

GARAGE MANAGEMENT SYSTEM

College Name: BISHOP APPASAMYCOLLEGE OF
ARTS AND SCIENCE

College Code:BRU3G

TEAM ID: NM2025TMID20453

TEAM MEMBERS:

Team Leader Name: Jeevanandham.v

Email:sanjayjeeva8870@gmail.com

Team Member 1: B.Laba rawal

Email:labarawaljr10@gmail.com

Team Member 2:R.Hafeez

Email:hafehafe23@gmail.com

Team Member 3:Ebinesarmano.B

Email:ebinasarmano2411@gmail.com

1.INTRODUCTION

The Garage Management System is a valuable tool for automotive repair facilities, helping them deliver top-notch service, increase operational efficiency, and build lasting customer relationships. With its user-friendly interface and powerful features, GMS empowers garages to thrive in a competitive market while ensuring a seamless and satisfying experience for both customers and staff.

DEVELOPMENT PHASE

Creating Developer Account: By using this URL

- [Developer Edition with Agentforce and Data Cloud - Salesforce.com](#)

Salesforce

Introduction:

Are you new to Salesforce? Not sure exactly what it is, or how to use it? Don't know where you should start on your learning journey? If you've answered yes to any of these questions, then you're in the right place. This module is for you.

Welcome to Salesforce! Salesforce is game-changing technology, with a host of productivity-boosting features, that will help you sell smarter and faster. As you work toward your badge for this module, we'll take you through these features and answer the question, "What is Salesforce, anyway?"

What Is Salesforce?

Salesforce is your customer success platform, designed to help you sell, service, market, analyze, and connect with your customers.

Salesforce has everything you need to run your business from anywhere. Using standard products and features, you can manage relationships with prospects and customers, collaborate and engage with employees and partners, and store your data securely in the cloud.

So what does that really mean? Well, before Salesforce, your contacts, emails, follow-up tasks, and prospective deals might have been organized something like this:

Object

1. What Is an Object?

2. Salesforce objects are database tables that permit you to store data that is specific to an organization. What are the types of Salesforce objects

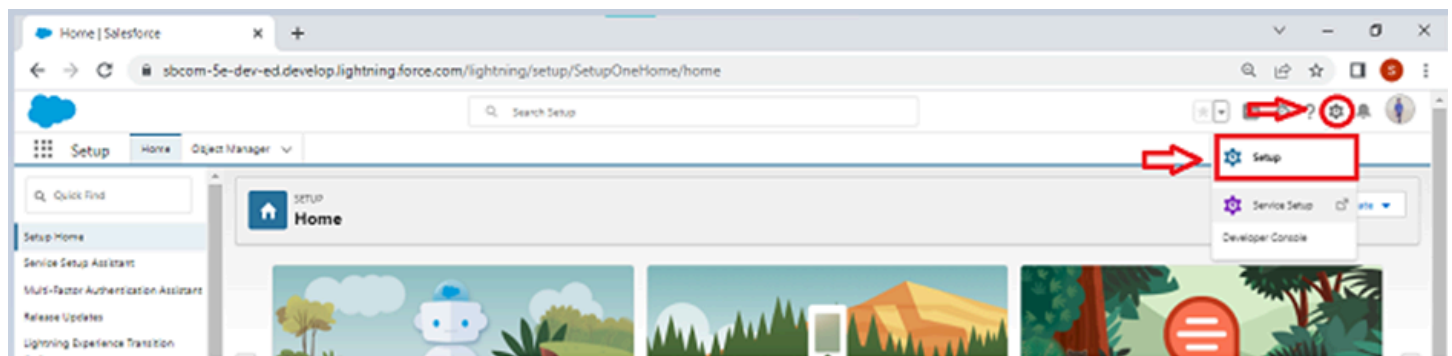
3. Salesforce objects are of two types:

1. **Standard Objects:** Standard objects are the kind of objects that are provided by salesforce.com such as users, contracts, reports, dashboards, etc.
2. **Custom Objects:** Custom objects are those objects that are created by users. They supply information that is unique and essential to their organization. They are the heart of any application and provide a structure for sharing data.

4.

To Navigate to Setup page:

5. Click on gear icon ? click setup.



1.

2.

To create an object:

1. From the setup page ? Click on Object Manager ? Click on Create ? Click on Custom Object.



1.

