**".
5. Make the **Formula Return Type** as **Number**.
6. Click **Next**.
7. Click **Advanced Formula**.
8. Click **Insert Field**.
9. Choose **Trip > Ending Mileage** and hit **Insert Field**.
10. Click **Insert Operator** and choose - **Subtract**.
11. Click **Insert Field**.
12. Choose **Trip > Starting Mileage** and hit **Insert Field**.
13. Click **Check Syntax** to check for errors
14. Select the **Treat Black Fields as Blanks** button.
15. Click **Next**.
16. Click **Next**.
17. Click **Save**.

Now we're going make a new application for your new **Trip** object.

1. Go to **Setup > App Manager**.
2. Click **New Lightning App**.
3. Name the app **Mileage Tracker**.
4. You can add an image, that is optional according to your preference.
5. Click **Next**.
6. Keep **Navigation Style** as **Standard Navigation**.
7. Click **Next**.
8. Click **Next**.
9. Search for your **Trip** object.
10. Press the **Right Arrow** to add it to the **Selected Items** column.
11. Search for **Accounts** object.
12. Press the **Right Arrow** to add it to the **Selected Items** column.
13. Click **Next**.
14. Select **All Profiles**.
15. Press the **Right Arrow** to add it to the **Selected Profiles** column.
16. Click **Save & Finish**.

Now if you go to your App Launcher, you can see your new app that you just created. Next, you will want to relate your new **Trip** object to the **Account** object.

1. Go to **Setup > Object Manager**.
2. Click the **Trip** object.
3. Click Fields & Relationships.
4. Click **New**.
5. Select Lookup Relationship as the Data Type.
6. Choose **Account** from the **Related to** picklist.
7. Label the field Account.
8. Click **Next**.
9. Click **Next**.
10. Click **Save**.

Now if you go to your **Trip Tab** via the Mileage Tracker app:

1. Click **New**.
2. The **Account** lookup should appear, if not, refresh your page then get back to this page.
3. Set the Starting Mileage and Ending Mileage.
4. Click on the **Account** related lookup list to add an account.
5. Click **Save**.

Notes:

Lesson 9 – Making the Mileage Tracker App Mobile

Once you have built the Mileage Tracker app, you can log in to the Salesforce Mobile app and see how it functions on your phone. Try and get hands on experience with your Mileage Tracker app when making trips and log your mileage. See if you can come up with enhancements and improvements to the app to customize it and make it more user-friendly and your own!

In this lesson, I demonstrate how to preview the Salesforce mobile app in your browser. I also share a Chrome App that simulates the Saleforce mobile app.

References:

Knowledge Article 204818 – Emulating a mobile browser with Google Chrome browser
https://help.salesforce.com/articleView?id=000204818&type=1
Salesforce Mobile Simulator App - https://chrome.google.com/webstore/detail/salesforce1-simulator/cknbjckicenodbiaejbmkjhldffonggp?hl=en-US

Notes:

Lesson 10 – Creating Validation Rules

In this lesson, we create a validation rule for our Mileage Tracking application. This rule is used to enforce that only positive distance traveled values are entered via the app.

Notes:

Lesson 11 – Tracking Trip Field History

In this lesson we track field history on our new custom Trip object by setting Field History Tracking. We also enable Feed Track for Chatter on the Trip object to also track field changes there as well.

I highlight a display glitch for Lightning in displaying the Chatter tab and we also adjust the Trip Page Layout to include the Trip History related list.

Notes:

Lesson 12 – Creating an Expenses Custom Object

In this lesson, we create a new custom object to track expenses. We add Lookup fields to Projects, Trips and Opportunities to be able to associate these expenses with.

Notes:

Lesson 13 – Creating an Expenses App

In this lesson, we create a new app, the Expenses app.

We previously created a custom object to track expenses. We added Lookup fields to Projects, Trips and Opportunities to be able to associate these expenses with. We also explore Lookup Filters and making them optional. I demonstrate how to override an optional Lookup Filter as well as encounter a limitation of Lookup Filters in Lightning vs. Classic.

References:
Lookup Filters Not Optional in Lightning? https://success.salesforce.com/answers?id=9063A000000lAjhQAE

Vote on this Idea
https://success.salesforce.com/ideaView?id=0873A000000TtRdQAK

Notes:

Module Assignments

Mobile Assignment
Log in to the Salesforce Mobile app and see how your new applications function on your phone or tablet. Try to get hands-on experience with your Mileage Tracker, Time Tracker, and Expense Tracker applications. See if you can come up with enhancements and improvements to the apps to customize them, make them more user-friendly and your own!

Stretch Assignment
Create an approval process for the Expense Tracker application. The use case is that each Expense needs to be submitted for approval by a Manager. In this instance, you could make another Salesforce user record as your manager in your Salesforce instance.

Reference:
Trailhead – Approval Processes

Notes and Action Plan