Phase: 5 Apex Programming

Part 1: Creating the Apex Files

First, we need to create each of the necessary Apex files in your Salesforce org.

Resource:- using the Developer Console.

Step 1.1: Open the Developer Console

1. Log in to your Salesforce org.
2. Click the **Gear Icon** in the top-right corner.
3. Select **Developer Console** from the dropdown menu.

Step 1.2: Create Each Apex File

For each of the six files listed below, you will follow the same process:

- In the Developer Console, click **File → New → Apex Class** (or **Apex Trigger** for the .trigger file).
- Enter the exact file name provided.
- Delete any default text in the editor window.
- Copy and paste the complete code for that file.
- Click **File → Save** or press Ctrl + S.

Create the files in this order:

1. **VolunteerTriggerHandler.apxc (Class)**

```apex
public class VolunteerTriggerHandler {
    private static Boolean isRunning = false;

    public static void updateVolunteerTotals(List<Volunteer__c> volunteers){
        if(isRunning) return;
        isRunning = true;

        Set<Id> volunteerIds = new Set<Id>();
        for(Volunteer__c v : volunteers){
            if(v.Id != null){
                volunteerIds.add(v.Id);
            }
        }
        if(volunteerIds.isEmpty()) {
            isRunning = false;
            return;
        }

        // 1. Aggregate Donations
        Map<Id, Decimal> donationTotals = new Map<Id, Decimal>();
        for(AggregateResult ar : [SELECT Linked_Volunteer__c vId, SUM(Amount__c) total FROM Donation__c WHERE Linked_Volunteer__c IN :volunteerIds GROUP BY Linked_Volunteer__c]){
            donationTotals.put((Id)ar.get('vId'), (Decimal)ar.get('total'));
        }

        // 2. Aggregate Hours using your confirmed field names
        Map<Id, Decimal> hourTotals = new Map<Id, Decimal>();
        for(AggregateResult ar : [SELECT Volunteer__c vId, SUM(Hours_Worked__c) totalHours FROM Volunteer_Task__c WHERE Volunteer__c IN :volunteerIds GROUP BY Volunteer__c]){
            hourTotals.put((Id)ar.get('vId'), (Decimal)ar.get('totalHours'));
        }

        // 3. Prepare updates
        List<Volunteer__c> updates = new List<Volunteer__c>();
        List<Volunteer__c> existingVolunteers = [SELECT Id, Total_Donations__c, Total_Hours__c FROM Volunteer__c WHERE Id IN :volunteerIds];

        for(Volunteer__c existing : existingVolunteers){
            Decimal newDonations = donationTotals.get(existing.Id) == null ? 0 : donationTotals.get(existing.Id);
            Decimal newHours = hourTotals.get(existing.Id) == null ? 0 : hourTotals.get(existing.Id);

            if(existing.Total_Donations__c != newDonations || existing.Total_Hours__c != newHours){
                Volunteer__c vUpdate = new Volunteer__c(Id = existing.Id);
                vUpdate.Total_Donations__c = newDonations;
                vUpdate.Total_Hours__c = newHours;
                updates.add(vUpdate);
            }
        }

        if(!updates.isEmpty()){
            update updates;
        }

        isRunning = false;
    }
}
```

2. VolunteerTrigger.apxt (Trigger)

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

VolunteerTrigger.apxt

Code Coverage: None ▾ API Version: 64

```apex
trigger VolunteerTrigger on Volunteer__c (after insert, after update) {
    if(Trigger.isAfter){
        VolunteerTriggerHandler.updateVolunteerTotals(Trigger.new);
    }
}
```

3. VolunteerTriggerTest.apxc (Class)

VolunteerTriggerTest.apxc

Code Coverage: None | API Version: 64

```
@isTest
public class VolunteerTriggerTest {
    @testSetup
    static void setupData() {
        Volunteer__c v = new Volunteer__c(
            Volunteer_Name__c = 'Test Volunteer',
            Email__c = 'vol@test.com',
            Phone_Number__c = '1234567890',
            Availability__c = 'Weekends',
            Status__c = 'Active'
        );
        insert v;

        Donation__c d1 = new Donation__c(
            Donor_Name__c = 'Donor One',
            Donor_Email__c = 'donor1@test.com',
            Amount__c = 100,
            Linked_Volunteer__c = v.Id,   // <-- ADD A COMMA HERE
            Donation_Date__c = Date.today() // <-- ADD A COMMA HERE
        );
            insert d1;
    }

    @isTest
    static void testVolunteerTotalsUpdate() {
        Volunteer__c v = [SELECT Id, Total_Donations__c, Total_Hours__c FROM Volunteer__c LIMIT 1];

        System.assertEquals(100, v.Total_Donations__c, 'Total Donations should be updated');
        System.assertEquals(0, v.Total_Hours__c, 'Total Hours should remain 0 since no Hours__c field exists on Task');
    }
}
```