1. On Custom object defining page:
2. Enter the label name, plural label name, click on Allow reports, Allow search.

New Custom Object | Salesforce

sbcom-5e-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/new

Setup Home Object Manager

SETUP New Custom Object

Custom Object Definition Edit Save Save & New Cancel

Custom Object Information Required Information

The singular and plural labels are used in tabs, page layouts, and records.

Label Example: Account

Plural Label Example: Accounts

Starts with vowel sound ☐

The Object Name is used when referencing the object via the API.

Object Name Example: Account

Description

Context-sensitive help setting ☒ Open the standard Salesforce.com Help & Training window
☐ Open a window using a Visualforce page

Context name

Enter Record Name Label and Format

The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always called "Name" when referenced via the API.

Record Name Example: Account Name

Date Type

Optional Features

☒ Allow Reports ☐ Allow Activities ☐ Track Field History ☐ Allow in Chatter Groups ☐ Enable Licensing

- 1.
- 2.

Optional Features

☒ Allow Reports ☐ Allow Activities ☐ Track Field History ☐ Allow in Chatter Groups ☐ Enable Licensing

Object Classification

When these settings are enabled, this object is classified as an Enterprise Application object. When these settings are disabled, this object is classified as a Light Application object. [Learn more.](#)

☒ Allow Sharing ☒ Allow Bulk API Access ☒ Allow Streaming API Access

Deployment Status [What is this?](#)

☐ In Development ☒ Deployed

Search Status

When this setting is enabled, your users can find records of this object type when they search. [Learn more.](#)

☒ Allow Search

Object Creation Options (Available only when custom object is first created)

☐ Add Notes and Attachments related list to default page layout
☐ Launch New Custom Tab Wizard after saving this custom object

Save Save & New Cancel

- 1.

Tabs

What is Tab : A tab is like a user interface that is used to build records for objects and to view the records in the objects.

Types of Tabs:

1.

Custom Tabs

Custom object tabs are the user interface for custom applications that you build in salesforce.com. They look and behave like standard salesforce.com tabs such as accounts, contacts, and opportunities.

1. **Web Tabs**

Web Tabs are custom tabs that display web content or applications embedded in the salesforce.com window. Web tabs make it easier for your users to quickly access content and applications they frequently use without leaving the salesforce.com application.

1. **Visualforce Tabs**

Visualforce Tabs are custom tabs that display a Visualforce page. Visualforce tabs look and behave like standard salesforce.com tabs such as accounts, contacts, and opportunities.

1. **Lightning Component Tabs**

Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app.

1. **Lightning Page Tabs**

Lightning Page Tabs let you add Lightning Pages to the mobile app navigation menu.

Lightning Page tabs don't work like other custom tabs. Once created, they don't show up on the All Tabs page when you click the Plus icon that appears to the right of your current tabs. Lightning Page tabs also don't show up in the Available Tabs list when you customise the tabs for your apps.

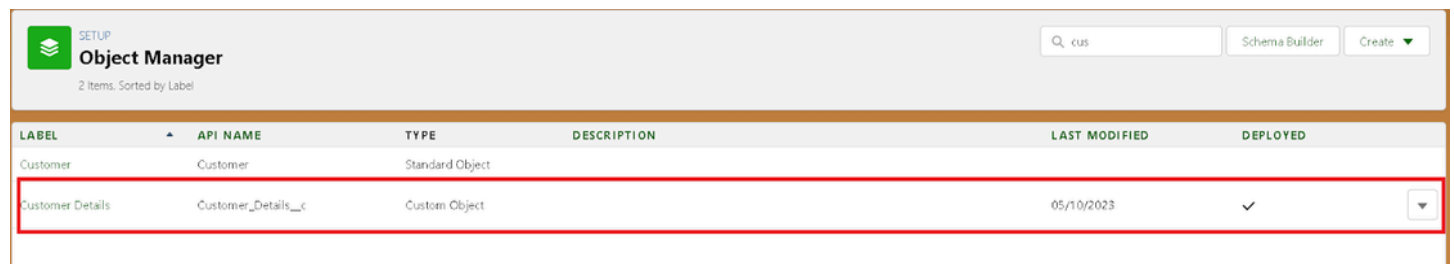
An app is a collection of items that work together to serve a particular function. In Lightning Experience, Lightning apps give your users access to sets of objects, tabs, and other items all in one convenient bundle in the navigation bar.

