**Phase 5: Apex Programming (Developer)**

**Classes & Objects:**

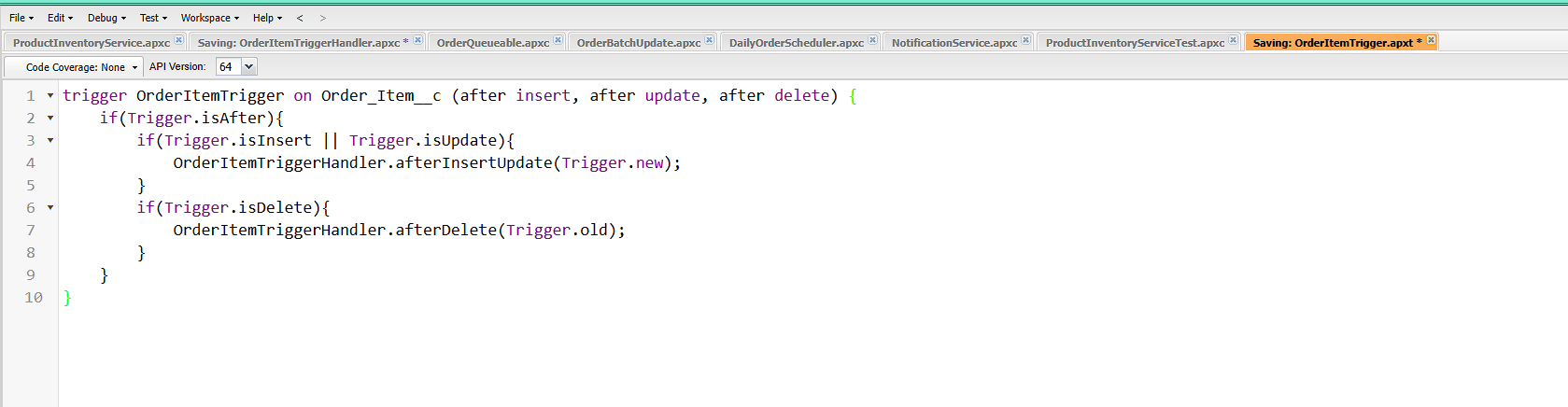
|  |  |  |
| --- | --- | --- |
| Class / Trigger | Type | Purpose |
| ProductInventoryService | **Apex Class** | **Core business logic for inventory updates.** |
| OrderItemTriggerHandler | **Apex Class** | **Trigger handler (keeps trigger clean).** |
| OrderItemTrigger | **Apex Trigger** | **Executes on Order Item insert/update/delete.** |
| OrderBatchUpdate | **Apex Class** | **Batch job to bulk update orders.** |
| OrderQueueable | **Apex Class** | **Queueable job for async processing.** |
| DailyOrderScheduler | **Apex Class** | **Schedules daily job execution.** |
| NotificationService | **Apex Class** | **Sends async emails via @future.** |
| ProductInventoryServiceTest | **Apex Test Class** | **Unit tests for inventory trigger logic.** |
| Inventory\_Threshold\_Config\_\_mdt | **Custom Metadata Type** | **Stores configurable reorder levels and notification emails.** |

