

**PROJECT REPORT**

**FOR**

**SALESFORCE DEVELOPER CATALYST**

**ON**

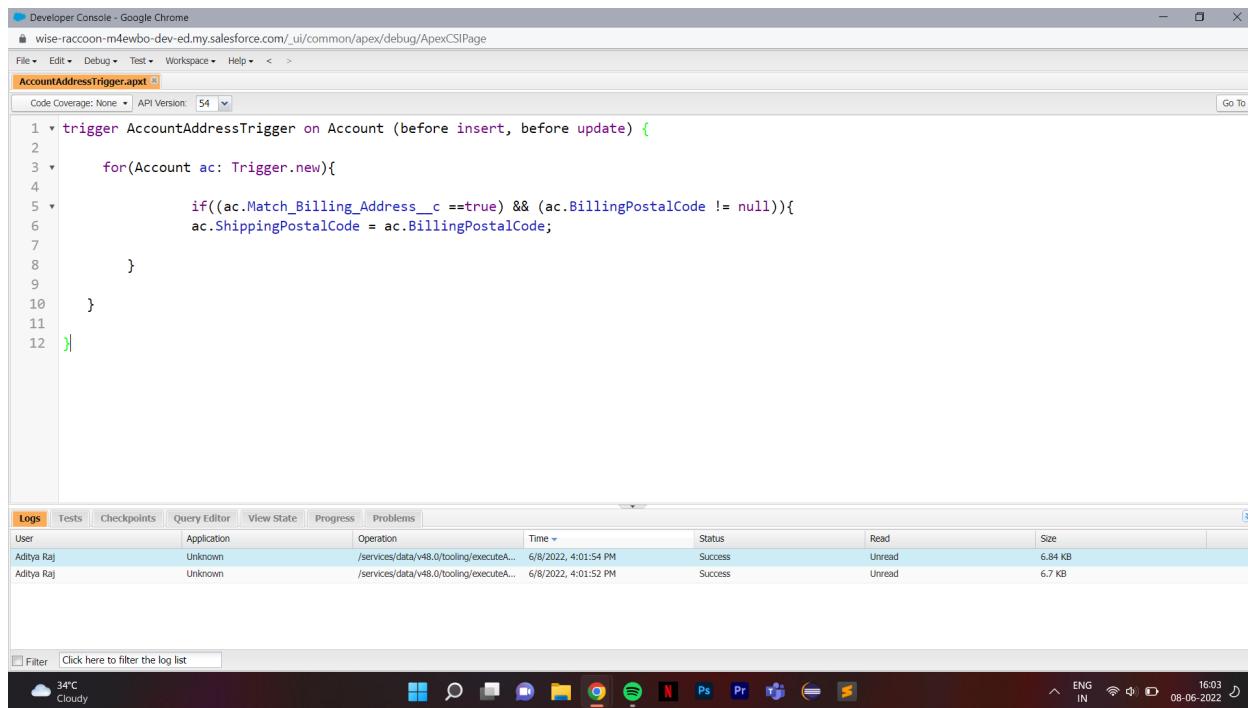
**APEX SPECIALIST SUPERBADGE**

**Submitted by: Aditya Raj**

# 1) APEX TRIGGERS

**Create an apex trigger** - I've created a trigger named AccountAddressTrigger of Account object. It sets an account's Shipping Postal Code to match the Billing Postal Code if the Match Billing Address option is selected. This trigger is fired before inserting an account or updating an account.

```
trigger AccountAddressTrigger on Account (before insert, before update) {  
    for(Account ac: Trigger.new){  
        if((ac.Match_Billing_Address__c ==true) && (ac.BillingPostalCode != null)){  
            ac.ShippingPostalCode = ac.BillingPostalCode;  
        }  
    }  
}
```



**Create a Bulk Apex trigger:** I've created a bulk apex trigger named ClosedOpportunityTrigger of Opportunity object that adds a follow-up task to an opportunity if its stage is Closed Won. This trigger is fired after inserting or updating an opportunity.

```
trigger ClosedOpportunityTrigger on Opportunity (before insert) {  
List<Task> taskList = new List <task>();  
for(Opportunity opp : Trigger.New){  
    if(opp.StageName == 'Closed Won'){  
        taskList.add(new Task(Subject = 'Follow Up Test Task', WhatId =  
opp.Id));  
    }  
}  
if(taskList.size()>0){  
    insert taskList;  
}  
}
```

The screenshot shows the Salesforce Developer Console in Google Chrome. The title bar reads "Developer Console - Google Chrome". The address bar shows the URL: "wise-raccoon-m4ewbo-dev-ed.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage". The tab bar has "ClosedOpportunityTrigger.apex" selected. The main content area displays the Apex trigger code. Below the code, there is a "Logs" section showing two log entries from "Aditya Raj". The logs table has columns: User, Application, Operation, Time, Status, Read, and Size. The first entry is for an "Unknown" application at 6/8/2022, 4:14:55 PM, status Success, unread, and size 291.78 KB. The second entry is for an "Unknown" application at 6/8/2022, 4:14:53 PM, status Success, unread, and size 8.26 KB. At the bottom, there is a "Logs" button, a "Filter" input field, and a system tray with icons for weather (34°C Cloudy), search, file explorer, browser, Netflix, Photoshop, and others. The date and time in the tray are 08-06-2022 16:16.

User	Application	Operation	Time	Status	Read	Size
Aditya Raj	Unknown	/services/data/v48.0/tooling/executeA...	6/8/2022, 4:14:55 PM	Success	Unread	291.78 KB
Aditya Raj	Unknown	/services/data/v48.0/tooling/executeA...	6/8/2022, 4:14:53 PM	Success	Unread	8.26 KB

## 2) APEX TESTING

**Create a Unit Test for a Simple Apex Class:** Created an apex class named VerifyDate and copied the github code. Created another apex class named TestVerifyDate.

```
@isTest public class TestVerifyDate {  
    @isTest static void testOldDate(){  
        Date dateTest = VerifyDate.CheckDates(Date.parse('01/01/2020'),  
Date.parse('01/03/2020'));  
        System.assertEquals(Date.parse('01/03/2020'), dateTest);  
    }  
    @isTest static void testLessThan30Days(){  
        Date dateTest = VerifyDate.CheckDates(Date.parse('01/01/2020'),  
Date.parse('03/03/2020'));  
        System.assertEquals(Date.parse('01/31/2020'), dateTest);  
    }  
}
```

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is [wise-raccoon-m4ewbo-dev-ed.my.salesforce.com/\\_ui/common/apex/debug/ApexCSIPage](https://wise-raccoon-m4ewbo-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage). The page displays the Test Verify Date class with two test methods: testOldDate() and testLessThan30Days(). The code coverage for the class is shown as 59%. The browser's taskbar at the bottom shows various open tabs and system icons.

Class	Percent	Lines
Overall	59%	
AccountAddressTrigger	0%	0/3
ClosedOpportunityTrigger	0%	0/6
VerifyDate	100%	13/13

**Create a Unit Test for a Simple Apex Trigger:** I've created an Apex trigger on the Contact object named RestrictContactByName and an Apex class to run the tests named TestRestrictContactByName.

```
@isTest public class TestRestrictContactByName {
```

```
    @isTest public static void testContact() {
```

```
        Contact c = new Contact();
```

```
        c.LastName = 'INVALIDNAME';
```

```
        Database.SaveResult res = Database.insert(c, false);
```

*System.assertEquals('The Last Name "INVALIDNAME" is not allowed for DML', res.getErrors()[0].getMessage());*

```
}
```

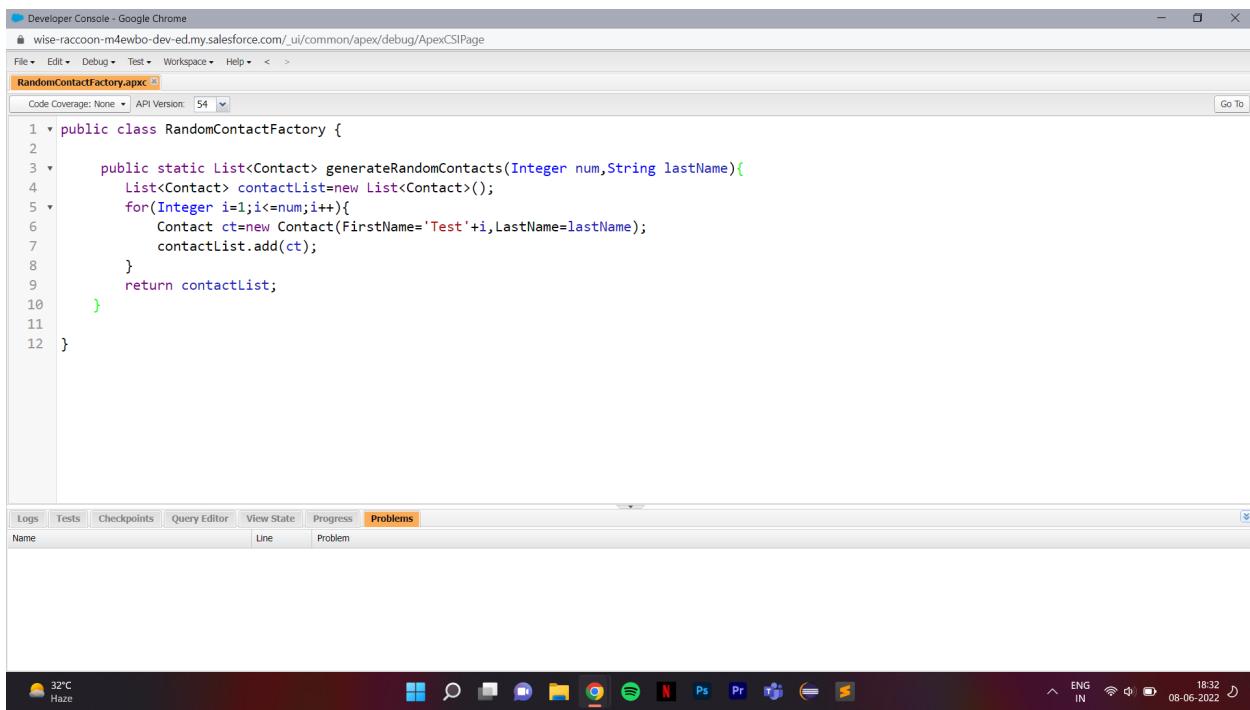
```
}
```

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is [wise-raccoon-m4ewbo-dev-ed.my.salesforce.com/\\_ui/common/apex/debug/ApexCSIPage](https://wise-raccoon-m4ewbo-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage). The page displays the code for the `TestRestrictContactByName` class. The code includes a test method `testContact` that creates a contact with a last name of 'INVALIDNAME', inserts it, and then asserts that an error message is returned. Below the code editor, the test runner interface is shown. It has tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Tests tab is selected, showing two test runs: 'TestRun @ 4:45:39 pm' and 'TestRun @ 6:19:23 pm', both of which have passed. An overall code coverage table is also present, showing 64% coverage across various classes, with `RestrictContactByName` at 100%.

Class	Percent	Lines
Overall	64%	
AccountAddressTrigger	0%	0/3
ClosedOpportunityTrigger	0%	0/6
RestrictContactByName	100%	3/3
VerifyDate	100%	13/13

**Create a Contact Test Factory:** In this, I've implemented apex testing by creating test data. Created an Apex class that returns a list of contacts based on two incoming parameters: the number of contacts to generate and the last name.

```
public class RandomContactFactory {  
    public static List<Contact> generateRandomContacts(Integer num, String  
lastName){  
        List<Contact> contactList=new List<Contact>();  
        for(Integer i=1;i<=num;i++){  
            Contact ct=new Contact(FirstName='Test'+i,LastName=lastName);  
            contactList.add(ct);  
        }  
        return contactList;  
    }  
}
```



The screenshot shows the Salesforce Developer Console interface. The title bar reads "Developer Console - Google Chrome". The URL is "wise-raccoon-n4ewbo-dev-ed.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage". The tab bar has "RandomContactFactory.apxc" selected. The code editor displays the Apex class code provided above. Below the code editor is a "Logs" tab. At the bottom, there's a system tray with icons for weather (32°C Haze), taskbar icons (File Explorer, Task View, File, Mail, Photos, Edge, Spotify, Netflix, Photoshop, Project, OneDrive, Task Manager), and system status indicators (language, battery, date/time).

```
1 public class RandomContactFactory {  
2     public static List<Contact> generateRandomContacts(Integer num, String lastName){  
3         List<Contact> contactList=new List<Contact>();  
4         for(Integer i=1;i<=num;i++){  
5             Contact ct=new Contact(FirstName='Test'+i,LastName=lastName);  
6             contactList.add(ct);  
7         }  
8         return contactList;  
9     }  
10    }  
11 }  
12 }
```

### 3) ASYNCHRONOUS APEX

**Create an Apex class that uses the @future annotation to update Account records:**

I've created an Apex class with a future method that accepts a List of Account IDs and updates a custom field on the Account object with the number of contacts associated to the Account.