Lightning apps let you brand your apps with a custom colour and logo. You can even include a utility bar and Lightning page tabs in your Lightning app. Members of your org can work more efficiently by easily switching between apps

Creation of fields for the Customer Details object

1. To create fields in an object:

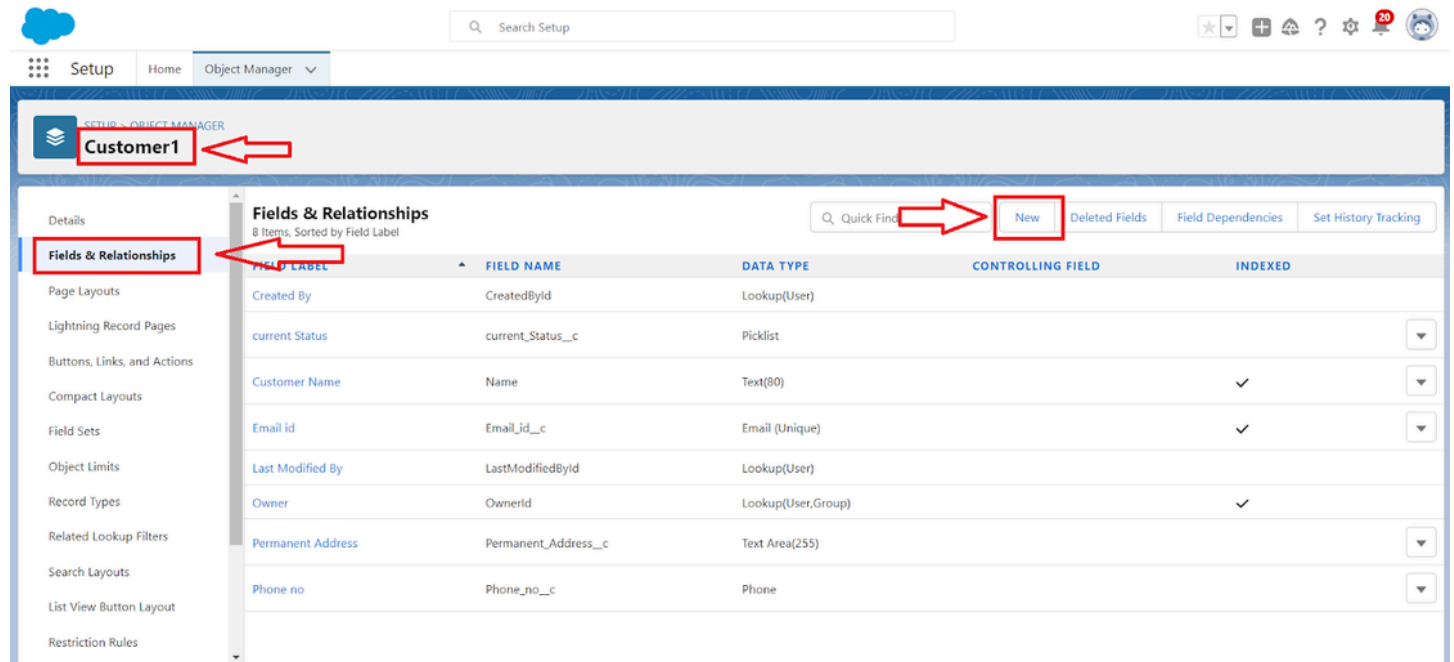
1. Go to setup >> click on Object Manager >> type object name(Customer Details) in search bar >> click on the object.



The screenshot shows the Salesforce Object Manager interface. At the top, there's a search bar with 'cus' entered. Below it, a table lists objects. The 'Customer Details' object is highlighted with a red box. It has an API name of 'Customer_Details__c', is a 'Custom Object', and was last modified on '05/10/2023'.

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Customer	Customer	Standard Object			
Customer Details	Customer_Details__c	Custom Object		05/10/2023	✓

1. Now click on “Fields & Relationships” >> New



The screenshot shows the 'Fields & Relationships' page for the 'Customer1' object. The 'Customer1' label is highlighted with a red box and an arrow. The 'Fields & Relationships' tab is selected in the left sidebar, also highlighted with a red box and an arrow. The 'New' button is highlighted with a red box and an arrow. The table below lists existing fields.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
current Status	current_Status__c	Picklist		
Customer Name	Name	Text(80)		✓
Email id	Email_Id__c	Email (Unique)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Permanent Address	Permanent_Address__c	Text Area(255)		
Phone no	Phone_no__c	Phone		

1. Select Data Type as a “Phone”

Setup > OBJECT MANAGER
Customer1

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

- ☐ Currency
- ☐ Date
- ☐ Date/Time
- ☐ Email
- ☐ Geolocation
- ☐ Number
- ☐ Percent
- ☒ **Phone**
- ☐ Picklist
- ☐ Picklist (Multi-Select)
- ☐ Text
- ☐ Text Area
- ☐ Text Area (Long)
- ☐ Text Area (Rich)
- ☐ Text (Encrypted)
- ☐ Time
- ☐ URL

Allows users to enter a dollar or other currency amount and automatically formats the text as a currency amount. This can be useful if you export data to Excel or another spreadsheet.