**Apex Triggers (before/after insert/update/delete):**

**Trigger Name:** OrderItemTrigger

**Events Handled:** after insert, after update, after delete

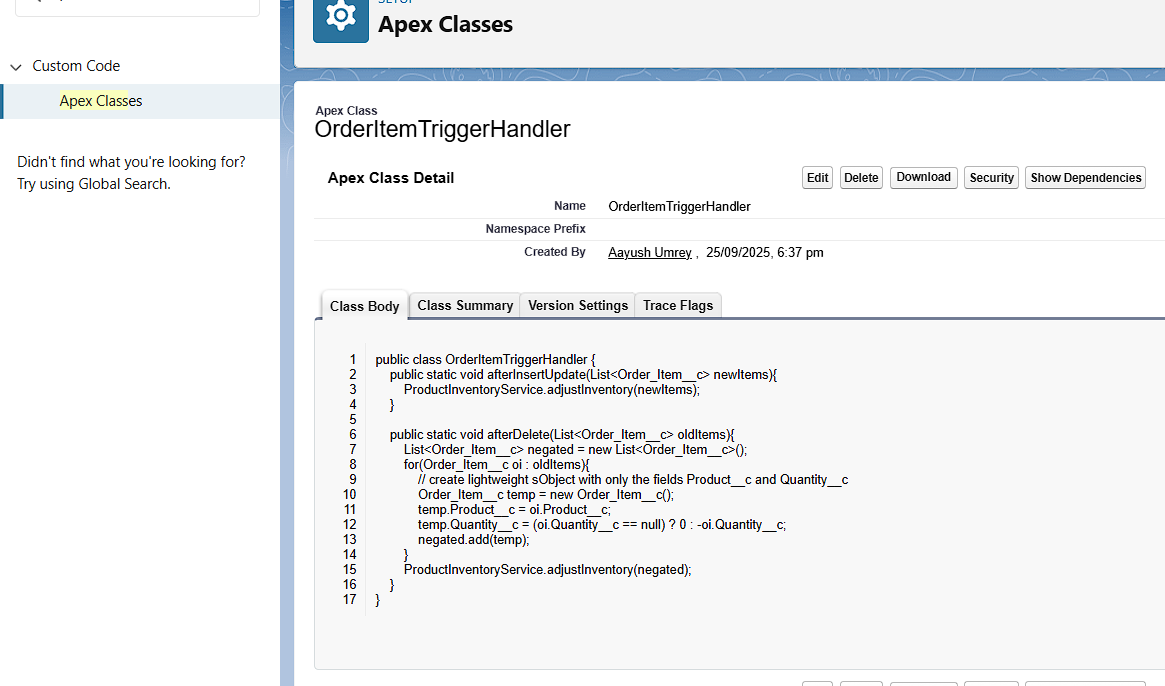
**Purpose:** Update reserved inventory whenever order items are inserted, updated, or delete

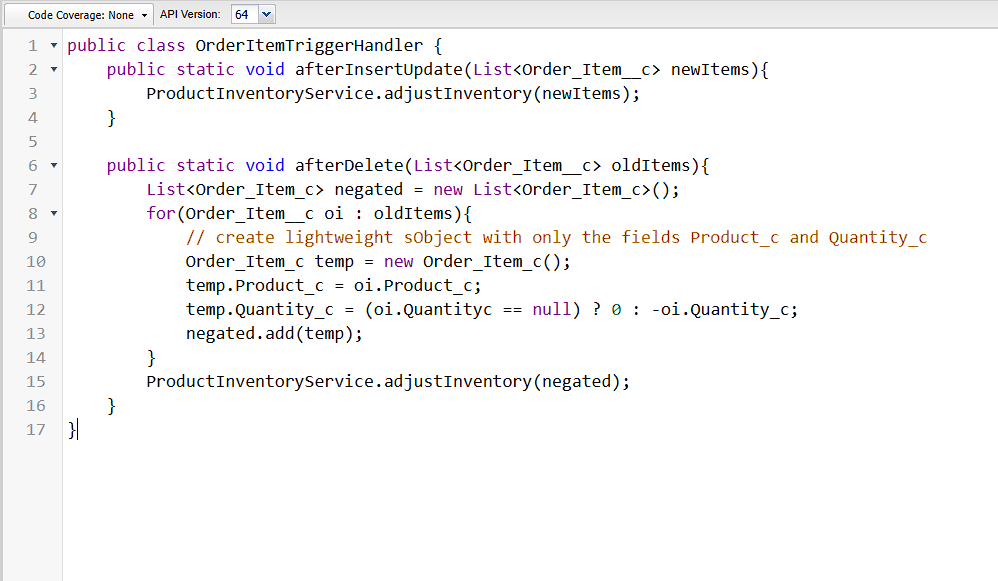


**Trigger Design Pattern:**

We implemented the Trigger Handler pattern by creating `OrderItemTriggerHandler`. This separates trigger logic from the trigger itself and keeps the code bulk-safe.