4. DonationTriggerHandler.apxc (Class)

DonationTriggerHandler.apxc

Code Coverage: None | API Version: 64

```
public class DonationTriggerHandler {
    public static void processDonations(List<Donation__c> donations){
        Set<Id> volunteerIds = new Set<Id>();
        for(Donation__c d : donations){
            if(d.Linked_Volunteer__c != null){
                volunteerIds.add(d.Linked_Volunteer__c);
            }
        }

        if(!volunteerIds.isEmpty()){
            VolunteerTriggerHandler.updateVolunteerTotals(
                [SELECT Id FROM Volunteer__c WHERE Id IN :volunteerIds]
            );
        }

        // Async email
        for(Donation__c d : donations){
            if(d.Donor_Email__c != null){
                System.enqueueJob(new DonationThankYouEmailQueue(d.Id));
            }
        }
    }
}
```

5. DonationTrigger.apxt (Trigger)

DonationTrigger.apxt

Code Coverage: None API Version: 64

```
trigger DonationTrigger on Donation__c (after insert, after update) {
    if(Trigger.isAfter){
        DonationTriggerHandler.processDonations(Trigger.new);
    }
}
```

6. DonationTriggerTest.apxc (Class)

File Edit Debug Test Workspace Help < >

DonationTriggerTest.apxc

Code Coverage: None API Version: 64

```
@isTest
public class DonationTriggerTest {
    @isTest
    static void testDonationProcessingAndEmail() {
        Volunteer__c v = new Volunteer__c(
            Volunteer_Name__c = 'Donor Volunteer',
            Email__c = 'donorvol@test.com',
            Phone_Number__c = '9999999999',
            Availability__c = 'Weekdays',
            Status__c = 'Active'
        );
        insert v;

        Test.startTest();
        Donation__c d = new Donation__c(
            Donor_Name__c = 'Test Donor',
            Donor_Email__c = 'thankyou@test.com',
            Amount__c = 200,
            Linked_Volunteer__c = v.Id,
            Donation_Date__c = Date.today()
        );
        insert d;
        Test.stopTest(); // ensures async Queueable job executes

        v = [SELECT Total_Donations__c FROM Volunteer__c WHERE Id = :v.Id];
        System.assertEquals(200, v.Total_Donations__c, 'Volunteer total donations should update after donation');
    }
}
```

7. DonationThankYouEmailQueue.apxc (Class)

DonationThankYouEmailQueue.apxc

Code Coverage: None | API Version: 64

```apex
public class DonationThankYouEmailQueue implements Queueable {
    private Id donationId;
    public DonationThankYouEmailQueue(Id donationId){
        this.donationId = donationId;
    }

    public void execute(QueueableContext context){
        Donation__c d = [
            SELECT Donor_Email__c, Donor_Name__c, Amount__c
            FROM Donation__c WHERE Id = :donationId LIMIT 1
        ];

        Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
        mail.setToAddresses(new String[]{d.Donor_Email__c});
        mail.setSubject('Thank you for your donation!');
        mail.setPlainTextBody('Dear ' + d.Donor_Name__c +
            ',\n\nThank you for your generous donation of $' + d.Amount__c +
            '. Your support is greatly appreciated.\n\n- Nonprofit Team');
        Messaging.sendEmail(new Messaging.SingleEmailMessage[]{mail});
    }
}
```

8. VolunteerTaskTriggerHandler.apxc (Class)

VolunteerTaskTriggerHandler.apxc

Code Coverage: None | API Version: 64

```apex
public class VolunteerTaskTriggerHandler {
    public static void updateVolunteerHours(List<Volunteer_Task__c> tasks){
        Set<Id> volunteerIds = new Set<Id>();
        for(Volunteer_Task__c t : tasks){
            if(t.Volunteer__c != null){
                volunteerIds.add(t.Volunteer__c);
            }
        }
        if(!volunteerIds.isEmpty()){
            VolunteerTriggerHandler.updateVolunteerTotals(
                [SELECT Id FROM Volunteer__c WHERE Id IN :volunteerIds]
            );
        }
    }
}
```