AccountProcessor

```
public class AccountProcessor {  
    @future public static void countContacts(List<Id> accountIds){  
        List<Account> accList = [Select Id, Number_Of_Contacts__c, (Select Id from  
        Contacts) from Account where Id in :accountIds];  
        for(Account act : accList){  
            act.Number_of_Contacts__c = act.Contacts.size();  
        }  
        update accList;  
    }  
}
```

```

1 public class AccountProcessor {
2
3     @future
4     public static void countContacts(List<Id> accountIds){
5
6         List<Account> accList = [Select Id, Number_of_Contacts__c, (Select Id from Contacts) from Account where Id in :accountIds];
7
8         for(Account act : accList){
9
10             act.Number_of_Contacts__c = act.Contacts.size();
11         }
12
13         update accList;
14     }
15 }

```

Logs Tests Checkpoints Query Editor View State Progress Problems

Status	Test Run	Enqueued Time	Duration	Failures	Total
✓	TestRun @ 6:54:00 pm			0	1

Overall Code Coverage

Class	Percent	Lines
Overall	63%	
AccountAddressTrigger	66%	2/3
AccountProcessor	100%	5/5
ClosedOpportunityTrigger	0%	0/6
RandomContactFactory	0%	0/6
RestrictContactByName	100%	3/3

## AccountProcessorTest

```
@isTest public class AccountProcessorTest {
    public static testMethod void testAccountProcessor() {
```

```
    Account acc = new Account();
```

```
    acc.Name = 'Test Account';
```

```
    insert acc;
```

```
    Contact c = new Contact();
```

```
    c.FirstName = 'Aditya';
```

```
    c.LastName = 'Raj';
```

```
    c.AccountId = acc.Id;
```

```
    insert c;
```

```
List<Id> accListId = new List<Id>();
```

```
accListId.add(acc.Id);
```

```
Test.startTest();
```

```
AccountProcessor.countContacts(accListId);
```

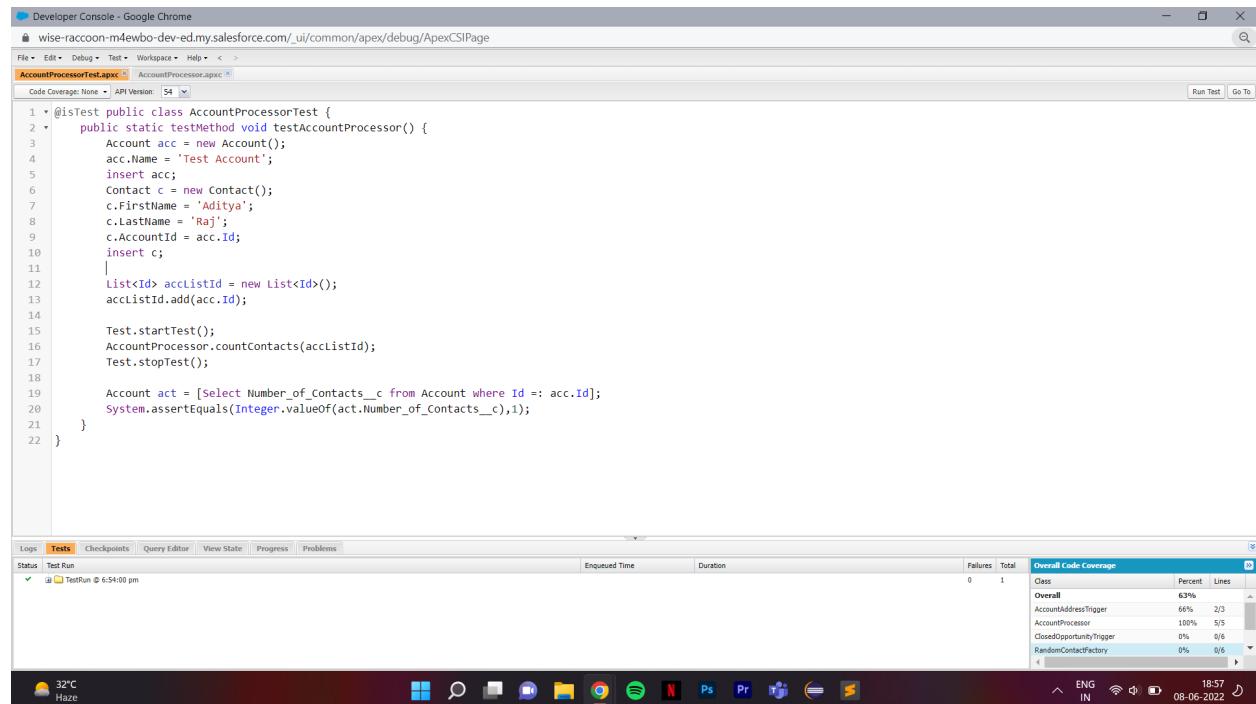
`Test.stopTest();`

`Account act = [Select Number_of_Contacts__c from Account where Id =:  
acc.Id];`

`System.assertEquals(Integer.valueOf(act.Number_of_Contacts__c),1);`

`}`

`}`



```
1 * @isTest public class AccountProcessorTest {
2     public static void testMethod() {
3         Account acc = new Account();
4         acc.Name = 'Test Account';
5         insert acc;
6         Contact c = new Contact();
7         c.FirstName = 'Aditya';
8         c.LastName = 'Raj';
9         c.AccountId = acc.Id;
10        insert c;
11    }
12    List<Id> accListId = new List<Id>();
13    accListId.add(acc.Id);
14
15    Test.startTest();
16    AccountProcessor.countContacts(accListId);
17    Test.stopTest();
18
19    Account act = [Select Number_of_Contacts__c from Account where Id =: acc.Id];
20    System.assertEquals(Integer.valueOf(act.Number_of_Contacts__c),1);
21 }
22 }
```

The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and a search bar. The main area displays the `AccountProcessorTest.apex` code. Below the code editor is a test results summary table:

Status	Tests	Checkpoints	Query Editor	View State	Progress	Problems
Test Run	1	0	0	0	0	0

The bottom status bar shows system information: 32°C Haze, ENG IN, battery level, signal strength, and the date/time 08-06-2022 18:57.

**Create an Apex class that uses Batch Apex to update Lead records:** Create an Apex class that implements the Database.Batchable interface to update all Lead records in the org with a specific LeadSource.

LeadProcessor

```
global class LeadProcessor implements Database.Batchable<sObject> {
    global Integer count = 0;
    global Database.QueryLocator start(Database.BatchableContext bc){
        return Database.getQueryLocator('SELECT ID, LeadSource FROM Lead');
    }
    global void execute(Database.BatchableContext bc, List<Lead> L_list){
        List<lead> L_list_new = new List<lead>();
        for(lead L:L_list){
            L.leadsource = 'Dreamforce';
            L_list_new.add(L);
            count += 1;
        }
        update L_list_new;
    }
    global void finish(Database.BatchableContext bc){
        System.debug('count = '+count);
    }
}
```

The screenshot shows the Salesforce Developer Console in Google Chrome. The top tab is 'Developer Console - Google Chrome' with the URL 'wise-raccoon-m4ewbo-dev-ed.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage'. Below the tabs are 'File', 'Edit', 'Debug', 'Test', 'Workspace', 'Help', and a dropdown for 'API Version: 54'. The main area displays two Apex classes:

```

1 * global class LeadProcessor implements Database.Batchable<sObject> {
2     global Integer count = 0;
3
4     global Database.QueryLocator start(Database.BatchableContext bc){
5         return Database.getQueryLocator('SELECT ID, LeadSource FROM Lead');
6     }
7
8     global void execute(Database.BatchableContext bc, List<Lead> L_list){
9         List<Lead> L_list_new = new List<Lead>();
10
11        for(Lead L:L_list){
12            L.leadsource = 'Dreamforce';
13            L_list_new.add(L);
14            count += 1;
15        }
16        update L_list_new;
17    }
18    global void finish(Database.BatchableContext bc){
19        System.debug('count = '+count);
20    }

```

The 'Logs' tab is selected, showing two test runs:

Status	Test Run	Enqueued Time	Duration	Failures	Total
✓	TestRun @ 6:54:00 pm			0	1
✓	TestRun @ 9:14:37 pm			0	1

The 'Overall Code Coverage' table shows the following data:

Class	Percent	Lines
AccountProcessor	100%	5/5
ClosedOpportunityTrigger	0%	0/6
LeadProcessor	100%	11/11
RandomContactFactory	0%	0/6
RestrictContactByName	100%	3/3
VerifyDate	100%	13/13

The bottom status bar shows system information: 31°C Haze, ENG IN, 21:14, 08-06-2022.

## LeadProcessorTest

```

@isTest public class LeadProcessorTest {
    @isTest
    public static void testIt() {
        List<Lead> L_list = new List<Lead>();
        for(Integer i =0; i<200;i++) {
            Lead L = new Lead();
            L.LastName = 'name' + i;
            L.Company = 'Company';
            L.Status = 'Random Status';
            L_list.add(L);
        }
        insert L_list;
        Test.startTest();
        LeadProcessor lp = new LeadProcessor();
        Id batchId = Database.executeBatch(lp);
        Test.stopTest();
    }
}

```

```
}
```

```
}
```

Developer Console - Google Chrome  
wise-raccoon-m4ewbo-dev-ed.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < ▾ >

LeadProcessor.apxc [ LeadProcessorTest.apxc ]

Code Coverage: None ▾ API Version: 54 ▾ Run Test Go To

```
1 * @isTest public class LeadProcessorTest {  
2  
3  
4     @isTest  
5     public static void testIt() {  
6         List<lead> L_list = new List<lead>();  
7         for(Integer i =0; i<200;i++) {  
8             Lead L = new lead();  
9             L.LastName = 'name' + i;  
10            L.Company = 'Company';  
11            L.Status = 'Random Status';  
12            L_list.add(L);  
13        }  
14        insert L_list;  
15  
16        Test.startTest();  
17        LeadProcessor lp = new LeadProcessor();  
18        Id batchId = Database.executeBatch(lp);  
19        Test.stopTest();  
20    }
```

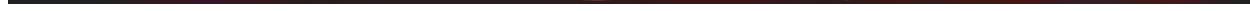
Logs Tests Checkpoints Query Editor View State Progress Problems

Status	Test Run	Enqueued Time	Duration	Failures	Total
✓	TestRun @ 6:54:00 pm			0	1
✓	TestRun @ 9:14:37 pm			0	1

Overall Code Coverage

Class	Percent	Lines
AccountProcessor	100%	5/5
ClosedOpportunityTrigger	0%	0/6
LeadProcessor	100%	11/11
RandomContactFactory	0%	0/6
RestrictContactByName	100%	3/3
VerifyDate	100%	13/13

31°C Haze 21:14 08-06-2022



## Create a Queueable Apex class that inserts Contacts for

**Accounts:** Create a Queueable Apex class that inserts the same Contact for each Account for a specific state.

AddPrimaryContact

```
public class AddPrimaryContact implements Queueable{
    Contact con;
    String state;

    public AddPrimaryContact(Contact con, String state){
        this.con = con;
        this.state = state;
    }
    public void execute(QueueableContext qc){
        List<Account> lstOfAccs = [SELECT Id FROM Account WHERE
BillingState = :state LIMIT 200];

        List<Contact> lstOfConts = new List<Contact>();
        for(Account acc : lstOfAccs){
            Contact conInst = con.clone(false,false,false,false);
            conInst.AccountId = acc.Id;

            lstOfConts.add(conInst);
        }

        INSERT lstOfConts;
    }
}
```

