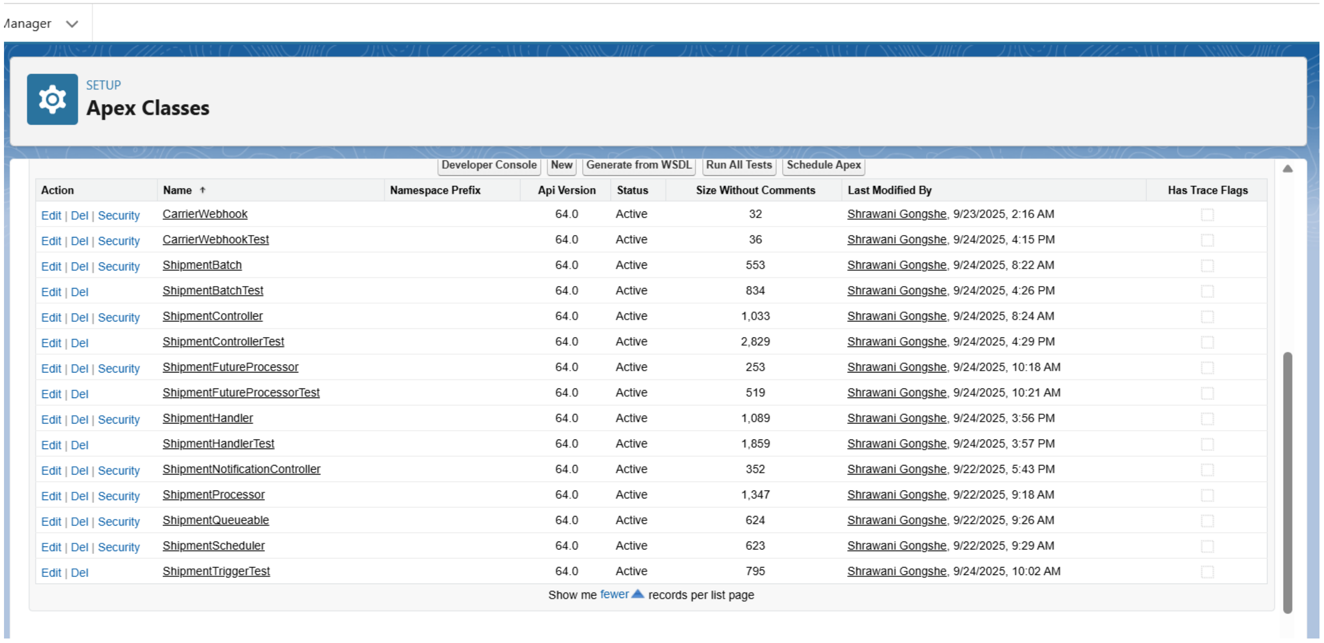
Phase 5: Apex Programming (Developer)

1. *Classes & Objects*

Apex Classes Implemented

- **CarrierWebhook / CarrierWebhookTest** → Handles integration with carrier system and validates through unit tests.
- **ShipmentBatch / ShipmentBatchTest** → Batch Apex class for processing large shipment records, with corresponding test class.
- **ShipmentController / ShipmentControllerTest** → Apex Controller for LWC/Visualforce handling shipment logic.
- **ShipmentFutureProcessor / ShipmentFutureProcessorTest** → Future methods to handle async callouts.
- **ShipmentHandler / ShipmentHandlerTest** → Trigger handler class following design pattern for scalable trigger logic.
- **ShipmentNotificationController** → Manages notifications (custom + email) for shipment events.
- **ShipmentProcessor** → Core business logic class for shipment operations.
- **ShipmentQueueable** → Queueable Apex for chained async job execution.
- **ShipmentScheduler** → Scheduled Apex for time-based automation (e.g., nightly shipment checks).
- **ShipmentTriggerTest** → Ensures trigger functionality and coverage through unit testing.