Allows users to enter a date or pick a date from a popup calendar.

Allows users to enter a date and time, or pick a date from a popup calendar. When users click a date in the pop-up, that date and the current time are entered into the Date/Time field.

Allows users to enter an email address, which is validated to ensure proper format. If this field is specified for a contact or lead, users can choose the address when clicking Send an Email. Note that custom email addresses cannot be used for mass emails.

Allows users to define locations. Includes latitude and longitude components, and can be used to calculate distance.

Allows users to enter any number. Leading zeros are removed.

Allows users to enter a percentage number, for example, "10" and automatically adds the percent sign to the number.

Allows users to enter any phone number. Automatically formats it as a phone number.

Allows users to select a value from a list you define.

Allows users to select multiple values from a list you define.

Allows users to enter any combination of letters and numbers.

Allows users to enter up to 255 characters on separate lines.

Allows users to enter up to 131,072 characters on separate lines.

Allows users to enter formatted text, add images and links. Up to 131,072 characters on separate lines.

Allows users to enter any combination of letters and numbers and store them in encrypted form.

Allows users to enter a local time. For example, "2:40 PM", "14:40", "14:40:00", and "14:40:50.600" are all valid times for this field.

Allows users to enter any valid website address. When users click on the field, the URL will open in a separate browser window.

1. Click on next.

Setup > OBJECT MANAGER
Customer1

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Scoping Rules

Triggers

Flow Triggers

Validation Rules

Edit Customer1 Custom Field
Phone no

Custom Field Definition Edit

Field Information

Field Label: Phone no

Field Name: Phone_no

Data Type: Phone

Help Text:

Data Owner: User

Field Usage: --None--

Data Sensitivity Level: --None--

Compliance Categorization

Available: PII, HIPAA, GDPR, PCI

Chosen:

General Options

Required: ☒ Always require a value in this field in order to save a record

Default Value: [Show Formula Editor](#)

Use formula syntax. Enclose text and portal value API names in double quotes ("the text"). Include numbers without quotes (20). Show percentages as decimals (0.75), and express date calculations in the standard format: {Today} + 7. To reference a field from a Custom Metadata type record use: {CustomMetadataType__mdt.RecordAPINameField__c}

Change Field Type Save Cancel

5. Fill the Above as following:

- Field Label: Phone number
- Field Name : gets auto generated
- Click on Next >> Next >> Save and new.

Note: Follow the above steps for the remaining field for the same object.

2. To create another fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Customer Details) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data type as a “Email” and Click on Next
4. Fill the Above as following:
 - Field Label : Gmail
 - Field Name : gets auto generated
 - Click on Next >> Next >> Save and new.

Validation rule

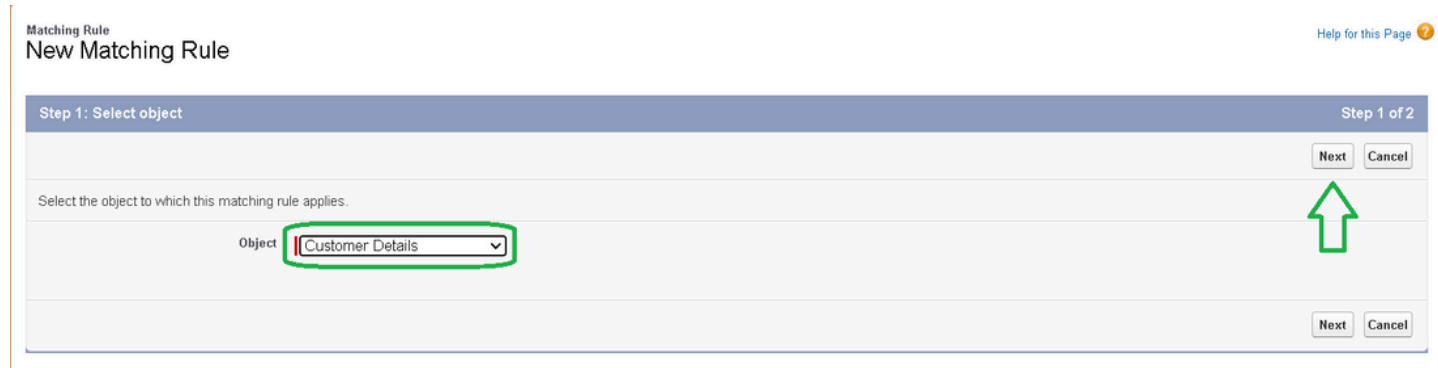
Validation rules are applied when a user tries to save a record and are used to check if the data meets specified criteria. If the criteria are not met, the validation rule triggers an error message and prevents the user from saving the record until the issues are resolved.