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is [wise-raccoon-m4ewbo-dev-ed.my.salesforce.com/\\_ui/common/apex/debug/ApexCSIPage](https://wise-raccoon-m4ewbo-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage). The page displays the code for the `AddPrimaryContact` class, which implements the `Queueable` interface. The code defines a constructor that takes a `Contact` and a `String state`, and an `execute` method that queries accounts based on the state and clones them into a list of contacts. A code coverage report at the bottom shows 100% coverage across several classes.

```

1  public class AddPrimaryContact implements Queueable{
2      Contact con;
3      String state;
4
5      public AddPrimaryContact(Contact con, String state){
6          this.con = con;
7          this.state = state;
8      }
9      public void execute(QueueableContext qc){
10         List<Account> lstOfAccs = [SELECT Id FROM Account WHERE BillingState = :state LIMIT 200];
11
12         List<Contact> lstOfConts = new List<Contact>();
13         for(Account acc : lstOfAccs){
14             Contact conInst = con.clone(false,false,false,false);
15             conInst.AccountId = acc.Id;
16
17             lstOfConts.add(conInst);
18         }
19     }
20

```

Status	Test Run	Enqueued Time	Duration	Failures	Total
✓	TestRun @ 6:54:00 pm			0	1
✓	TestRun @ 9:14:37 pm			0	1
✓	TestRun @ 9:26:40 pm			0	1

Overall Code Coverage		
Class	Percent	Lines
AccountProcessor	100%	5/5
AddPrimaryContact	100%	11/11
ClosedOpportunityTrigger	0%	0/6
LeadProcessor	100%	11/11
RandomContactFactory	0%	0/6
RestrictContactByName	100%	3/3

## AddPrimaryContactTest

`@isTest`

`public class AddPrimaryContactTest{`

`@testSetup`

`static void setup(){`

`List<Account> lstOfAcc = new List<Account>();`

`for(Integer i = 1; i <= 100; i++){`

`if(i <= 50)`

`lstOfAcc.add(new Account(name='AC'+i, BillingState = 'NY'));`

`else`

`lstOfAcc.add(new Account(name='AC'+i, BillingState = 'CA'));`

`}`

`INSERT lstOfAcc;`

`}`

`static testmethod void testAddPrimaryContact(){`

```

Contact con = new Contact(LastName = 'TestCont');
AddPrimaryContact addPCIns = new AddPrimaryContact(CON , 'CA');

Test.startTest();
System.enqueueJob(addPCIns);
Test.stopTest();

System.assertEquals(50, [select count() from Contact]);
}

}

```

The screenshot shows the Salesforce Developer Console interface in Google Chrome. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and tabs for ApexCSPage, AddPrimaryContact.apcpx, and AddPrimaryContactTest.apcpx. The API Version is set to 54.

The main area displays the Apex code for the `AddPrimaryContactTest` class. The code includes a setup method that creates 100 accounts (50 NY, 50 CA) and an insert statement. It also contains a test method that initializes a contact and adds it to an account using the `AddPrimaryContact` class.

Below the code, the "Tests" tab is selected in the navigation bar. The "Logs" section shows three successful test runs. The "Overall Code Coverage" table provides coverage details for various classes:

Class	Percent	Lines
AccountProcessor	100%	5/5
AddPrimaryContact	100%	11/11
ClosedOpportunityTrigger	0%	0/6
LeadProcessor	100%	11/11
RandomContactFactory	0%	0/6
RestrictContactByName	100%	3/3

The system status bar at the bottom indicates a temperature of 31°C, Haze, ENG IN, 21:26, and the date 08-06-2022.

**Create an Apex class that uses Scheduled Apex to update Lead record:** Create an Apex class that implements the Schedulable interface to update Lead records with a specific LeadSource.

DailyLeadProcessor

```
global class DailyLeadProcessor implements Schedulable {  
  
    global void execute(SchedulableContext ctx) {  
        List<Lead> IList = [Select Id, LeadSource from Lead where LeadSource = null];  
  
        if(!IList.isEmpty()) {  
            for(Lead l: IList) {  
                l.LeadSource = 'Dreamforce';  
            }  
            update IList;  
        }  
    }  
}
```

```

1 global class DailyLeadProcessor implements Schedulable {
2
3     global void execute(SchedulableContext ctx) {
4         List<Lead> llist = [Select Id, LeadSource from Lead where LeadSource = null];
5
6         if(!llist.isEmpty()) {
7             for(Lead l: llist) {
8                 l.LeadSource = 'Dreamforce';
9             }
10            update llist;
11        }
12    }
13 }

```

Overall Code Coverage		
Class	Percent	Lines
AddPrimaryContact	100%	11/11
ClosedOpportunityTrigger	0%	0/6
DailyLeadProcessor	100%	6/6
LeadProcessor	100%	11/11
RandomContactFactory	0%	0/6
RestrictContactByName	0%	0/3

## DailyLeadProcessorTest

*@isTest*

```
public class DailyLeadProcessorTest{
```

```
static testMethod void testMethod1()
{
```

```
    Test.startTest();
```

```
List<Lead> lstLead = new List<Lead>();
```

```
for(Integer i=0 ;i <200;i++)
```

```
{
```

```
    Lead led = new Lead();
```

```
    led.FirstName ='FirstName';
```

```
    led.LastName ='LastName'+i;
```

```
    led.Company ='demo'+i;
```

```
    lstLead.add(led);
```

```
}
```

```

    insert lstLead;

DailyLeadProcessor ab = new DailyLeadProcessor();
String jobId = System.schedule('jobName','0 5 * * * ? ',ab);
Test.stopTest();
}

}

```

Developer Console - Google Chrome  
wise-raccoon-m4ewbo-dev-ed.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < ▾ >

DailyLeadProcessorTest.apxc DailyLeadProcessor.apxc

Code Coverage: None ▾ API Version: 54 ▾ Run Test Go To

```

1  @isTest
2  public class DailyLeadProcessorTest{
3
4      static testMethod void testMethod1()
5      {
6          Test.startTest();
7
8          List<Lead> lstLead = new List<Lead>();
9          for(Integer i=0 ;i <200;i++)
10         {
11             Lead led = new Lead();
12             led.FirstName ='FirstName';
13             led.LastName ='LastName'+i;
14             led.Company ='demo'+i;
15             lstLead.add(led);
16         }
17
18         insert lstLead;
19
20         DailyLeadProcessor ab = new DailyLeadProcessor();

```

Logs Tests Checkpoints Query Editor View State Progress Problems

Status	Test Run	Enqueued Time	Duration	Failures	Total
✓	TestRun @ 9:42:48 pm			0	1

Overall Code Coverage

Class	Percent	Lines
AddPrimaryContact	100%	11/11
ClosedOpportunityTrigger	0%	0/6
DailyLeadProcessor	100%	6/6
LeadProcessor	100%	11/11
RandomContactFactory	0%	0/6
RestrictContactByName	0%	0/3

30°C Cloudy 21:43 08-06-2022

## 4) APEX INTEGRATION SERVICES

### Create an Apex class that calls a REST endpoint and write a test

**class:** Create an Apex class that calls a REST endpoint to return the name of an animal, write unit tests that achieve 100% code coverage for the class using a mock response, and run your Apex tests.

AnimalLocator

```
public class AnimalLocator
{
    public static String getAnimalNameById(Integer id)
    {
        Http http = new Http();
        HttpRequest request = new HttpRequest();
        request.setEndpoint('https://th-apex-http-
callout.herokuapp.com/animals/' + id);
        request.setMethod('GET');
        HttpResponse response = http.send(request);
        String strResp = '';
        system.debug('*****response ' + response.getStatusCode());
        system.debug('*****response ' + response.getBody());
        // If the request is successful, parse the JSON response.
        if (response.getStatusCode() == 200)
        {
            // Deserializes the JSON string into collections of primitive data types.
            Map<String, Object> results = (Map<String, Object>)
            JSON.deserializeUntyped(response.getBody());
            // Cast the values in the 'animals' key as a list
            Map<string,object> animals = (map<string,object>)
            results.get('animal');
```

```

        System.debug('Received the following animals:' + animals );
        strResp = string.valueOf(animals.get('name'));
        System.debug('strResp >>>>' + strResp );
    }
    return strResp ;
}
}

```

Developer Console - Google Chrome  
mindful-impala-eyjwvs-a-dev-ed.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >

Animallocator.apxc \* AnimallocatorTest.apxc \* AnimallocatorMock.apxc

Code Coverage: All Tests 100% ▾ API Version 54 ▾ Go To

```

1 public class Animallocator
2 {
3     public static String getAnimalNameById(Integer id)
4     {
5         Http http = new Http();
6         HttpRequest request = new HttpRequest();
7         request.setEndpoint('https://th-apex-http-callout.herokuapp.com/animals/' + id);
8         request.setMethod('GET');
9         HttpResponse response = http.send(request);
10        String strResp = '';
11        system.debug('*****response ' + response.getStatusCode());
12        system.debug('*****response ' + response.getBody());
13        // If the request is successful, parse the JSON response.
14        if (response.getStatusCode() == 200)
15        {
16            // Deserializes the JSON string into collections of primitive data types.
17            Map<String, Object> results = (Map<String, Object>) JSON.deserializeUntyped(response.getBody());
18            // Cast the values in the 'animals' key as a list

```

Logs Tests Checkpoints Query Editor View State Progress Problems

User	Application	Operation	Time	Status	Read	Size
Aditya Raj	Unknown	/services/data/v48.0/tooling/executeA...	6/9/2022, 12:11:45 AM	Success	Unread	6.49 KB
Aditya Raj	Unknown	ApexTestHandler	6/9/2022, 12:11:29 AM	Success	Unread	9.68 KB
Aditya Raj	Unknown	ApexTestHandler	6/9/2022, 12:11:29 AM	Success	Unread	10.66 KB
Aditya Raj	Unknown	ApexTestHandler	6/9/2022, 12:11:29 AM	Success	Unread	1004 bytes
Aditya Raj	Unknown	/services/data/v54.0/tooling/runTestsS...	6/9/2022, 12:08:56 AM	Success	Unread	10.59 KB

Filter Click here to filter the log list

Cloudy 28°C

Windows taskbar icons: File Explorer, Task View, Taskbar Settings, Taskbar Icons, Taskbar Buttons, Taskbar Buttons.

System tray icons: ENG IN, 00:11, 09-06-2022, battery icon.