9. VolunteerTaskTrigger (Trigger)

VolunteerTaskTrigger.apxt

Code Coverage: None API Version: 64

```
trigger VolunteerTaskTrigger on Volunteer_Task__c (after insert, after update, after delete) {
    if(Trigger.isAfter){
        VolunteerTaskTriggerHandler.updateVolunteerHours(
            Trigger.isDelete ? Trigger.old : Trigger.new
        );
    }
}
```

10. VolunteerTaskTriggerTest (Class)

VolunteerTaskTrigger.apxt

Code Coverage: None API Version: 64

```
trigger VolunteerTaskTrigger on Volunteer_Task__c (after insert, after update, after delete) {
    if(Trigger.isAfter){
        VolunteerTaskTriggerHandler.updateVolunteerHours(
            Trigger.isDelete ? Trigger.old : Trigger.new
        );
    }
}
```

11. VolunteerTotalBatch.apxc (Class)

VolunteerTotalsBatch.apxc

Code Coverage: None API Version: 64

```
public class VolunteerTotalsBatch implements Database.Batchable<SObject>, Database.Stateful {
    public Database.QueryLocator start(Database.BatchableContext bc) {
        return Database.getQueryLocator('SELECT Id FROM Volunteer__c');
    }

    public void execute(Database.BatchableContext bc, List<Volunteer__c> scope) {
        VolunteerTriggerHandler.updateVolunteerTotals(scope);
    }

    public void finish(Database.BatchableContext bc) {
        System.debug('Volunteer totals recalculation completed.');
    }
}
```