To create a matching rule to an Customer details Object

1. Go to quick find box in setup and search for matching Rule.
2. Click on matching rule >> click on New Rule.



1. Select the object as Customer details and click Next.



1. Give the Rule name : Matching customer details
2. Unique name : is auto populated
3. Define the matching criteria as
4. Field Matching Method

1. Gmail Exact

2. Phone Number Exact

1. Click save.
2. After Saving Click on Activate.

SaveCancel

Rule Details

ObjectCustomer Details

Rule Namematching Customer deta

Unique Namematching_Customer_det

Description

Matching Criteria

Tell the rule which fields to compare and how.

Field

Gmail

Phone Number

--None--

--None--

--None--

Matching Method

Exact

Exact

Exact

Exact

Exact

Match Blank Fields

☐

☐

☐

☐

☐

AND

AND

AND

AND

Add Filter Logic...

SaveCancel

Matching Rule

matching Customer details

Help for this Page

Matching Rule Detail

EditDeleteCloneActivate

ObjectCustomer Details

Rule Namematching Customer details

Unique Namematching_Customer_details

Description

Matching Criteria(Customer Details: Gmail EXACT MatchBlank = FALSE) AND (Customer Details: Phone_Number EXACT MatchBlank = FALSE)

StatusInactive

Created Byproject2, 25/09/2023, 10:15 am

Modified Byproject2, 10/10/2023, 3:32 pm

Apex Trigger

Apex can be invoked by using triggers. Apex triggers enable you to perform custom actions before or after changes to Salesforce records, such as insertions, updates, or deletions.

A trigger is Apex code that executes before or after the following types of operations:

- insert
- update
- delete
- merge
- upsert
- undelete

For example, you can have a trigger run before an object's records are inserted into the database, after records have been deleted, or even after a record is restored from the Recycle Bin.

You can define triggers for top-level standard objects that support triggers, such as a Contact or an Account, some standard child objects, such as a CaseComment, and custom objects. To define a trigger, from the object management settings for the object whose triggers you want to access, go to Triggers.

There are primarily two types of Apex Triggers:

Before Trigger: This type of trigger in Salesforce is used either to update or validate the values of a record before they can be saved into the database. So, basically, the before trigger validates the record first and then saves it. Some criteria or code can be set to check data before it gets ready to be inserted into the database.

After Trigger: This type of trigger in Salesforce is used to access the field values set by the system and affect any change in the record. In other words, the after trigger makes changes to the value from the data inserted in some other record.

creating records

To create a record in the follow objects follow these steps

1. Click on the app launcher located at the left side of the screen.
2. Search for “ **Garage Management**” and click on it.
3. Click on the “ **Consumer details** tab”.
4. Click on new and fill the details as shown below figs, and click save.

The screenshot shows a web application window titled "New Customer Detail". In the top right corner, there is a legend: "* = Required Information". Below this, a section titled "Information" contains three input fields, each with a red asterisk and a back arrow icon to its right:

- * Customer Name**: The input field contains the text "Mac".
- * Phone number**: The input field contains the text "5678765567".
- * Gmail**: The input field contains the text "mac@gmail.com".

To the right of these fields, under the heading "Owner", there is a small blue circular icon with a white robot face, followed by the text "Annapurna SmartBridge". At the bottom of the form, there are three buttons: "Cancel" (light blue), "Save & New" (light blue), and "Save" (dark blue).

Now, Create the Appointment Record

1. Click on the “**Appointment** tab”.
2. Enter the customer details as created, while entering Appointment Date enter the date less than the created date.
3. Match the validation while entering the vehicle number plate.
4. Select the services you need.
5. Click on save to see the Service Amount.