Apex Classes

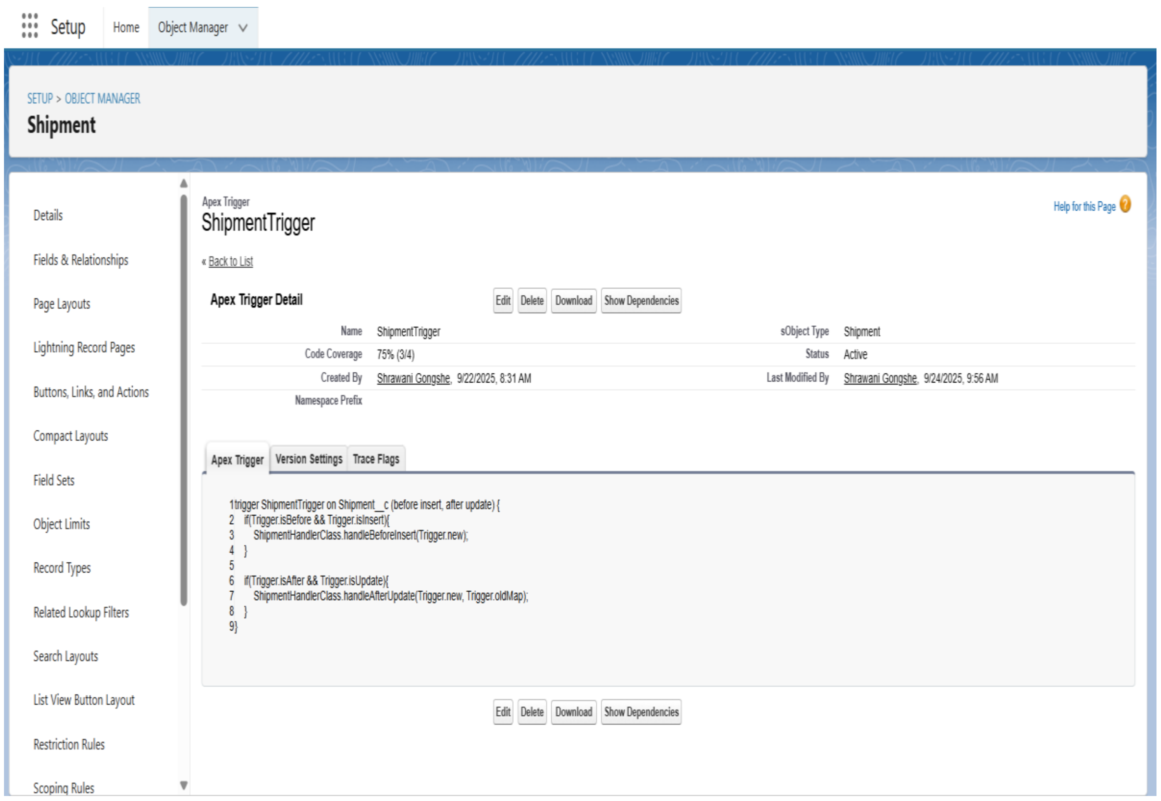
Action	Name ↑	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit | Del | Security	CarrierWebhook		64.0	Active	32	Shrawani Gongshe, 9/23/2025, 2:16 AM	☐
Edit | Del | Security	CarrierWebhookTest		64.0	Active	36	Shrawani Gongshe, 9/24/2025, 4:15 PM	☐
Edit | Del | Security	ShipmentBatch		64.0	Active	553	Shrawani Gongshe, 9/24/2025, 8:22 AM	☐
Edit | Del	ShipmentBatchTest		64.0	Active	834	Shrawani Gongshe, 9/24/2025, 4:26 PM	☐
Edit | Del | Security	ShipmentController		64.0	Active	1,033	Shrawani Gongshe, 9/24/2025, 8:24 AM	☐
Edit | Del	ShipmentControllerTest		64.0	Active	2,829	Shrawani Gongshe, 9/24/2025, 4:29 PM	☐
Edit | Del | Security	ShipmentFutureProcessor		64.0	Active	253	Shrawani Gongshe, 9/24/2025, 10:18 AM	☐
Edit | Del	ShipmentFutureProcessorTest		64.0	Active	519	Shrawani Gongshe, 9/24/2025, 10:21 AM	☐
Edit | Del | Security	ShipmentHandler		64.0	Active	1,089	Shrawani Gongshe, 9/24/2025, 3:56 PM	☐
Edit | Del	ShipmentHandlerTest		64.0	Active	1,859	Shrawani Gongshe, 9/24/2025, 3:57 PM	☐
Edit | Del | Security	ShipmentNotificationController		64.0	Active	352	Shrawani Gongshe, 9/22/2025, 5:43 PM	☐
Edit | Del | Security	ShipmentProcessor		64.0	Active	1,347	Shrawani Gongshe, 9/22/2025, 9:18 AM	☐
Edit | Del | Security	ShipmentQueueable		64.0	Active	624	Shrawani Gongshe, 9/22/2025, 9:26 AM	☐
Edit | Del | Security	ShipmentScheduler		64.0	Active	623	Shrawani Gongshe, 9/22/2025, 9:29 AM	☐
Edit | Del	ShipmentTriggerTest		64.0	Active	795	Shrawani Gongshe, 9/24/2025, 10:02 AM	☐

Show me fewer ▲ records per list page

2. Apex Triggers (before/after insert, update, delete)

ShipmentTrigger

- Runs on **Shipment__c** (before insert, after update).
- Uses **Handler Pattern** (`ShipmentHandlerClass`) for logic.
- Ensures clean separation of trigger and business logic.
- **Code Coverage:** 75%.