12. VolunteerTotalsScheduler.cls (Class)

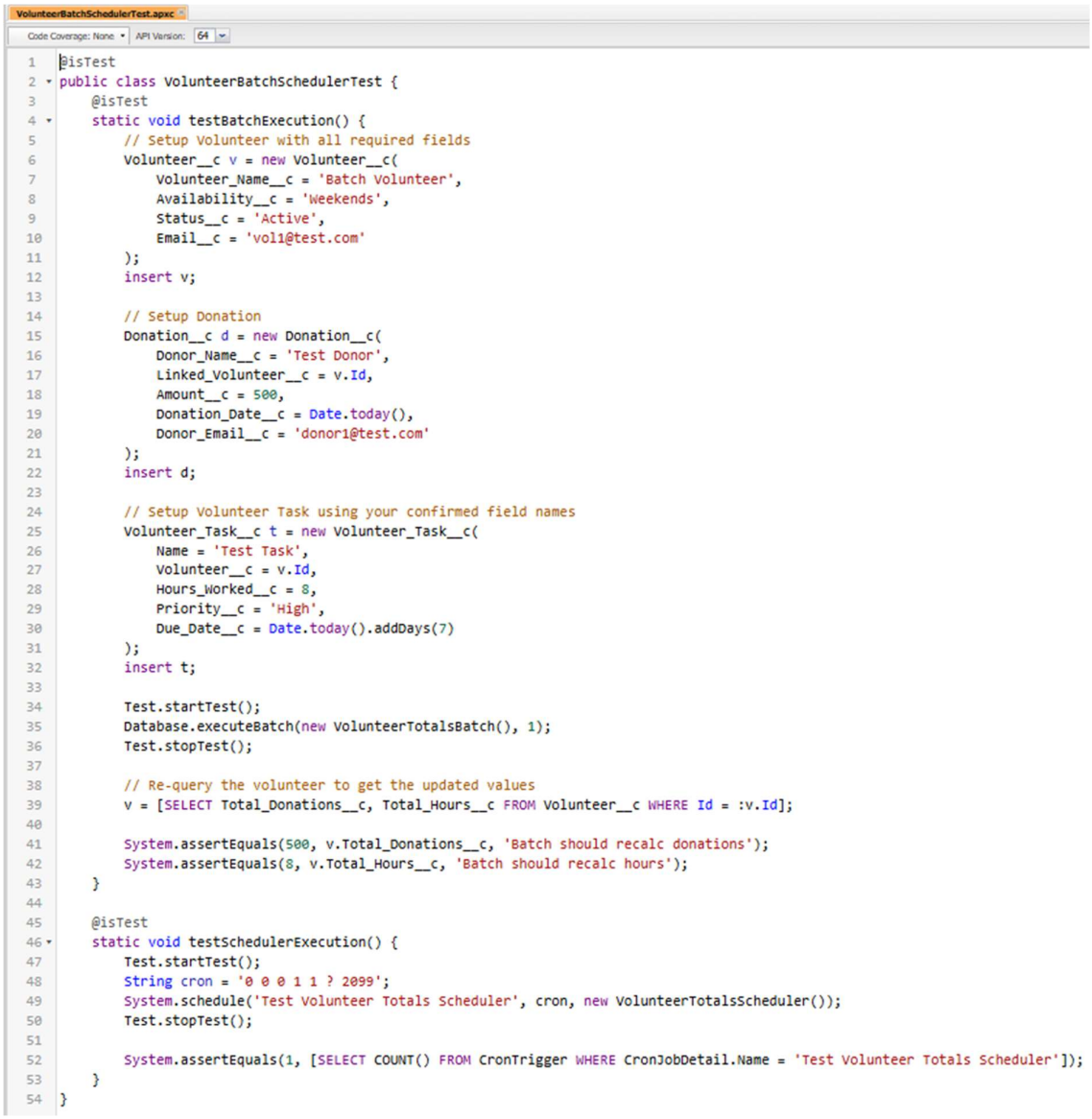
File Edit Debug Test Workspace Help < >

VolunteerTotalsScheduler.apxc

Code Coverage: None API Version: 64

```
public class VolunteerTotalsScheduler implements Schedulable {
    public void execute(SchedulableContext sc) {
        Database.executeBatch(new VolunteerTotalsBatch(), 200);
    }
}
```

13. **VolunteerTotalsScheduler.cls** (Class)




```
@isTest
public class VolunteerBatchSchedulerTest {
    @isTest
    static void testBatchExecution() {
        // Setup Volunteer with all required fields
        Volunteer__c v = new Volunteer__c(
            Volunteer_Name__c = 'Batch Volunteer',
            Availability__c = 'Weekends',
            Status__c = 'Active',
            Email__c = 'vol1@test.com'
        );
        insert v;

        // Setup Donation
        Donation__c d = new Donation__c(
            Donor_Name__c = 'Test Donor',
            Linked_Volunteer__c = v.Id,
            Amount__c = 500,
            Donation_Date__c = Date.today(),
            Donor_Email__c = 'donor1@test.com'
        );
        insert d;

        // Setup Volunteer Task using your confirmed field names
        Volunteer_Task__c t = new Volunteer_Task__c(
            Name = 'Test Task',
            Volunteer__c = v.Id,
            Hours_Worked__c = 8,
            Priority__c = 'High',
            Due_Date__c = Date.today().addDays(7)
        );
        insert t;

        Test.startTest();
        Database.executeBatch(new VolunteerTotalsBatch(), 1);
        Test.stopTest();

        // Re-query the volunteer to get the updated values
        v = [SELECT Total_Donations__c, Total_Hours__c FROM Volunteer__c WHERE Id = :v.Id];

        System.assertEquals(500, v.Total_Donations__c, 'Batch should recalc donations');
        System.assertEquals(8, v.Total_Hours__c, 'Batch should recalc hours');
    }

    @isTest
    static void testSchedulerExecution() {
        Test.startTest();
        String cron = '0 0 0 1 1 ? 2099';
        System.schedule('Test Volunteer Totals Scheduler', cron, new VolunteerTotalsScheduler());
        Test.stopTest();

        System.assertEquals(1, [SELECT COUNT() FROM CronTrigger WHERE CronJobDetail.Name = 'Test Volunteer Totals Scheduler']);
    }
}
```

After saving all 13 files without any errors, you are ready to run the tests.

Part 2: Running Tests and Capturing Results

Now we will execute the test classes to verify all the logic works correctly and to check your code coverage.

