
TripAdvisor E-Management

-SalesForce

Hardware Required:

System with advance configuration

Software Required:

Salesforce Platform

System Required:

Good Configuration

The TripAdvisor E-Management app with the salesforce works with your all-in-one travel companion, empowering you to plan, book, and make the most of the trips. Discover millions of reviews and insights shared by fellow travelers, helping you make informed decisions for every aspect of your journey. Whether you're seeking the perfect hotel, top-rated restaurants, must-visit attractions, or the best travel deals, TripAdvisor has you covered

Submitted by

au711221104026

au711221104006

au711221104015

au711221104053

1.Create Object

Hotel Object is created to ensure that when a new Food Option is added or updated with the necessary information.

The screenshot shows the Salesforce Setup interface for creating a custom object. The browser address bar displays a URL from jsreccom6-dev-ed. The top navigation bar includes the Setup icon, a search bar, and various utility icons. The left sidebar lists navigation options: 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, and Object Access. The main content area is titled 'Hotel' and contains the 'Custom Object Definition Edit' form. This form includes sections for 'Custom Object Information' (with fields for Label, Plural Label, and Object Name, each with an example) and 'Enter Record Name Label and Format' (with a field for Record Name and an example). A 'Context-Sensitive Help Setting' section allows users to choose between opening the standard Salesforce.com Help & Training window or a window using a Visualforce page. The form also includes a 'Description' text area and a 'Content Name' dropdown menu.

jsreccom6-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01WU000001fm3j/edit?address=%2F01WU000001fm3j%2Fe%3FretURL%3D%252Fsetup%252Fobject%252F01WU000001f...

Setup Home Object Manager

SETUP > OBJECT MANAGER

Hotel

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

Object Access

Edit Custom Object

Hotel

Help for this Page

Custom Object Definition Edit

Save Save & New Cancel

Custom Object Information

The singular and plural labels are used in tabs, page layouts, and reports. Be careful when changing the name or label as it may affect existing integrations and merge templates.

Label Hotel Example: Account

Plural Label Hotels Example: Accounts

Starts with vowel sound ☐

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

Object Name Hotel Example: Account

Description

Context-Sensitive Help Setting

☒ Open the standard Salesforce.com Help & Training window

☐ Open a window using a Visualforce page

Content Name --None--

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 Hotel Name Example: Account Name


2.Create Fields for Hotel Object

Sr. No.	Field Name	Data Type
1	TotalFoodOptions	Number
2	Date	Date

← → ↺

jsreccom6-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/011WU000001fm3j/FieldsAndRelationships/view

☆ ⚙ Error



⚙ Setup

Home

Object Manager ▾

SETUP > OBJECT MANAGER

Hotel

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

Object Access

Fields & Relationships

9 Items, Sorted by Field Label

New

Deleted Fields

Field Dependencies

Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED	
Created By	CreatedById	Lookup(User)			
Date	Date__c	Date			▼
Food Amount	Food_Amount__c	Currency(18, 0)			▼
Hotel	Hotel__c	Lookup(Hotel)		✓	▼
Hotel Name	Name	Text(80)		✓	▼
Last Modified By	LastModifiedById	Lookup(User)			
Name	Name__c	Text(18)			▼
Owner	OwnerId	Lookup(User,Group)		✓	
TotalFoodOptions	TotalFoodOptions__c	Number(18, 0)			▼

3.Create Fields for Food Option

Sr. No.	Field Name	Data Type
1	Name	Text
2	Hotel	Hotel(Lookup)
3	Food Amount	Currency

jsreccom6-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01IWU000001gg77/FieldsAndRelationships/view

Search Setup

Setup Home Object Manager

SETUP > OBJECT MANAGER

Food Option

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Restriction Rules

Scoping Rules

Object Access

Triggers

Flow Triggers

Fields & Relationships

8 Items, Sorted by Field Label


Quick Find

New Deleted Fields Field Dependencies Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Food Amount	Food_Amount__c	Currency(18, 0)		
Food Option Name	Name	Text(80)		✓
Food Options	Food_Options__c	Currency(18, 0)		
Hotel	Hotel__c	Lookup(Hotel)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Name	Name__c	Text(18)		
Owner	OwnerId	Lookup(User,Group)		✓