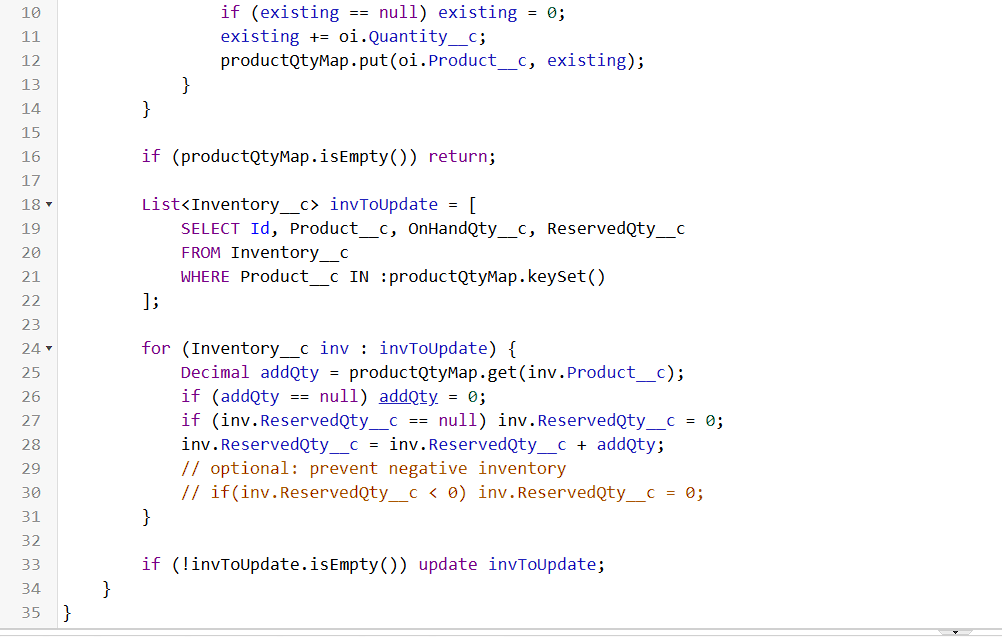
**Handler:** OrderItemTriggerHandler

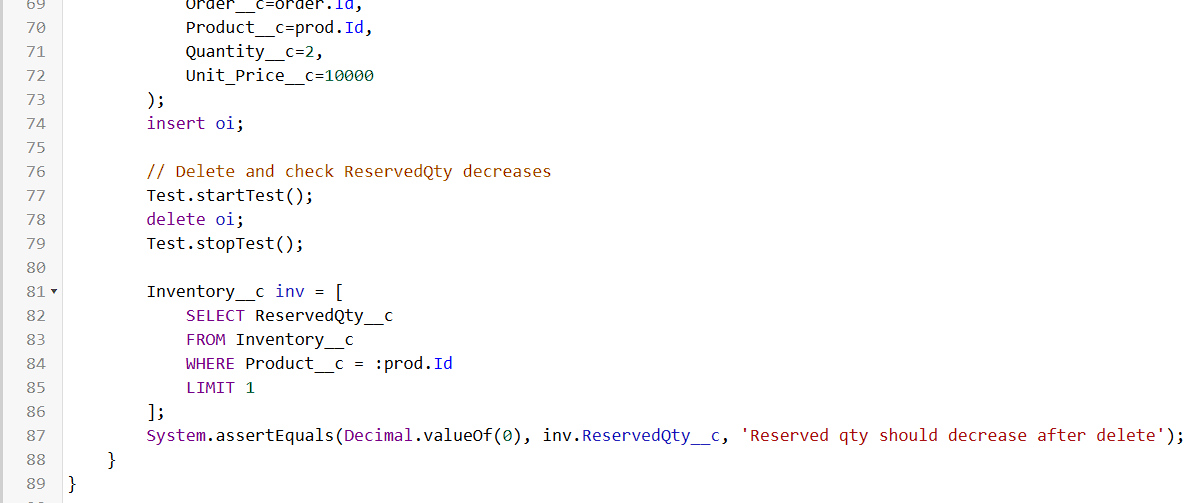




**SOQL & SOSL:**

**We used SOQL queries in both `ProductInventoryService and `ProductInventoryService Test` .**





**Collections: List, Set, Map:**

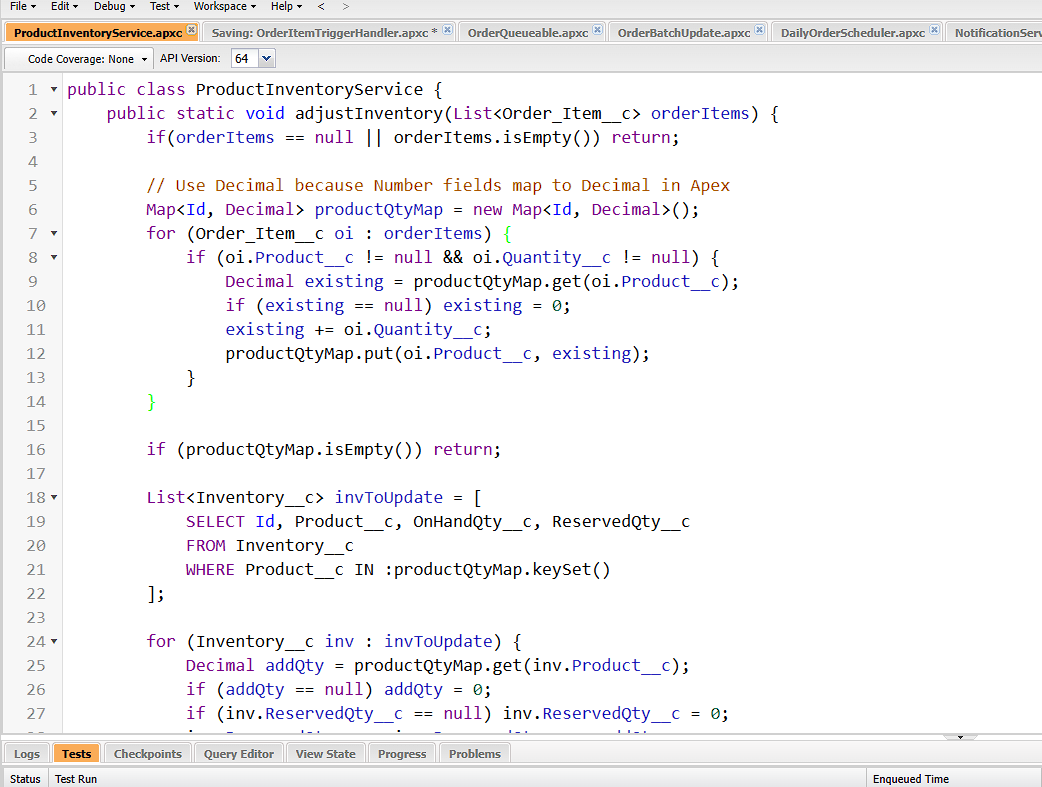
**List:** Store ordered values.