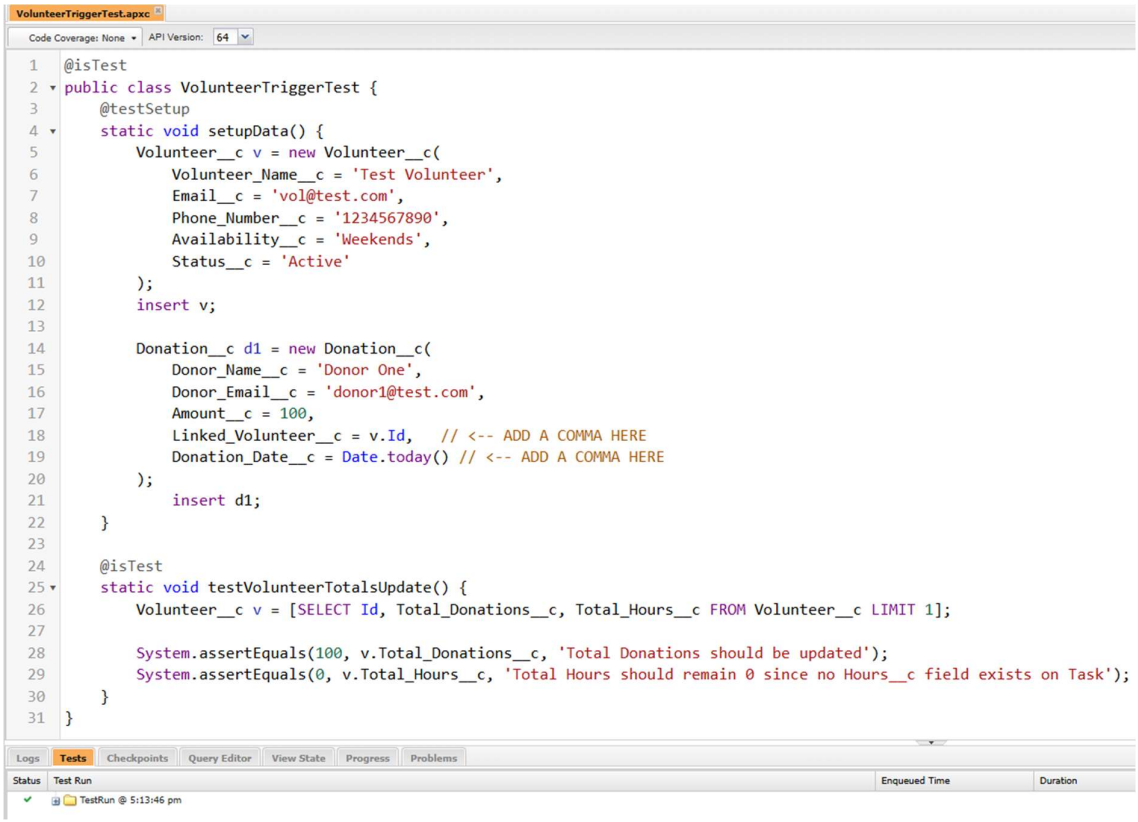
Step 2.1: Navigate to the Test Execution Tab

1. In the Developer Console, click on the **Test** tab in the menu bar.
2. Select **New Run**.

Step 2.2: Run the Trigger Test

1. From the "Test Classes" list, select **VolunteerTriggerTest**.
2. Click the **Add Selected** button to move it to the right-hand column.
3. Click the **Run** button.