Apex Trigger Detail

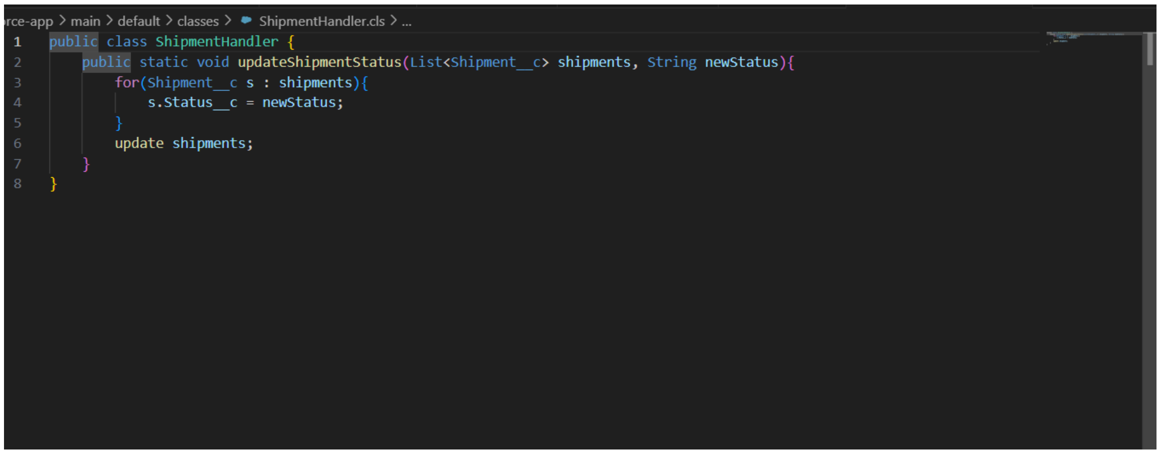
Name	ShipmentTrigger	sObject Type	Shipment
Code Coverage	75% (3/4)	Status	Active
Created By	Shrawani Gongshe, 9/22/2025, 8:31 AM	Last Modified By	Shrawani Gongshe, 9/24/2025, 9:56 AM
Namespace Prefix			

```
trigger ShipmentTrigger on Shipment__c (before insert, after update) {
  if(Trigger.isBefore && Trigger.isInsert){
    ShipmentHandlerClass.handleBeforeInsert(Trigger.new);
  }

  if(Trigger.isAfter && Trigger.isUpdate){
    ShipmentHandlerClass.handleAfterUpdate(Trigger.new, Trigger.oldMap);
  }
}
```

3. *Trigger Design Pattern (Handler approach for scalability)*

ShipmentHandler

- **Method:** `updateShipmentStatus(List<Shipment__c> shipments, String newStatus)`
- **Logic:** Iterates over shipment records and updates their `Status__c` field to the provided status.
- **Usage:** Supports **trigger** and other Apex classes for centralized shipment status updates.

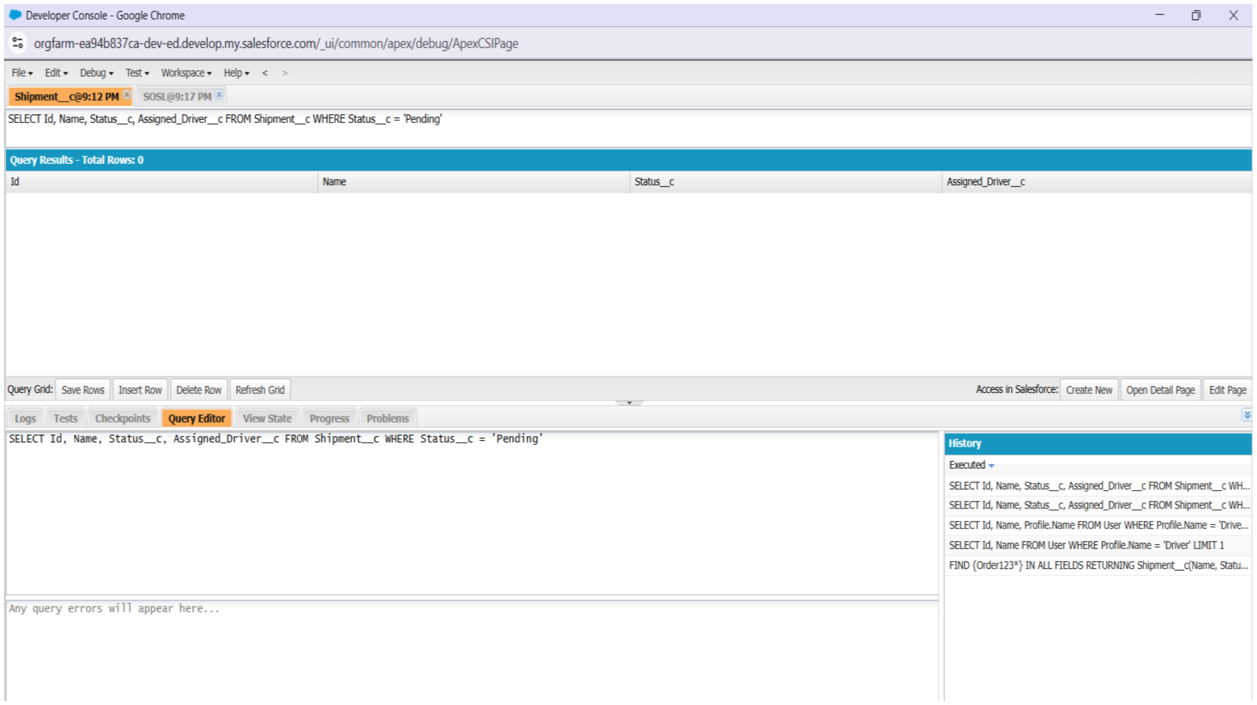

```
rce-app > main > default > classes > ShipmentHandler.cls > ...
public class ShipmentHandler {
    public static void updateShipmentStatus(List<Shipment__c> shipments, String newStatus){
        for(Shipment__c s : shipments){
            s.Status__c = newStatus;
        }
        update shipments;
    }
}
```

4. SOQL & SOSL (data retrieval and search)

- **SOQL Query:**

```
SELECT Id, Name, Status__c, Assigned_Driver__c
FROM Shipment__c
WHERE Status__c = 'Pending'
```

→ Retrieves shipments with status = *Pending*.