4.Create Fields in the Flight Object

Sr. No.	Field Name	Data Type
1	Name	Date/Time
2	DepartureDateTime	Hotel(Lookup)



Setup

Home

Object Manager

SETUP > OBJECT MANAGER

Flight

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Restriction Rules

Scoping Rules

Object Access

Triggers

Flow Triggers

Fields & Relationships

8 Items, Sorted by Field Label

New

Deleted Fields

Field Dependencies

Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
ContactEmail	ContactEmail__c	Email		
Created By	CreatedById	Lookup(User)		
Departure Date/Time	Departure_Date_Time__c	Date/Time		
DepartureDateTime	DepartureDateTime__c	Lookup(Hotel)		✓
Flight Name	Name	Text(80)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Name	Name__c	Date/Time		
Owner	OwnerId	Lookup(User,Group)		✓

5.Create Fields in the Customer Object

Sr. No.	Field Name	Data Type
1	Customer Name	Name
2	Discount Amount	Formula (Currency)
3	Discount Percent	Percentage

← → ↺ jsreccom6-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/Customer/FieldsAndRelationships/view

Search Setup

Setup Home Object Manager

SETUP > OBJECT MANAGER

Customer

Details

Fields & Relationships

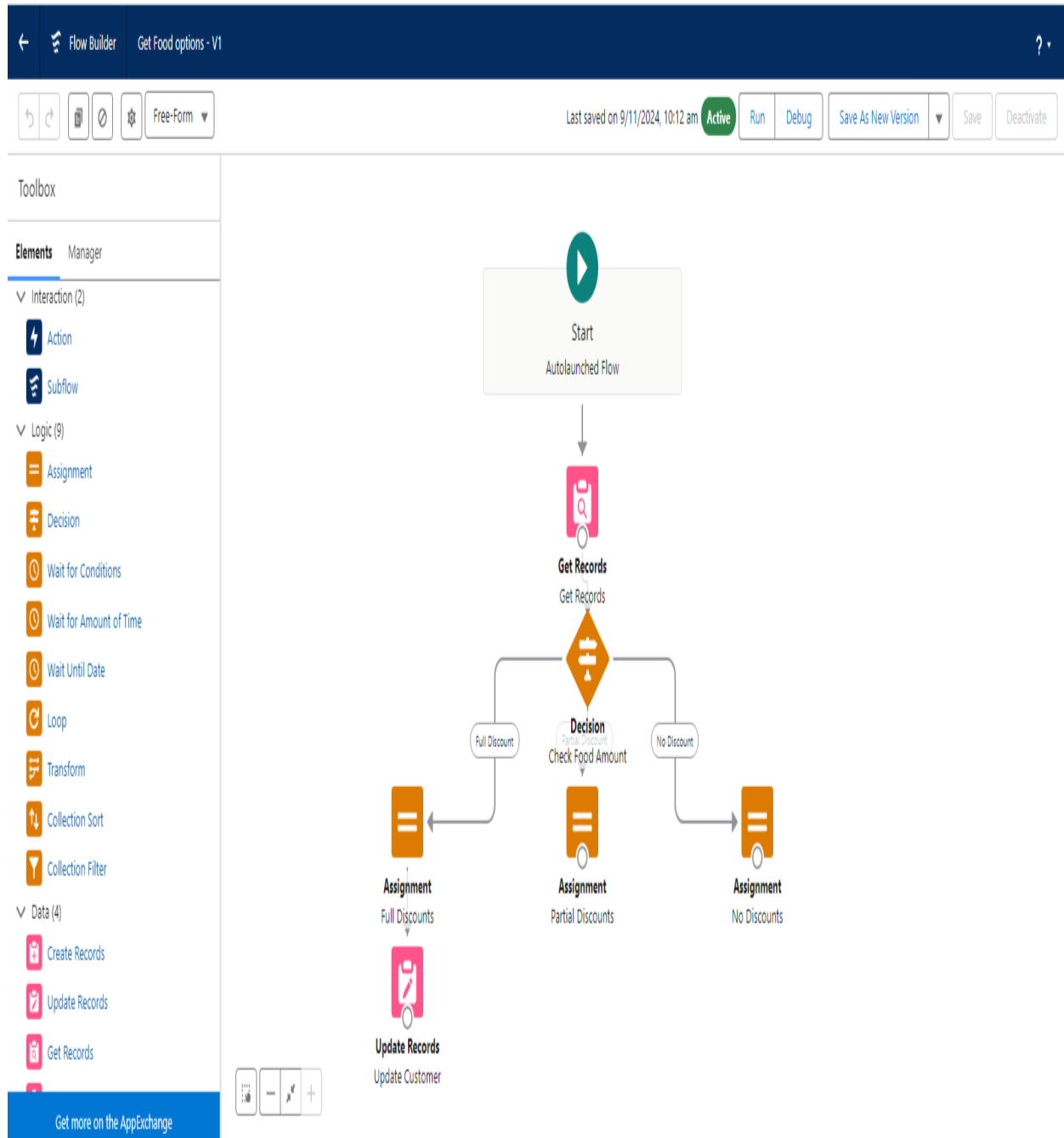
12 Items, Sorted by Field Label

Quick Find New Deleted Fields Field Dependencies Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		✓
Created Date	CreatedDate	Date/Time		✓
Customer Name	Customer_Name__c	Text(18)		
Customer Status Type	CustomerStatusType	Picklist		
Discount Amount	Discount_Amount__c	Formula (Percent)		
Discount Percent	Discount_Percent__c	Percent(18, 0)		
Last Modified By	LastModifiedById	Lookup(User)		
Last Modified Date	LastModifiedDate	Date/Time		
Name	Name	Text(255)		✓
Owner Name	OwnerId	Lookup(User/Group)		✓
Party	PartyId	Lookup(Individual)		✓
Total Life Time Value	TotalLifeTimeValue	Number(9, 0)		