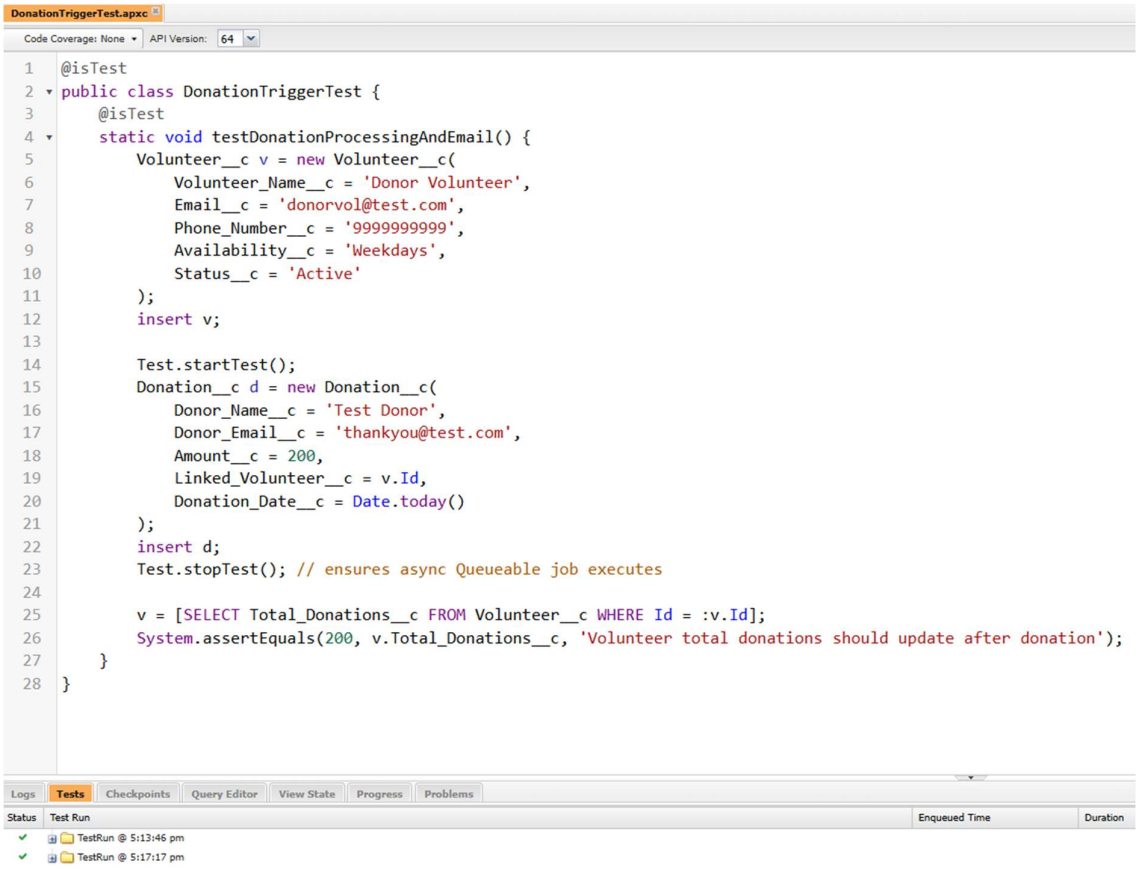
The test will execute, and the results will appear in the "Tests" tab.



Step 2.3: Run the Trigger Test

1. From the "Test Classes" list, select **DonationTriggerTest**.
2. Click the **Add Selected** button to move it to the right-hand column.
3. Click the **Run** button.

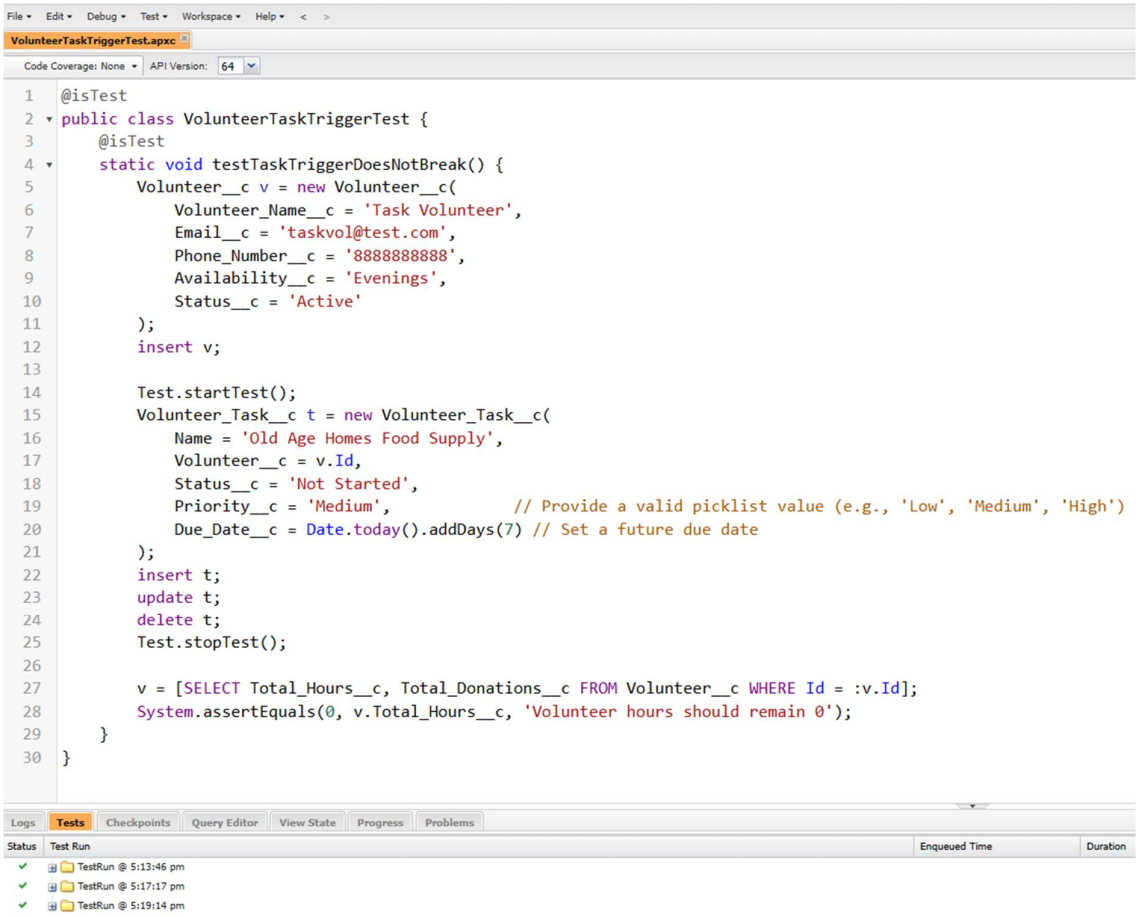
The test will execute, and the results will appear in the "Tests" tab.