Developer Console - Google Chrome
orgfarm-ea94b837ca-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help < >
Shipment__c@9:12 PM
SOSL@9:17 PM

```
SELECT Id, Name, Status__c, Assigned_Driver__c FROM Shipment__c WHERE Status__c = 'Pending'
```

Query Results - Total Rows: 0

Id	Name	Status__c	Assigned_Driver__c

Query Grid: Save Rows Insert Row Delete Row Refresh Grid
Access in Salesforce: Create New Open Detail Page Edit Page
Logs Tests Checkpoints Query Editor View State Progress Problems

```
SELECT Id, Name, Status__c, Assigned_Driver__c FROM Shipment__c WHERE Status__c = 'Pending'
```

```
Any query errors will appear here...
```

History

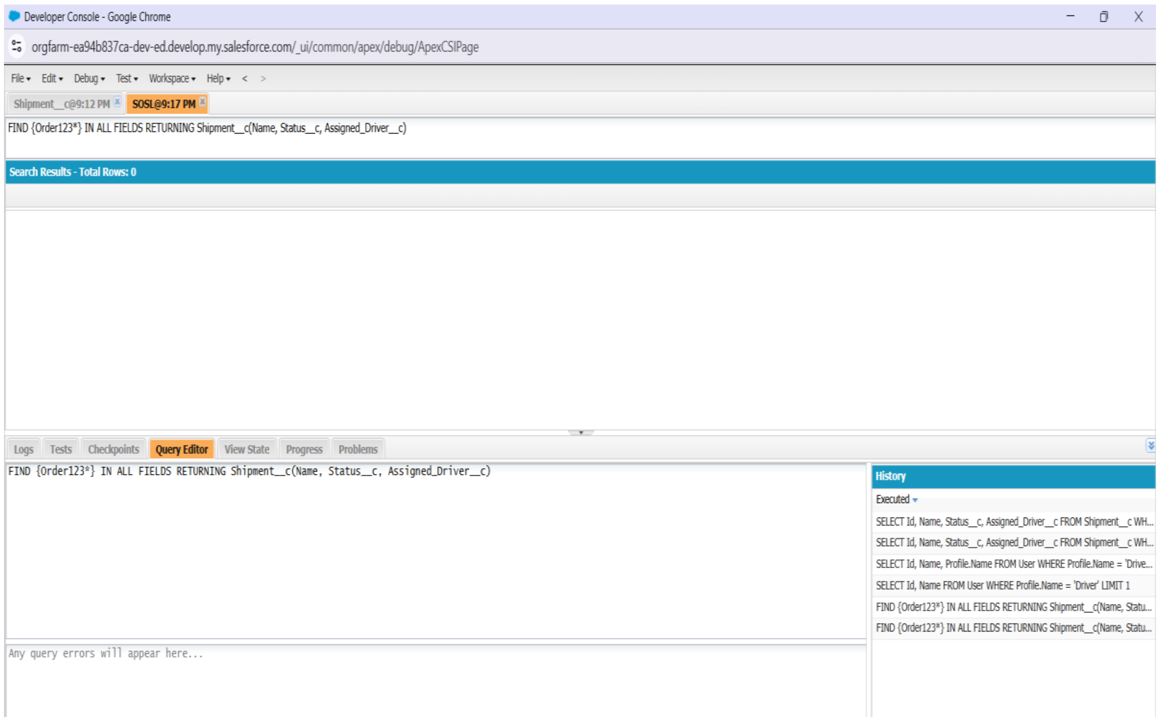
Executed

```
SELECT Id, Name, Status__c, Assigned_Driver__c FROM Shipment__c WH...
SELECT Id, Name, Status__c, Assigned_Driver__c FROM Shipment__c WH...
SELECT Id, Name, Profile.Name FROM User WHERE Profile.Name = 'Drive...
SELECT Id, Name FROM User WHERE Profile.Name = 'Driver' LIMIT 1
FIND {Order123*} IN ALL FIELDS RETURNING Shipment__c(Name, Statu...
```

SOSL Query:

```
FIND {Order123*} IN ALL FIELDS
RETURNING Shipment__c (Name, Status__c, Assigned_Driver__c)
```

→ Searches across all fields to locate shipments related to *Order123*.