## AnimalLocatorTest

@isTest

```
private class AnimalLocatorTest{
```

```
    @isTest static void AnimalLocatorMock1() {
```

```
        Test.SetMock(HttpCallOutMock.class, new AnimalLocatorMock());
```

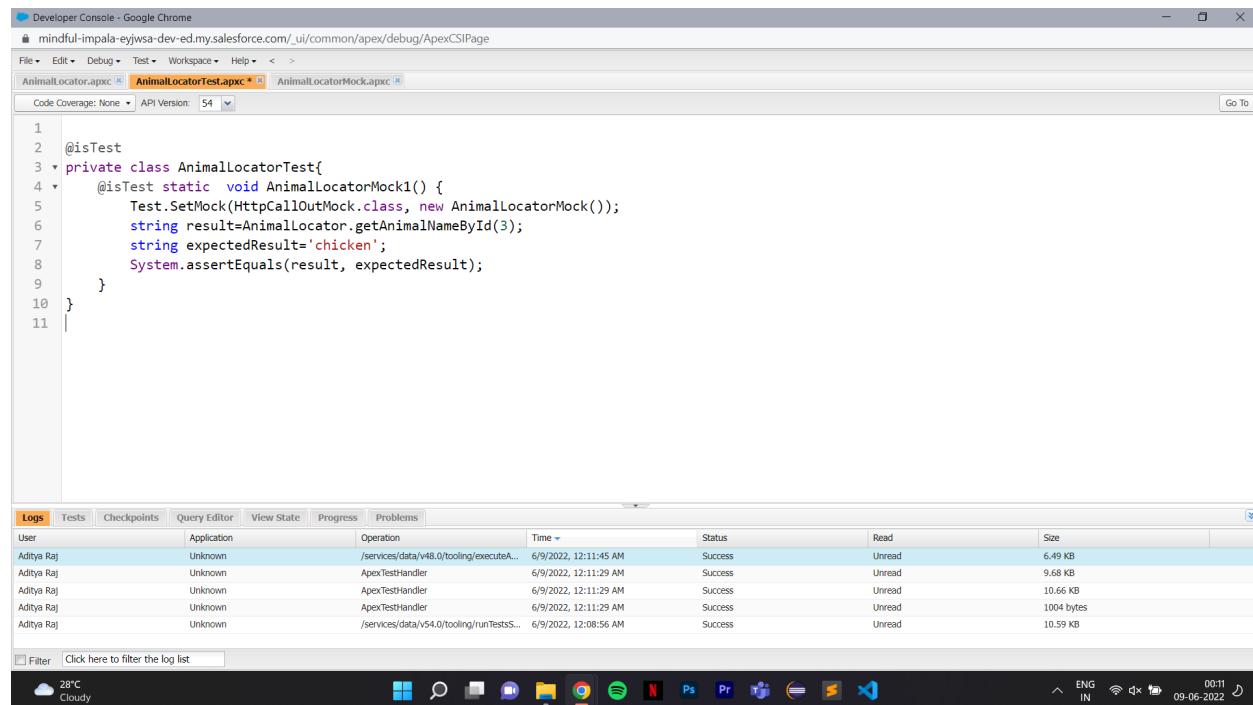
```
        string result=AnimalLocator.getAnimalNameById(3);
```

```
        string expectedResult='chicken';
```

```
        System.assertEquals(result, expectedResult);
```

```
}
```

```
}
```



```
1
2     @isTest
3     private class AnimalLocatorTest{
4         @isTest static void AnimalLocatorMock1() {
5             Test.SetMock(HttpCallOutMock.class, new AnimalLocatorMock());
6             string result=AnimalLocator.getAnimalNameById(3);
7             string expectedResult='chicken';
8             System.assertEquals(result, expectedResult);
9         }
10    }
```

The screenshot shows a developer console window in Google Chrome. The URL is mindful-impala-eyjwsa-dev-ed.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage. The tabs at the top are 'Animallocator.apxc' (selected), 'AnimallocatorTest.apxc' (highlighted in orange), and 'AnimallocatorMock.apxc'. The status bar shows 'Code Coverage: None' and 'API Version: 54'. The bottom part of the window displays a log table with 5 rows of data, a filter input, and a system tray with weather, taskbar icons, and system status.

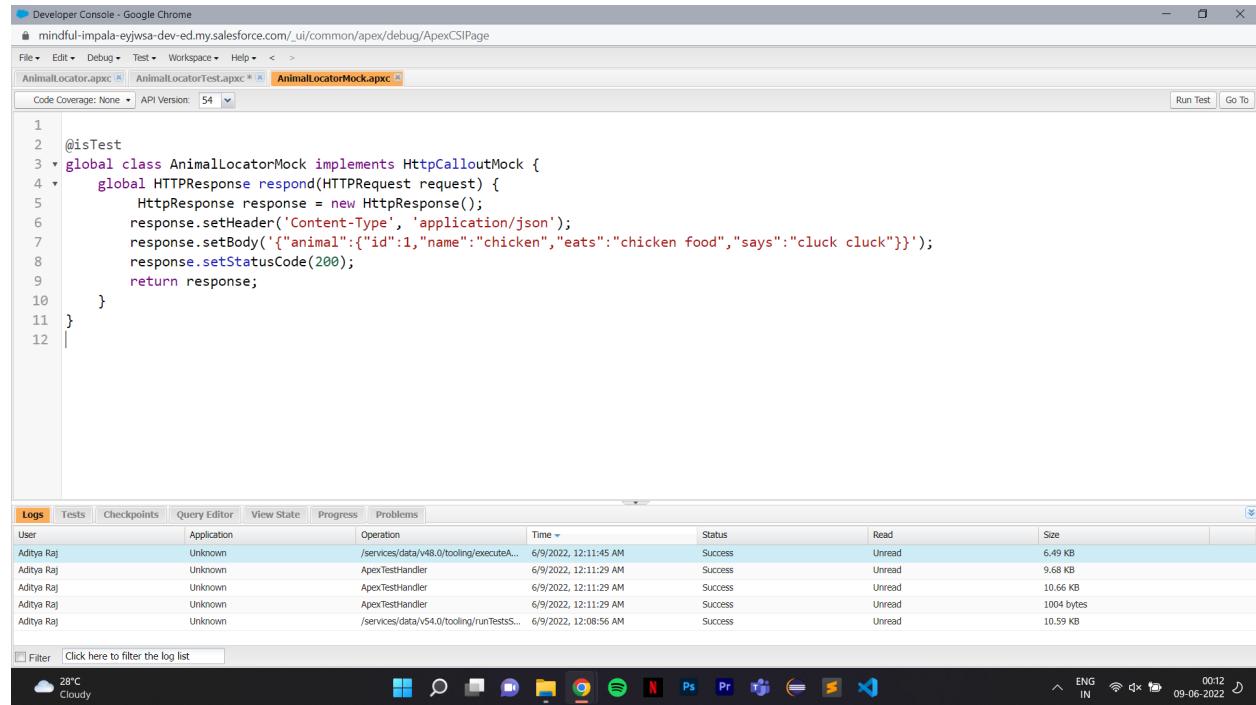
User	Application	Operation	Time	Status	Read	Size
Aditya Raj	Unknown	/services/data/v48.0/tooling/executeA...	6/9/2022, 12:11:45 AM	Success	Unread	6.49 KB
Aditya Raj	Unknown	ApexTestHandler	6/9/2022, 12:11:29 AM	Success	Unread	9.68 KB
Aditya Raj	Unknown	ApexTestHandler	6/9/2022, 12:11:29 AM	Success	Unread	10.66 KB
Aditya Raj	Unknown	ApexTestHandler	6/9/2022, 12:11:29 AM	Success	Unread	1004 bytes

Filter: Click here to filter the log list

Cloudy 28°C ENG IN 00:11 09-06-2022

## AnimalLocatorMock

```
@isTest
global class AnimalLocatorMock implements HttpCalloutMock {
    global HTTPResponse respond(HTTPRequest request) {
        HttpResponse response = new HttpResponse();
        response.setHeader('Content-Type', 'application/json');
        response.setBody('{"animal":{"id":1,"name":"chicken","eats":"chicken
food","says":"cluck cluck"}');
        response.setStatusCode(200);
        return response;
    }
}
```

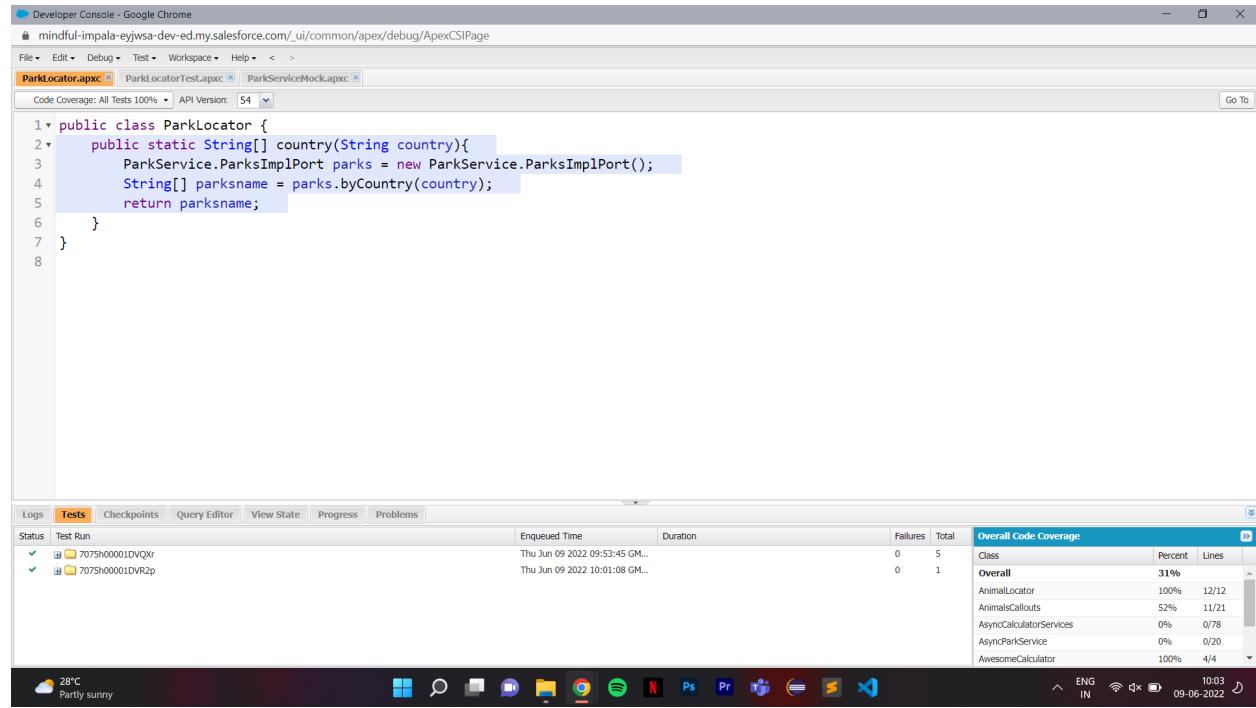


## Generate an Apex class using WSDL2Apex and write a test class:

Generate an Apex class using WSDL2Apex for a SOAP web service, write unit tests that achieve 100% code coverage for the class using a mock response, and run your Apex tests.

### ParkLocator

```
public class ParkLocator {  
    public static String[] country(String country){  
        ParkService.ParksImplPort parks = new ParkService.ParksImplPort();  
        String[] parksname = parks.byCountry(country);  
        return parksname;  
    }  
}
```



### ParkLocatorTest

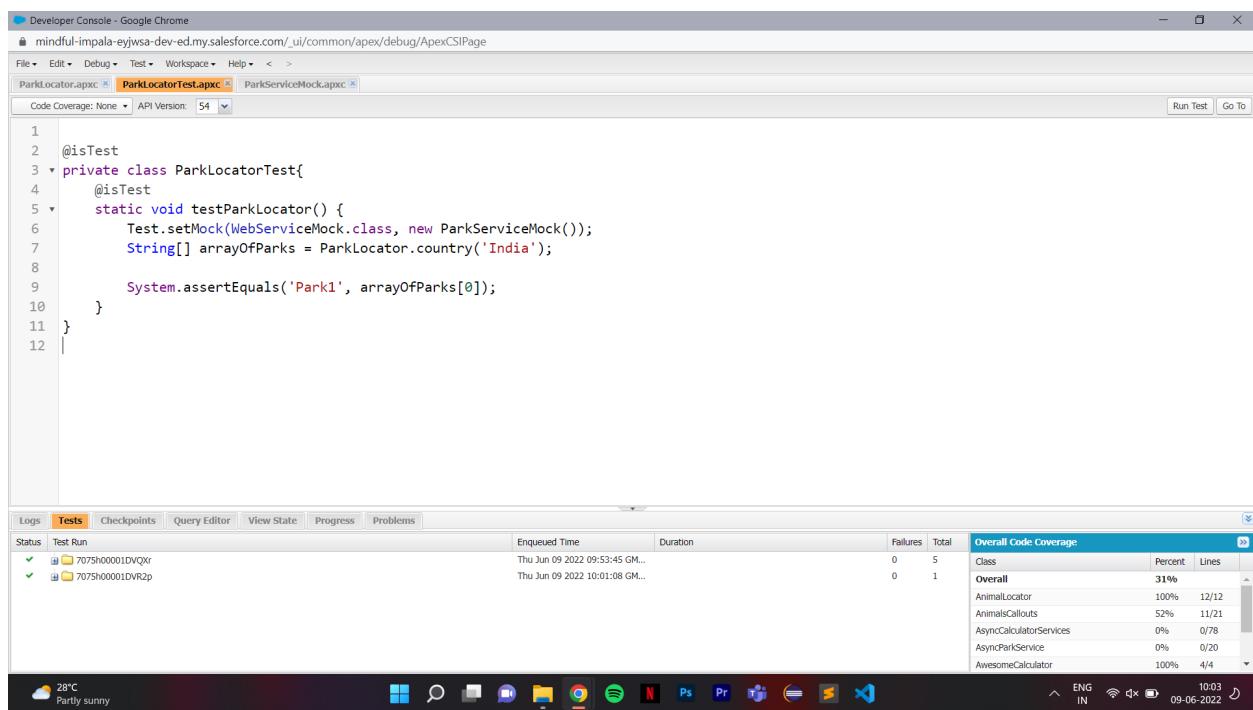
```
@isTest  
private class ParkLocatorTest{
```

```

@isTest
static void testParkLocator() {
    Test.setMock(WebServiceMock.class, new ParkServiceMock());
    String[] arrayOfParks = ParkLocator.country('India');

    System.assertEquals('Park1', arrayOfParks[0]);
}
}