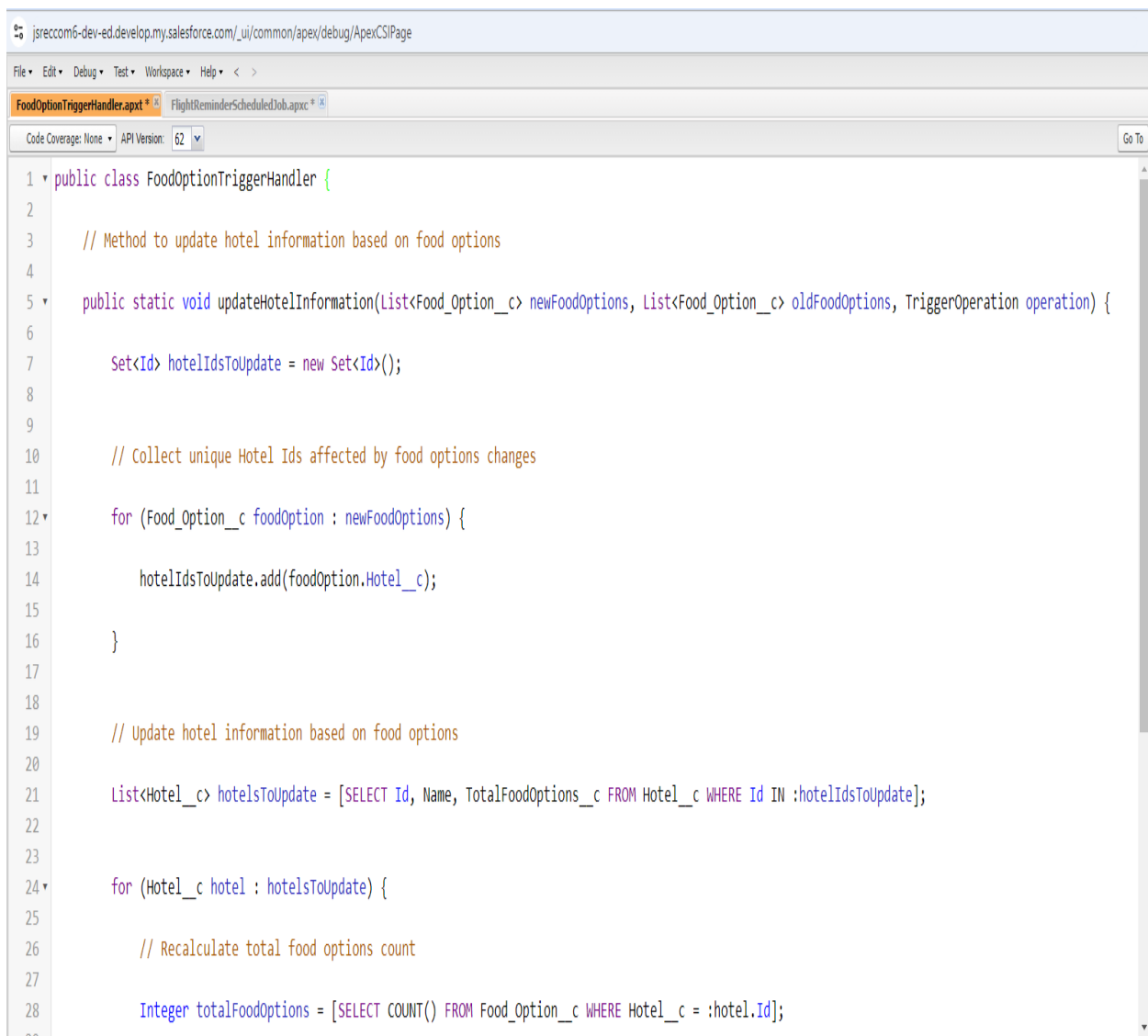
6.Create Flow

Create the Flow for the discount for customer when the Amount is greater than 3000 some Amount of Discounts will be there if the Amount is between 1500 to 3000 so Some Amount of Discount will be there for them.



7.Apex Triggers

Scenario: In the Hotel you have to ensure that when a new Food Option is added or updated, the corresponding Hotel's information is updated accordingly. For example, you might want to maintain a total count of food options for each hotel. To manage the things properly with perspective to the Hotel things should be clearly manageable for making the food options available with respect to hotels.



```
jsrecomm6-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File • Edit • Debug • Test • Workspace • Help • < >
FoodOptionTriggerHandler.apxt * FlightReminderScheduledJob.apxc *
Code Coverage: None • API Version: 62 • Go To

1 public class FoodOptionTriggerHandler {
2
3     // Method to update hotel information based on food options
4
5     public static void updateHotelInformation(List<Food_Option__c> newFoodOptions, List<Food_Option__c> oldFoodOptions, TriggerOperation operation) {
6
7         Set<Id> hotelIdsToUpdate = new Set<Id>();
8
9
10        // Collect unique Hotel Ids affected by food options changes
11
12        for (Food_Option__c foodOption : newFoodOptions) {
13
14            hotelIdsToUpdate.add(foodOption.Hotel__c);
15
16        }
17
18
19        // Update hotel information based on food options
20
21        List<Hotel__c> hotelsToUpdate = [SELECT Id, Name, TotalFoodOptions__c FROM Hotel__c WHERE Id IN :hotelIdsToUpdate];
22
23
24        for (Hotel__c hotel : hotelsToUpdate) {
25
26            // Recalculate total food options count
27
28            Integer totalFoodOptions = [SELECT COUNT() FROM Food_Option__c WHERE Hotel__c = :hotel.Id];
29
30        }
```


Apex Trigger:(code)