Developer Console - Google Chrome
orgfarm-ea94b837ca-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help < >
Shipment__c@9:12 PM
SOSL@9:17 PM
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems
History
Executed

```
FIND {Order123*} IN ALL FIELDS RETURNING Shipment__c(Name, Status__c, Assigned_Driver__c)
```

Search Results - Total Rows: 0

```
FIND {Order123*} IN ALL FIELDS RETURNING Shipment__c(Name, Status__c, Assigned_Driver__c)
```

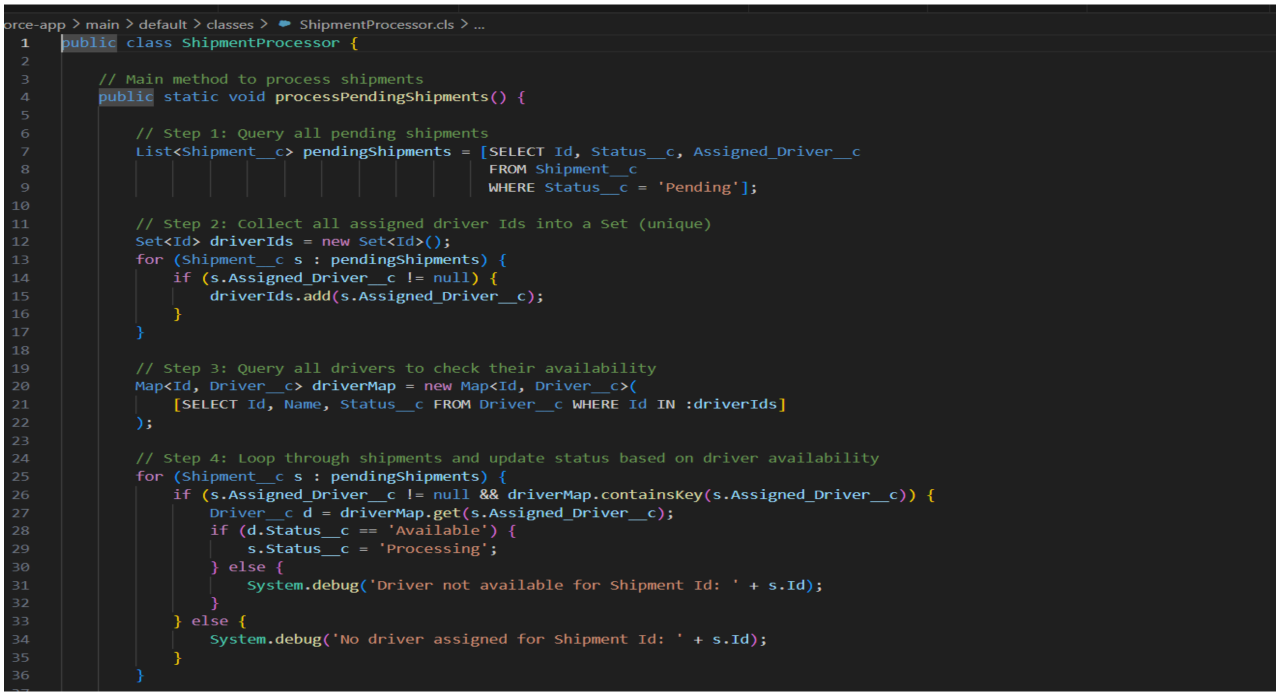
Any query errors will appear here...

```
SELECT Id, Name, Status__c, Assigned_Driver__c FROM Shipment__c WH...
SELECT Id, Name, Status__c, Assigned_Driver__c FROM Shipment__c WH...
SELECT Id, Name, Profile.Name FROM User WHERE Profile.Name = 'Drive...
SELECT Id, Name FROM User WHERE Profile.Name = 'Driver' LIMIT 1
FIND {Order123*} IN ALL FIELDS RETURNING Shipment__c(Name, Statu...
FIND {Order123*} IN ALL FIELDS RETURNING Shipment__c(Name, Statu...
```