```



## ParkServiceMock

```

@isTest
global class ParkServiceMock implements WebServiceMock {
    global void doInvoke(
        Object stub,
        Object request,
        Map<String, Object> response,

```

```

    String endpoint,
    String soapAction,
    String requestName,
    String responseNS,
    String responseName,
    String responseType) {
    ParkService.byCountryResponse response_x = new
    ParkService.byCountryResponse();
    List<String> lstOfDummyParks = new List<String>
    {'Park1','Park2','Park3'};
    response_x.return_x = lstOfDummyParks;
    response.put('response_x', response_x);
}
}

```

The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and tabs for Parklocator.apxc, ParkLocatorTest.apxc, and ParkServiceMock.apxc. The current tab is ParkServiceMock.apxc.

The code editor displays the following Apex class:

```

@isTest
global class ParkServiceMock implements WebServiceMock {
    global void doInvoke(
        Object stub,
        Object request,
        Map<String, Object> response,
        String endpoint,
        String soapAction,
        String requestName,
        String responseNS,
        String responseName,
        String responseType) {
        ParkService.byCountryResponse response_x = new ParkService.byCountryResponse();
        List<String> lstOfDummyParks = new List<String> {'Park1','Park2','Park3'};
        response_x.return_x = lstOfDummyParks;

        response.put('response_x', response_x);
    }
}

```

The bottom section of the interface shows the Test Results table:

Status	Test Run	Enqueued Time	Duration	Failures	Total	Overall Code Coverage
✓	7075H00001DVQxr	Thu Jun 09 2022 09:53:45 GM...		0	5	Class: 31%
✓	7075H00001DVR2p	Thu Jun 09 2022 10:01:08 GM...		0	1	Overall: 31%

On the right side, there is a detailed "Overall Code Coverage" table:

Class	Percent	Lines
Overall	31%	
AnimalsLocator	100%	12/12
AnimalsCallouts	52%	11/21
AsyncCalculatorServices	0%	0/78
AsyncParkService	0%	0/20
AwesomeCalculator	100%	4/4

The system status bar at the bottom shows weather (28°C, Partly sunny), language (ENG IN), battery level (1003), and date (09-06-2022).

**Create an Apex REST service that returns an account and its contacts:** Create an Apex REST class that is accessible at /Accounts/<Account\_ID>/contacts. The service will return the account's ID and name plus the ID and name of all contacts associated with the account. Write unit tests that achieve 100% code coverage for the class and run your Apex tests.

AccountManager

```
@RestResource(urlMapping='/Accounts/*/contacts')
global with sharing class AccountManager{
    @HttpGet
    global static Account getAccount(){
        RestRequest r = RestContext.request;
        String accId = r.requestURI.substringBetween('Accounts/', '/contacts');
        Account acc = [SELECT Id, Name, (SELECT Id, Name FROM Contacts)
                      FROM Account WHERE Id = :accId];
        return acc;
    }
}
```

The screenshot shows the Salesforce Developer Console in Google Chrome. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and tabs for AccountManager.apxc and AccountManagerTest.apxc. The main area displays the Apex code for AccountManager. Below the code editor is a table showing the status of three test runs. The bottom right corner shows system status icons and the date/time.

Status	Test Run	Enqueued Time	Duration	Failures	Total
✓	7075h00001DVQXr	Thu Jun 09 2022 09:53:45 GM...		0	5
✓	7075h00001DVR2p	Thu Jun 09 2022 10:01:08 GM...		0	1
✓	7075h00001DVREI	Thu Jun 09 2022 10:12:33 GM...		0	7

**Overall Code Coverage**

Class	Percent	Lines
Overall	32%	
AccountManager	100%	6/6
AnimalsLocator	100%	12/12
AnimalsCallouts	52%	11/21
AsyncCalculatorServices	0%	0/78
AsyncParkService	0%	0/20

## AccountManagerTest

```
@IsTest
private class AccountManagerTest{
    @isTest static void testAccountManager(){
        Id recordId = getTestId();

        RestRequest request = new RestRequest();
        request.requestUri =
            'https://ap5.salesforce.com/services/apexrest/Accounts/'+recordId
            +'contacts';
        request.httpMethod = 'GET';
        RestContext.request = request;

        Account acc = AccountManager.getAccount();

        System.assert(acc != null);
```

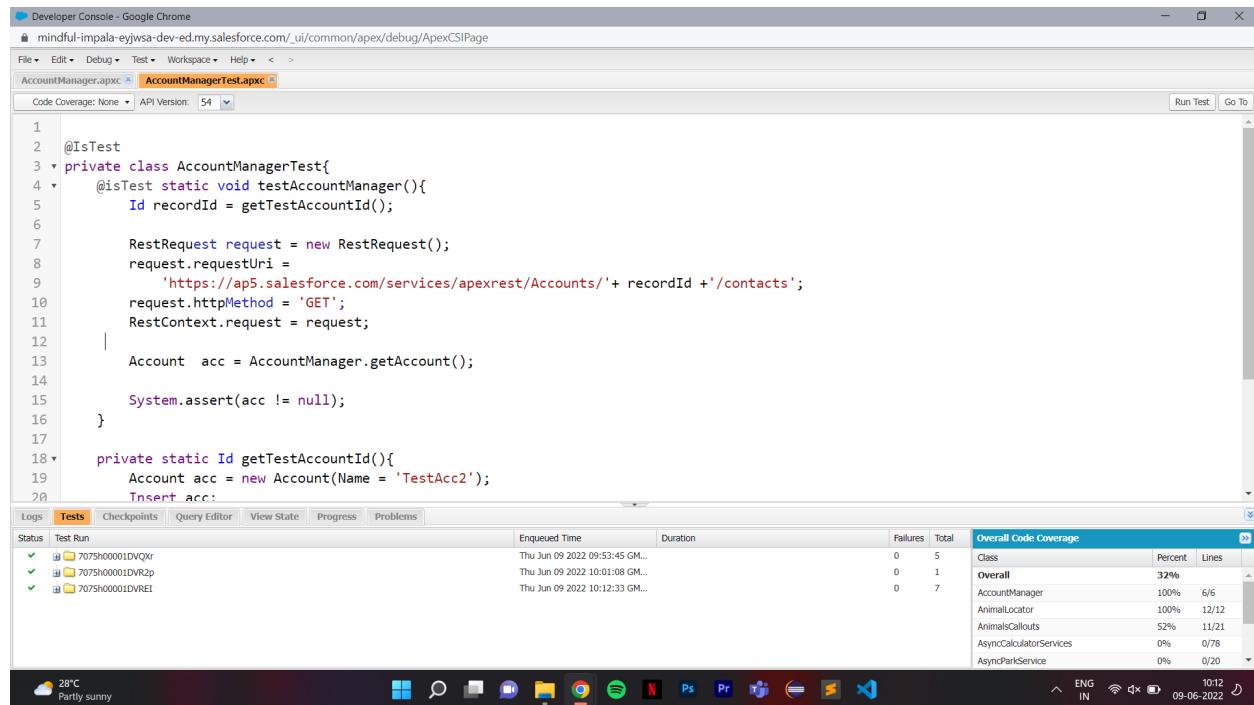
```
}
```

```
private static Id getTestId(){  
    Account acc = new Account(Name = 'TestAcc2');  
    Insert acc;
```

```
Contact c = new Contact(LastName = 'TestCont2', AccountId = acc.Id);  
Insert c;
```

```
return acc.Id;
```

```
}  
}
```



The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is mindful-impala-eyjwsa-dev-ed.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage. The tab bar shows AccountManager.apxc and AccountManagerTest.apxc. The code editor contains the provided Apex code. The bottom section shows the Test Results table and the Overall Code Coverage table.

Status	Test Run	Enqueued Time	Duration	Failures	Total
✓	7075h00001DVQXr	Thu Jun 09 2022 09:53:45 GM...		0	5
✓	7075h00001DV2p2	Thu Jun 09 2022 10:01:08 GM...		0	1
✓	7075h00001DVREI	Thu Jun 09 2022 10:12:33 GM...		0	7

Overall Code Coverage		
Class	Percent	Lines
Overall	32%	
AccountManager	100%	6/6
AnimalLocator	100%	12/12
AnimalsCallouts	52%	11/21
AsyncCalculatorServices	0%	0/78
AsyncParkService	0%	0/20