Step 2.3: Run the Trigger Test

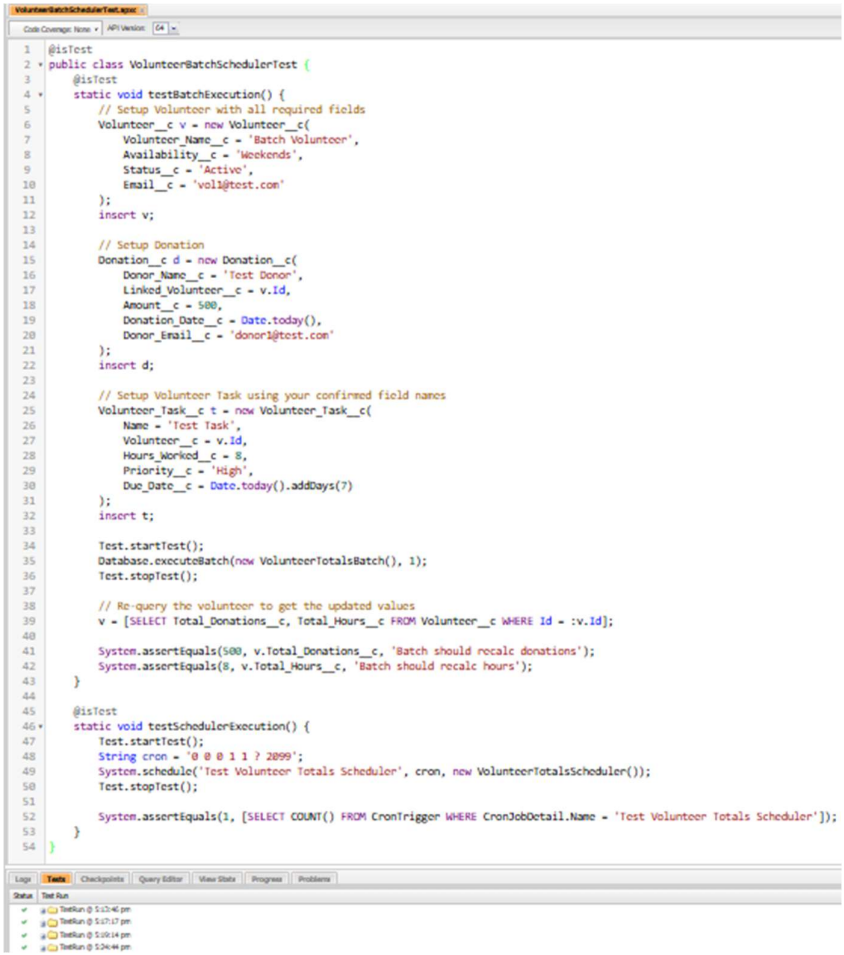
1. From the "Test Classes" list, select **VolunteerTaskTriggerTest**.
2. Click the **Add Selected** button to move it to the right-hand column.
3. Click the **Run** button.