Garage Manageme...

Customer Details

Appointments

Service records

Billing details and feedback

Reports

Dashboards

Appointment

app-016

Appointment Name

app-016

Owner

Annapurna SmartBridge

Customer Details

Mac

* Appointment Date

13/11/2024

Maintenance service

☒

Repairs

☒

Replacement Parts

☐

Service Amount

* Vehicle number plate

TS30EU0443

Created By

Annapurna SmartBridge, 18/11/2024, 3:28 pm

Cancel

Save

Last Modified By

Annapurna SmartBridge, 18/11/2024, 3:28 pm

Now, Create a service Record

1. Click on the “**Service record** tab”.
2. Enter the Appointment, and started is selected as default.
3. Click on save.

New Service record

* = Required Information

Information

Service Record Name

Owner

Annapurna SmartBridge

* Appointment

app-016

Quality Check Status

☐

Service Status

Started


Cancel

Save & New


Save

1. Open the record and click on Quality check status as true.
2. Click on save.

Service Record Name
ser-010

Owner
 Annapurna SmartBridge

* Appointment

 app-016 ×

Quality Check Status


☒

↶

Service Status


Started ▼

service date
18/11/2024
This field is calculated upon save

Created By
 Annapurna SmartBridge, 18/11/2024, 4:32 pm

Cancel

Save



Modified By
 Annapurna SmartBridge, 18/11/2024, 4:34 pm


1. Now automatically Service status will be moved to completed.

Related

Details


Service Record Name
ser-010


Owner
 [Annapurna SmartBridge](#) 

Appointment
[app-016](#) 


Quality Check Status


☒



Service Status
Completed 

service date
18/11/2024

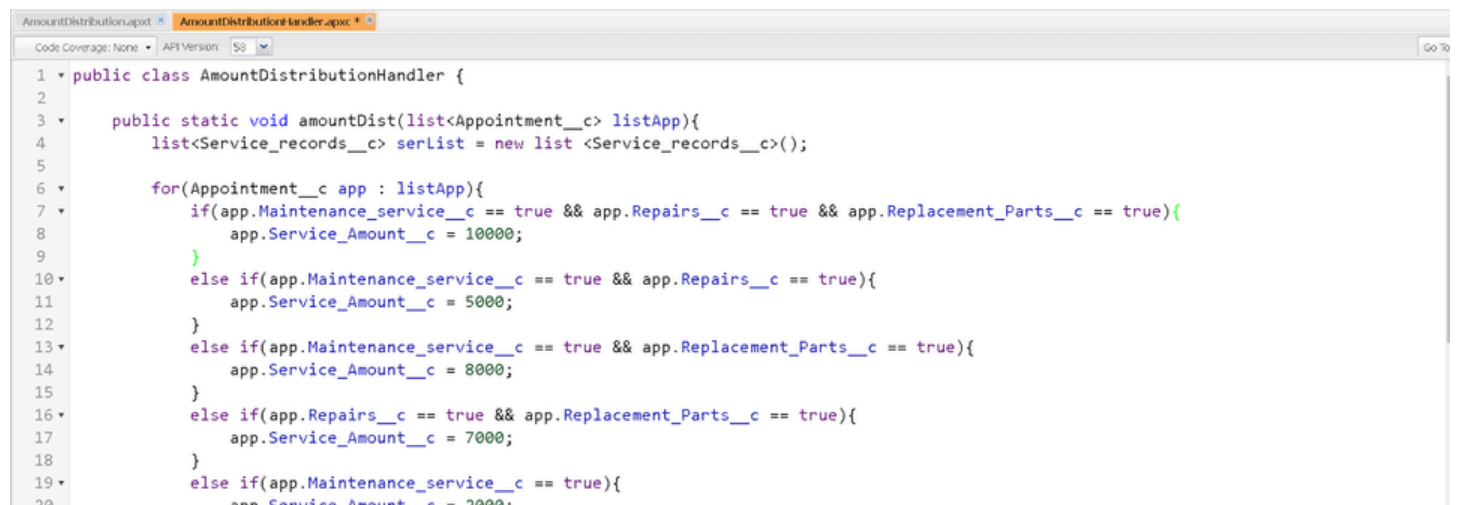
Created By
 [Annapurna SmartBridge](#), 18/11/2024, 4:32 pm

Last Modified By
 [Annapurna SmartBridge](#), 18/11/2024, 4:34 pm

Apex handler

UseCase : This use case works for Amount Distribution for each Service the customer selected for there Vehicle.