## **APEX SPECIALIST SUPERBADGE**

### **Automate record creation:**

MaintenanceRequestHelper class:

```
public with sharing class MaintenanceRequestHelper {  
    public static void updateWorkOrders(List<Case> updWorkOrders,  
    Map<Id,Case> nonUpdCaseMap) {  
        Set<Id> validIds = new Set<Id>();  
        For (Case c : updWorkOrders){  
            if (nonUpdCaseMap.get(c.Id).Status != 'Closed' && c.Status == 'Closed'){  
                if (c.Type == 'Repair' || c.Type == 'Routine Maintenance'){  
                    validIds.add(c.Id);  
                }  
            }  
        }  
        if (!validIds.isEmpty()){  
            List<Case> nCases = new List<Case>();  
            Map<Id,Case> closedCasesM = new Map<Id,Case>([SELECT Id,  
Vehicle__c, ProductId, Product.Maintenance_Cycle__c,(SELECT  
Id,Equipment__c,Quantity__c FROM Equipment_Maintenance_Items__r)  
                FROM Case WHERE Id IN :validIds]);  
            Map<Id,Decimal> maintenanceCycles = new Map<ID,Decimal>();  
            AggregateResult[] results = [SELECT Maintenance_Request__c,  
MIN(Equipment__r.Maintenance_Cycle__c)cycle FROM  
Equipment_Maintenance_Item__c WHERE Maintenance_Request__c IN  
:ValidIds GROUP BY Maintenance_Request__c];  
  
            for (AggregateResult ar : results){  
                maintenanceCycles.put((Id) ar.get('Maintenance_Request__c'), (Decimal)  
ar.get('cycle'));  
            }  
        }  
    }  
}
```

```

for(Case cc : closedCasesM.values()){
    Case nc = new Case (
        ParentId = cc.Id,
        Status = 'New',
        Subject = 'Routine Maintenance',
        Type = 'Routine Maintenance',
        Vehicle__c = cc.Vehicle__c,
        ProductId =cc.ProductId,
        Origin = 'Web',
        Date_Reported__c = Date.Today()
    );

    If (maintenanceCycles.containsKey(cc.Id)){
        nc.Date_Due__c = Date.today().addDays((Integer)
maintenanceCycles.get(cc.Id));
    } else {
        nc.Date_Due__c = Date.today().addDays((Integer)
cc.Product.maintenance_Cycle__c);
    }

    nCases.add(nc);
}

insert nCases;

List<Equipment_Maintenance_Item__c> clonedWPs = new
List<Equipment_Maintenance_Item__c>();
for (Case nc : nCases){
    for (Equipment_Maintenance_Item__c wp :
closedCasesM.get(nc.ParentId).Equipment_Maintenance_Items__r){

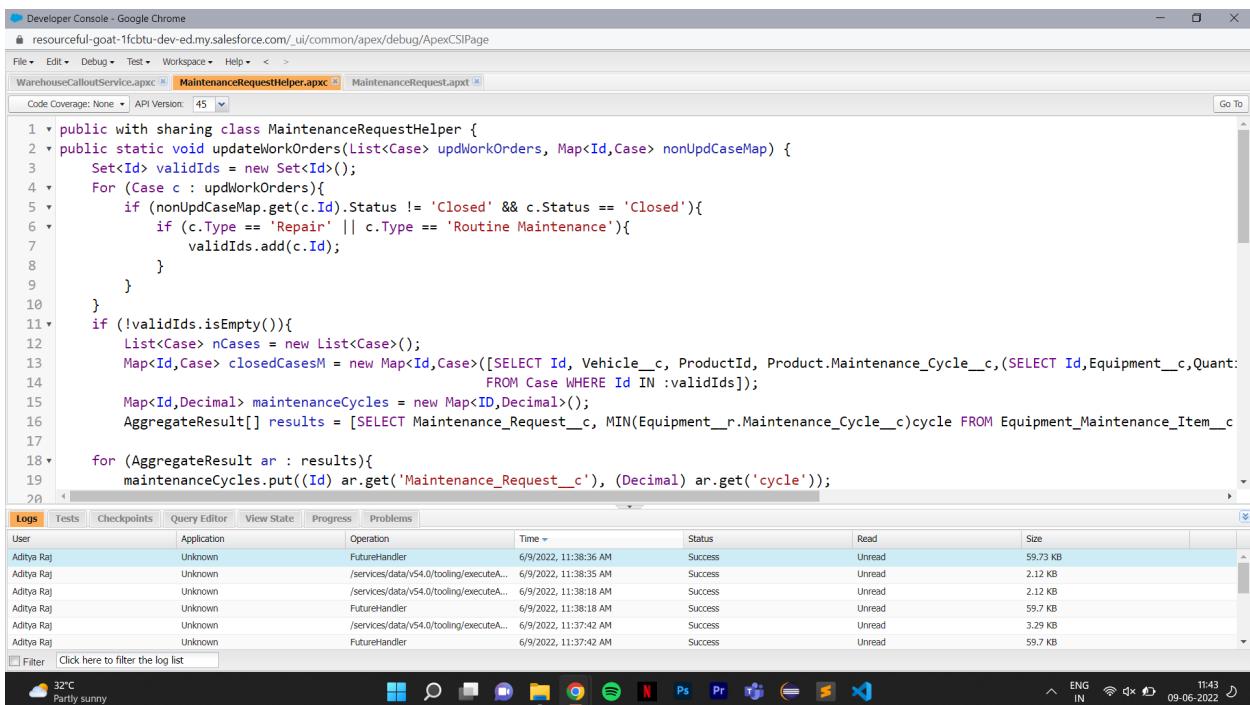
```

```
Equipment_Maintenance_Item__c wpClone = wp.clone();
wpClone.Maintenance_Request__c = nc.Id;
ClonedWPs.add(wpClone);

}
}

insert ClonedWPs;
}

}
```



## MaintenanceRequest Trigger:

```
trigger MaintenanceRequest on Case (before update, after update) {
```

```
if(Trigger.isUpdate && Trigger.isAfter){
```

```
    MaintenanceRequestHelper.updateWorkOrders(Trigger.New,  
    Trigger.OldMap);
```

```
}
```

The screenshot shows the Salesforce Developer Console in Google Chrome. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and tabs for WarehouseCalloutService.apxc, MaintenanceRequestHelper.apxc, and MaintenanceRequest.apxt. The code editor displays the trigger code:

```
trigger MaintenanceRequest on Case (before update, after update) {  
if(Trigger.isUpdate && Trigger.isAfter){  
    MaintenanceRequestHelper.updateWorkOrders(Trigger.New, Trigger.OldMap);  
}
```

Below the code editor is a log viewer with tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Logs tab is selected, showing the following log entries:

User	Application	Operation	Time	Status	Read	Size
Aditya Raj	Unknown	FutureHandler	6/9/2022, 11:38:36 AM	Success	Unread	59.73 KB
Aditya Raj	Unknown	/services/data/v54.0/tooling/executeA...	6/9/2022, 11:38:35 AM	Success	Unread	2.12 KB
Aditya Raj	Unknown	/services/data/v54.0/tooling/executeA...	6/9/2022, 11:38:18 AM	Success	Unread	2.12 KB
Aditya Raj	Unknown	FutureHandler	6/9/2022, 11:38:18 AM	Success	Unread	59.7 KB
Aditya Raj	Unknown	/services/data/v54.0/tooling/executeA...	6/9/2022, 11:37:42 AM	Success	Unread	3.29 KB
Aditya Raj	Unknown	FutureHandler	6/9/2022, 11:37:42 AM	Success	Unread	59.7 KB

The bottom status bar shows the weather as 32°C Partly sunny, system time as 11:45, and date as 09-06-2022.

## Synchronize Salesforce data with an external system:

WarehouseCalloutService

```
public with sharing class WarehouseCalloutService implements Queueable {  
    private static final String WAREHOUSE_URL = 'https://th-superbadge-  
apex.herokuapp.com/equipment';  
    @future(callout=true)  
    public static void runWarehouseEquipmentSync(){  
        Http http = new Http();  
        HttpRequest request = new HttpRequest();  
  
        request.setEndpoint(WAREHOUSE_URL);  
        request.setMethod('GET');  
        HttpResponse response = http.send(request);  
  
        List<Product2> warehouseEq = new List<Product2>();  
  
        if (response.getStatusCode() == 200){  
            List<Object> jsonResponse =  
(List<Object>)JSON.deserializeUntyped(response.getBody());  
            System.debug(response.getBody());  
  
            //class maps the following fields: replacement part (always true), cost,  
            current inventory, lifespan, maintenance cycle, and warehouse SKU  
            //warehouse SKU will be external ID for identifying which equipment  
            records to update within Salesforce  
            for (Object eq : jsonResponse){  
                Map<String, Object> mapJson = (Map<String, Object>)eq;  
                Product2 myEq = new Product2();  
                myEq.Replacement_Part__c = (Boolean)mapJson.get('replacement');
```

```
    myEq.Name = (String) mapJson.get('name');
    myEq.Maintenance_Cycle__c = (Integer)
mapJson.get('maintenanceperiod');
    myEq.Lifespan_Months__c = (Integer) mapJson.get('lifespan');
    myEq.Cost__c = (Integer) mapJson.get('cost');
    myEq.Warehouse_SKU__c = (String) mapJson.get('sku');
    myEq.Current_Inventory__c = (Double) mapJson.get('quantity');
    myEq.ProductCode = (String) mapJson.get('_id');
    warehouseEq.add(myEq);
}

if (warehouseEq.size() > 0){
    upsert warehouseEq;
    System.debug('Your equipment was synced with the warehouse
one');
}
}

public static void execute (QueueableContext context){
    runWarehouseEquipmentSync();
}

}
```

Developer Console - Google Chrome  
resourceful-goat-1fcbtu-dev-ed.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >

WarehouseCalloutService.apxc MaintenanceRequestHelper.apxc MaintenanceRequest.apxt

Code Coverage: None ▾ API Version: 45 ▾ Go To

```
1 public with sharing class WarehouseCalloutService implements Queueable {
2     private static final String WAREHOUSE_URL = 'https://th-superbadge-apex.herokuapp.com/equipment';
3     @future(callout=true)
4     public static void runWarehouseEquipmentSync(){
5         Http http = new Http();
6         HttpRequest request = new HttpRequest();
7
8         request.setEndpoint(WAREHOUSE_URL);
9         request.setMethod('GET');
10        HttpResponse response = http.send(request);
11
12        List<Product2> warehouseEq = new List<Product2>();
13
14        if (response.getStatusCode() == 200){
15            List<Object> jsonResponse = (List<Object>)JSON.deserializeUntyped(response.getBody());
16            System.debug(response.getBody());
17
18            //class maps the following fields: replacement part (always true), cost, current inventory, lifespan, maintenance cycle, and warehouse
19            //warehouse SKU will be external ID for identifying which equipment records to update within Salesforce
20    }
```

Logs Tests Checkpoints Query Editor View State Progress Problems

User	Application	Operation	Time	Status	Read	Size
Aditya Raj	Unknown	FutureHandler	6/9/2022, 11:38:36 AM	Success	Unread	59.73 KB
Aditya Raj	Unknown	/services/data/v54.0/tooling/executeA...	6/9/2022, 11:38:35 AM	Success	Unread	2.12 KB
Aditya Raj	Unknown	/services/data/v54.0/tooling/executeA...	6/9/2022, 11:38:18 AM	Success	Unread	2.12 KB
Aditya Raj	Unknown	FutureHandler	6/9/2022, 11:38:18 AM	Success	Unread	59.7 KB
Aditya Raj	Unknown	/services/data/v54.0/tooling/executeA...	6/9/2022, 11:37:42 AM	Success	Unread	3.29 KB
Aditya Raj	Unknown	FutureHandler	6/9/2022, 11:37:42 AM	Success	Unread	59.7 KB

Filter Click here to filter the log list

32°C Partly sunny

Cloudy icon

Windows taskbar icons: File Explorer, Search, Task View, Chat, Google Chrome, Netflix, Photoshop, Project, Microsoft Edge, Task Manager, Snipping Tool, Camera, Speaker.

System tray icons: ENG IN, Wi-Fi, Battery, 11:45, 09-06-2022, and a circular arrow icon.

## Schedule synchronization:

WarehouseSyncSchedule class:

*global with sharing class WarehouseSyncSchedule implements Schedulable{*

*global void execute(SchedulableContext ctx){*

*System.enqueueJob(new WarehouseCalloutService());*

*}*

*}*

The screenshot shows the Salesforce Developer Console interface. At the top, there's a header bar with 'Developer Console - Google Chrome' and the URL 'resourceful-goat-1fcbtu-dev-ed.my.salesforce.com/.ui/common/apex/debug/ApexCSIPage'. Below the header are standard browser navigation buttons. The main area contains the Apex code for the 'WarehouseSyncSchedule' class. The code is as follows:

```
1 * global with sharing class WarehouseSyncSchedule implements Schedulable{
2     global void execute(SchedulableContext ctx){
3         System.enqueueJob(new WarehouseCalloutService());
4     }
5 }
```

Below the code editor is a tabs bar with 'Logs' selected, followed by 'Tests', 'Checkpoints', 'Query Editor', 'View State', 'Progress', and 'Problems'. The 'Logs' tab displays a table of API logs:

User	Application	Operation	Time	Status	Read	Size
Aditya Raj	Unknown	FutureHandler	6/9/2022, 11:38:36 AM	Success	Unread	59.73 KB
Aditya Raj	Unknown	/services/data/v54.0/tooling/executeA...	6/9/2022, 11:38:35 AM	Success	Unread	2.12 KB
Aditya Raj	Unknown	/services/data/v54.0/tooling/executeA...	6/9/2022, 11:38:18 AM	Success	Unread	2.12 KB
Aditya Raj	Unknown	FutureHandler	6/9/2022, 11:38:18 AM	Success	Unread	59.7 KB
Aditya Raj	Unknown	/services/data/v54.0/tooling/executeA...	6/9/2022, 11:37:42 AM	Success	Unread	3.29 KB
Aditya Raj	Unknown	FutureHandler	6/9/2022, 11:37:42 AM	Success	Unread	59.7 KB

At the bottom of the screen, there's a system tray with icons for weather (32°C Partly sunny), taskbar icons (Windows, File Explorer, Task View, Google Chrome, Netflix, Photoshop, Project, Teams, Edge, and a speaker icon), and system status indicators (ENG IN, battery level, signal strength, and date/time 09-06-2022 11:49).