5. Collections: List, Set, Map

Collections Used in ShipmentProcessor

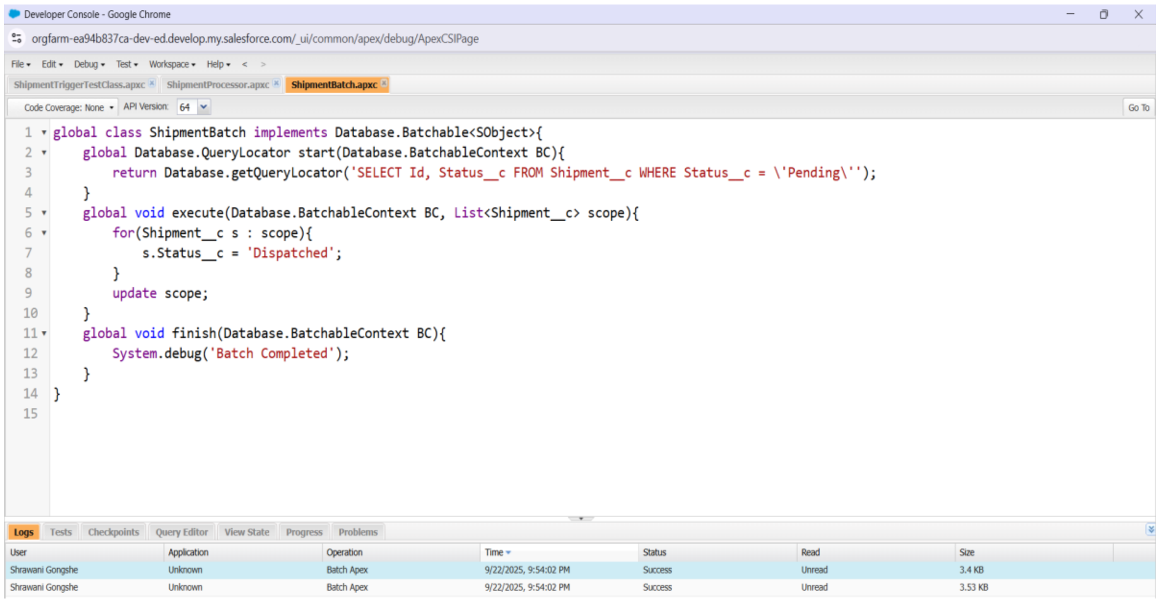
- **List →**
- `List<Shipment__c> pendingShipments`
    - Stores all shipments with `Status__c = 'Pending'`.
- **Set →**
- `Set<Id> driverIds`
    - Collects unique driver IDs from pending shipments.
- **Map →**
- `Map<Id, Driver__c> driverMap`
    - Maps driver IDs to driver records for quick availability lookup


orce-app > main > default > classes > ShipmentProcessor.cls > ...

```
public class ShipmentProcessor {

    // Main method to process shipments
    public static void processPendingShipments() {

        // Step 1: Query all pending shipments
        List<Shipment__c> pendingShipments = [SELECT Id, Status__c, Assigned_Driver__c
                                              FROM Shipment__c
                                              WHERE Status__c = 'Pending'];

        // Step 2: Collect all assigned driver Ids into a Set (unique)
        Set<Id> driverIds = new Set<Id>();
        for (Shipment__c s : pendingShipments) {
            if (s.Assigned_Driver__c != null) {
                driverIds.add(s.Assigned_Driver__c);
            }
        }

        // Step 3: Query all drivers to check their availability
        Map<Id, Driver__c> driverMap = new Map<Id, Driver__c>(
            [SELECT Id, Name, Status__c FROM Driver__c WHERE Id IN :driverIds]
        );

        // Step 4: Loop through shipments and update status based on driver availability
        for (Shipment__c s : pendingShipments) {
            if (s.Assigned_Driver__c != null && driverMap.containsKey(s.Assigned_Driver__c)) {
                Driver__c d = driverMap.get(s.Assigned_Driver__c);
                if (d.Status__c == 'Available') {
                    s.Status__c = 'Processing';
                } else {
                    System.debug('Driver not available for Shipment Id: ' + s.Id);
                }
            } else {
                System.debug('No driver assigned for Shipment Id: ' + s.Id);
            }
        }
```

6. Batch Apex (process large data volumes)

- **Class:** `ShipmentBatch` implements `Database.Batchable<SObject>`.
- **Logic:**
    - **Start:** Retrieves all shipments with `Status__c = 'Pending'`.
    - **Execute:** Updates each batch of records → sets status to **Dispatched**.
    - **Finish:** Logs `"Batch Completed"` after processing.
- **Purpose:** Efficiently processes **large volumes of shipments** in chunks without hitting governor limits.


Developer Console - Google Chrome
orgfarm-ea94b837ca-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help < >
ShipmentTriggerTestClass.apxc
ShipmentProcessor.apxc
ShipmentBatch.apxc
Code Coverage: None
API Version: 64
Go To
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems

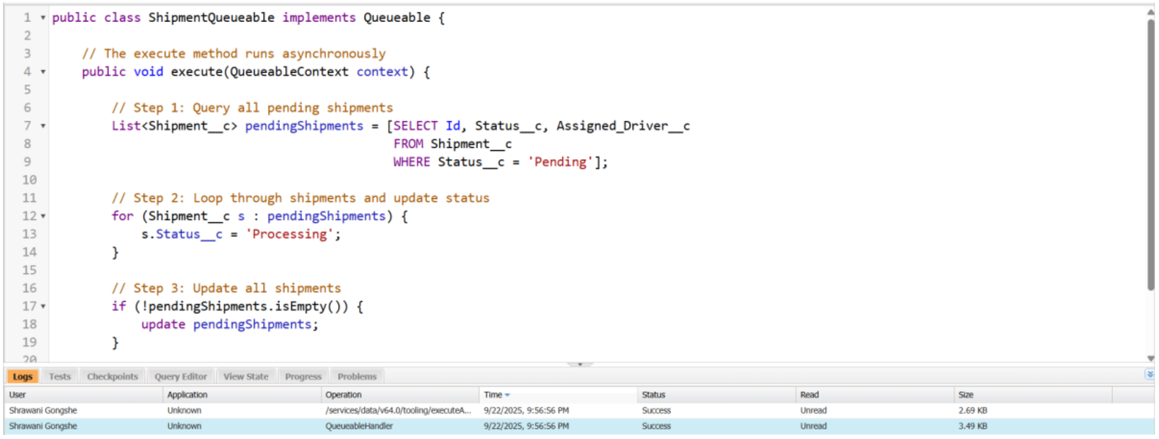
```
global class ShipmentBatch implements Database.Batchable<SObject>{
    global Database.QueryLocator start(Database.BatchableContext BC){
        return Database.getQueryLocator('SELECT Id, Status__c FROM Shipment__c WHERE Status__c = \'Pending\'');
    }
    global void execute(Database.BatchableContext BC, List<Shipment__c> scope){
        for(Shipment__c s : scope){
            s.Status__c = 'Dispatched';
        }
        update scope;
    }
    global void finish(Database.BatchableContext BC){
        System.debug('Batch Completed');
    }
}
```