1. Login to the respective trailhead account and navigate to the gear icon in the top right corner.
2. Click on the Developer console. Now you will see a new console window.
3. In the toolbar, you can see FILE. Click on it and navigate to new and create New apex class.
4. Name the class as “AmountDistributionHandler”.



```
1 public class AmountDistributionHandler {
2
3     public static void amountDist(list<Appointment__c> listApp){
4         list<Service_records__c> serList = new list <Service_records__c>();
5
6         for(Appointment__c app : listApp){
7             if(app.Maintenance_service__c == true && app.Repairs__c == true && app.Replacement_Parts__c == true){
8                 app.Service_Amount__c = 10000;
9             }
10            else if(app.Maintenance_service__c == true && app.Repairs__c == true){
11                app.Service_Amount__c = 5000;
12            }
13            else if(app.Maintenance_service__c == true && app.Replacement_Parts__c == true){
14                app.Service_Amount__c = 8000;
15            }
16            else if(app.Repairs__c == true && app.Replacement_Parts__c == true){
17                app.Service_Amount__c = 7000;
18            }
19            else if(app.Maintenance_service__c == true){
20                app.Service_Amount__c = 2000;
```

Code:

```
public class AmountDistributionHandler {
```

```
    public static void amountDist(list<Appointment__c> listApp){
```

```
        list<Service_records__c> serList = new list <Service_records__c>();
```

```
for(Appointment__c app : listApp){

if(app.Maintenance_service__c == true && app.Repairs__c == true && app.Replacement_Parts__c == true){

app.Service_Amount__c = 10000;

}

else if(app.Maintenance_service__c == true && app.Repairs__c == true){

app.Service_Amount__c = 5000;

}

else if(app.Maintenance_service__c == true && app.Replacement_Parts__c == true){

app.Service_Amount__c = 8000;

}

else if(app.Repairs__c == true && app.Replacement_Parts__c == true){

app.Service_Amount__c = 7000;

}

else if(app.Maintenance_service__c == true){

app.Service_Amount__c = 2000;

}

else if(app.Repairs__c == true){

app.Service_Amount__c = 3000;

}

else if(app.Replacement_Parts__c == true){

app.Service_Amount__c = 5000;

}

}

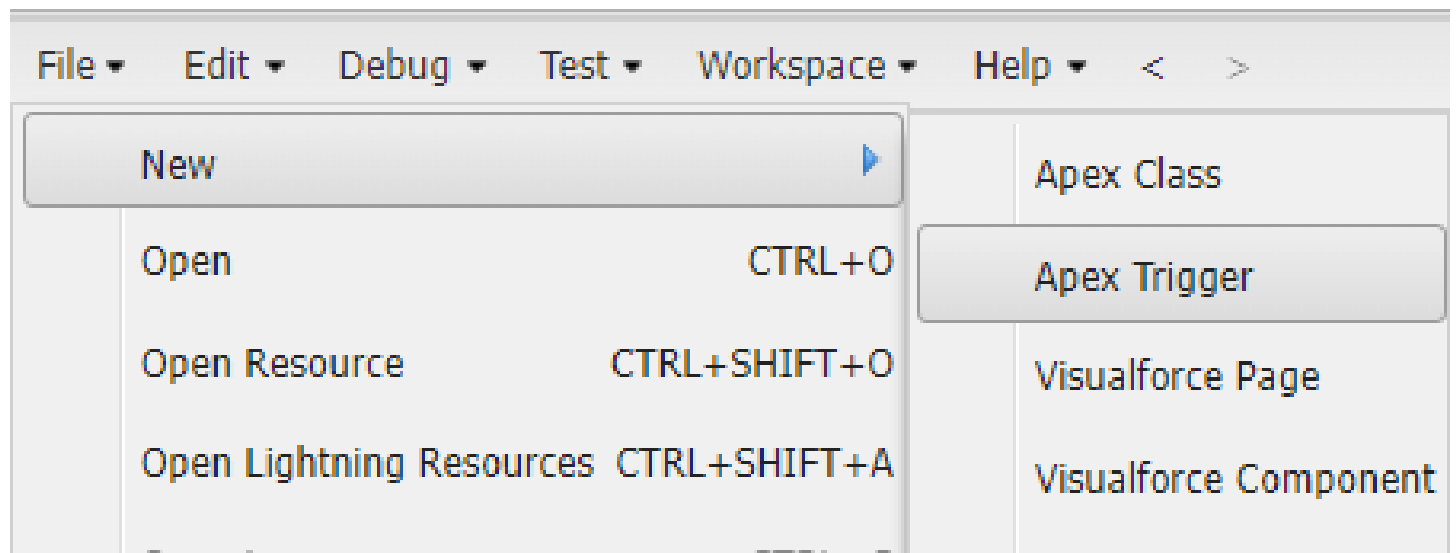
}

}
```