## Test automation logic:

MaintenanceRequestHelperTest

@istest

```
public with sharing class MaintenanceRequestHelperTest {
```

```
    private static final string STATUS_NEW = 'New';
    private static final string WORKING = 'Working';
    private static final string CLOSED = 'Closed';
    private static final string REPAIR = 'Repair';
    private static final string REQUEST_ORIGIN = 'Web';
    private static final string REQUEST_TYPE = 'Routine Maintenance';
    private static final string REQUEST SUBJECT = 'Testing subject';
```

```
PRIVATE STATIC Vehicle__c createVehicle(){
```

```
    Vehicle__c Vehicle = new Vehicle__C(name = 'SuperTruck');
    return Vehicle;
}
```

```
PRIVATE STATIC Product2 createEq(){
```

```
    product2 equipment = new product2(name = 'SuperEquipment',
        lifespan_months__C = 10,
        maintenance_cycle__C = 10,
        replacement_part__c = true);
    return equipment;
}
```

```
PRIVATE STATIC Case createMaintenanceRequest(id vehicleId, id
equipmentId){
```

```
    case cs = new case(Type=REPAIR,
        Status=STATUS_NEW,
```

```
    Origin=REQUEST_ORIGIN,  
    Subject=REQUEST SUBJECT,  
    ProductId=equipmentId,  
    Vehicle_c=vehicleId);  
  
    return cs;  
}
```

```
PRIVATE STATIC Equipment_Maintenance_Item__c createWorkPart(id  
equipmentId,id requestId){  
    Equipment_Maintenance_Item__c wp = new  
Equipment_Maintenance_Item__c(Equipment__c = equipmentId,  
                                Maintenance_Request__c =  
requestId);  
    return wp;  
}
```

```
@istest  
private static void testMaintenanceRequestPositive(){  
    Vehicle__c vehicle = createVehicle();  
    insert vehicle;  
    id vehicleId = vehicle.Id;  
  
    Product2 equipment = createEq();  
    insert equipment;  
    id equipmentId = equipment.Id;  
  
    case somethingToUpdate =  
createMaintenanceRequest(vehicleId,equipmentId);  
    insert somethingToUpdate;  
  
    Equipment_Maintenance_Item__c workP =
```

```
createWorkPart(equipmentId,somethingToUpdate.id);  
insert workP;
```

```
test.startTest();  
somethingToUpdate.status = CLOSED;  
update somethingToUpdate;  
test.stopTest();
```

```
Case newReq = [Select id, subject, type, ProductId, Date_Reported__c,  
Vehicle__c, Date_Due__c  
from case  
where status =:STATUS_NEW];
```

```
Equipment_Maintenance_Item__c workPart = [select id  
from Equipment_Maintenance_Item__c  
where Maintenance_Request__c =:newReq.Id];
```

```
system.assert(workPart != null);  
system.assert(newReq.Subject != null);  
system.assertEquals(newReq.Type, REQUEST_TYPE);  
SYSTEM.assertEquals(newReq.ProductId, equipmentId);  
SYSTEM.assertEquals(newReq.Vehicle__c, vehicleId);  
SYSTEM.assertEquals(newReq.Date_Reported__c, system.today());  
}
```

```
@istest  
private static void testMaintenanceRequestNegative(){
```

```
Vehicle__C vehicle = createVehicle();  
insert vehicle;  
id vehicleId = vehicle.Id;
```

```
product2 equipment = createEq();
```

```

insert equipment;
id equipmentId = equipment.Id;

case emptyReq = createMaintenanceRequest(vehicleId,equipmentId);
insert emptyReq;

Equipment_Maintenance_Item__c workP = createWorkPart(equipmentId,
emptyReq.Id);
insert workP;

test.startTest();
emptyReq.Status = WORKING;
update emptyReq;
test.stopTest();

list<case> allRequest = [select id
                           from case];

Equipment_Maintenance_Item__c workPart = [select id
                                             from Equipment_Maintenance_Item__c
                                             where Maintenance_Request__c = :emptyReq.Id];

system.assert(workPart != null);
system.assert(allRequest.size() == 1);

}

@istest
private static void testMaintenanceRequestBulk(){
list<Vehicle__C> vehicleList = new list<Vehicle__C>();
list<Product2> equipmentList = new list<Product2>();
list<Equipment_Maintenance_Item__c> workPartList = new
list<Equipment_Maintenance_Item__c>();

```

```
list<case> requestList = new list<case>();
list<id> oldRequestIds = new list<id>();

for(integer i = 0; i < 300; i++){
    vehicleList.add(createVehicle());
    equipmentList.add(createEq());
}
insert vehicleList;
insert equipmentList;

for(integer i = 0; i < 300; i++){
    requestList.add(createMaintenanceRequest(vehicleList.get(i).id,
equipmentList.get(i).id));
}
insert requestList;

for(integer i = 0; i < 300; i++){
    workPartList.add(createWorkPart(equipmentList.get(i).id,
requestList.get(i).id));
}
insert workPartList;

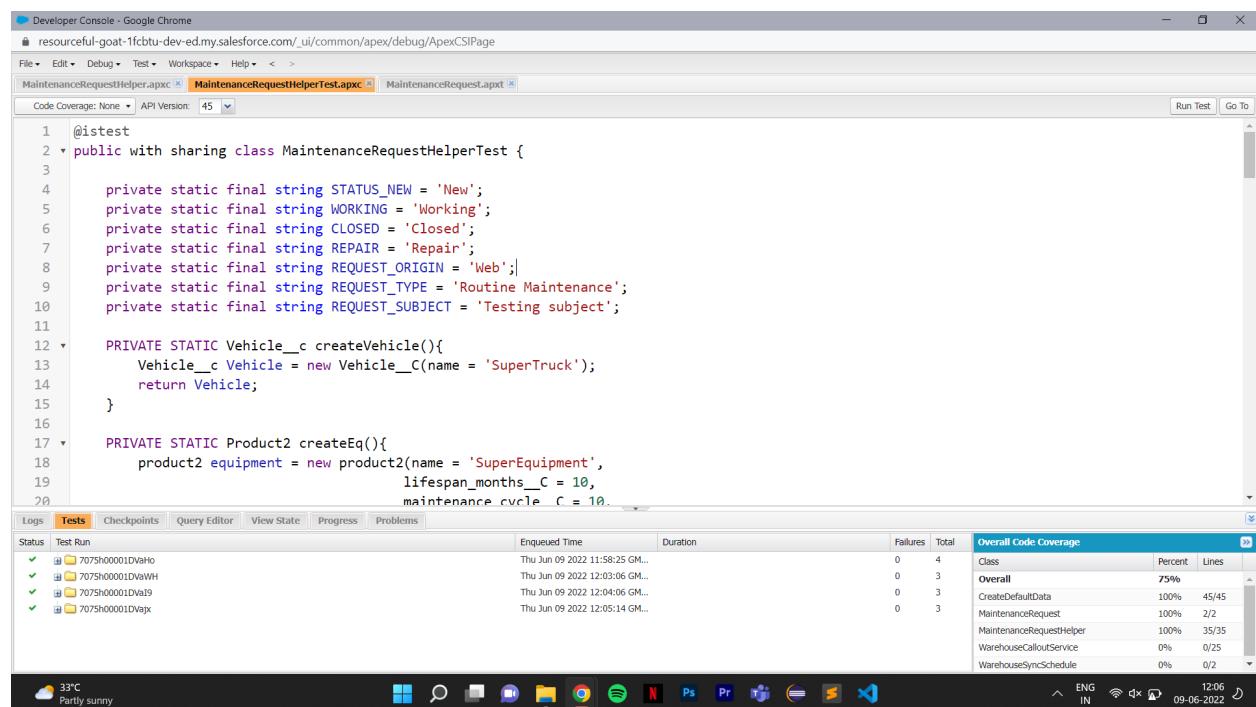
test.startTest();
for(case req : requestList){
    req.Status = CLOSED;
    oldRequestIds.add(req.Id);
}
update requestList;
test.stopTest();

list<case> allRequests = [select id
                           from case
```

*where status =: STATUS\_NEW];*

```
list<Equipment_Maintenance_Item__c> workParts = [select id  
from Equipment_Maintenance_Item__c  
where Maintenance_Request__c in:  
oldRequestIds];
```

```
    system.assert(allRequests.size() == 300);  
}
```



# 100% coverage

The screenshot shows the Salesforce Developer Console in Google Chrome. The title bar reads "Developer Console - Google Chrome" and the URL is "resourceful-goat-1fcbtu-dev-ed.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage". The menu bar includes File, Edit, Debug, Test, Workspace, Help, and a Go To button. The tabs at the top are "MaintenanceRequestHelper.apxc", "MaintenanceRequestHelperTest.apxc", and "MaintenanceRequestApex". The main area displays the Apex code for the helper class:

```
30     Origin = 'Web',
31     Date_Reported__c = Date.Today()
32
33 );
34
35 v   If (maintenanceCycles.containsKey(cc.Id)){
36     nc.Date_Due__c = Date.today().addDays((Integer) maintenanceCycles.get(cc.Id));
37   }
38
39   nCases.add(nc);
40 }
41
42 insert nCases;
43
44 List<Equipment_Maintenance_Item__c> clonedWPs = new List<Equipment_Maintenance_Item__c>();
45 v   for (Case nc : nCases){
46 v     for (Equipment_Maintenance_Item__c wp : closedCasesM.get(nc.ParentId).Equipment_Maintenance_Items__r){
47       Equipment_Maintenance_Item__c wpClone = wp.clone();
48       wpClone.Maintenance_Request__c = nc.Id;
49 }
```

Below the code editor is a test results table:

Status	Test Run	Enqueued Time	Duration	Failures	Total
✓	7075H00001DWh0	Thu Jun 09 2022 11:58:25 GM...		0	4
✓	7075H00001DWbWH	Thu Jun 09 2022 12:03:05 GM...		0	3
✓	7075H00001DWh19	Thu Jun 09 2022 12:04:06 GM...		0	3
✓	7075H00001DWhgj	Thu Jun 09 2022 12:05:14 GM...		0	3

At the bottom right, there is a weather icon (33°C, Partly sunny), system status icons (ENG IN, battery, signal, etc.), and the date/time (12:06 09-06-2022).

## Test callout logic:

WarehouseCalloutServiceMock

```
@isTest
global class WarehouseCalloutServiceMock implements HttpCalloutMock {
    // implement http mock callout
    global static HttpResponse respond(HttpRequest request){
        System.assertEquals('https://th-superbadge-
apex.herokuapp.com/equipment', request.getEndpoint());
        System.assertEquals('GET', request.getMethod());

        // Create a fake response
        HttpResponse response = new HttpResponse();
        response.setHeader('Content-Type', 'application/json');

        response.setBody('[{"_id":"55d66226726b611100aaf741","replacement":false,
"quantity":5,"name":"Generator 1000
kW","maintenanceperiod":365,"lifespan":120,"cost":5000,"sku":"100003"}]');
        response.setStatusCode(200);
        return response;
    }
}
```

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is [resourceful-goat-1fcbtu-dev-ed.my.salesforce.com/\\_ui/common/apex/debug/ApexCSIPage](https://resourceful-goat-1fcbtu-dev-ed.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage). The tabs at the top are "WarehouseCalloutService.apxc" (selected), "WarehouseCalloutServiceMock.apxc" (highlighted in orange), and "WarehouseCalloutServiceTest.apxc". The status bar shows "Code Coverage None | API Version 45". The main area displays the Apex code for `WarehouseCalloutServiceMock`, which implements `HttpCalloutMock` and overrides the `respond` method to return a fake response. Below the code is a table of test results for the `WarehouseCalloutServiceTest` class, showing 7 successful tests and 3 failures. The "Overall Code Coverage" table shows 98% coverage across various methods. The bottom of the screen shows the Windows taskbar with icons for File Explorer, Task View, Start, Taskbar settings, and other applications like Google Chrome, Photoshop, and Microsoft Word.

```

1  @isTest
2  v global class WarehouseCalloutServiceMock implements HttpCalloutMock {
3      // implement http mock callout
4  v     global static HttpResponse respond(HttpRequest request){
5
6         System.assertEquals('https://th-superbadge-apex.herokuapp.com/equipment', request.getEndpoint());
7         System.assertEquals('GET', request.getMethod());
8
9         // Create a fake response
10        HttpResponse response = new HttpResponse();
11        response.setHeader('Content-Type', 'application/json');
12        response.setBody('[{"_id": "55d66226726b611100aaf741", "replacement": false, "quantity": 5, "name": "Generator 1000 kW", "maintenanceperiod": 365, "li...}');
13        response.setStatusCode(200);
14        return response;
15    }
16 }