User	Application	Operation	Time	Status	Read	Size
Shrawani Gongshe	Unknown	Batch Apex	9/22/2025, 9:54:02 PM	Success	Unread	3.4 KB
Shrawani Gongshe	Unknown	Batch Apex	9/22/2025, 9:54:02 PM	Success	Unread	3.53 KB

7. Queueable Apex (chained async jobs)

- **Class:** `ShipmentQueueable` implements `Queueable`.
- **Logic:**

- **Step 1:** Queries all shipments with `Status__c = 'Pending'`.
- **Step 2:** Iterates shipments → updates status to **Processing**.
- **Step 3:** Updates all modified shipment records.
- **Purpose:** Runs asynchronously, supports **chaining multiple jobs**, and is lighter than Batch Apex for smaller async tasks.



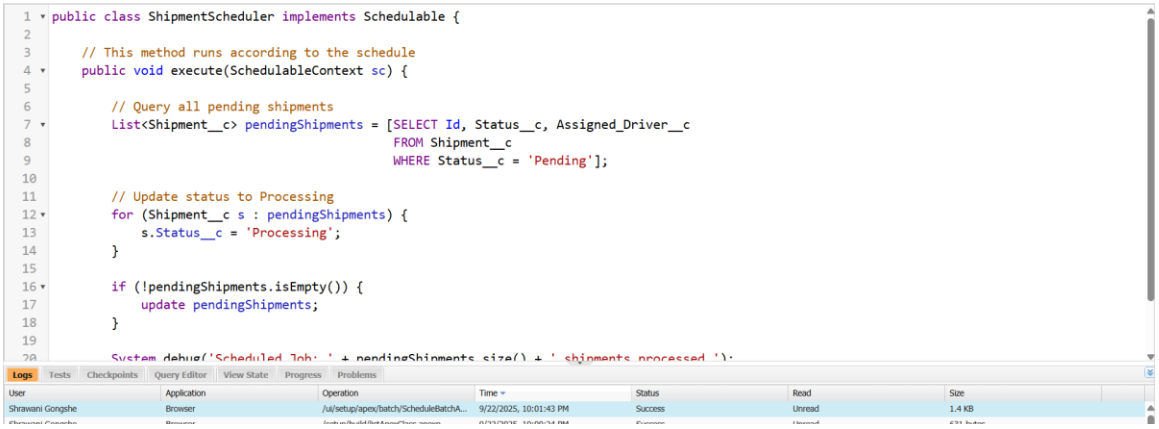
```
public class ShipmentQueueable implements Queueable {

    // The execute method runs asynchronously
    public void execute(QueueableContext context) {

        // Step 1: Query all pending shipments
        List<Shipment__c> pendingShipments = [SELECT Id, Status__c, Assigned_Driver__c
                                              FROM Shipment__c
                                              WHERE Status__c = 'Pending'];

        // Step 2: Loop through shipments and update status
        for (Shipment__c s : pendingShipments) {
            s.Status__c = 'Processing';
        }

        // Step 3: Update all shipments
        if (!pendingShipments.isEmpty()) {
            update pendingShipments;
        }
```

Logs | Tests | Checkpoints | Query Editor | View State | Progress | Problems

User	Application	Operation	Time	Status	Read	Size
Shrawani Gongshe	Unknown	/services/data/v64.0/tooling/executeA...	9/22/2025, 9:56:56 PM	Success	Unread	2.69 KB
Shrawani Gongshe	Unknown	QueueableHandler	9/22/2025, 9:56:56 PM	Success	Unread	3.49 KB

8. Scheduled Apex (time-based job execution)

- **Class:** `ShipmentScheduler` implements `Schedulable`.
- **Logic:**
- Queries shipments with `Status__c = 'Pending'`.
- Updates their status to **Processing**.
- **Execution:** Runs automatically based on a defined **CRON schedule**.
- **Purpose:** Automates shipment processing at fixed times (e.g., nightly updates).