Trigger Handler :

How to create a new trigger :

1. While still in the trailhead account, navigate to the gear icon in the top right corner.
2. Click on developer console and you will be navigated to a new console window.
3. Click on File menu in the tool bar, and click on new? Trigger.
4. Enter the trigger name and the object to be triggered.
5. Name : AmountDistribution
6. sObject : Appointment__c



New Apex Trigger

Name:

sObject: ▼

Syntax For creating trigger :

The syntax for creating trigger is :

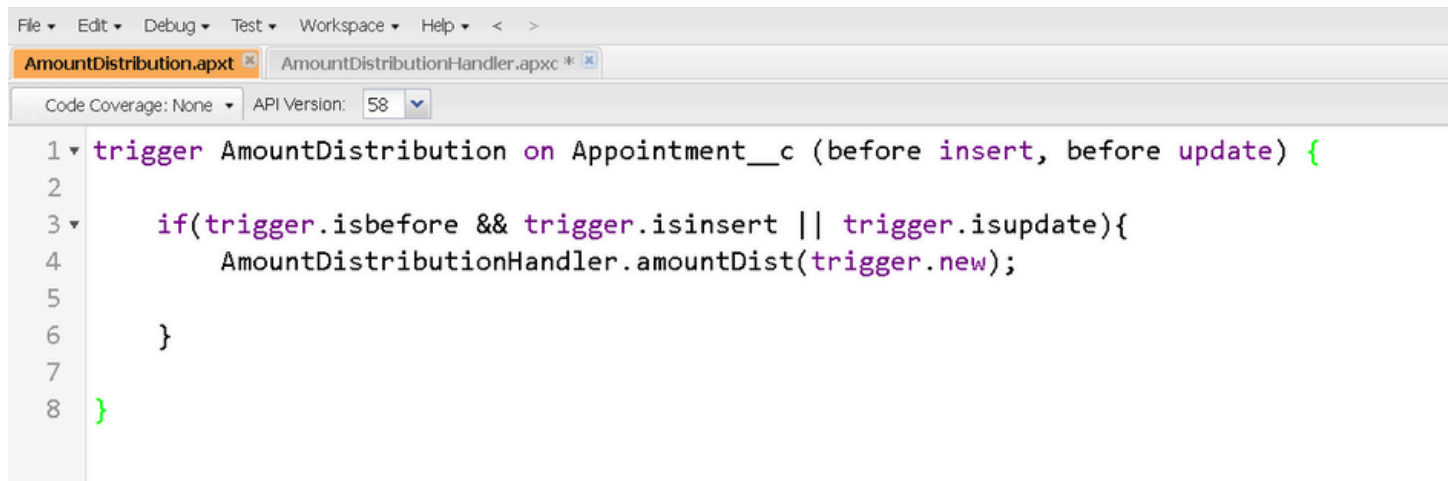
Trigger [trigger name] on [object name](Before/After event)

```
{  
  
}
```

In this project , trigger is called whenever the particular records sum exceed the threshold i.e minimum business requirement value. Then the code in the trigger will get executed.

1.

Handler for the Appointment Object

A screenshot of an IDE window showing a trigger definition. The window has a menu bar with File, Edit, Debug, Test, Workspace, and Help. Below the menu bar are two tabs: 'AmountDistribution.apxt' (active) and 'AmountDistributionHandler.apxc'. Below the tabs is a status bar showing 'Code Coverage: None' and 'API Version: 58'. The main editor area contains the following code:

```
1 trigger AmountDistribution on Appointment__c (before insert, before update) {  
2  
3     if(trigger.isbefore && trigger.isinsert || trigger.isupdate){  
4         AmountDistributionHandler.amountDist(trigger.new);  
5     }  
6 }  
7  
8 }
```

Code:

```
trigger AmountDistribution on Appointment__c (before insert, before update) {
```

```
if(trigger.isbefore && trigger.isinsert || trigger.isupdate){
```

```
AmountDistributionHandler.amountDist(trigger.new);
```

}

}