```

Status	Test Run	Enqueued Time	Duration	Failures	Total
✓	7075h00001DVdnh	Thu Jun 09 2022 12:51:36 GM...		0	3
✓	7075h00001DVjx	Thu Jun 09 2022 12:05:14 GM...		0	7
✓	7075h00001DwA9	Thu Jun 09 2022 12:04:06 GM...		0	7
✓	7075h00001DwAHI	Thu Jun 09 2022 12:03:06 GM...		0	7
✓	7075h00001DwH0	Thu Jun 09 2022 11:58:25 GM...		0	4
✓	7075h00001DVdmr	Thu Jun 09 2022 12:52:34 GM...		0	3

Overall Code Coverage		
Class	Percent	Lines
Overall	98%	
CreateDefaultData	100%	45/45
MaintenanceRequest	100%	2/2
MaintenanceRequestHelper	100%	35/35
WarehouseCalloutService	100%	23/23
WarehouseSyncSchedule	0%	0/2

## WarehouseCalloutServiceTest

### @isTest

```

private class WarehouseCalloutServiceTest {
    @isTest
    static void testWareHouseCallout(){
        Test.startTest();
        // implement mock callout test here
        Test.setMock(HTTPCalloutMock.class, new
        WarehouseCalloutServiceMock());
        WarehouseCalloutService.runWarehouseEquipmentSync();
        Test.stopTest();
        System.assertEquals(1, [SELECT count() FROM Product2]);
    }
}

```

Developer Console - Google Chrome  
 resourceful-goat-1fcbtu-dev-ed.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >

WarehouseCalloutService.apxc ▾ WarehouseCalloutServiceMock.apxc ▾ WarehouseCalloutServiceTest.apxc

Code Coverage: None ▾ API Version: 45 ▾ Run Test Go To

```

1  @isTest
2
3  private class WarehouseCalloutServiceTest {
4      @isTest
5      static void testWareHouseCallout(){
6          Test.startTest();
7          // implement mock callout test here
8          Test.setMock(HTTPCalloutMock.class, new WarehouseCalloutServiceMock());
9          WarehouseCalloutService.runWarehouseEquipmentsSync();
10         Test.stopTest();
11         System.assertEquals(1, [SELECT count() FROM Product2]);
12     }
13 }
```

Logs Tests Checkpoints Query Editor View State Progress Problems

Status	Test Run	Enqueued Time	Duration	Failures	Total
✓	7075h00001DVdnh	Thu Jun 09 2022 12:51:36 GM...		0	3
✓	7075h00001DVnjx	Thu Jun 09 2022 12:05:14 GM...		0	7
✓	7075h00001DwA19	Thu Jun 09 2022 12:04:06 GM...		0	7
✓	7075h00001DwAWh	Thu Jun 09 2022 12:03:06 GM...		0	7
✓	7075h00001DwH0	Thu Jun 09 2022 11:58:25 GM...		0	4
✓	7075h00001Dvdmr	Thu Jun 09 2022 12:52:34 GM...		0	3

Overall Code Coverage

Class	Percent	Lines
<b>Overall</b>	<b>98%</b>	
CreateDefaultData	100%	45/45
MaintenanceRequest	100%	2/2
MaintenanceRequestHelper	100%	35/35
WarehouseCalloutService	100%	23/23
WarehouseSyncSchedule	0%	0/2

Cloud Partly sunny 33°C ENG IN 12:54 09-06-2022

100% coverage

Developer Console - Google Chrome  
 resourceful-goat-1fcbtu-dev-ed.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >

WarehouseCalloutService.apxc ▾ WarehouseCalloutServiceMock.apxc ▾ WarehouseCalloutServiceTest.apxc

Code Coverage: All Tests 100% ▾ API Version: 45 ▾ Go To

```

1  public with sharing class WarehouseCalloutService {
2      private static final String WAREHOUSE_URL = 'https://th-superbadge-apex.herokuapp.com/equipment';
3      //@future(callout=true)
4      public static void runWarehouseEquipmentSync(){
5          Http http = new Http();
6          HttpRequest request = new HttpRequest();
7
8          request.setEndpoint(WAREHOUSE_URL);
9          request.setMethod('GET');
10         HttpResponse response = http.send(request);
11
12         List<Product2> warehouseEq = new List<Product2>();
13
14         if (response.getStatusCode() == 200){
15             List<Object> jsonResponse = (List<Object>)JSON.deserializeUntyped(response.getBody());
16             System.debug(response.getBody());
17
18             //class maps the following fields: replacement part (always true), cost, current inventory, lifespan, maintenance cycle, and warehouse
19             //warehouse SKU will be external ID for identifying which equipment records to update within Salesforce
20     }
```

Logs Tests Checkpoints Query Editor View State Progress Problems

Status	Test Run	Enqueued Time	Duration	Failures	Total
✓	7075h00001DVdnh	Thu Jun 09 2022 12:51:36 GM...		0	3
✓	7075h00001DVnjx	Thu Jun 09 2022 12:05:14 GM...		0	7
✓	7075h00001DwA19	Thu Jun 09 2022 12:04:06 GM...		0	7
✓	7075h00001DwAWh	Thu Jun 09 2022 12:03:06 GM...		0	7
✓	7075h00001DwH0	Thu Jun 09 2022 11:58:25 GM...		0	4
✓	7075h00001Dvdmr	Thu Jun 09 2022 12:52:34 GM...		0	3

Overall Code Coverage

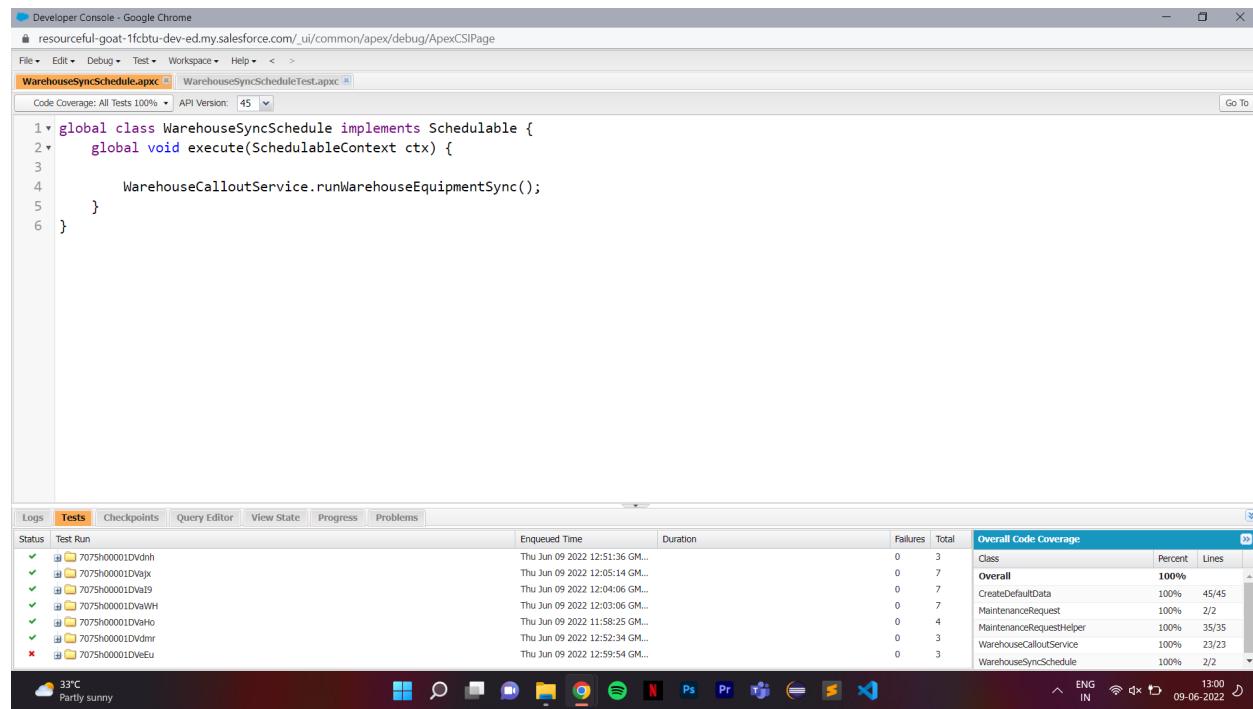
Class	Percent	Lines
<b>Overall</b>	<b>98%</b>	
CreateDefaultData	100%	45/45
MaintenanceRequest	100%	2/2
MaintenanceRequestHelper	100%	35/35
WarehouseCalloutService	100%	23/23
WarehouseSyncSchedule	0%	0/2

Cloud Partly sunny 33°C ENG IN 12:52 09-06-2022

## Scheduling logic:

### WarehouseSyncSchedule

```
global class WarehouseSyncSchedule implements Schedulable {  
    global void execute(SchedulableContext ctx) {  
  
        WarehouseCalloutService.runWarehouseEquipmentSync();  
    }  
}
```



### WarehouesSyncScheduleTest

```
@isTest  
public class WarehouseSyncScheduleTest {
```

```
@isTest static void WarehousescheduleTest(){
```

```

String scheduleTime = '00 00 01 * * ?';
Test.startTest();
Test.setMock(HttpCalloutMock.class, new
WarehouseCalloutServiceMock());
String jobID=System.schedule('Warehouse Time To Schedule to Test',
scheduleTime, new WarehouseSyncSchedule());
Test.stopTest();
//Contains schedule information for a scheduled job. CronTrigger is
similar to a cron job on UNIX systems.
// This object is available in API version 17.0 and later.
CronTrigger a=[SELECT Id FROM CronTrigger where NextFireTime >
today];
System.assertEquals(jobID, a.Id,'Schedule ');
}
}

```

Developer Console - Google Chrome  
resourceful-goat-1fbtu-dev-ed.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < ▾

WarehouseSyncScheduleTest.apxc ▾ WarehouseSyncScheduleTest.apxc

Code Coverage: None ▾ API Version: 45 ▾ Run Test Go To

```

1 @isTest
2 public class WarehouseSyncScheduleTest {
3
4     @isTest static void WarehousescheduleTest(){
5         String scheduleTime = '00 00 01 * * ?';
6         Test.startTest();
7         Test.setMock(HttpCalloutMock.class, new WarehouseCalloutServiceMock());
8         String jobID=System.schedule('Warehouse Time To Schedule to Test', scheduleTime, new WarehouseSyncSchedule());
9         Test.stopTest();
10        //Contains schedule information for a scheduled job. CronTrigger is similar to a cron job on UNIX systems.
11        // This object is available in API version 17.0 and later.
12        CronTrigger a=[SELECT Id FROM CronTrigger where NextFireTime > today];
13        System.assertEquals(jobID, a.Id,'Schedule ');
14
15    }
16
17 }

```

Logs Tests Checkpoints Query Editor View State Progress Problems

Status	Test Run	Enqueued Time	Duration	Failures	Total	Overall Code Coverage
✓	7075h00001DVdhh	Thu Jun 09 2022 12:51:36 GM...		0	3	Class
✓	7075h00001DwAjx	Thu Jun 09 2022 12:05:14 GM...		0	7	Overall 100%
✓	7075h00001DVd19	Thu Jun 09 2022 12:04:06 GM...		0	7	CreateDefaultData
✓	7075h00001DVdWh	Thu Jun 09 2022 12:03:06 GM...		0	7	MaintenanceRequest
✓	7075h00001DVdHo	Thu Jun 09 2022 11:58:25 GM...		0	4	MaintenanceRequestHelper
✓	7075h00001DVdMr	Thu Jun 09 2022 12:52:34 GM...		0	3	WarehouseCalloutService
⌚	7075h00001DwEu	Thu Jun 09 2022 12:59:54 GM...		1	6	WarehouseSyncSchedule

33°C Party sunny ENG IN 13:00 09-06-2022