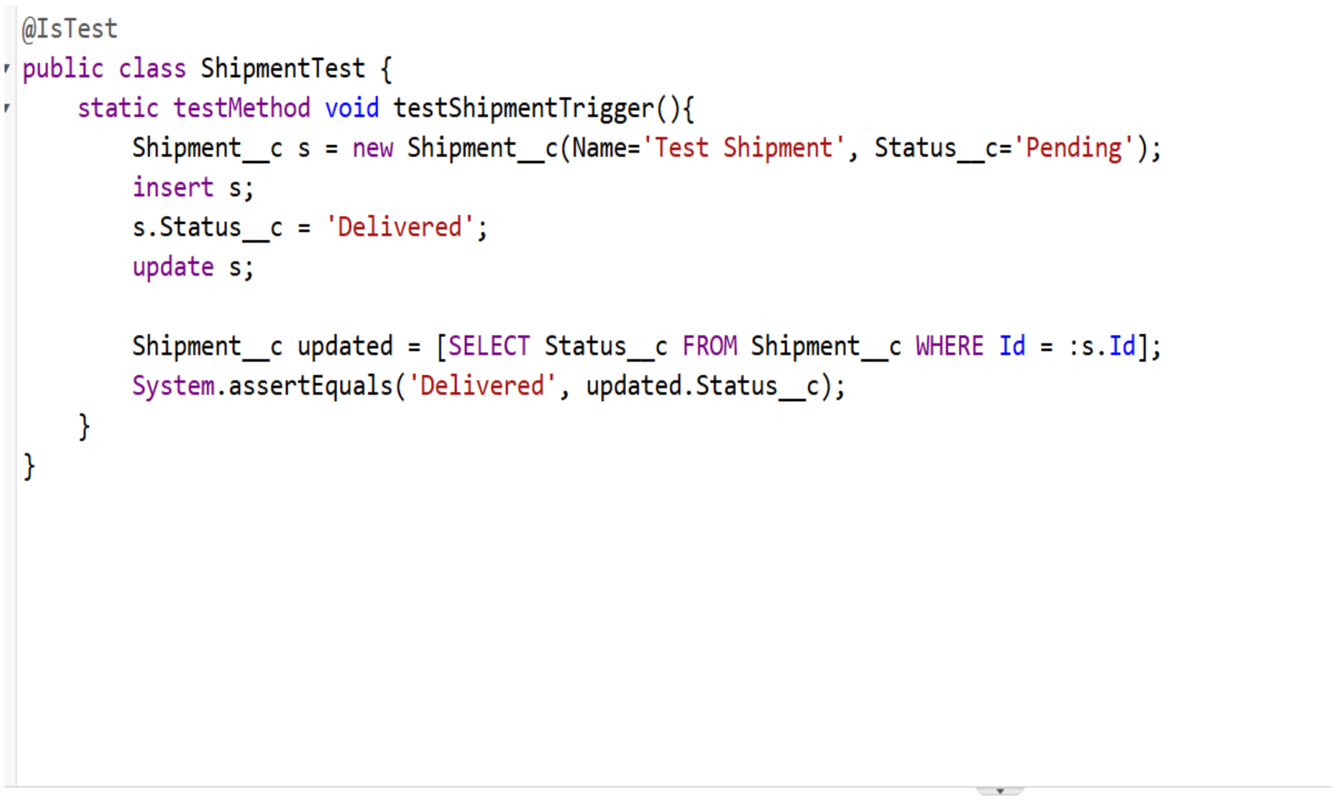
```
public class ShipmentScheduler implements Schedulable {

    // This method runs according to the schedule
    public void execute(SchedulableContext sc) {

        // Query all pending shipments
        List<Shipment__c> pendingShipments = [SELECT Id, Status__c, Assigned_Driver__c
                                              FROM Shipment__c
                                              WHERE Status__c = 'Pending'];

        // Update status to Processing
        for (Shipment__c s : pendingShipments) {
            s.Status__c = 'Processing';
        }

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

        System.debug('Scheduled Job: ' + pendingShipments.size() + ' shipments processed.');
```

Logs | Tests | Checkpoints | Query Editor | View State | Progress | Problems

User	Application	Operation	Time	Status	Read	Size
Shrawani Gongshe	Browser	/ui/setup/apex/batch/ScheduleBatchA...	9/22/2025, 10:01:43 PM	Success	Unread	1.4 KB
Shrawani Gongshe	Browser	/setup/build/listApexClass.apexp	9/22/2025, 10:00:24 PM	Success	Unread	631 bytes

9. Test Classes (unit testing with coverage)

- **Class:** `ShipmentTest` marked with `@IsTest`.
- **Logic:**
  - Inserts a test shipment with `Status__c = 'Pending'`.
  - Updates status to **Delivered**.
  - Uses `System.assertEquals()` to validate expected outcome.
- **Purpose:** Ensures **trigger logic** works correctly and contributes to **code coverage** for deployments.



```
@IsTest
public class ShipmentTest {
    static testMethod void testShipmentTrigger(){
        Shipment__c s = new Shipment__c(Name='Test Shipment', Status__c='Pending');
        insert s;
        s.Status__c = 'Delivered';
        update s;

        Shipment__c updated = [SELECT Status__c FROM Shipment__c WHERE Id = :s.Id];
        System.assertEquals('Delivered', updated.Status__c);
    }
}
```