**Set:** Store unique product IDs.

**Map: Product →** ReservedQty updates.

**Control Statements:**

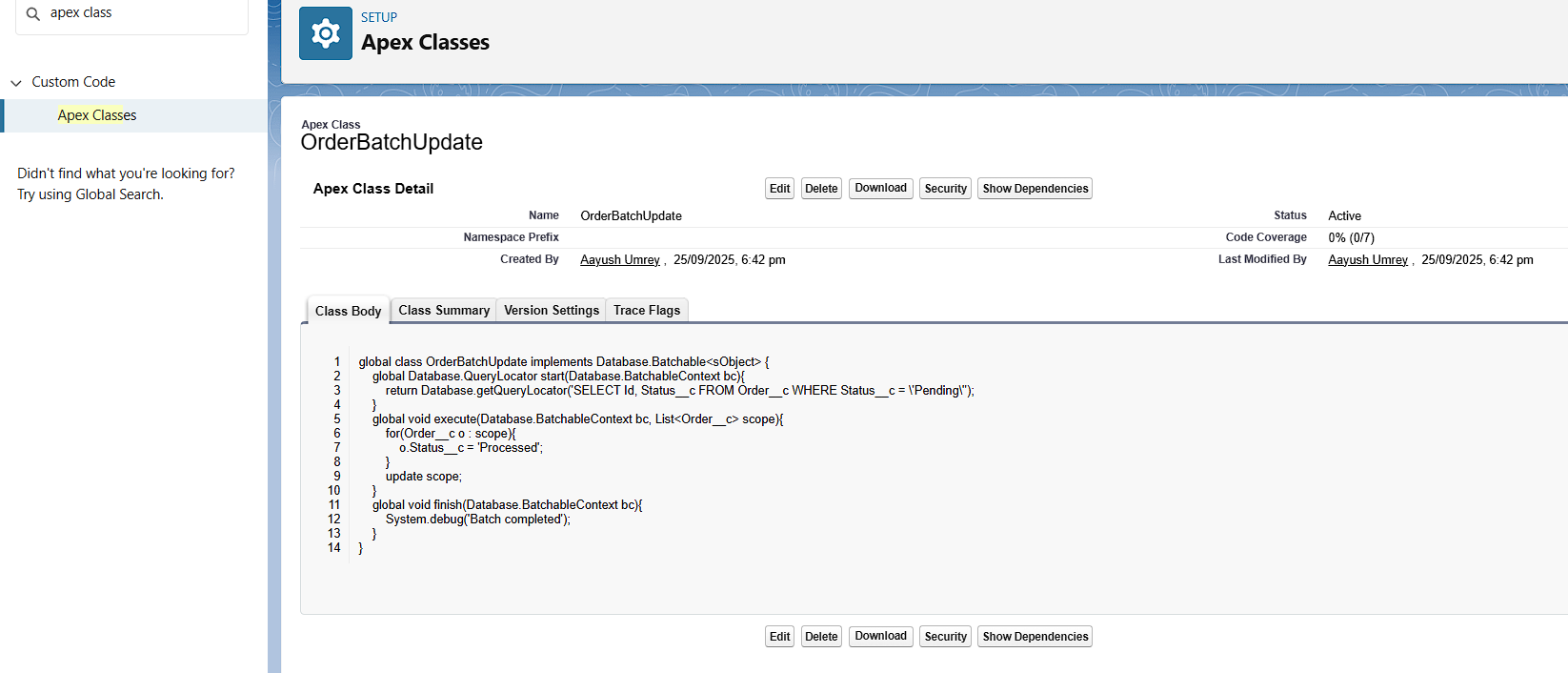
Used for looping and condition handling.

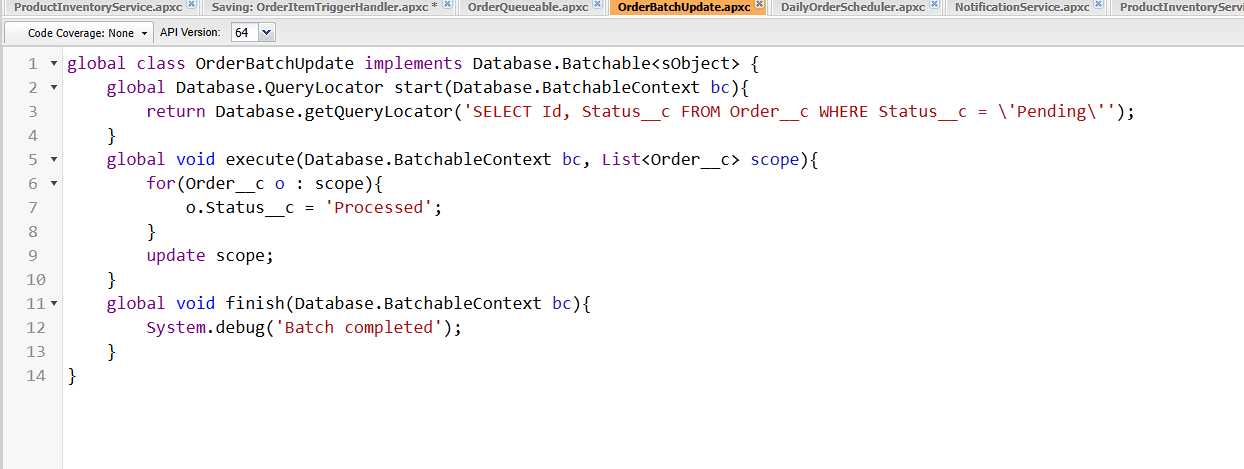


**Batch Apex:**

Used to process large records asynchronously.