1.Apex Trigger With Handler

```
public class FoodOptionTriggerHandler {
    // Method to update hotel information based on food options
    public static void updateHotelInformation(List<Food_Option__c>
newFoodOptions, List<Food_Option__c> oldFoodOptions, TriggerOperation
operation) {
        Set<Id> hotelIdsToUpdate = new Set<Id>();

        // Collect unique Hotel Ids affected by food options changes
        for (Food_Option__c foodOption : newFoodOptions) {
            hotelIdsToUpdate.add(foodOption.Hotel__c);
        }

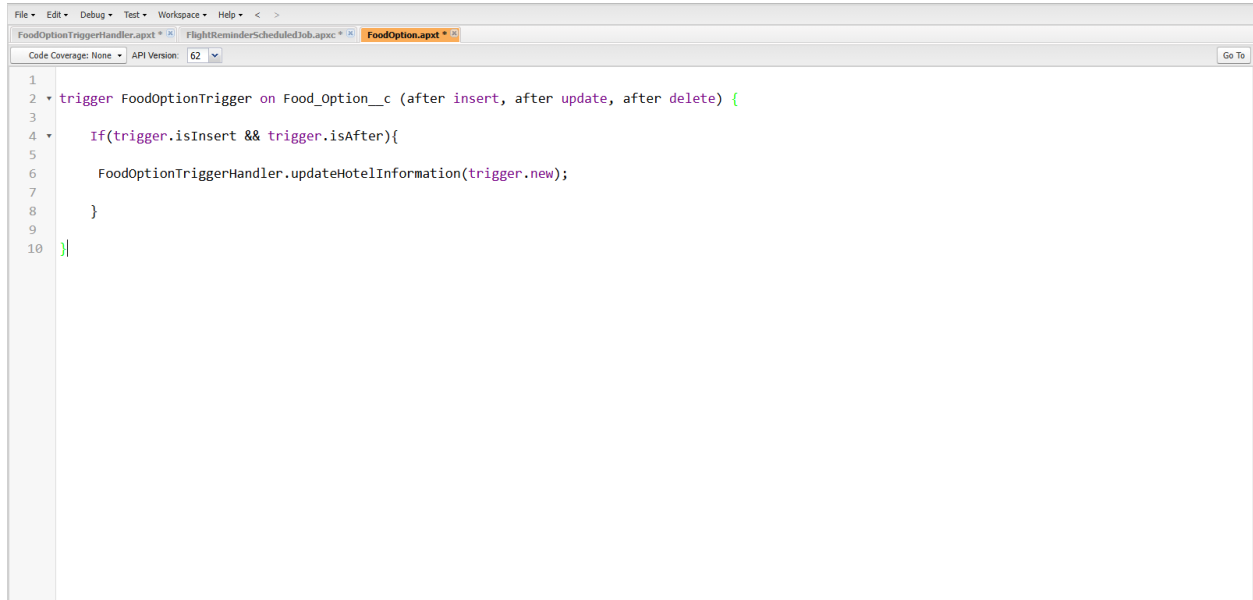
        // Update hotel information based on food options
        List<Hotel__c> hotelsToUpdate = [SELECT Id, Name,
TotalFoodOptions__c FROM Hotel__c WHERE Id IN :hotelIdsToUpdate];

        for (Hotel__c hotel : hotelsToUpdate) {
            // Recalculate total food options count
            Integer totalFoodOptions = [SELECT COUNT() FROM Food_Option__c
WHERE Hotel__c = :hotel.Id];
            hotel.TotalFoodOptions__c = totalFoodOptions;
        }

        // Update hotels with new total food options count
        update hotelsToUpdate;
    }
}
```

2.

```
trigger FoodOptionTrigger on Food_Option__c (after insert, after update,
after delete) {
    If(trigger.isInsert && trigger.isAfter){
        FoodOptionTriggerHandler.updateHotelInformation(trigger.new);
    }
}
```