The test will execute, and the results will appear in the "Tests" tab.



Step 2.4: Run the Batch and Scheduler Test

1. Just like before, click **Test → New Run**.
2. This time, select **VolunteerBatchSchedulerTest** from the list.
3. Click **Add Selected**, then click **Run**.

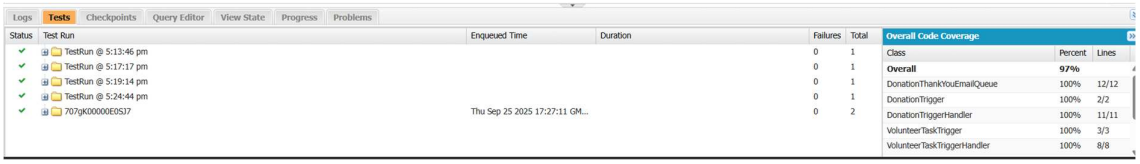


```
@isTest
public class VolunteerBatchSchedulerTest {
    @isTest
    static void testBatchExecution() {
        // Setup Volunteer with all required fields
        Volunteer__c v = new Volunteer__c(
            Volunteer_Name__c = 'Batch Volunteer',
            Availability__c = 'Weekends',
            Status__c = 'Active',
            Email__c = 'vol1@test.com'
        );
        insert v;

        // Setup Donation
        Donation__c d = new Donation__c(
            Donor_Name__c = 'Test Donor',
            Linked_Volunteer__c = v.Id,
            Amount__c = 500,
            Donation_Date__c = Date.today(),
            Donor_Email__c = 'donor1@test.com'
        );
        insert d;

        // Setup Volunteer Task using your confirmed field names
        Volunteer_Task__c t = new Volunteer_Task__c(
            Name = 'Test Task',
            Volunteer__c = v.Id,
            Hours_Worked__c = 8,
            Priority__c = 'High',
            Due_Date__c = Date.today().addDays(7)
        );
        insert t;

        Test.startTest();
        Database.executeBatch(new VolunteerTotalsBatch(), 1);
        Test.stopTest();

        // Re-query the volunteer to get the updated values
        v = [SELECT Total_Donations__c, Total_Hours__c FROM Volunteer__c WHERE Id = :v.Id];

        System.assertEquals(500, v.Total_Donations__c, 'Batch should recalc donations');
        System.assertEquals(8, v.Total_Hours__c, 'Batch should recalc hours');
    }

    @isTest
    static void testSchedulerExecution() {
        Test.startTest();
        String cron = '0 0 0 1 1 ? 2099';
        System.schedule('Test Volunteer Totals Scheduler', cron, new VolunteerTotalsScheduler());
        Test.stopTest();

        System.assertEquals(1, [SELECT COUNT() FROM CronTrigger WHERE CronJobDetail.Name = 'Test Volunteer Totals Scheduler']);
    }
}
```

Step 2.4: Check Overall Code Coverage

After all tests have passed, you can view your organization's overall Apex code coverage.

1. In the "Tests" tab of the Developer Console, look at the top-right corner. You will see "Overall Code Coverage".
2. Click the name of a class to see which lines were covered by your tests.



Part 3: Scheduling the Nightly Job

The final step is to run the script that officially schedules your batch job to run every night.

Step 3.1: Run the Anonymous Apex Script

1. In the Developer Console, click **Debug → Open Execute Anonymous Window**.
2. Copy and paste the following script into the window:

Apex

```
VolunteerTotalsScheduler job = new VolunteerTotalsScheduler();
String cron = '0 0 2 * * ?'; // Runs daily at 2 AM
System.schedule('Nightly Volunteer Totals Job', cron, job);
```

3. Click the **Execute** button at the bottom right.

Step 3.2: Verify the Job is Scheduled

1. Go back to the main Salesforce window (not the Developer Console).
2. Click the **Gear Icon** (⚙) and select **Setup**.
3. In the "Quick Find" box on the left, type Scheduled Jobs.
4. Click on the **Scheduled Jobs** link under the "Jobs" heading.

You will see a list of all scheduled jobs in your org.