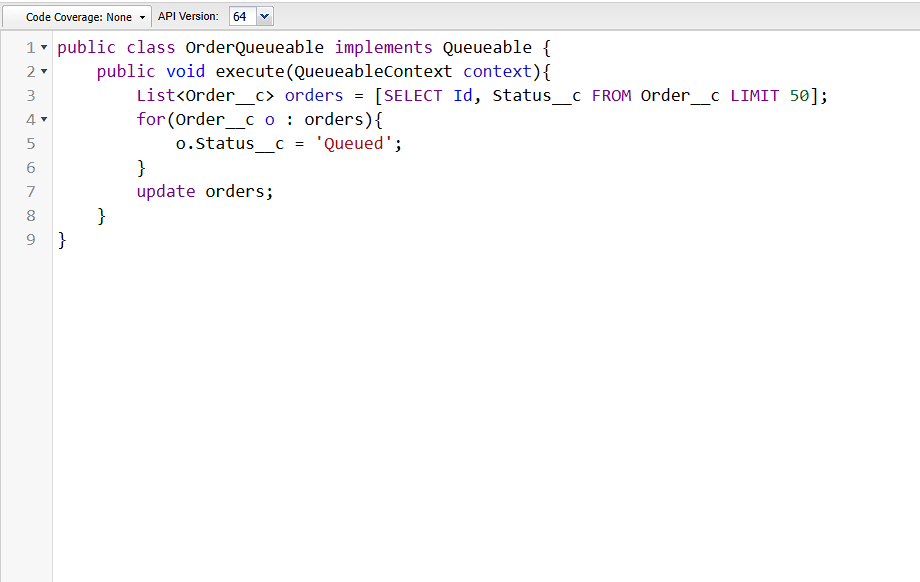
We have created “OrderBatchUpdate” Apex class