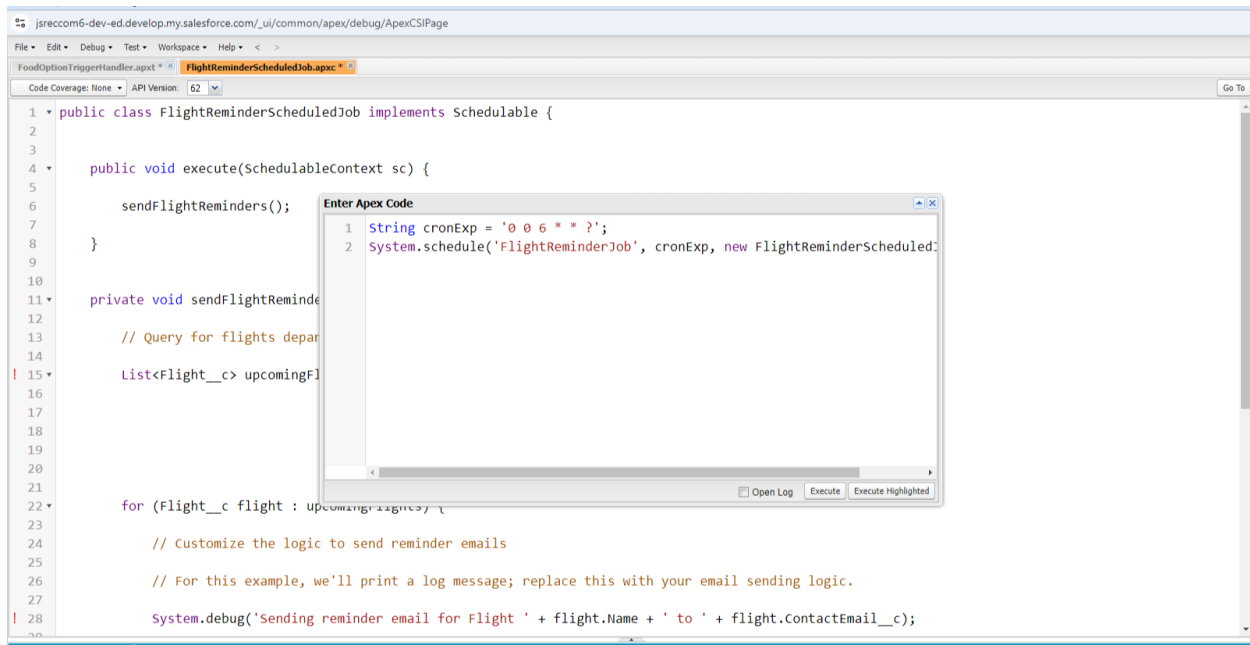


8.Apex Schedule

Create the Reminder mail for the customer who has booked the flight according to that booking set the Apex schedule so mail will be sent prior to 24hrs.
Note: Please create the required field for Scheduled Apex Code

Create the Apex code in an anonymous Window to execute the Apex Code

```
// Schedule the job to run every day at a specific time (e.g., 6 AM)
String cronExp = '0 0 6 * * ?';
System.schedule('FlightReminderJob', cronExp, new FlightReminderScheduledJob());
```