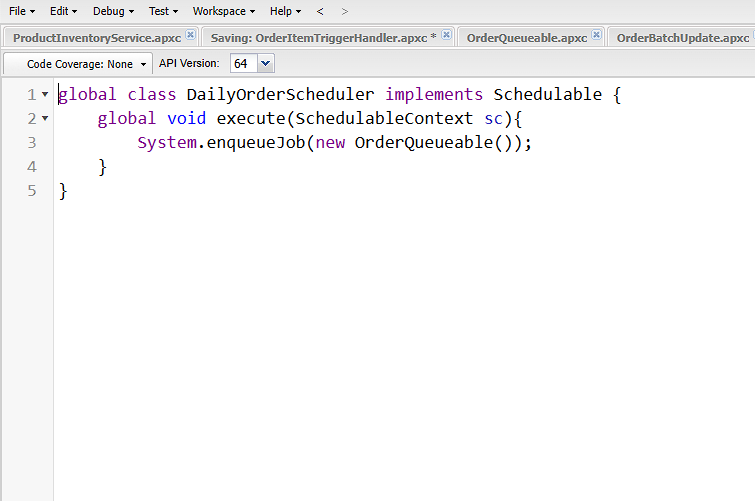
**Queueable Apex:**

We have Created “OrderQueueable” Apex class



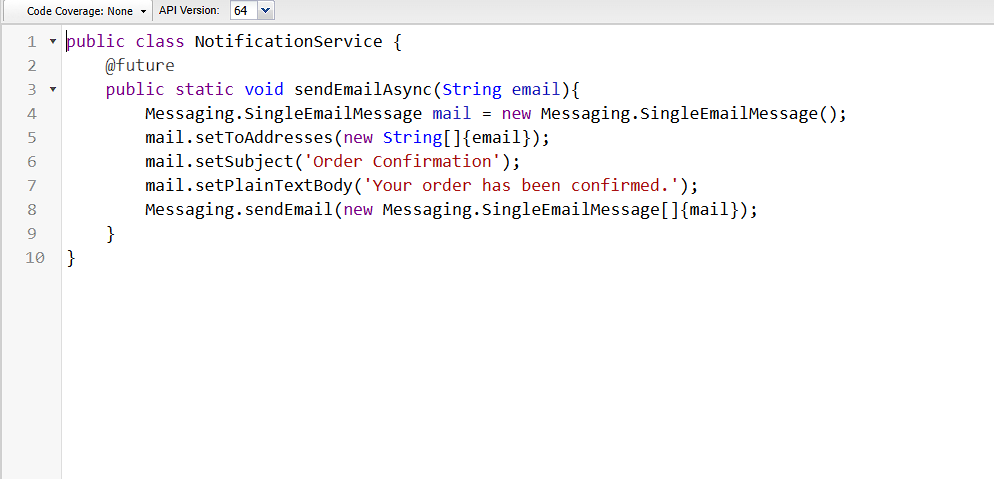
**Scheduled Apex:**

We have created “DailyOrderScheduler” Apex class



**Future Methods:**

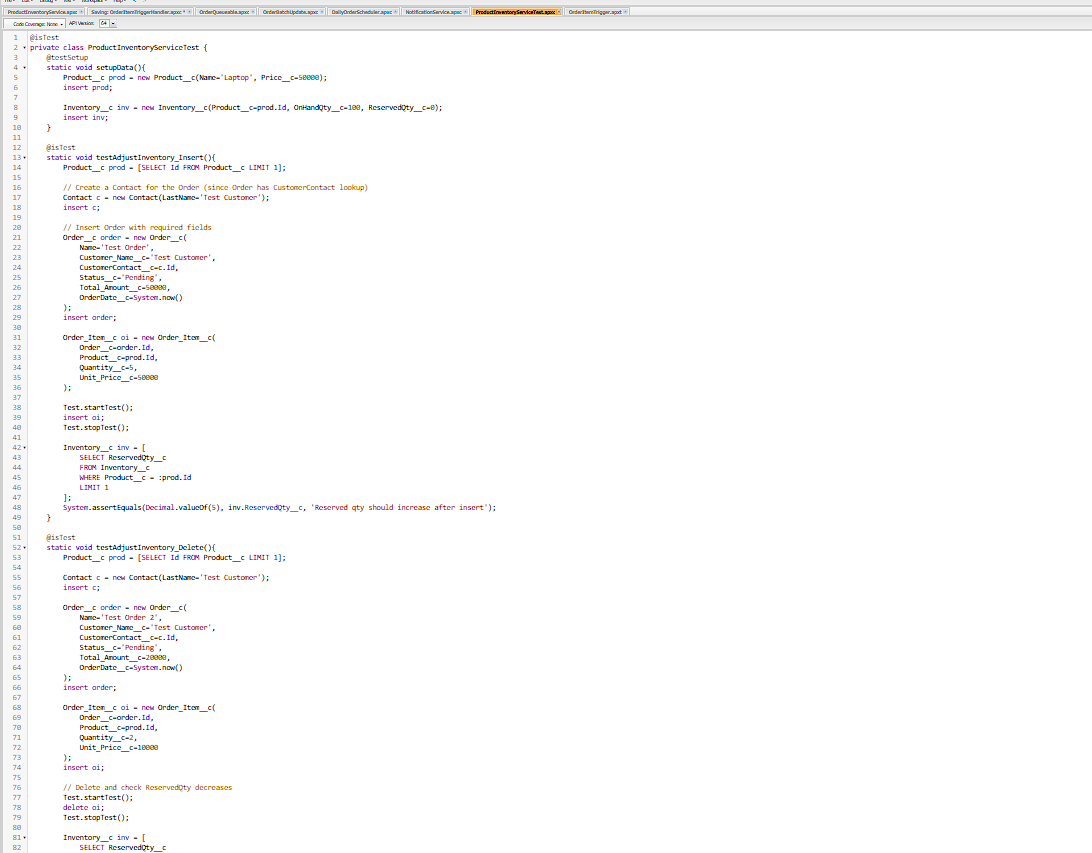
For Future Method we have created “NotificationService” Apex Class



**Test Classes:**

Written to cover triggers, services, and async processes.

We have created “ProductInventoryServiceTest” Apex Test Class.



**Asynchronous Processing:**

**Batch Apex →** Large volume updates.

**Queueable Apex →** Chain jobs.

**Scheduled Apex →** Run daily/weekly jobs.

**Future Methods →** Lightweight async tasks (e.g., email, callouts).