Apex Schedule Class Solution

```

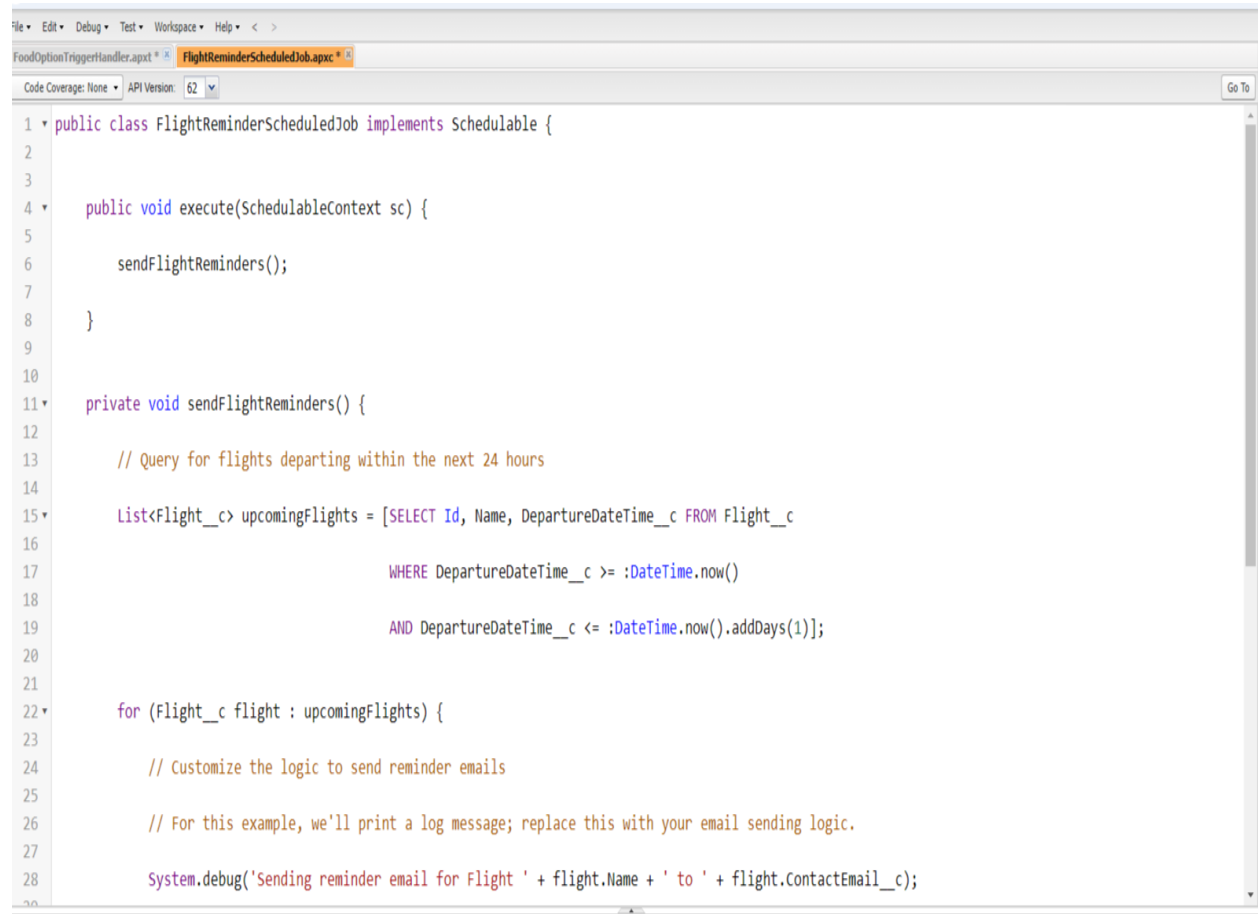
public class FlightReminderScheduledJob implements Schedulable {
    public void execute(SchedulableContext sc) {
        sendFlightReminders();
    }

    private void sendFlightReminders() {
        List<Flight__c> upcomingFlights = [SELECT Id, Name,
        DepartureDateTime__c FROM Flight__c
        WHERE DepartureDateTime__c >= :DateTime.now()
        AND DepartureDateTime__c <=
        :DateTime.now().addDays(1)];
        for (Flight__c flight : upcomingFlights) {
            System.debug('Sending reminder email for Flight ' +
            flight.Name + ' to ' + flight.ContactEmail__c);
            Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
            email.setToAddresses(new List<String>{ flight.ContactEmail__c
            });
            email.setSubject('Flight Reminder: ' + flight.Name);

            email.setPlainTextBody('This is a reminder for your upcoming flight ' +
            flight.Name +
            ' departing on ' +
            flight.DepartureDateTime__c);
        }
    }
}

```

```
        Messaging.sendEmail(new List<Messaging.SingleEmailMessage>{
            email });
    }
}
}
```



The screenshot shows an IDE window with the file `FlightRemindersScheduledJob.apxc` open. The code is written in Apex and implements the `Schedulable` interface. It includes a `execute` method that calls `sendFlightReminders`, which in turn queries for flights departing within the next 24 hours and logs the details for each flight.

```
1 public class FlightRemindersScheduledJob implements Schedulable {
2
3
4     public void execute(SchedulableContext sc) {
5
6         sendFlightReminders();
7
8     }
9
10
11     private void sendFlightReminders() {
12
13         // Query for flights departing within the next 24 hours
14
15         List<Flight__c> upcomingFlights = [SELECT Id, Name, DepartureDateTime__c FROM Flight__c
16
17             WHERE DepartureDateTime__c >= :DateTime.now()
18
19             AND DepartureDateTime__c <= :DateTime.now().addDays(1)];
20
21
22         for (Flight__c flight : upcomingFlights) {
23
24             // Customize the logic to send reminder emails
25
26             // For this example, we'll print a log message; replace this with your email sending logic.
27
28             System.debug('Sending reminder email for Flight ' + flight.Name + ' to ' + flight.ContactEmail__c);
29
30 }
```